# UNIVERSITI POLY-TECH MALAYSIA

# UPTM ACADEMIC AI ASSISTANT TOOLS

MUHAMMAD ZULHUSNI BIN ROSDAN

BACHELOR OF INFORMATION TECHNOLOGY (HONS) IN COMPUTER APPLICATION DEVELOPMENT

# UNIVERSITI POLY-TECH MALAYSIA

## Faculty of Computing & Multimedia

## UPTM ACADEMIC AI ASSISTANT TOOLS

**MUHAMMAD ZULHUSNI BIN ROSDAN**

**AM2405015965**

**FYP4144**

**DECEMBER 2025**

# Declaration of Originality

This project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet, etc.) has been acknowledged within the main report to an item in the References or Bibliography lists.

I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

## Copyright Acknowledgement

I acknowledge that the copyright of this project and report belongs to Universiti Poly-Tech Malaysia.

Signed: Date: 14/11/2025

Office Stamp

# Abstract

The UPTM Academic AI Assistant Tools is a web-based system developed to support academic activities among students and lecturers at Universiti Poly-Tech Malaysia (UPTM), particularly within the Faculty of Computing and Multimedia (FCOM). The system integrates artificial intelligence (AI) technology to assist users in generating, organizing, and understanding academic content more effectively. It was designed in response to the challenges faced when using general-purpose AI tools, such as complex prompt creation, lack of academic focus, and security risks associated with external data storage. Developed using the PHP Laravel framework for backend management and HTML, CSS, and Bootstrap for the frontend interface, the system offers a user-friendly, secure, and responsive environment. Key features include built-in academic templates, which simplify academic writing, learning and teaching tasks. The system also emphasizes security through authentication, role-based access control, password hashing, and session management to ensure data protection and privacy within UPTM's internal environment. The project's implementation follows a structured software development process, involving requirement gathering, design, development, and testing. Testing results confirmed that all modules, including AI integration and user management, function as intended. Overall, the system successfully achieves its objectives by providing ready-to-use academic AI tools that enhance productivity, support teaching and learning, and maintain data security, making it a reliable academic support platform tailored to institutional needs.

# TABLE OF CONTENTS

*1 INTRODUCTION.....16*

**1.1 Introduction..... 16**

**1.2 Project Background ..... 16**

**1.3 Problem Statement..... 17**

1.3.1 Difficulty in using general AI tools .....17

1.3.2 Lack of academic focused features.....18

1.3.3 Lack of security and reliability .....18

**1.4 Project Objectives ..... 19**

1.4.1 To provide ready-to-use academic AI tools.....19

1.4.2 To develop an AI system for productive academic tasks .....19

1.4.3 To provide a secure and institution-friendly AI system for education .....19

**1.5 Scope and Target User ..... 20**

1.5.1 Project Scope .....20

1.5.2 Product Scope.....21

1.5.3 Target User .....21

**1.6 Overview of This Report ..... 22**

*2 LITERATURE REVIEW.....24*

**2.1 Introduction..... 24**

**2.2 Investigation ..... 24**

2.2.1 Academic AI Assistant Tools.....24

2.2.2 Prompt Engineering for Academic Use .....25

2.2.3 Data Privacy and Security in Educational AI Systems .....26

**2.3 Related Works..... 28**

2.3.1 Google Classroom (with Gemini AI Tools for Educators).....28

2.3.2 ChatGPT (OpenAI) ..... 31
2.3.3 DeepSeek ..... 33
2.3.4 Scribbr ..... 35
**2.4 Comparison ..... 37**
**2.5 Discussion ..... 39**
**2.6 Conclusion ..... 41**
***3 METHODOLOGY ..... 42***
**3.1 Introduction ..... 42**
**3.2 Agile Methodology ..... 42**
**3.3 Phases in Agile Methodology ..... 43**
3.3.1 Phase 1: Plan ..... 43
3.3.2 Phase 2: Design ..... 44
3.3.3 Phase 3: Develop ..... 45
3.3.4 Phase 4: Test ..... 45
3.3.5 Phase 5: Deploy ..... 46
3.3.6 Phase 6: Review ..... 46
**3.4 Requirement ..... 47**
3.4.1 Data Gathering Techniques ..... 47
3.4.2 Functional Requirement ..... 48
3.4.3 Non-Function Requirement ..... 52
3.4.4 System Requirement ..... 53
**3.5 Analysis ..... 55**
3.5.1 Data Gathering Analysis ..... 55
3.5.2 Use Case Model ..... 79
3.5.3 BPNM (Business Process Modelling Notation) ..... 83
**3.6 Conclusion ..... 86**

*4 DESIGN.....87*

**4.1 Introduction..... 87**

**4.2 Interface Design..... 87**

**4.3 Database Design..... 104**

4.3.1 Data Dictionary.....104

4.3.2 Data Flow Diagram (DFD).....111

4.3.3 Entity Relational Diagram (ERD).....115

**4.4 Flow of the System..... 117**

**4.5 Implementation..... 120**

4.5.1 Execution Platform.....120

4.5.2 Implementation Tools.....121

4.5.3 System Interface.....123

4.5.3 Significant Functions.....134

**4.6 Conclusion..... 145**

*5 FINDING.....146*

**5.1 Introduction..... 146**

**5.2 Testing..... 146**

**5.3 Unit Testing..... 147**

**5.4 Integration Testing..... 152**

**5.5 System Testing..... 155**

**5.6 Acceptance Testing..... 164**

5.6.1 Client Acceptance Testing.....164

5.6.2 User Acceptance Testing.....167

*6 CONCLUSION.....169*

**6.1 Introduction ..... 169**

**6.2 Project Schedule ..... 169**

6.2.1 Work Breakdown Structure ..... 170

6.2.2 Gantt Chart ..... 173

**6.3 Risk Management ..... 177**

**6.4 Achievement ..... 178**

6.4.1 To Provide Ready-to-Use Academic AI Tools ..... 178

6.4.2 To Develop an AI System for Productive Academic Tasks ..... 179

6.4.3 To Provide a Secure and Institution-Friendly AI System ..... 180

**6.5 Constraint and Limitation ..... 181**

**6.6 Future Work and Recommendation ..... 182**

**6.7 Conclusion ..... 183**

***Appendix A – Requirements Specification Document ..... 184***

***Appendix B – User Manual ..... 184***

***Appendix C – Turnitin Result ..... 191***

***Appendix D – Log Book ..... 199***

***References ..... 202***

# LIST OF FIGURES

Figure 2.1 Google Classroom Gemini Dashboard ..... 26
Figure 2.2 ChatGPT ..... 29
Figure 2.3 DeepSeek ..... 31
Figure 2.4 Scribbr ..... 33
Figure 3.1 Agile Methodology Diagram (Hari, 2025) ..... 40
Figure 3.2 Questionnaire Question 1 ..... 54
Figure 3.3 Questionnaire Question 2 ..... 55
Figure 3.4 Questionnaire Question 2 ..... 55
Figure 3.5 Questionnaire Question 3 ..... 56
Figure 3.6 Questionnaire Question 4 ..... 58
Figure 3.7 Questionnaire Question 5 ..... 59
Figure 3.8 Questionnaire Question 6 ..... 61
Figure 3.9 Questionnaire Question 7 ..... 63
Figure 3.10 Questionnaire Question 8 ..... 65
Figure 3.11 Questionnaire Question 9 ..... 66
Figure 3.12 Questionnaire Question 10 ..... 67
Figure 3.13 Questionnaire Question 11 ..... 68
Figure 3.14 Questionnaire Question 12 ..... 69
Figure 3.15 Questionnaire Question 13 ..... 71
Figure 3.16 Use Case Diagram 1.0 ..... 77
Figure 3.17 Use Case Diagram 2.0 ..... 78
Figure 3.18 Use Case Diagram 3.0 ..... 79
Figure 3.19 Business Process Modelling Notation (BPMN) ..... 81
Figure 4.1 Login (Users/Admin/SuperAdmin) ..... 85
Figure 4.2 Sign Up (Users) ..... 86
Figure 4.3 Forgot Password (Users) ..... 86
Figure 4.4 Reset New Password (Users) ..... 86

Figure 4.5 Dashboard (SuperAdmin) ..... 87
Figure 4.6 Dashboard (Admin) ..... 88
Figure 4.7 Dashboard (User) ..... 89
Figure 4.8 Profile (Users/Admin/SuperAdmin) ..... 90
Figure 4.9 Change Password (Users/Admin/SuperAdmin) ..... 90
Figure 4.10 All Users (Admin) ..... 91
Figure 4.11 Manage Users (SuperAdmin) ..... 91
Figure 4.12 Delete Confirmation Modal (Admin/SuperAdmin) ..... 92
Figure 4.13 Edit Modal (SuperAdmin) ..... 92
Figure 4.14 Reset User Password (SuperAdmin) ..... 93
Figure 4.15 Create New User (SuperAdmin) ..... 93
Figure 4.16 Add New Template (Admin/SuperAdmin) ..... 94
Figure 4.17 All Template (SuperAdmin) ..... 94
Figure 4.18 All Template (Admin) ..... 95
Figure 4.19 All Template (Users) ..... 95
Figure 4.20 Preview Template (Users/Admin/SuperAdmin) ..... 96
Figure 4.21 Change Status Template (SuperAdmin) ..... 96
Figure 4.22 Delete Template Confirmation Modal ..... 97
Figure 4.23 Edit Template (Admin/SuperAdmin) ..... 97
Figure 4.24 Admin Activities ..... 98
Figure 4.25 Activity Details ..... 98
Figure 4.26 Activity Log Cleanup ..... 99
Figure 4.27 Document Cleanup ..... 99
Figure 4.28 Document (Users/Admin/SuperAdmin) ..... 100
Figure 4.29 Edit Document (Users/Admin/SuperAdmin) ..... 100
Figure 4.30 View Generated Output Modal (Users/Admin/SuperAdmin) ..... 101
Figure 4.31 Delete Document Confirmation Modal ..... 101
Figure 4.15 Data Flow Diagram Level 0 ..... 109
Figure 4.16 Data Flow Diagram Level 1 ..... 111
Figure 4.17 Entity Relational Diagram ..... 113

Figure 4.18 Flow of the System ..... 115
Figure 4.19 Login ..... 121
Figure 4.20 Sign Up ..... 121
Figure 4.21 Forgot Password ..... 122
Figure 4.22 Reset Password ..... 122
Figure 4.23 SuperAdmin Dashboard ..... 122
Figure 4.24 Personal Profile ..... 123
Figure 4.25 Change Password ..... 123
Figure 4.26 Manage All Users (SuperAdmin/Admin) ..... 123
Figure 4.27 Create New User (SuperAdmin) ..... 124
Figure 4.28 Add New Template (SuperAdmin/Admin) ..... 124
Figure 4.29 Add New Template (SuperAdmin/Admin) ..... 124
Figure 4.30 All Template ..... 125
Figure 4.31 Template Preview ..... 125
Figure 4.32 Template Preview ..... 125
Figure 4.33 Edit Template (SuperAdmin/Admin) ..... 126
Figure 4.34 Edit Template (SuperAdmin/Admin) ..... 126
Figure 4.35 Change Status (SuperAdmin/Admin) ..... 126
Figure 4.36 Delete Template (SuperAdmin/Admin) ..... 127
Figure 4.37 Download Template ..... 127
Figure 4.38 Download Template ..... 127
Figure 4.39 Admin Activity Tracking (SuperAdmin) ..... 128
Figure 4.40 Log Cleanup (SuperAdmin) ..... 128
Figure 4.41 Admin Activity Export to CSV (SuperAdmin) ..... 128
Figure 4.42 Document Cleanup (SuperAdmin) ..... 129
Figure 4.43 Documents ..... 129
Figure 4.44 View Document ..... 129
Figure 4.45 Edit Document ..... 130
Figure 4.46 Delete Document ..... 130
Figure 4.47 User Dashboard ..... 130

Figure 4.48 All Template ..... 131
Figure 4.49 User Role Attribute Defined In Database Migration ..... 132
Figure 4.50 Role Selection Option During User Registration ..... 132
Figure 4.51 Admin Middleware Validating User Role Before Access Is Granted ..... 133
Figure 4.52 Superadmin Middleware Validating User Role Before Access Is Granted ..... 133
Figure 4.53 User Middleware Validating User Role Before Access Is Granted ..... 134
Figure 4.54 User Routes Protected Using Isuser Middleware ..... 134
Figure 4.55 Admin Dashboard Routes Restricted To Admin Role ..... 134
Figure 4.56 Superadmin Routes Restricted To Highest-Level Role ..... 134
Figure 4.57 Input Configuration Panel ..... 137
Figure 4.58 Generate Content Button ..... 137
Figure 4.59 Step 1 And Step 2 ..... 137
Figure 4.60 Step 3 And Step 4 ..... 138
Figure 4.61 Step 4 ..... 138
Figure 4.62 Get AI Suggestion Button (Frontend) ..... 140
Figure 4.63 Event Trigger (Frontend) ..... 140
Figure 4.64 Processing And Prompt Engineering (Backend) ..... 140
Figure 4.65 Processing And Prompt Engineering (Backend) ..... 141
Figure 4.66 Processing And Prompt Engineering (Backend) 2.0 ..... 141
Figure 4.67 Processing And Prompt Engineering (Backend) 3.0 ..... 141
Figure 4.68 User Selection And Input Update ..... 142
Figure 5.1 Client Acceptance Testing with Clients ..... 163
Figure 6.1 WBS 1.0 ..... 168
Figure 6.2 WBS 2.0 ..... 168
Figure 6.3 Gantt Chart 1.0 ..... 171
Figure 6.4 Gantt Chart 2.0 ..... 171
Figure 6.5 Gantt Chart 3.0 ..... 172
Figure 6.6 Gantt Chart 4.0 ..... 172
Figure 6.7 AI Templates ..... 176
Figure 6.8 Student and Lecturer Features ..... 177

Figure 6.9 Sample Output ..... 177

Figure 6.10 AI Interaction with Users ..... 178

# LIST OF TABLES

Table 2.1 Comparison of Existing Project ..... 36

Table 3.1 Functional Requirements for SuperAdmin ..... 46

Table 3.2 Functional Requirements for Admin ..... 48

Table 3.3 Functional Requirements for Users (Lecturer/Student) ..... 49

Table 3.4 Non-Functional Requirements ..... 50

Table 3.5 Hardware Requirements ..... 51

Table 3.6 Software Requirements ..... 52

Table 3.7 Question 1 Interview Analysis ..... 73

Table 3.8 Question 2 Interview Analysis ..... 74

Table 3.9 Question 3 Interview Analysis ..... 74

Table 3.10 Question 4 Interview Analysis ..... 75

Table 3.11 Question 5 Interview Analysis ..... 75

Table 3.12 Question 6 Interview Analysis ..... 76

Table 4.1 Users Data Dictionary ..... 102

Table 4.2 Templates Data Dictionary ..... 104

Table 4.3 Template Input Fields Data Dictionary ..... 105

Table 4.4 Generated Contents Data Dictionary ..... 106

Table 4.5 Admin Activities Data Dictionary ..... 107

Table 5.1 Unit Testing Results (SuperAdmin) ..... 145

Table 5.2 Unit Testing Results (Admin) ..... 147

Table 5.3 Unit Testing Results (Users) ..... 149

Table 5.4 Integration Testing for UPTM Academic AI Assistant Tools ..... 150

Table 5.5 Functional Testing (SuperAdmin) ..... 153

Table 5.6 Functional Testing (Admin) ..... 156

Table 5.7 Functional Testing (Users: Lecturer/Student) ..... 158

Table 5.8 Non-Functional Testing ..... 160

Table 5.9 Client Acceptance Testing ..... 163

Table 5.10 User Acceptance Testing ..... 165

Table 6.1 Risk Management ..... 175

# Acknowledgements

First and foremost, I would like to express my deepest gratitude to Allah S.W.T for His blessings, guidance, and strength throughout the development of this project. Without His will, the completion of this report and system would not have been possible. I would like to extend my heartfelt appreciation to my supervisor, Mr. Muhammad Amin bin Mustapa, for his continuous guidance, valuable feedback, and encouragement throughout the entire process. His advice and support have been instrumental in helping me understand and improve both the technical and conceptual aspects of this project. My sincere thanks also go to the Faculty of Computing and Multimedia (FCOM) and Universiti Poly-Tech Malaysia (UPTM) for providing the resources and platform to complete this project successfully. I am also grateful to Majlis Amanah Rakyat (MARA) for their continuous educational support and opportunity to pursue my studies. Lastly, I would like to express my appreciation to my family and friends for their endless support, patience, and encouragement. Their motivation and understanding have been a constant source of strength and inspiration throughout my journey.

# 1 INTRODUCTION

## 1.1 Introduction

This chapter will provide a brief overview of UPTM Academic AI Assistant Tools. There will be six parts to this chapter, and it involves the project background, problems statements, project objectives, its scopes and target users alongside the whole report outline.

## 1.2 Project Background

The fast evolution of digital technology, especially artificial intelligence (AI), has significantly changed the way people live, study, and work in the modern world. AI technologies continue to reshape communication, knowledge acquisition, and problem-solving processes across various sectors, including education. With the integration of the internet and machine learning algorithms, educators now have access to digital tools that can streamline teaching, automate academic tasks, and enhance learning outcomes (Zipp et al., 2025; USAHS, 2025). These innovations have allowed institutions to adopt AI-driven systems that support efficiency, interactivity, and personalized learning experiences (USAHS, 2025).

A review of recent studies and existing AI platforms reveals both opportunities and challenges in applying AI to education. General-purpose AI tools such as ChatGPT, DeepSeek, and Gemini AI have demonstrated significant capabilities in content generation, reasoning, and personalized assistance. However, their broad and non-specialized nature makes them less effective for academic environments where precision, credibility, and ethical considerations are essential (Stanford HAI, 2025; Yan et al., 2023). Research also highlights common issues including the complexity of crafting effective prompts, over-reliance on unverified outputs, and risks related to data privacy when using third-party cloud systems (FeedbackFruits, 2023; Yan et al., 2023). These limitations suggest the need for more context-specific AI systems that address academic demands directly while ensuring institutional data security.

Within the context of Universiti Poly-Tech Malaysia (UPTM), particularly the Faculty of Computing and Multimedia (FCOM), both lecturers and students are beginning to explore AI tools to improve their academic efficiency. However, challenges such as complex prompt engineering, lack of academic focus, and limited data control remain barriers to effective adoption. In response, this project introduces the UPTM Academic AI Assistant Tools, a web-

based system designed exclusively for FCOM's lecturers and students. It provides built-in academic templates that simplify task execution and ensure AI-generated results remain relevant to academic standards. The system is developed using PHP Laravel framework with HTML, CSS, and Bootstrap for frontend responsiveness. Its scope is limited to internal institutional use, meaning that access will be to UPTM's environment to protect academic data and maintain compliance with university policies. While the system focuses on academic productivity and secure usage, it does not cover broader AI applications such as entertainment or creative writing.

## 1.3 Problem Statement

Although Artificial Intelligence (AI) provides new opportunities for education, most existing tools are not designed specifically for academic use. Users often face challenges such as difficulty in usage, lack of academic-focused features, and concerns about security and reliability. These limitations reduce the effectiveness of AI in supporting teaching, learning, and research.

### 1.3.1 Difficulty in using general AI tools

Many students and lecturers struggle when using general AI tools because these platforms are not straightforward. To get good results, users often need to write very detailed prompts and provide scenarios before the AI produces something useful. This process is not only time consuming but also confusing for users who do not have experience in prompt engineering. For instance, a student who wants to generate a thesis title may need to try several different prompts until the AI gives an answer that sounds academic. This trial-and-error process can be frustrating and may lead to irrelevant or poor-quality results. As a result, instead of saving time, students and lecturers may waste valuable hours trying to fix the AI output, which reduces efficiency in academic work (Whiteman, 2025).

### 1.3.2 Lack of academic focused features

Most AI tools available to the public are designed for general use, not specifically for academic environments. They can generate essays, poems, or creative writing, but they are not tailored to important academic needs. Because of this limitation, lecturers and students often need to manually adjust the AI's responses to meet academic standards, which requires additional editing and checking. For example, a student may find that summaries from AI tools miss key academic concepts. This lack of educational focus reduces productivity and makes the tools less effective for teaching, learning, and research (Dean, 2023).

### 1.3.3 Lack of security and reliability

Public AI tools also bring significant concerns about security, trust, and institutional alignment. When students or lecturers use these tools, sensitive academic data such as research ideas, data institution, or student information may be exposed to third-party platforms. This creates risks of data privacy breaches. In addition, the reliability of outputs from general AI tools is questionable because they may provide incorrect, or unverified information that does not meet university standards. For example, a student relying on an AI-generated may unknowingly receive false references. Institutions also require tools that are consistent and safe for official academic use. Without strong safeguards in place, general AI tools are not suitable for use in universities, as they may compromise both data privacy and academic quality (Farhat, 2025).

## 1.4 Project Objectives

This project is designed to develop an AI assistant tool that directly addresses the needs of academic environments. Its main objectives are to provide ready-to-use academic AI tools through built-in templates, support students and lecturers in completing productive academic tasks efficiently and ensure a secure system that protects sensitive data while remaining institution-friendly.

### 1.4.1 To provide ready-to-use academic AI tools

The first objective of this project is to make academic AI tools simple and ready for immediate use. Instead of requiring students and lecturers to write long and complex prompts, the system will already have built-in academic templates designed by the admin. For example, a student can directly select the tools they want to use and enter a topic, without needing to explain the whole scenario. This approach makes the system much easier to use, especially for those unfamiliar with prompt engineering, and ensures faster and more relevant results.

### 1.4.2 To develop an AI system for productive academic tasks

The second objective is to develop an AI system that enhances productivity in academic writing, teaching, and learning activities. The system primarily focuses on educational contexts, supporting both students and lecturers in completing academic work more efficiently. By automating key processes related to content generation, comprehension, and improvement, the system allows users to save time and concentrate on higher-level thinking and creativity. This contributes to improved academic performance, teaching effectiveness, and overall learning quality.

### 1.4.3 To provide a secure and institution-friendly AI system for education

The final objective is to ensure that the system is secure, reliable, and aligned with institutional needs. Unlike public AI tools, this project will not transmit or store sensitive academic data on external platforms through open APIs. Instead, all data will be stored and managed within the institution's own database, minimizing the risk of unauthorized access. This approach reduces the chances of data breaches and ensures stronger

protection of user information, giving both students and lecturers confidence in using it. For instance, sensitive content such as institutional data can be safely generated and utilized within the system without being exposed outside the institution. This makes the system more institution-friendly, and a safer, more reliable alternative compared to general AI tools.

## 1.5 Scope and Target User

The following sections will detail two distinct scopes, outlining the specific objectives and deliverables associated with each element, as well as identifying the target users involved.

### 1.5.1 Project Scope

The UPTM Academic AI Assistant Tools project is designed to support academic tasks by providing AI-driven features tailored for lecturers and students within the Faculty of Computing and Multimedia (FCOM). The project will focus on developing a web-based platform using PHP with the Laravel framework as the main programming language, supported by HTML, CSS, and Bootstrap for the frontend design. This technology stack ensures a responsive, secure, and user-friendly interface that simplifies interaction between users and the system.

The system will primarily offer ready-to-use academic templates, reduce the complexity of prompt engineering and ensure that outputs are directly relevant to academic needs. One key limitation of this project is that its features are restricted to academic-focused tools and do not cover broader AI applications such as creative writing or entertainment. Additionally, the system will be implemented within the institutional environment of UPTM, which means external access will not be prioritized in the initial development phase. The project is therefore scoped to meet academic productivity needs in a secure and controlled university setting.

### 1.5.2 Product Scope

The UPTM Academic AI Assistant Tools is designed to provide a secure and user-friendly platform specifically for students and lecturers, focusing on enhancing academic productivity and supporting teaching and learning tasks. The system includes features such as the Academic Paragraph Generator, which help students create clear and well-structured research content. For revision and study purposes, tools like the Academic Summary Generator, and Concept Explainer allow students to understand, organize, and compare academic concepts more effectively. Lecturers are supported through features such as Classroom Activity Ideas, Teaching Material Simplifier, Exam Review Notes Creator, and Feedback Generator, which streamline lesson preparation, simplify complex content, and provide structured guidance to students. All academic data is securely stored within UPTM's database, ensuring privacy and compliance with institutional standards. The system emphasizes reliability, ease of use, and targeted academic support rather than replacing general-purpose AI tools, delivering a focused and practical solution for higher education.

### 1.5.3 Target User

The primary users of the UPTM Academic AI Assistant Tools are lecturers and students at Universiti Poly-Tech Malaysia (UPTM), particularly within the Faculty of Computing and Multimedia (FCOM).

Lecturers will benefit from features such as Classroom Activity Ideas, Teaching Material Simplifier, Exam Review Notes Creator, and Feedback Generator, which streamline lesson planning, simplify complex academic content, and provide structured feedback for students. These tools save time and enhance the quality of teaching materials, allowing lecturers to focus more on interactive and student-centered learning.

Students will use features like the Academic Paragraph Generator, Academic Summary Generator, Concept Explainer, and Comparison Assistant to generate clear research content, understand difficult concepts, and organize study materials effectively. This is especially helpful for students with limited experience in AI tools or prompt writing, providing guidance in a structured and accessible way.

The system is designed to accommodate users with varying technical skills, ensuring that even those with minimal digital experience can navigate the platform using pre-built academic templates. By targeting these users, the project ensures that the UPTM Academic AI Assistant Tools meets the specific academic needs of both lecturers and students, improving productivity while safeguarding sensitive institutional data.

## 1.6 Overview of This Report

This report is organized into six main chapters, each focusing on different aspects of the project development process. It provides a systematic explanation of how the project was conceptualized, designed, implemented, and evaluated.

Chapter 1: Introduction presents an overview of the project, including its background, problem statements, objectives, scope, and target users. This chapter helps readers understand the motivation behind developing the system and the intended outcomes.

Chapter 2: Literature Review discusses previous studies, existing systems, and related technologies that guided the development of this project. It also highlights comparisons and justifications for the chosen approach or system features.

Chapter 3: Methodology explains the development approach used throughout the project, such as the Agile methodology. It outlines the phases involved, the requirement-gathering process, and the analytical methods applied to understand user needs and system functions.

Chapter 4: Design describes the overall architecture and system design, including wireframes, database design, entity-relationship diagrams, and system flow. It also covers security considerations and implementation tools used during development.

Chapter 5: Findings focus on the testing process, presenting the results of unit, integration, system, and user acceptance testing. It evaluates how well the system meets its intended requirements and objectives.

Chapter 6: Conclusion provides a summary of the project's achievements, risk management strategies, and reflections on the development process. It also discusses the project's limitations, potential future improvements, and recommendations for further enhancement.

Overall, this structure ensures that each phase of the project from initial research to implementation and evaluation is clearly documented and justified, providing a complete understanding of the system's development lifecycle.

# 2 LITERATURE REVIEW

## 2.1 Introduction

A literature review helps explore previous studies, provide context, and identify gaps for improvement. This review focuses on the use of Artificial Intelligence (AI) in education, particularly in supporting learning, teaching, and academic management. It highlights the benefits and challenges of AI in higher education and underscores the need for a tailored solution like the UPTM Academic AI Assistant Tools, designed to meet the institution's specific needs.

## 2.2 Investigation

The purpose of this investigation is to review related studies and concepts that form the foundation of this project. Since the project focuses on building an Academic AI Assistant system, three key areas of investigation are highlighted: Academic AI Assistant Tools, Prompt Engineering for Academic Use, and Data Privacy and Security in Educational AI Systems. Each area is examined in detail to understand current practices, challenges, and opportunities for improvement. By analyzing these areas, the project ensures that the proposed system addresses real educational needs, improves usability for both lecturers and students, and aligns with institutional values.

### 2.2.1 Academic AI Assistant Tools

Academic AI Assistant Tools are AI-driven platforms designed specifically to enhance teaching, learning, and academic administration within educational environments. These tools aim to provide focused support by automating processes such as lecture summarization, academic writing assistance, study note generation, and explanation of complex concepts. Unlike general-purpose AI systems, which cater to a broad audience, academic-focused tools are tailored to meet the unique requirements of students and lecturers, ensuring that outputs align with learning outcomes and institutional goals (Sajja et al., 2023; López-Galisteo et al., 2025).

One of the main advantages of academic AI assistants is their ability to provide structured outputs that are more relevant and reliable than those generated by generic AI platforms. Research shows that general-purpose tools often produce vague responses that require significant editing before they can be used in academic work

(López-Galisteo et al., 2025; Sajja et al., 2023). This creates additional workload for users instead of reducing it. Academic AI tools, however, integrate pre-designed academic templates, institution-specific guidelines, and quality control features to ensure that the generated information is more useful. For example, platforms developed under instruction-based design approaches highlight the importance of creating AI systems that do not simply provide information but also foster deeper understanding and improve educational efficiency (López-Galisteo et al., 2025).

Furthermore, the adoption of academic AI assistant tools contributes to improving productivity and consistency in both teaching and learning. Lecturers can use them to develop teaching materials more efficiently, while students benefit from simplified content explanations and structured study aids. According to a review of AI tools in 2025, certain AI assistants are increasingly recognized in academic settings as tools supporting institutional missions rather than optional add-ons (TechRadar, 2025). Thus, investigating academic AI assistant tools provides a clear understanding of how they fill the gaps left by general-purpose AI platforms and set the foundation for a more targeted system.

### 2.2.2 Prompt Engineering for Academic Use

Prompt engineering refers to the process of designing and structuring prompts in a way that allows AI systems to generate relevant and meaningful outputs. In the context of education, prompt engineering is crucial because students and lecturers may lack the technical knowledge required to phrase prompts effectively. Research has shown that general AI systems are highly sensitive to how queries are formulated, with vague prompts often leading to irrelevant or misleading results (Chen et al., 2025; Google Research, 2024). This makes prompt engineering a significant barrier for users who want to apply AI tools in academic tasks.

In higher education, this problem becomes more evident because academic tasks require precision, context-awareness, and alignment with scholarly standards. For example, a poorly constructed prompt may generate content that is too general, plagiarized, or not aligned with a given curriculum (Hewing & Leinhos, 2024). To address this, researchers and developers have proposed integrating pre-built academic templates and structured prompt options within AI systems. EDUCAUSE (2023) argues that reducing the complexity of user input can greatly enhance

accessibility, especially for students who may not have strong command of academic language. Instead of requiring detailed instructions, systems can offer guided selections such as “Generate a thesis title,” “Summarize this lecture,” or “Explain this concept in simple terms” (EDUCAUSE, 2023; Stanford University, 2024).

Studies also emphasize that prompt engineering for academic purposes should not only focus on technical accuracy but also on pedagogy. According to Stanford HAI (2023), effective AI integration in education requires prompts that encourage critical thinking and conceptual clarity rather than simple fact retrieval. Similarly, MIT RAISE (2024) highlights that user-friendly prompts can reduce learning anxiety and free up more time for meaningful academic engagement. By investigating prompt engineering in academic use, this project acknowledges that ease of interaction is as important as the quality of AI output. This ensures that the proposed system does not merely provide answers but actively supports students and lecturers in achieving their academic goals.

### 2.2.3 Data Privacy and Security in Educational AI Systems

Data privacy and security are major concerns in the adoption of AI systems in education. Unlike general usage scenarios, academic environments handle sensitive information such as student records, institutional documents, unpublished research, and assessment data. Using external AI platforms raises significant risks since much of this data may be transmitted and stored on third-party servers without adequate protection (Devineni, 2024; EDUCAUSE, 2024). This exposes institutions to potential issues such as unauthorized access, academic misconduct, and breaches of confidentiality. As a result, many universities are cautious in adopting AI technologies without proper data governance strategies.

Research suggests that educational AI systems must prioritize secure data handling to ensure long-term trust and compliance with institutional policies. Secure academic AI should be designed with strict controls such as local data storage, role-based access, and encrypted communication to prevent information leakage (European Data Protection Board, 2025). Some institutions are increasingly adopting “closed AI ecosystems,” where all processing occurs internally rather than via open external APIs, minimizing external exposure and ensuring alignment with institutional values (Parambil et al., 2024).

The importance of security also extends to ethical and legal aspects. Sensitive academic content, such as examination questions or student evaluations, must be safeguarded against misuse or tampering. Failure to protect such data could result in reputational damage for institutions and loss of trust from students and faculty (Devineni, 2024). By investigating data privacy and security in educational AI systems, this project underscores its commitment to developing a platform that is safe, reliable, and institution friendly. This ensures that users can confidently engage with the system without fear of data exploitation, making it a more sustainable solution for higher education compared to public AI platforms.

## 2.3 Related Works

Related works examine existing platforms that integrate artificial intelligence (AI) into academic or educational contexts. The purpose of studying these systems is not only to understand their design and functionality but also to critically evaluate their strengths, weaknesses, and the implications of their use in higher education. By analyzing these tools, this project identifies gaps in current technologies and justifies the development of the proposed Academic AI Assistant Tools. Four main platforms were selected for review: Google Classroom with Gemini AI, ChatGPT by OpenAI, DeepSeek, and Scribbr.

### 2.3.1 Google Classroom (with Gemini AI Tools for Educators)

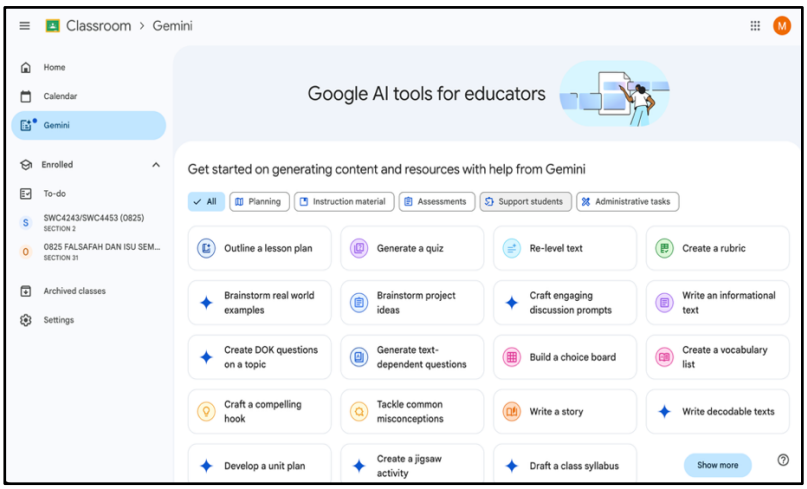



**Figure 2.1:** Google Classroom Gemini Dashboard

Google Classroom is a cloud-based learning management system (LMS) that is part of the Google Workspace for Education ecosystem. It enables lecturers to distribute assignments, share learning materials, and communicate effectively with students in both online and hybrid settings (Google Support, 2025; Google for Education, 2025). The addition of Gemini AI, Google's generative AI tool, expands its capability by integrating artificial intelligence into academic workflows. The main users of this system are lecturers, students, and educational administrators who rely on Google Workspace for teaching, assignment management, and academic collaboration. Gemini AI specifically benefits educators by providing intelligent automation and content generation that simplify daily teaching tasks (Google Workspace with Gemini, 2025; Google Workspace Updates, 2025).

Google Classroom operates through centralized cloud storage and synchronization within the Google ecosystem. Educators create virtual classrooms where they can upload materials, assign work, grade submissions, and communicate with students through comments or announcements (Google Support, 2025; Google Classroom Features, 2025). When Gemini AI is activated, it functions as an assistant within this ecosystem, analyzing classroom content, generating lesson ideas, and suggesting improvements (Google for Education, 2025; Google Workspace with Gemini, 2025). For example, a lecturer can input a lesson topic, and Gemini AI can automatically produce outlines, quizzes, or related questions. However, Gemini AI is not always fully embedded directly into the classroom interface; often it opens a separate workspace where content must be manually imported, which can interrupt workflow continuity and reduce efficiency (Burwood Blog, 2025; Google Workspace Updates, 2025).

Gemini AI introduces several powerful features designed to support educators. These include AI-generated lesson plans, quiz and exam creation, teaching material drafting, summarization of lecture content, and student feedback suggestions (Google for Education, 2025; Google Workspace Updates, 2025). The integration also supports multilingual translation, real-time document collaboration, and automatic grading through Google Forms (Google Support, 2025; Google for Education, 2025). Together, these features streamline academic preparation and assessment, allowing educators to focus more on teaching quality rather than administrative workload. Despite its convenience, many of these AI-driven features are limited to premium users or institutions subscribed to certain Google Workspace editions, reducing accessibility for smaller institutions (Google Workspace Updates, 2025; Google Support, 2025).

Google Classroom and Gemini AI rely heavily on Google Cloud's security infrastructure. Data is encrypted using AES-256 encryption (for data at rest) and TLS (for data in transit) (Google for Education, 2025). Access control mechanisms such as two-factor authentication (2FA) and role-based access management are implemented to safeguard user information (Google for Education, 2025; Google Workspace Privacy Notice, 2025). However, all data including assignments, grades, and AI-generated materials is stored and processed on Google's remote servers. While this ensures reliability and uptime, it also raises privacy concerns because sensitive academic information leaves the institution's jurisdiction (Google Workspace with Gemini FAQ, 2025; Burwood Blog, 2025). This dependence on external servers can lead to compliance challenges with privacy regulations such as GDPR and FERPA, especially

for universities that must retain full control of their data (Google Workspace with Gemini FAQ, 2025; Google for Education, 2025).

The greatest advantage of Google Classroom with Gemini AI lies in its seamless integration within the Google ecosystem, providing familiar tools, real-time collaboration, and automated assistance that enhance teaching efficiency. It reduces administrative workload, encourages interactive learning, and introduces AI-driven content generation that personalizes education. However, the disadvantages are significant: Gemini AI is not fully integrated within the classroom interface, requiring users to switch between platforms, which disrupts productivity. Additionally, data sovereignty and privacy risks arise since all user data is stored externally, beyond institutional control. These weaknesses highlight the need for an institution-controlled AI platform such as the UPTM Academic AI Assistant Tools, which can provide similar AI-driven efficiency while ensuring that all user data and AI-generated outputs are securely stored within the university's own database.

### 2.3.2 ChatGPT (OpenAI)

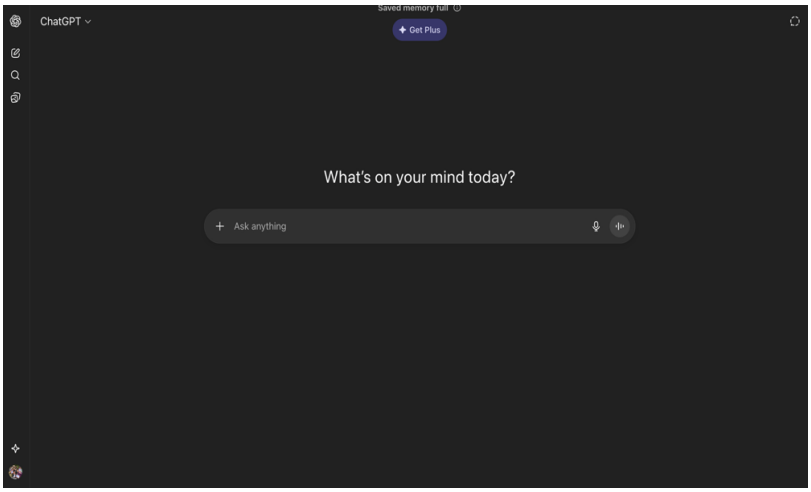



**Figure 2.2:** ChatGPT

ChatGPT, developed by OpenAI, is an advanced conversational AI model based on the Generative Pre-trained Transformer (GPT) architecture. It is one of the most popular AI tools worldwide, used for both academic and general purposes. The system interacts with users through natural language, allowing students, lecturers, and professionals to communicate with it conversationally to obtain information, explanations, or written content. In the academic context, students use ChatGPT to summarize texts, generate essays, or clarify complex theories, while lecturers utilize it to prepare lesson outlines, exam questions, or teaching notes (Yazdani Motlagh et al., 2023).

ChatGPT operates through a large-scale language model trained on massive datasets comprising books, academic papers, and internet sources. When a user submits a prompt, the model processes the text, predicts the most contextually relevant response, and generates coherent answers in real time. This process is powered by deep neural networks optimized for natural language understanding and generation. The model learns relationships between words and meanings, enabling it to produce detailed explanations, paraphrases, or summaries depending on the user's input. ChatGPT's adaptability allows it to function across domains, from general conversation to highly specialized academic fields, although it lacks internal citation verification or fact-checking mechanisms (Zhang et al., 2023).

The tool includes several features that make it versatile in educational environments. These include content generation (essays, summaries, research outlines), code

writing, translation, and question-answering. Advanced versions like GPT-4 also provide improvements in reasoning and domain understanding (Sohail et al., 2023). Through these functions, ChatGPT assists users in writing, brainstorming, and conceptual understanding. However, since it is designed as a general-purpose model, it does not specialize in academic writing conventions, referencing styles, or institutional learning management integration. As a result, outputs may require extensive human revision to meet academic expectations (Buruk, 2023).

From a security perspective, OpenAI applies industry-standard encryption protocols to protect data in transit and at rest (OpenAI, “Business data privacy”, 2025). For instance, ChatGPT Enterprise ensures that conversations are encrypted in transit using TLS and encrypted at rest with AES-256. In enterprise settings, data is not used to train models, and role-based control, domain verification, and other security controls are offered (OpenAI, “Introducing ChatGPT Enterprise”, n.d.). However, even with these protections, all user interactions are processed and stored on OpenAI’s external servers, raising concerns about data privacy, ownership, and compliance with institutional data governance policies (OpenAI, “Security & Privacy”, n.d.). Sensitive academic content such as student work, research drafts, or assessment materials may therefore be exposed to third-party processing, which can be unsuitable for institutions prioritizing data sovereignty and confidentiality.

The main advantages of ChatGPT lie in its accessibility, versatility, and strong language generation ability. It significantly improves productivity, helps users learn new concepts, and accelerates the writing process. However, the disadvantages are notable: ChatGPT sometimes produces inaccurate or biased responses due to limited fact-verification (Motlagh et al., 2023), struggles with consistent citation formatting, and lacks institutional-level data control. These limitations make it risky for academic environments that require high data privacy and adherence to educational standards. Consequently, the development of the UPTM Academic AI Assistant Tools is inspired by ChatGPT’s conversational efficiency but designed to overcome its weaknesses by offering localized academic templates, verified reference integration, and internal data storage managed entirely within the institution’s own secure database.

### 2.3.3 DeepSeek

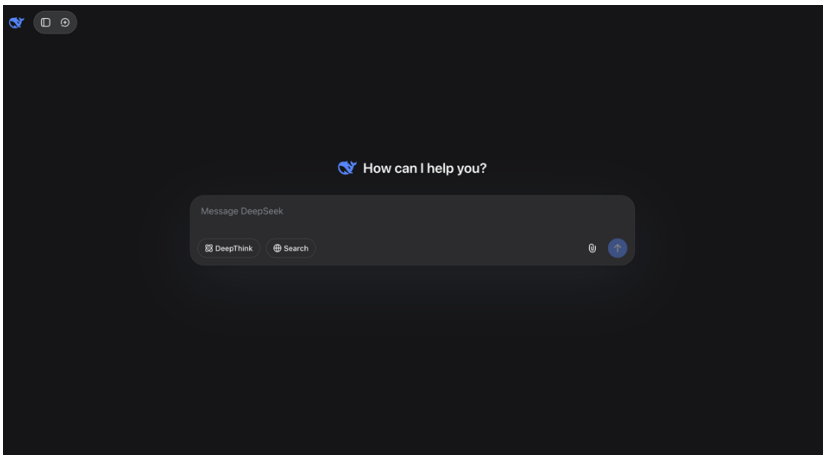



**Figure 2.3:** DeepSeek

DeepSeek is a generative artificial intelligence model developed by a Chinese research team that attracted global attention in early 2025 for combining high reasoning performance with relatively low cost (Nature, 2025; Reuters, 2025). It was primarily designed to serve students, researchers, and professionals who need AI support in academic, technical, and research-related tasks. Its main users include those working in science, technology, engineering, and mathematics (STEM) fields, where structured problem-solving and coding capabilities are essential (Nature, 2025). The platform gained popularity due to its ability to deliver strong performance while maintaining competitiveness in pricing.

DeepSeek's evolution is documented in recent announcements: for example, DeepSeek released an upgraded model (V3) with support for domestic chip architectures and efficiency improvements (Reuters, 2025). The model's design is oriented toward reasoning tasks and cost-effective inference (Euronews, 2025). However, technical details such as mixture-of-experts (MoE) architectures or performance tradeoffs beyond what's publicly shared remain less validated in accessible literature.

The system includes several features suited for technical purposes, such as reasoning support, code generation, and domain-specific analysis (Scientific American, 2025). While some sources describe DeepSeek's focus on logical precision and cost-efficiency, explicit claims about image generation modules, multilingual support, or integration with academic systems are less well documented in public sources.

From a security and governance standpoint, DeepSeek's data infrastructure being located in China has raised concerns about data sovereignty, transparency, and compliance with international regulations (Euronews, 2025; Reuters, 2025; SCMP, 2025). A German data protection official urged app store removals over unlawful data transfer to China (Euronews, 2025). Moreover, CAISI (a U.S. standards body) evaluated DeepSeek's models and found shortcomings in performance, security, and adoption compared to leading U.S. models (NIST CAISI, 2025). These issues underscore risks around privacy, auditability, and institutional control.

Analyzing DeepSeek's advantages and disadvantages, observe that its strengths include relative affordability, strong reasoning capability, and a challenge to the dominance of Western AI models. But its weaknesses lack of transparency, governance uncertainty, privacy risks, and limited institutional control are substantial. These challenges affirm the motivation behind developing an institutional AI assistant (like UPTM's) that provides similar effectiveness while retaining full data control and compliance.

### 2.3.4 Scribbr

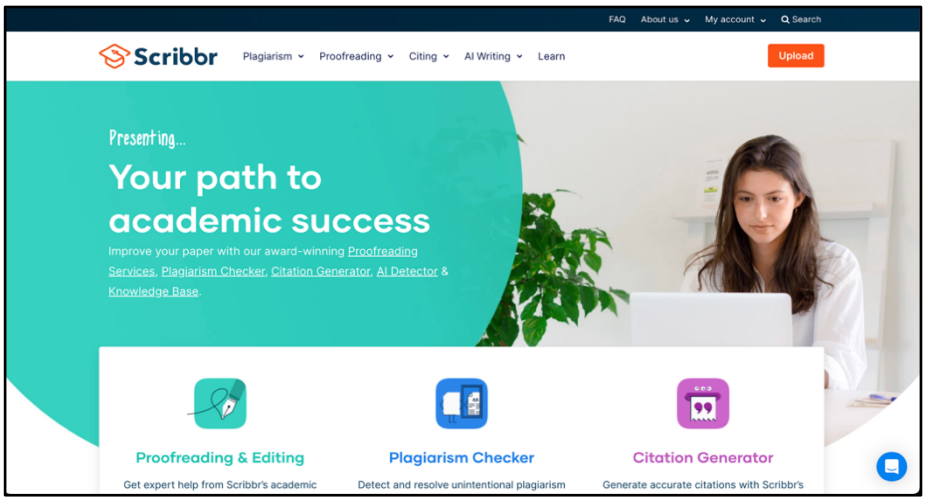



**Figure 2.4:** Scribbr

Scribbr is an academic tool developed to help students, researchers, and educators maintain academic integrity by ensuring their written work follows proper citation and originality standards. It is mainly used by university students to check for plagiarism, generate accurate citations, and improve the overall credibility of their assignments and research papers. The platform is widely recognized in academic environments for promoting honest writing practices and helping users comply with institutional referencing guidelines (Scribbr, 2024a).

Scribbr operates through a cloud-based system that allows users to upload their documents directly to its web platform. Once uploaded, Scribbr's algorithm scans the text against a large database containing academic journals, publications, websites, and previously submitted student papers. The system highlights matched content, provides similarity percentages, and identifies areas that require citation or rewriting. This process helps users detect unintentional plagiarism before submitting their work, ensuring originality and proper source acknowledgment (Scribbr, 2024b).

The main features of Scribbr include a plagiarism checker, citation generator, and proofreading services. The plagiarism checker compares text using advanced similarity detection, while the citation generator supports multiple styles such as APA, MLA, and Chicago, making it easier for students to cite accurately (Scribbr, 2024c). In addition, Scribbr offers an automated grammar and structure check to enhance writing clarity and academic tone. These integrated tools simplify the writing process,

especially for students who are still learning how to format and reference their academic work correctly (Scribbr, 2024d).

In terms of security, Scribbr uses encrypted data transfer to protect uploaded documents and ensures they are stored in a secure database. It employs HTTPS and SSL encryption to safeguard user data from unauthorized access during submission and analysis. However, users have limited control over how long their documents remain stored on Scribbr's servers, as the company retains submissions to improve its plagiarism detection database. Although this improves accuracy for future users, the unclear data retention policy raises some privacy concerns for sensitive academic materials (Scribbr, 2024e).

The advantages of Scribbr include its high accuracy in plagiarism detection, support for multiple citation styles, and user-friendly interface, which make it suitable for academic writing. It effectively helps students maintain integrity and meet institutional writing standards. However, its disadvantages include a limited range of functions compared to broader AI tools, and uncertainty regarding data retention policies. While Scribbr performs well in its specialized domain, it lacks broader writing assistance capabilities such as paraphrasing or conceptual explanations, leading users to rely on multiple platforms for complete academic support.

## 2.4 Comparison

This section presents a comparison between existing academic AI-related platforms which is Google Classroom, ChatGPT, DeepSeek, and Scribbr against the proposed UPTM Academic AI Assistant Tools. The comparison is based on several criteria including features, users, navigation, security elements, scalability, and cost-effectiveness. This analysis highlights the strengths and weaknesses of current systems and shows how the proposed solution addresses gaps to better suit UPTM's academic environment.

**Table 2.1**: Comparison of Existing Project

| Criteria | Google Classroom (with Gemini AI) | ChatGPT (OpenAI) | DeepSeek | Scribbr | UPTM Academic AI Assistant Tools |
|---|---|---|---|---|---|
| **Features** | Classroom management, assignment submission, Gemini AI for lesson/quiz generation | Conversational AI, text generation, academic Q&A, content drafting | Affordable AI chatbot, strong reasoning in STEM, open-source | Plagiarism detection, citation generation (APA, MLA, Chicago) | Academic-focused AI assistant, smart Q&A, learning support, and secure academic data handling |
| **Users** | Lecturers and students in institutions using Google Workspace | General public, students, lecturers, professionals | Students, researchers, and general users (mostly in Asia) | Students writing academic papers | UPTM lecturers, students, and academic staff/admin |

| | | | | | |
|---|---|---|---|---|---|
| **Navigation (Ease of Use)** | Seamless integration with Google tools, but AI sometimes redirects outside workflow | Easy-to-use chat interface, but may generate non-academic or irrelevant content | Simple interface, but limited localization and support issues | Straightforward, but limited to plagiarism and referencing functions | User-friendly dashboard, responsive design, direct academic workflow integration |
| **Security Element** | Data stored on Google servers, raising privacy concerns for sensitive academic work | Data processed on OpenAI servers risk in handling confidential materials | Security concerns due to regional restrictions and data privacy issues | Secure plagiarism database but limited coverage outside referencing | Secure database hosted under UPTM control, complies with institutional privacy and data policies |
| **Scalability** | Scales well for global education institutions | Highly scalable via cloud but not tailored for institutions | Scales at low cost but limited governance in some regions | Limited scalability as tool is function-specific | Designed to scale within UPTM and potentially other academic institutions |
| **Cost-effectiveness** | Free basic plan, premium features require institutional licenses | Free and subscription-based | Very affordable, open-source alternative | Paid service per plagiarism/citation check | Cost-effective as institution-owned, API purchased and integrated by UPTM |

## 2.5 Discussion

Based on the comparison in the previous section, the development of the UPTM Academic AI Assistant Tools has been carefully designed by adapting selected features and security mechanisms from existing systems. Among the reviewed platforms, the most relevant references were Google Classroom with Gemini AI, ChatGPT by OpenAI, Scribbr, and DeepSeek. These systems demonstrated strengths in supporting academic workflows but also revealed weaknesses that became the main trigger for creating features in the proposed system.

From ChatGPT, the system adopts the core idea of AI-powered content generation. Features such as the Thesis Title Generator, Academic Paragraph Generator, Academic Summary Generator, Concept Explainer, and Comparison Assistant are all inspired by ChatGPT's natural language processing capabilities. The trigger here is that while ChatGPT can generate a wide variety of content, it often lacks academic focus and requires users to craft long and complex prompts to get usable results. This limitation motivated the creation of ready-to-use academic templates in the UPTM system, which reduce the burden of prompt engineering and ensure that outputs remain aligned with academic standards. By narrowing the scope, the system reduces the risks of irrelevant or incorrect answers that are common in general-purpose AI models.

From Google Classroom with Gemini AI, the project adopts features that support lecturers in teaching activities. These include Classroom Activity Ideas, Teaching Material Simplifier, Exam Review Notes Creator, and Feedback Generator. The reason for adopting these features is that lecturers often struggle to prepare engaging activities and simplified learning materials within limited timeframes. Although Gemini AI provides such assistance, it functions as a separate add-on rather than being fully embedded in the teaching workflow. This gap triggered the decision to integrate similar features directly into the UPTM Academic AI Assistant Tools, making the platform more seamless and practical for lecturers when preparing academic resources.

From Scribbr, the project adopts the principle of academic integrity and secure text handling. The inclusion of the Academic Proofreader reflects Scribbr's strength in maintaining scholarly quality through grammar checking and plagiarism detection. However, Scribbr is limited because it mainly focuses on proofreading and referencing without offering broader AI-powered academic support. This limitation triggered the decision to adapt only the integrity

aspect from Scribbr, while expanding functionality to include proofreading alongside summarization, generation, and teaching support. At the same time, privacy and security mechanisms are emphasized to ensure that sensitive documents such as assignments, theses, and feedback are not exposed to third parties. This ensures that the UPTM system remains both academically reliable and institutionally safe.

From DeepSeek, the project adopts two main principles: efficiency and data security awareness. DeepSeek gained attention for its resource-efficient architecture and strong reasoning capabilities, especially in STEM-related tasks. However, the system also raised major concerns regarding data privacy, regional data governance, and ethical transparency, as user information could potentially be exposed to third-party access or regional authorities. These issues highlight a critical gap in data ownership and user trust. To address this, the UPTM Academic AI Assistant Tools will implement a self-hosted security architecture where all AI interactions user prompts and generated outputs are securely stored within UPTM's own database. No external API calls will be made for sensitive data, ensuring full control over institutional information. This approach not only resolves the data privacy problems seen in DeepSeek but also strengthens user confidence in system reliability and compliance with academic ethics.

## 2.6 Conclusion

In conclusion, the literature review demonstrates that while numerous AI-based academic platforms such as Google Classroom with Gemini AI, ChatGPT, DeepSeek, and Scribbr have significantly contributed to the modernization of education, none fully address the comprehensive academic needs of both lecturers and students in a secure institutional environment. Each reviewed system presents notable strengths Google Classroom excels in structured learning management, ChatGPT offers powerful content generation through natural language understanding, DeepSeek showcases computational reasoning and cost efficiency, while Scribbr upholds academic integrity through plagiarism detection and citation management. However, their collective limitations including data privacy concerns, limited academic specialization, external dependency, and fragmented functionality reveal the need for a more integrated and institution-controlled solution tailored to local educational contexts such as UPTM.

The investigation into academic AI assistant tools, prompt engineering, and data security further reinforces the importance of designing a system that is both pedagogically effective and technically safe. Existing tools often focus on individual functions, leaving gaps between writing assistance, learning enhancement, and institutional compliance. Moreover, the reliance of most AI systems on third-party cloud storage introduces potential risks in data sovereignty and ethical management, which are particularly sensitive in academic environments. Addressing these issues requires not just technological advancement, but also ethical and contextual awareness to ensure AI is implemented responsibly within higher education.

Therefore, the development of the UPTM Academic AI Assistant Tools represents a strategic step toward bridging these gaps by combining the strengths of existing systems while mitigating their weaknesses. The proposed system emphasizes academic relevance through pre-designed templates, enhanced usability via structured prompt interfaces, and institutional data security through internal hosting within UPTM's infrastructure. It is envisioned not merely as a convenience tool but as an academic innovation that aligns with educational integrity, digital ethics, and institutional governance. Ultimately, this review justifies the necessity of a dedicated academic AI platform that empowers both lecturers and students to engage in smarter, safer, and more meaningful academic work within a controlled and trusted environment.

# 3 METHODOLOGY

## 3.1 Introduction

This chapter discusses the methodology adopted for developing the UPTM Academic AI Assistant Tools. A suitable methodology is essential to ensure the project is completed efficiently and meets user expectations. The methodology defines the framework, approach, and structured steps taken throughout the development cycle. In software development, the chosen methodology determines how requirements are gathered, features are implemented, and testing is conducted (Pressman & Maxim, 2020).

For this project, the Agile methodology was selected because it promotes flexibility, user collaboration, and iterative improvement. Agile allows continuous interaction between developers, supervisors, and users, ensuring that every feature is developed, tested, and refined in manageable sprints. This approach supports continuous feedback and aligns with the dynamic academic requirements of the system (Highsmith, 2021).

## 3.2 Agile Methodology

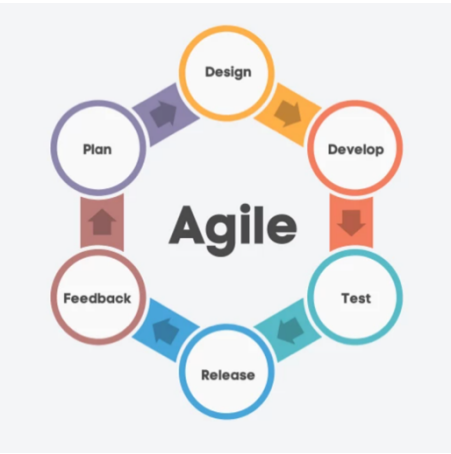



**Figure 3.1:** Agile Methodology Diagram (Hari, 2025)

The Agile methodology was chosen for this project due to its adaptability and user-centered focus. Unlike traditional methods such as the Waterfall model, Agile emphasizes iterative progress, continuous testing, and close collaboration between the development team, supervisors, and potential users (students and lecturers). This ensures that the system

remains flexible and can be modified based on real feedback from stakeholders (Beck et al., 2001).

Agile development follows a cyclical process where each module is planned, designed, developed, tested, and reviewed in short iterations known as sprints. Each sprint produces a working version of the system that can be tested by users. This continuous feedback loop helps to identify problems early, improve user satisfaction, and deliver a more effective and reliable final system.

For the UPTM Academic AI Assistant Tools, Agile is ideal because it involves frequent interaction with academic supervisors and potential users. These interactions ensure that each function meets the actual academic requirements. The project will undergo multiple sprints where each phase planning, designing, developing, testing, deployment, and review will result in measurable deliverables and incremental system improvement.

## 3.3 Phases in Agile Methodology

The Agile development cycle for this project is divided into six main phases: Plan, Design, Develop, Test, Deploy, and Review. Each phase has specific objectives, activities, and deliverables that contribute to the continuous enhancement of the system.

### 3.3.1 Phase 1: Plan

The planning phase serves as the foundation of the Agile development cycle and plays a crucial role in ensuring the success of the UPTM Academic AI Assistant Tools. During this phase, the project objectives, scope, and deliverables are carefully defined to ensure that the system meets the needs of its intended users lecturers and students at Universiti Poly-Tech Malaysia (UPTM), specifically within the Faculty of Computing and Multimedia (FCOM). The phase begins with identifying key academic challenges faced when using general-purpose AI tools, such as difficulty in produce quality academic content, complex prompt requirements, and lack of data security. Through discussions with supervisors, lecturers, and students, the project requirements are refined, and the system features are prioritized based on their importance and practicality. Additionally, the appropriate technology stack is selected to ensure smooth development and deployment. PHP with the Laravel framework for the backend, MySQL for database management, and HTML, CSS, and Bootstrap for the frontend

interface. The planning phase also involves preparing detailed documentation including the problem statement, project proposal, system requirements specification, and a review of related studies. These deliverables provide a solid foundation for the following phases, ensuring that every development step aligns with the academic and institutional goals of UPTM.

### 3.3.2 Phase 2: Design

The design phase is one of the most critical stages in the Agile development process, as it translates the project requirements gathered during the planning stage into structured system blueprints. For the UPTM Academic AI Assistant Tools, this phase focuses on transforming conceptual ideas into a clear, visual, and technical representation of how the system will function. The first step involves creating a wireframe, which outlines the layout and structure of the user interface (UI). The wireframe serves as a visual guide showing how lecturers and students will interact with the system, including navigation flow, main dashboard elements, and AI feature accessibility. After establishing the UI structure, a Data Dictionary is developed to define each data element used in the system such as user information, AI-generated responses, and session history to ensure consistency, accuracy, and clarity across all modules. Next, a Data Flow Diagram (DFD) is designed to illustrate how data moves between users, processes, and the database, providing a clear understanding of how information is stored, retrieved, and processed. Alongside the DFD, an Entity Relationship Diagram (ERD) is constructed to model the database structure, defining the relationships between key entities such as users, logs, and AI interaction records. Furthermore, a Business Process Model and Notation (BPMN) diagram is developed to represent the overall system workflow, showing how lecturers and students engage with various features in a sequence of logical steps. This phase ensures that both the technical and visual aspects of the system are well-documented and aligned with user expectations. By completing these design artifacts, the development team gains a comprehensive understanding of how to implement the system efficiently in the next phase, reducing potential errors and improving overall project quality.

### 3.3.3 Phase 3: Develop

The development phase is where the actual implementation of the UPTM Academic AI Assistant Tools begins. This stage involves converting the design specifications into a fully functional system through iterative sprints, following the Agile methodology. Each sprint focuses on developing specific system modules, starting with fundamental components such as user registration, login authentication, and role-based access control. Once the basic structure is complete, more advanced features like academic AI template generation, thesis title suggestion, and study notes summarization are developed. Laravel is used to build the backend framework, providing secure handling of API requests and efficient database communication through MySQL. The frontend is implemented using HTML, CSS, and Bootstrap to create a responsive and user-friendly interface. Throughout the development phase, coding standards and version control practices are strictly followed to maintain system stability and collaboration efficiency. Continuous feedback from supervisors and users ensures that the system evolves according to academic requirements and user preferences. Each sprint ends with a mini review to test functionality before moving on to the next module. This iterative process helps to detect and fix issues early while maintaining flexibility and improving system performance progressively.

### 3.3.4 Phase 4: Test

The testing phase is one of the most essential components in the Agile development cycle, as it ensures that the UPTM Academic AI Assistant Tools functions correctly, securely, and efficiently before deployment. This phase aims to identify and fix any errors, bugs, or performance issues that may arise during system operation. The process begins with testing preparation testing**,** where test cases are planned and documented based on the system requirements and design specifications. This step ensures that each feature is verified against its intended functionality. The first level of testing conducted is Unit Testing**,** which focuses on testing individual components or modules of the system, such as user authentication, AI query processing, and data retrieval functions. This allows early detection of minor coding issues within each function. Once individual units are verified, Integration Testing is performed to ensure that different modules such as the frontend, backend, and database work together seamlessly. This step checks data communication and interaction between modules to prevent logical errors or data mismatch problems. After successful integration, System Testing is conducted to validate the complete system. During this stage, all

functionalities are tested in an environment like the actual deployment, ensuring that the system meets both functional and non-functional requirements such as performance, usability, and security. Following system testing, Acceptance Testing is carried out to evaluate the system's readiness for real use. It consists of two sub-stages, Client Acceptance Testing, where project supervisors or evaluators review the system to ensure it meets project objectives and institutional standards, and User Acceptance Testing, which involves lecturers and students testing the system in real scenarios to confirm that it fulfills their academic needs effectively. Feedback collected from users is analyzed to make final improvements before deployment. Overall, the testing phase ensures the system's reliability, usability, and performance, guaranteeing that it is ready for real-world implementation with minimal risk of failure.

### 3.3.5 Phase 5: Deploy

Once testing is successfully completed, the project proceeds to the deployment phase. This phase involves preparing the system for real-world use by lecturers and students within UPTM. The UPTM Academic AI Assistant Tools will be deployed in a secure and controlled environment, preferably hosted on the university's internal server or a restricted-access cloud platform to maintain data privacy and prevent unauthorized access. Deployment activities include setting up the production database, migrating all necessary configurations, and integrating the AI model securely within the system. API connections are configured to ensure they function properly while maintaining strict security to prevent external exposure. During deployment, performance testing is conducted again to ensure that the system operates efficiently under real usage conditions, especially when multiple users access it simultaneously. Documentation such as deployment logs and user manuals are also prepared to assist administrators and users in operating the system effectively. The success of this phase ensures that the system is fully operational, accessible to the intended users, and capable of supporting academic activities in a secure and efficient manner.

### 3.3.6 Phase 6: Review

The review phase is the final stage of the Agile development cycle, focusing on evaluating system performance and identifying areas for improvement. After the UPTM Academic AI Assistant Tools has been deployed, feedback is collected from supervisors, lecturers, and students who actively use the system. This phase involves

analyzing user feedback, performance metrics, and potential issues encountered during usage. The collected insights are used to refine the system and plan for future updates or additional features. The review process ensures continuous improvement and aligns the system with the evolving needs of the academic community. Additionally, this phase assesses the project's overall success in meeting its objectives such as improving academic productivity, ensuring data security, and simplifying AI usage for educational purposes. Based on the feedback, enhancement logs and revision plans are prepared for subsequent development cycles. This reflective phase reinforces the Agile principle of continuous learning and adaptation, ensuring that the UPTM Academic AI Assistant Tools remains effective, reliable, and relevant in supporting the academic ecosystem of UPTM.

## 3.4 Requirement

The requirement phase plays a vital role in understanding the needs and expectations of users to ensure that the system is developed according to real academic demands. In this project, the process of gathering requirements for the UPTM Academic AI Assistant Tools focused on identifying the academic challenges faced by both lecturers and students, as well as the features they expect from an institutional AI-based platform. The primary aim of this phase was to collect reliable information that would guide the design and development process, ensuring that the final product supports the teaching, learning, and research needs within Universiti Poly-Tech Malaysia (UPTM). To achieve this, two main data collection techniques were used a questionnaire survey and an interview which provided both quantitative and qualitative insights into user requirements. The feedback gathered helped define the system's functional, non-functional, and system requirements, which serve as the blueprint for system implementation and testing.

### 3.4.1 Data Gathering Techniques

In this project, two data gathering techniques were applied Questionnaire Analysis and Interview Analysis**.** The questionnaire was distributed to both lecturers and students to obtain a wide range of feedback and opinions. It aimed to identify their main challenges when using general AI tools in academic tasks. The questionnaire also explored users preferred academic AI features and their expectations regarding security**,** privacy**,** and reliability of data. Questions were designed in multiple-choice and likert scale formats

to capture data and personal opinions. This approach to analyze trends and preferences among different user groups.

Meanwhile, the interview method provided a deeper understanding of the lecturer's experience, perspectives, and expectations regarding the use of AI in academic environments. The interview was conducted with a lecturer who teaches Artificial Intelligence (AI), focusing on several key discussion areas. The conversation began with exploring the lecturer's general experience with AI tools and their understanding of academic AI assistant platforms. The lecturer also shared the types of tools currently used to support academic tasks, along with examples of how these tools impact teaching and learning. During the session, the proposed system's features including academic AI templates were introduced and discussed. The lecturer was asked for feedback on whether these features align with academic needs and how they could be improved for better usability and productivity. The discussion concluded with suggestions for future improvements. Overall, the interview findings were valuable in refining the system's features and ensuring that its design not only meets real user expectations but also enhances academic efficiency.

### 3.4.2 Functional Requirement

Functional requirements define the specific operations and behaviors that the UPTM Academic AI Assistant Tools must perform to meet the users' needs. The system consists of three main user roles: Superadmin, Admin and User (Lecturer/Student). Each role has different access levels and capabilities, as shown below.

**Table 3.1**: Functional Requirements for SuperAdmin

| No | Functional Requirement | Description |
|---|---|---|
| 1. | Login | Allows superadmin to securely access the system using authentication credentials. |
| 2. | Sign Up | Enables superadmin to register an account using authorized institutional credentials. |

| | | |
|---|---|---|
| 3. | Manage Account | Superadmin can manage their own account settings. |
| 4. | Update Profile | Allows superadmin to update personal information such as name and email. |
| 5. | Change Password | Enables superadmin to change or reset passwords to maintain security. |
| 6. | Manage All Users | Superadmin can view all admin, lecturer, and student accounts details, update user status (active/inactive), and delete user accounts if necessary. |
| 7. | Add New User | Superadmin can add new users to the system and assign roles such as admin, lecturer, or student. |
| 8. | Templates Management | Superadmin can create, update, delete, and control the visibility of AI templates. |
| 9. | View All Templates | Allows superadmin to view all templates available in the system. |
| 10. | Admin Activity Tracking | Enables tracking of admin activities related to template modifications for monitoring and accountability purposes and log cleanup to remove old logs of administrator activity in the database if necessary. |
| 11. | Document Cleanup | Superadmin can delete old user-generated documents to optimize database storage. |
| 12. | Document Management | Superadmin can view and manage the history of documents generated by the AI, allowing tracking and quality assurance. |

| | | |
|---|---|---|
| 13. | Edit/Delete Document | Superadmin can modify or remove generated document. |

**Table 3.2**: Functional Requirements for Admin

| No | Functional Requirement | Description |
|---|---|---|
| 14. | Login | Allows admin to access the system using secure authentication credentials. |
| 15. | Sign Up | Enables admin to register a new account using institutional credentials. |
| 16. | Manage Account | Admin can manage all user accounts including viewing details and deleting unauthorized users. |
| 17. | Update Profile | Admin can update personal information such as name, email, or contact details. |
| 18. | Change Password | Admin can update or reset passwords to maintain account security. |
| 19. | Templates Management | Admin can create, update, delete, and control the visibility of AI templates. |
| 20. | View All Templates | Admin can view all templates available in the system to ensure they meet academic standards. |
| 21. | Document Management | Admin can view and manage the history of documents generated by the AI, allowing tracking and quality assurance. |
| 22. | Edit/Delete Document | Admin can modify or remove generated document. |

**Table 3.3** Functional Requirements for Users (Lecturer/Student)

| No | Functional Requirement | Description |
|---|---|---|
| 1. | Login | Allows lecturers and students to access the system securely using their registered credentials. |
| 2. | Sign Up | Enables lecturers and students to create an account within the institution's environment. |
| 3. | Manage Account | Users can update their personal profile information to keep their account details up to date. |
| 4. | Change Password | Users can reset or change their passwords to maintain account privacy. |
| 5. | View Account Details | Users can view their personal details stored in the system. |
| 6. | Use Template Provided by Admin | Users can select and use academic AI templates created by the admin for generating academic content. |
| 7. | Document Management | Users can view their document history, including outputs generated by the AI assistant. |
| 8. | Edit/Delete Document | Users can modify or delete previously generated documents to manage their workspace efficiently. |

### 3.4.3 Non-Function Requirement

Non-functional requirements define the quality attributes that describe how the system operates rather than what it does. These requirements ensure that the UPTM Academic AI Assistant Tools performs efficiently, securely, and reliably for all users.

**Table 3.4**: Non-Functional Requirements

| No | Category | Description |
|---|---|---|
| 1. | Performance | The system should generate AI-based academic responses efficiently within seconds, ensuring smooth performance even when accessed by multiple users simultaneously. Optimization of backend API calls and database queries will support this goal. |
| 2. | Usability | Designed with simplicity and clarity, the user interface ensures that both lecturers and students can operate the system easily without requiring advanced technical skills. Clear navigation and minimal steps enhance the user experience. |
| 3. | Reliability | The system is expected to maintain stable performance with minimal downtime. Regular maintenance schedules and backup strategies will be implemented to ensure system consistency and dependability. |
| 4. | Security | Security is prioritized through authentication, role-based access control, and encrypted communication (HTTPS). These measures prevent unauthorized access and protect academic data from leaks or misuse. |
| 5. | Privacy | Sensitive user and information data will be stored securely within the institutional server, preventing external data exposure. User-generated content will not be shared with third-party services. |

### 3.4.4 System Requirement

The development of this system requires both hardware and software resources to ensure smooth implementation, testing, and deployment. The following section outlines the detailed hardware and software requirements necessary for the system's operation. The project is developed using a MacBook Air M1, which provides the necessary performance and compatibility for both frontend and backend development tasks.

#### 3.4.4.1 Hardware Requirement

The project is developed using a MacBook Air M1, which provides the necessary performance and compatibility for both frontend and backend development tasks.

**Table 3.5**: Hardware Requirements

| Hardware Component | Specification / Description |
|---|---|
| Processor | Apple M1 Chip with 8-core CPU |
| Memory (RAM) | 8 GB Unified Memory |
| Storage | 256 GB SSD |
| Display | 13.3-inch Retina Display |
| Operating System | macOS Monterey or later |
| Internet Connection | Required for API access (ChatGPT API, GitHub version control) |

This setup provides a stable and efficient environment for code compilation, local server setup, and browser-based testing.

#### 3.4.4.2 Software Requirement

The system integrates multiple technologies to support both frontend and backend development. These tools and technologies are essential for system functionality, testing, and version management.

**Table 3.6**: Software Requirements

| Category | Software / Tools | Description / Purpose |
|---|---|---|
| Frontend Technologies | HTML, CSS, Bootstrap | Used to design and style the user interface, ensuring responsiveness and accessibility across devices. |
| Backend Technologies | PHP, Laravel Framework | PHP handles server-side logic and data interaction, while Laravel provides a structured MVC framework for routing, authentication, and database management. |
| API Integration | OpenAI API | Enables AI-powered responses and intelligent generation features within the system. |
| Development Tools | Visual Studio Code, MAMP, GitHub | VS Code serves as the primary code editor, MAMP is used to host the local server environment for PHP and MySQL, GitHub provides version control for backup. |
| Testing Tool | Postman | Used to test API endpoints (GET, POST, PUT, DELETE) to verify correct system responses and ensure stable API integration. |
| Browser | Google Chrome | Used to test and run the web application during the development and debugging phases. |

## 3.5 Analysis

The analysis phase is essential for understanding user needs and defining the system structure before moving into design and development. It focuses on analyzing the data collected through questionnaires and interviews to identify challenges, preferences, and requirements related to academic AI usage. From this analysis, several system models such as the Use Case Model, BPMN (Business Process Modelling Notation), and Entity Relationship Diagram (ERD) are developed to visualize the system's flow, user interactions, and data structure. This phase ensures that the UPTM Academic AI Assistant Tools are designed based on user-centered information, promoting a more reliable and efficient development process.

### 3.5.1 Data Gathering Analysis

The data gathering analysis focuses on interpreting information obtained through two main techniques, questionnaires and interviews. The questionnaire was distributed to both lecturers and students to understand their challenges when using AI tools, their preferred academic AI features, and their views on security and reliability. Meanwhile, the interview was conducted with a lecturer who teaches Artificial Intelligence (AI) to gain deeper insights into user expectations and system improvements. Together, these methods provided a clear understanding of user requirements, which served as the foundation for defining both functional and non-functional system requirements.

#### 3.5.1.1 Questionnaire Analysis

The questionnaire analysis was carried out to collect quantitative data from students and lecturers regarding their experience and perception of using Artificial Intelligence (AI) tools for academic purposes. The objective of this data collection was to identify the level of AI tool usage, challenges faced without AI assistance, and opinions on the proposed system features. The responses gathered provided useful insights into user needs and expectations, which will help guide the system's design and functionality to ensure it effectively supports academic tasks.

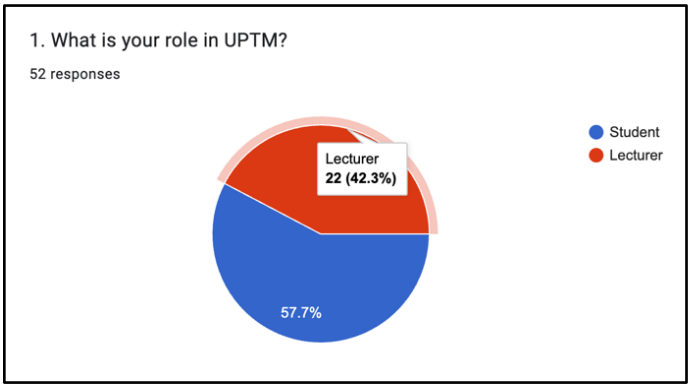



**Figure 3.2:** Questionnaire Question 1

Question 1: What is your role in UPTM?

The majority of respondents were students, accounting for 57.7%, while lecturers made up 42.3% of the total participants. This balanced distribution provides a strong foundation for comparing perspectives between both roles, ensuring that the analysis reflects the real academic environment within UPTM from both the learner's and educator's viewpoints.

The higher number of student respondents is expected since students represent the largest user group of the proposed UPTM Academic AI Assistant Tools. Their responses are particularly important for evaluating how the system can support learning tasks such as summarizing lecture notes, simplifying complex theories, and generating revision materials.

Meanwhile, the significant participation of lecturers (42.3%) adds valuable insights into how the system could also benefit teaching-related tasks, such as preparing classroom activities, providing feedback, and creating teaching materials efficiently. Having both roles represented allows for more reliable analysis in later questions such as challenges, feature preferences, and security expectations, where differences between students and lecturers may highlight how the system can serve each user type effectively.

| Row Labels | Lecturer | Student | Grand Total |
|---|---|---|---|
| Daily | 14 | 22 | 36 |
| Weekly | 8 | 8 | 16 |
| Grand Total | 22 | 30 | 52 |

**Figure 3.3:** Questionnaire Question 2

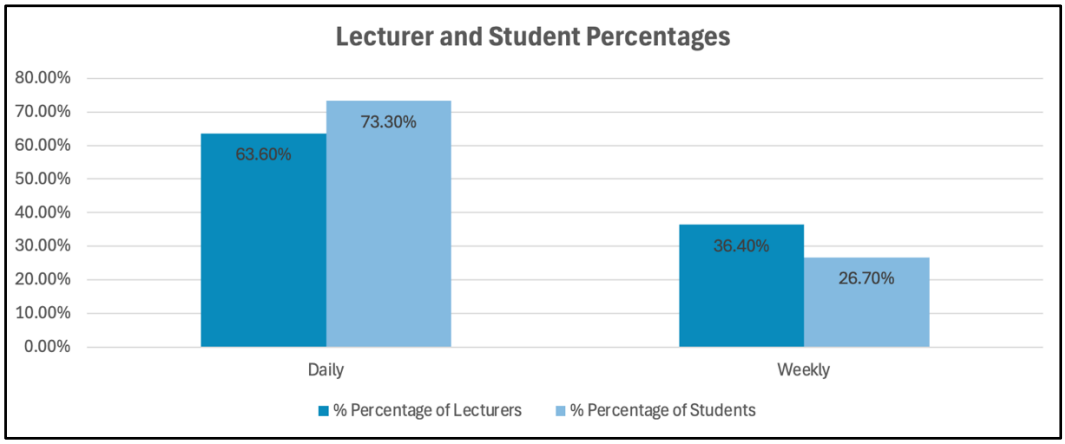



**Figure 3.4:** Questionnaire Question 2

Question 2: How often do you currently use general AI tools such as ChatGPT or Gemini for academic purposes?

Based on the responses, a majority of participants (69.2%) reported using general AI tools such as ChatGPT or Gemini on a daily basis, while 30.8% use them weekly. None indicated monthly or less frequent use, highlighting that AI tools have already become a consistent part of academic routines for both students and lecturers at UPTM.

When analyzed by role, students (22 out of 30, or 73.3%) use AI tools on a daily basis compared to lecturers (14 out of 22, or 63.6%). This suggests that students rely slightly more on AI for learning-related purposes such as summarizing materials, generating explanations, or preparing assignments.

For lecturers, 8 out of 22 (36.4%) reported using AI tools weekly, compared to 8 out of 30 (26.7%) among students. This shows that while lecturers may not use AI as frequently, they still maintain steady engagement, possibly for preparing lecture materials, generating assessments, or exploring innovative teaching approaches.

Overall, this result demonstrates a strong adoption rate of AI tools among both user groups. For the proposed UPTM Academic AI Assistant Tools, this is a positive indicator it confirms that users are already familiar and comfortable with AI technologies, allowing the system to focus on providing academic-specific features such as research support, content summarization, and guided explanations, rather than introducing basic AI concepts.

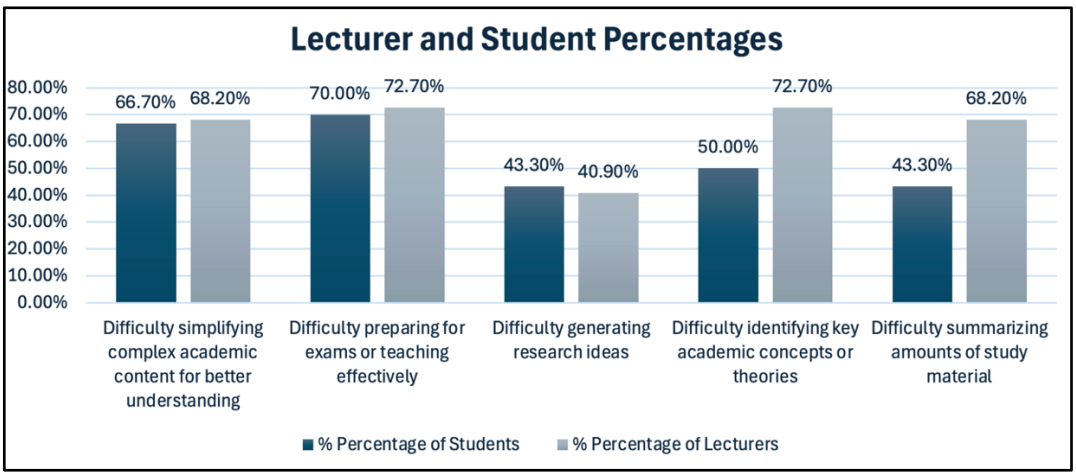



**Figure 3.5:** Questionnaire Question 3

Question 3: What difficulties do you usually encounter in completing academic tasks without AI tools?

A total of 52 respondents participated in this survey, comprising 30 students and 22 lecturers. The question aimed to identify the common difficulties faced by both groups when completing academic tasks without the help of AI tools. Since this was a checkbox question, respondents could select more than one difficulty, allowing a broader understanding of the challenges they experience. Overall, the findings show that both students and lecturers face similar academic struggles, particularly in simplifying complex content and preparing for exams or teaching effectively. These results highlight the strong need for AI-powered academic tools to support learning and teaching processes more effectively.

Among students, the most commonly selected difficulty was preparing for exams or teaching effectively, chosen by 21 out of 30 students (70%). This was followed by difficulty simplifying complex academic content (66.7%), identifying key academic concepts or theories (50%), generating research ideas (43.3%), and summarizing

large amounts of study material (43.3%). These findings indicate that students primarily struggle with understanding complex topics and managing exam preparation, suggesting that AI tools can play an essential role in simplifying learning materials and helping students study more efficiently.

For lecturers, the most frequently chosen difficulties were preparing for exams or teaching effectively (16 out of 22, 72.7%) and identifying key academic concepts or theories (16 out of 22, 72.7%), followed closely by difficulty simplifying complex academic content (68.2%) and summarizing large amounts of study material (68.2%). Meanwhile, generating research ideas was selected by 9 out of 22 lecturers (40.9%). This shows that lecturers mainly encounter challenges in delivering lessons effectively, identifying key knowledge points, and simplifying complex academic content for their students.

Overall, the results clearly justify the importance of the UPTM Academic AI Assistant Tools. The Concept Explainer feature can help users simplify complex content, while the Exam Review Notes Creator and Classroom Activity Ideas support exam preparation and teaching. Title Generator and Research Content Support address challenges in generating research ideas, and the Key Takeaways and Summary Generator features assist in identifying important theories and summarizing materials. Therefore, the data strongly supports the system’s purpose to reduce academic struggles and enhance both teaching and learning effectiveness through AI assistance.

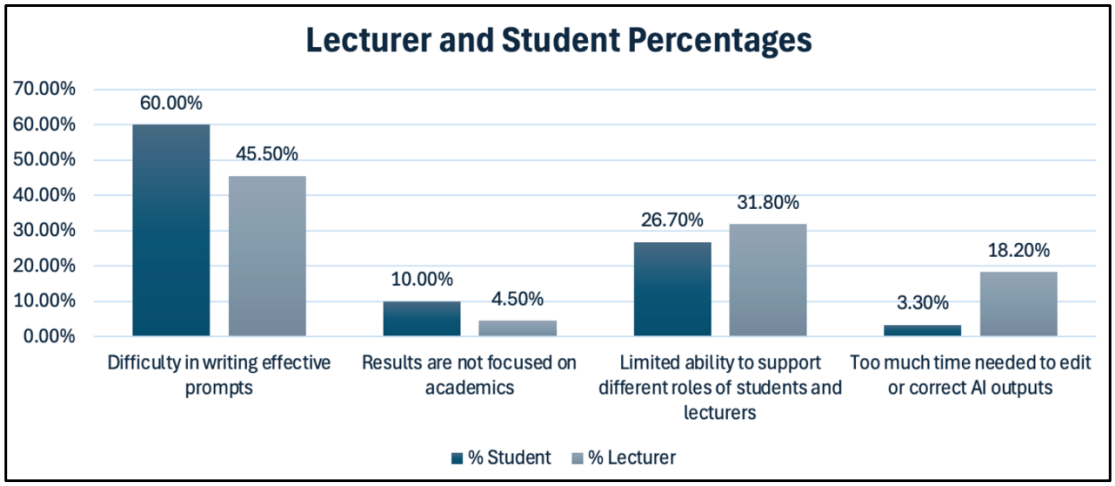



**Figure 3.6:** Questionnaire Question 4

Question 4: What challenges do you face when using general AI tools?

From the responses collected, the most common challenge faced by both students and lecturers when using general AI tools is difficulty in writing effective prompts. Among the 52 total respondents, 18 students (60%) and 10 lecturers (45.5%) selected this issue. This indicates that many users struggle to give precise instructions to AI systems to get accurate results. The second most reported challenge among students is the limited ability of general AI tools to support the different roles of students and lecturers, chosen by 8 students (26.7%) and 7 lecturers (31.8%). This shows that many users feel general AI tools fail to differentiate between academic needs, where students require learning aids like summaries and quizzes, while lecturers need tools for class preparation and evaluation. Meanwhile, only a small number of respondents, 3 students (10%) and 1 lecturer (4.5%), mentioned that AI results are not focused on academics, highlighting that while most users can adapt, some still find public AI tools too broad or informal for academic use. Finally, 1 student (3.3%) and 4 lecturers (18.2%) noted that using AI often takes too much time to edit or correct outputs, especially when the results are unstructured or not written in an academic tone.

Overall, these findings emphasize that users mainly struggle with both input and output problems when using general AI tools. The input issue, difficulty in writing prompts supports the need for the UPTM Academic AI Assistant's ready-made academic templates, which simplify the process of getting relevant results. Meanwhile, the output issue messy or unfocused answers validates the importance of having a customized academic-focused system that delivers concise, structured, and role-specific outputs

for both students and lecturers. This analysis proves that the proposed UPTM system addresses key weaknesses of general AI tools by providing features tailored to academic use and user roles.

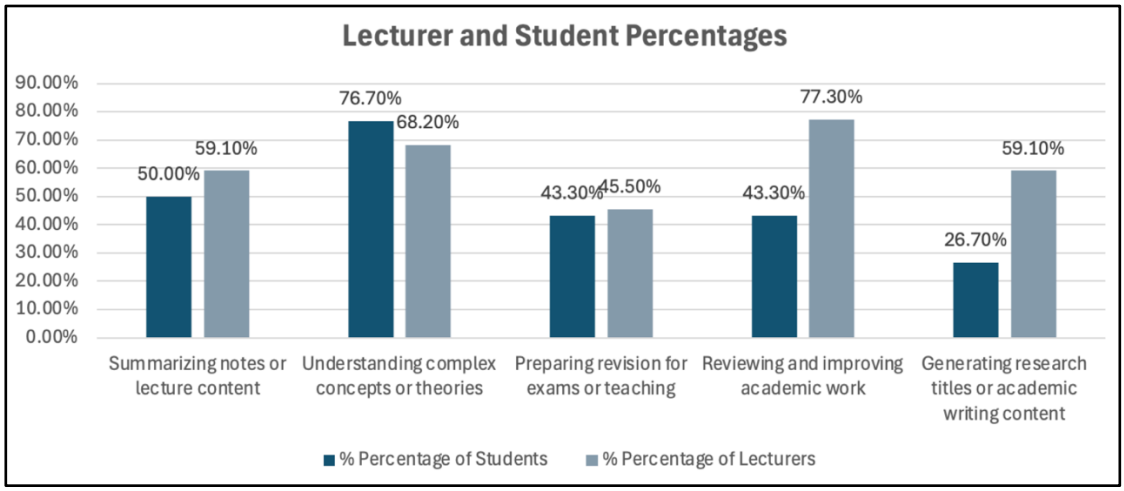



**Figure 3.7:** Questionnaire Question 5

Question 5: In your academic work, which area requires the most effort or consumes the most time?

The responses for this question provide valuable insight into which academic activities are the most time-consuming and demanding for both students and lecturers. Out of 52 respondents (30 students and 22 lecturers), the majority from both groups selected understanding complex concepts or theories as the most effortful task, with 76.7% of students and 68.2% of lecturers choosing this option. This finding highlights a shared challenge in grasping academic theories and content, supporting the need for the Concept Explainer and Teaching Material Simplifier features within the proposed UPTM Academic AI system. These tools aim to break down complex academic material into simpler, more understandable explanations, making learning and teaching more effective.

For summarizing notes or lecture content, 50% of students and 59.1% of lecturers reported this as a major time-consuming task. This aligns with the Academic Summary Generator, which is designed to assist users in quickly summarizing large volumes of lecture notes or study materials. The consistent response from both groups shows that summarization remains a key academic pain point that directly affects productivity and efficiency.

Additionally, 43.3% of students and 45.5% of lecturers indicated preparing revision for exams or teaching as a challenge. This emphasizes the usefulness of the Exam Review Notes Creator and Flashcard Creator, which can generate structured and concise study aids, helping students prepare more effectively and lecturers to create revision materials with less effort.

Another significant finding is that reviewing and improving academic work was chosen by 43.3% of students and as many as 77.3% of lecturers, making it one of the top concerns among academic staff. This clearly validates the need for AI-driven features like the Feedback Generator and Academic Proofreader, which can automate proofreading, improve writing quality, and provide quick academic feedback to enhance teaching and learning outcomes.

Finally, generating research titles or academic writing content was reported by 26.7% of students and 59.1% of lecturers. This indicates that while students face challenges in idea generation, lecturers also find research and writing tasks demanding. The Academic Paragraph Generator and Thesis Title Generator directly address this issue by guiding users through the early stages of research development and assisting in producing well-structured academic content.

Overall, the findings from this question are crucial for feature prioritization in the UPTM Academic AI system. They reveal that both students and lecturers struggle most with understanding complex concepts, summarizing materials, and reviewing academic work, all areas that the proposed AI system is specifically designed to improve.

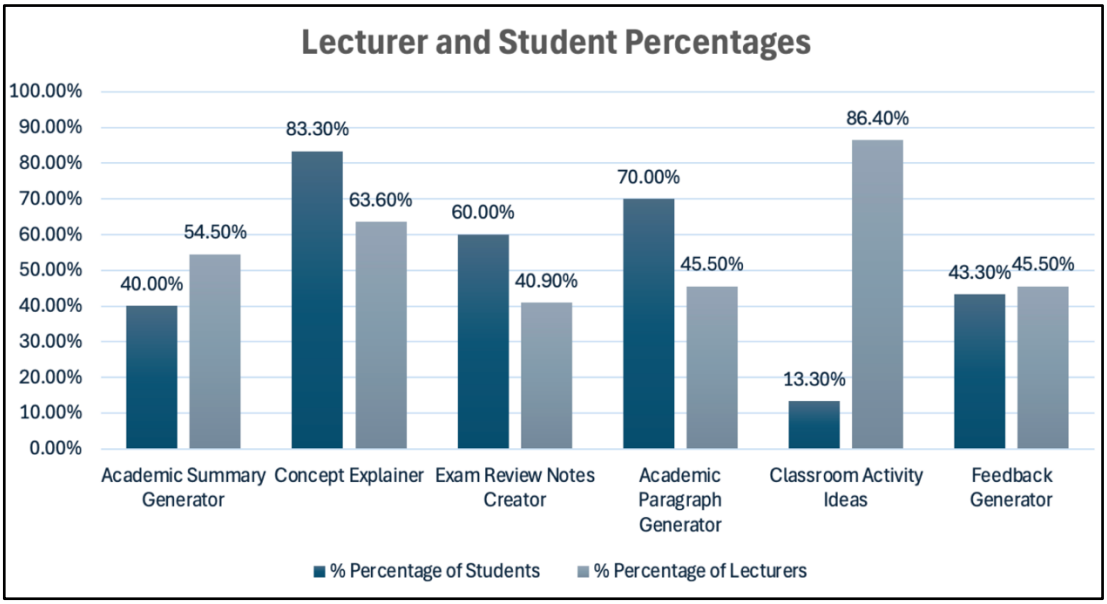



**Figure 3.8:** Questionnaire Question 6

Question 6: Which AI features would you find most useful and likely to use in your academic work?

This question identifies the specific AI features that respondents consider most valuable for supporting their academic activities. Out of 52 respondents (30 students and 22 lecturers), the majority highlighted Concept Explainer and Academic Paragraph Generator as the most useful features. This finding directly supports the system's goal of enhancing academic efficiency by simplifying complex content and improving the quality of written outputs.

Among students, Concept Explainer received the highest response with 25 out of 30 students (83.3%), followed by Academic Paragraph Generator at 70%, and Exam Review Notes Creator at 60%. These results show that students value features that help them understand difficult topics, generate structured academic writing, and prepare effectively for exams. This aligns closely with the learning challenges previously identified in Question 5, where understanding complex concepts was also the most time-consuming academic activity. Therefore, the system's Concept Explainer and Exam Review Notes Creator are confirmed as essential tools for improving comprehension and study productivity.

For lecturers, the pattern shows slightly different priorities. The Classroom Activity Ideas feature received the highest response with 19 out of 22 lecturers (86.4%),

indicating strong interest in tools that assist with lesson planning and teaching creativity. Following that, Concept Explainer (63.6%) and Academic Summary Generator (54.5%) were also considered valuable. These findings suggest that lecturers prefer tools that not only simplify complex topics but also help them deliver content more effectively and engage students during lectures.

The Feedback Generator feature, chosen by 43.3% of students and 45.5% of lecturers, further highlights the demand for automated feedback tools that enhance learning quality and reduce grading or revision workload. Meanwhile, Academic Summary Generator was equally selected by both students and lecturers (40%), showing its universal usefulness in summarizing academic materials for both teaching and learning.

Overall, the data reveals that students prioritize comprehension, writing support, and exam preparation, while lecturers focus more on teaching aids and classroom management support. From a system development perspective, this analysis provides clear direction for feature prioritization the UPTM Academic AI system should emphasize the Concept Explainer, Academic Paragraph Generator, and Classroom Activity Ideas as its core functionalities. These features directly align with user expectations, ensuring high system adoption and long-term value across both user roles.

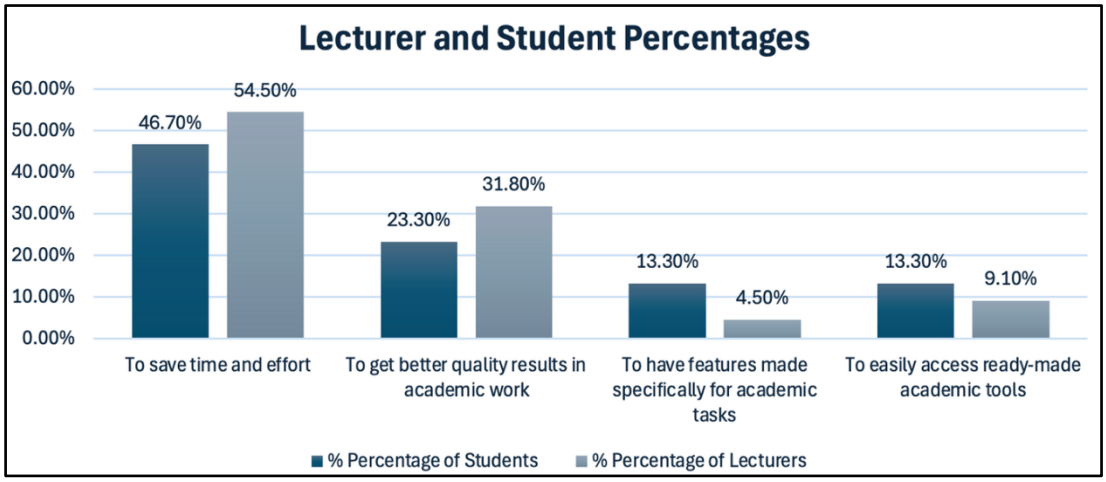



**Figure 3.9:** Questionnaire Question 7

Question 7: Which benefit would motivate you most to use academic AI features regularly?

Among students, the most common motivation for using academic AI tools is to save time and effort, with 14 out of 30 students selecting this option. This shows that students highly value convenience and the ability to complete academic tasks more efficiently. Many students likely view AI as a way to reduce time spent on repetitive work such as summarizing notes, or preparing revision materials.

The second most selected option is to get better quality results in academic work (7 students). This indicates that students also believe AI can enhance the quality of their academic output and help them produce more refined results. Meanwhile, 4 students chose to have features made specifically for academic tasks, and another 4 selected to easily access ready-made academic tools. These responses show that while customization and accessibility are appreciated, the main student focus remains on saving time and improving work quality.

For lecturers, the results show a similar trend. The majority (12 out of 22) also selected to save time and effort, suggesting that educators value efficiency when preparing lectures, grading, or creating teaching materials. This highlights that time-saving AI features can support their teaching workload and lesson preparation.

Another 7 lecturers (equal to the student group) selected to get better quality results, reflecting their interest in maintaining high-quality academic output, possibly through

tools that enhance clarity and accuracy in academic writing or feedback. A smaller portion of lecturers (1 respondent) preferred features made specifically for academic tasks, while 2 chose to easily access ready-made academic tools. This shows that while lecturers value specialized and accessible AI systems, their main motivation similar to students is still focused on saving time and improving quality.

In summary, both students and lecturers share a common motivation for adopting academic AI tools, time efficiency and quality improvement. This insight supports the system's design objective to provide AI features that simplify academic work while maintaining or enhancing academic standards. It also reinforces that automation-focused features, such as academic summarizers, content generators, and feedback assistants, should be prioritized in development to meet user expectations effectively.

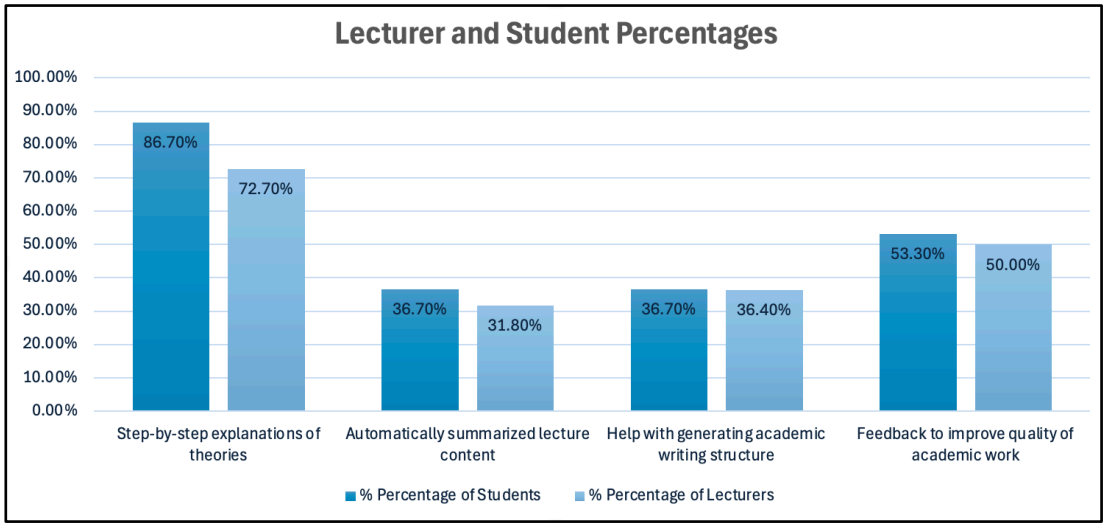



**Figure 3.10:** Questionnaire Question 8

Question 8: Which type of guidance would you want most from an academic AI system?

Among students, the majority (26 out of 30) prefer step-by-step explanations of theories, showing a strong need for detailed, guided understanding rather than simple answers. This result indicates that many students struggle with complex academic content and would benefit from a system that provides clear, logical explanations. In the UPTM Academic AI Assistant, this directly aligns with the Concept Explainer feature, where AI breaks down theories into simpler parts. The system already supports this by using hidden prompts behind each feature to ensure explanations are detailed and easy to follow.

The next popular choices among students are feedback to improve academic work (16 students) and help with generating academic writing structure (11 students). This shows that many students also value personalized improvement suggestions and writing guidance important for essays, or reports. Meanwhile, automatically summarized lecture content (11 students) also gained interest, showing that students appreciate tools that can save time during revision by turning long notes into concise summaries.

For lecturers, step-by-step explanations of theories also received the highest responses (16 out of 22), highlighting that even educators value AI support that can simplify and explain difficult theories clearly for classroom delivery. This indicates the

system's teaching material simplification and AI-assisted concept explanation features will be useful in their daily teaching routines.

The second-highest lecturer preference was feedback to improve quality of academic work (11 respondents). This demonstrates that lecturers see value in using AI for evaluating or refining academic materials. Additionally, help with generating academic writing structure (8 lecturers) and automatically summarized lecture content (7 lecturers) suggest that many lecturers appreciate structured AI support to make their teaching or research preparation more efficient.

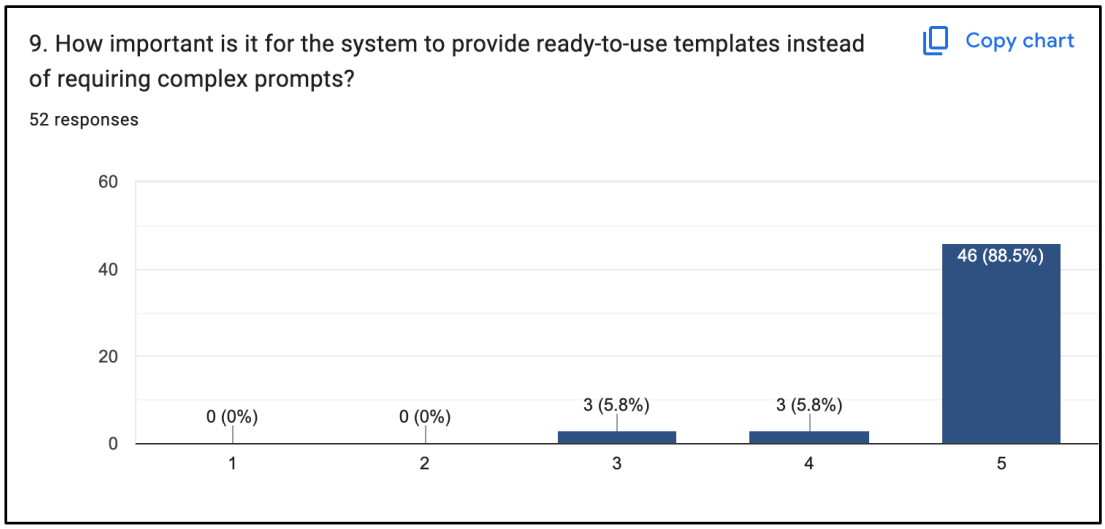



**Figure 3.11:** Questionnaire Question 9

Question 9: How important is it for the system to provide ready-to-use templates instead of requiring complex prompts?

The majority of respondents (88.5%) strongly agree that having ready-to-use templates is highly important, while 5.8% agree and another 5.8% remain neutral. Notably, none of the respondents disagreed, which clearly demonstrates that users strongly prefer a system that minimizes the need for complex prompt writing.

This result highlights a significant demand for efficiency and simplicity in academic AI systems. Users appreciate solutions that allow them to quickly generate academic outputs without spending time crafting long and specific prompts.

From an analytical standpoint, this finding reinforces the core strength of the UPTM Academic AI Assistant, which is built around pre-designed academic templates. These ready-made templates directly address the users' top priority which is saving time while maintaining academic quality.

In conclusion, the data confirms that ready-to-use academic templates are not just preferred but essential. They provide a smoother, faster, and more focused AI experience for both students and lecturers, supporting the system's main goal of simplifying academic work while maintaining professional output.

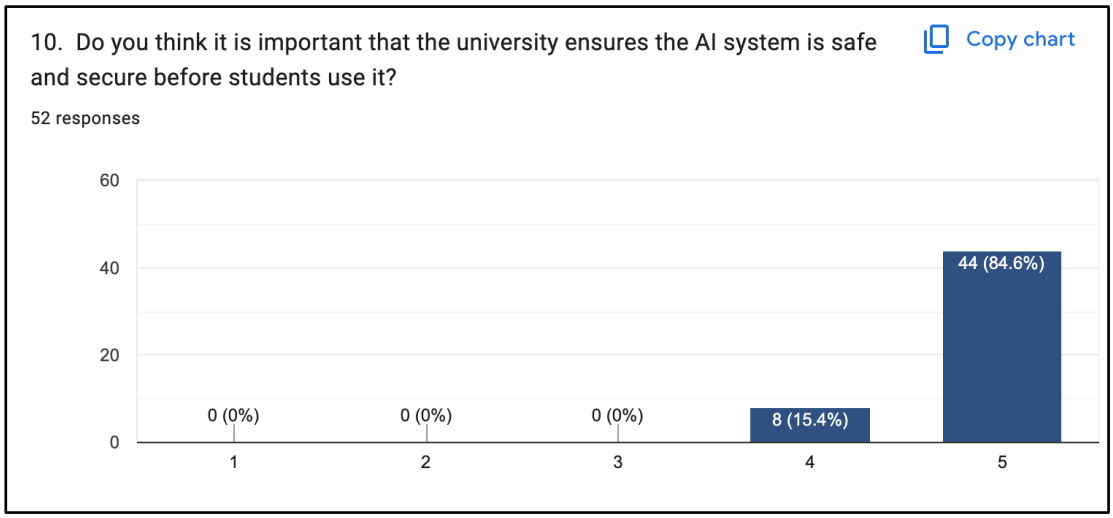



**Figure 3.12:** Questionnaire Question 10

Question 10: Do you think it is important that the university ensures the AI system is safe and secure before students use it?

The results reveal an overwhelming agreement among respondents, with 84.6% strongly agreeing and 15.4% agreeing that it is important for the university to ensure the AI system is safe and secure before being used by students. No respondents disagreed or remained neutral, indicating a unanimous belief in the importance of institutional oversight.

This strong consensus demonstrates that both students and lecturers place high trust in the university's role in managing and monitoring AI tools. It suggests that users are more confident in using AI systems that are officially developed, regulated, and

safeguarded by UPTM, compared to general public AI platforms that may raise privacy or data protection concerns.

From an analytical perspective, these findings support the argument that UPTM's control and implementation of its own academic AI assistant system enhances user confidence and ensures responsible AI use within the educational environment. The emphasis on safety and security also highlights the importance of ethical AI governance, ensuring that academic integrity, data privacy, and user trust remain central to the system's deployment.

In conclusion, the data confirms that institutional trust and system security are key factors in encouraging users to adopt AI tools, reinforcing the need for a university-managed AI platform that prioritizes reliability and protection over general, unverified alternatives.

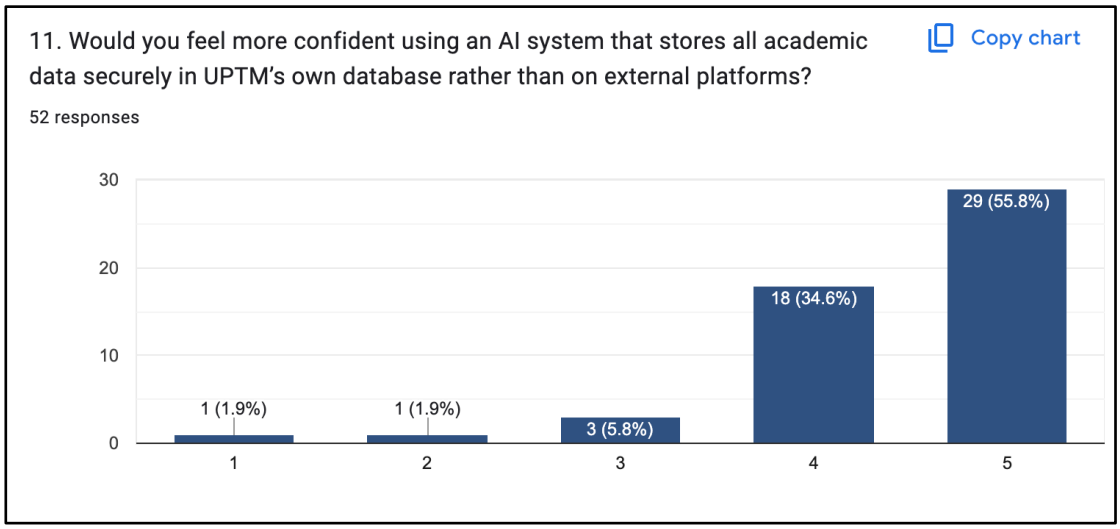



**Figure 3.13:** Questionnaire Question 11

Question 11: Would you feel more confident using an AI system that stores all academic data securely in UPTM's own database rather than on external platforms?

The results show that a large majority of respondents (90.4%) either agreed or strongly agreed that they would feel more confident using an AI system that stores academic data securely within UPTM's own database. Only a small portion expressed disagreement (3.8%), while 5.8% remained neutral.

This clearly demonstrates that data ownership and security are key trust factors for users when interacting with AI systems in academic settings. The strong preference for internal data storage reflects concerns over privacy, data misuse, and reliability when using external AI platforms which store data on third-party servers.

From an analytical standpoint, these findings strongly support the decision to host and secure all academic data under UPTM's infrastructure. This approach aligns perfectly with user expectations and builds greater confidence in the system's reliability and ethical standards.

In conclusion, the data provides strong evidence that UPTM's locally managed AI system is not only preferred but also essential for ensuring user trust, data integrity, and institutional control. This reinforces the strategic importance of developing the AI assistant within the university's own secure environment.

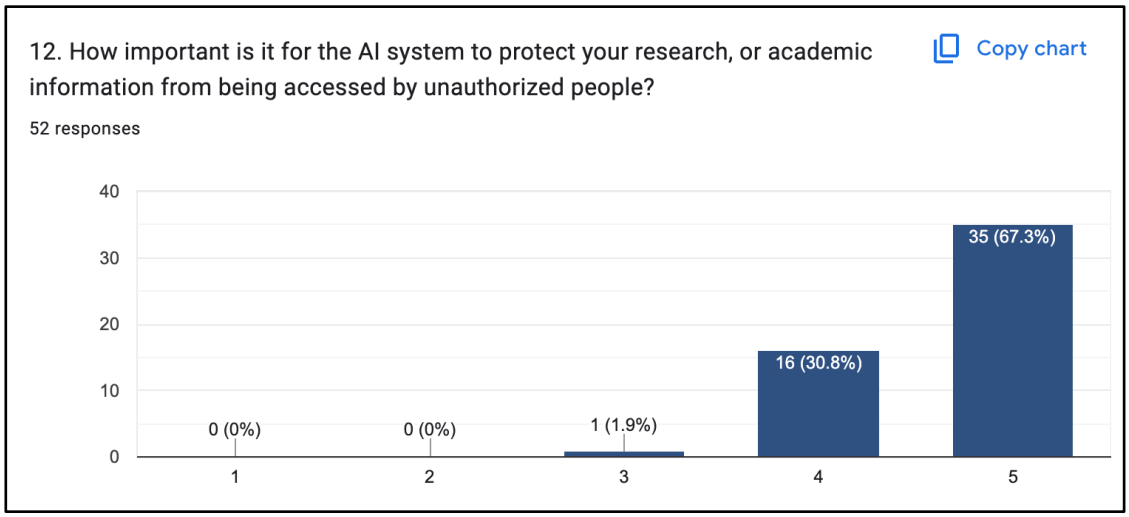



**Figure 3.14:** Questionnaire Question 12

Question 12: How important is it for the AI system to protect your research, or academic information from being accessed by unauthorized people?

The data reveals that an overwhelming 98.1% of respondents rated the importance of data protection as either Agree or Strongly Agree. In particular, 67.3% strongly agreed, indicating that the majority view security and privacy as a top priority in any academic AI system. Only 1.9% remained neutral, and no participants expressed disagreement.

This result provides strong support for prioritizing privacy protection within the proposed UPTM Academic AI Assistant Tools. Users clearly expect that all their research data, assignments, and academic interactions with the AI must be kept confidential and safeguarded against unauthorized access.

From an analytical perspective, this finding reinforces the need for secure data handling policies, such as storing both AI inputs and outputs (conversations, or generated content) within UPTM's own database instead of external platforms. By implementing robust access controls and institutional data ownership, the system can ensure academic integrity, confidentiality, and trustworthiness, all of which are critical to user acceptance and long-term reliability.

In summary, the responses demonstrate a clear expectation that the AI system must guarantee high standards of privacy and security, confirming that this is a fundamental factor for both students and lecturers when adopting AI technology in academic work.

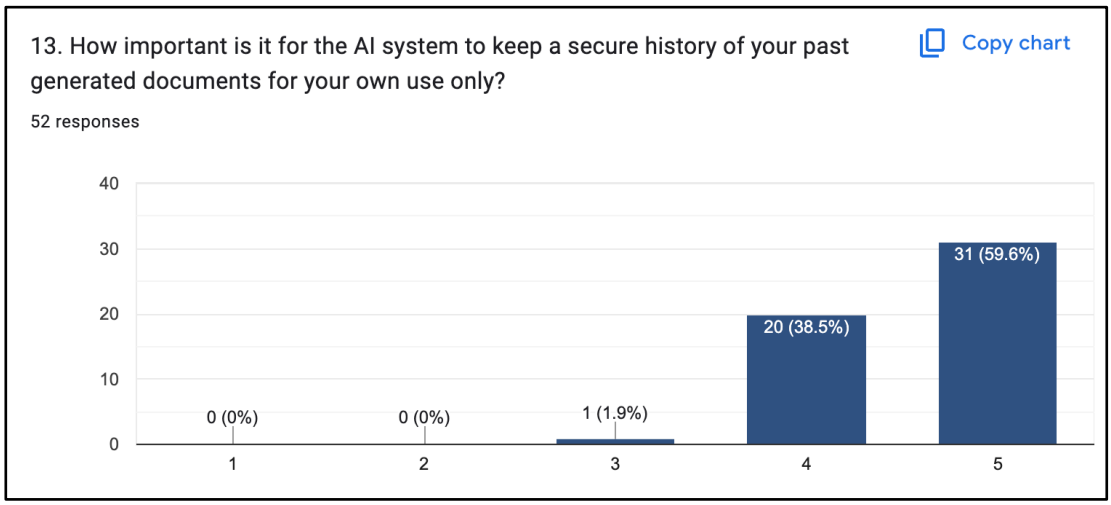



**Figure 3.15:** Questionnaire Question 13

Question 13: How important is it for the AI system to keep a secure history of your past generated documents for your own use only?

The responses indicate that all participants agree on the importance of having a secure and private document history, with 59.6% strongly agreeing and 38.5% agreeing. Only 1.9% of respondents remained neutral, and none disagreed. This clearly shows that users place high value on the ability to safely access their previous AI-generated work without sharing it publicly.

This finding is particularly relevant to the document history feature of the UPTM Academic AI Assistant Tools. It confirms that users both students and lecturers want a personalized and protected archive where they can retrieve their generated notes, summaries, or teaching materials securely. The emphasis on “for your own use only” highlights a strong preference for privacy and individual ownership over academic content produced with AI.

From an analytical perspective, this outcome supports the inclusion of secure user accounts and history management within the system. It shows that users not only value security but also expect convenience and usability through easy access to their own stored results.

In conclusion, this question reinforces that combining security with user-centered accessibility is key to making the AI system both trustworthy and practical. It proves

that users want to maintain control over their generated academic materials, ensuring both data privacy and long-term usability.

#### 3.5.1.2 Interview Analysis

The interview analysis was conducted to obtain qualitative insights from an expert in the field of Artificial Intelligence (AI) to support the development of the UPTM Academic AI Assistant Tools. The purpose of this interview was to understand the lecturer's experience with AI, her perspective on AI assistance tools, and her opinion on the proposed system features. The discussion also aimed to gather professional feedback on potential improvements and user expectations for AI integration in academic environments. The insights obtained were valuable in refining the system's design, aligning it with educational needs, and ensuring that it meets the standards of academic integrity and usability.

**Interview Details**

**Name:** Ms. Nadiah Yusof Position: Artificial Intelligence (AI) Lecturer, Universiti Poly-Tech Malaysia (UPTM)
**Platform:** Microsoft Teams
**Date:** 24 October 2025
**Time:** 4:00 PM – 5:00 PM

**Table 3.7**: Question 1 Interview Analysis

| | |
|---|---|
| **Question 1** | Can you share your general experience or knowledge about Artificial Intelligence (AI)? |
| **Answer** | AI is a field that focuses on humanizing machines, systems, and computers so that they can perform tasks that normally require human intelligence. |
| **Analysis** | From this response, it can be understood that the lecturer views AI as a technology designed to simulate human thinking and behavior. This aligns with the project's goal of integrating AI into an academic setting to assist in performing cognitive tasks such as summarizing study notes or generating academic ideas. Her perspective highlights the importance of designing an AI system that behaves intelligently and naturally, helping users interact with technology more effectively while maintaining academic standards. |

**Table 3.8**: Question 2 Interview Analysis

| | |
|---|---|
| **Question 2** | What is your understanding of AI assistance tools? |
| **Answer** | AI assistance tools are ready-to-use systems that help refine, or correct content efficiently. |
| **Analysis** | The lecturer's understanding reflects that AI tools serve as supportive assistants that enhance the quality and presentation of work rather than replace human input. This viewpoint supports the project's aim to develop a system that assists users in improving academic writing and organization. It also suggests that the tool should prioritize usability and accessibility, allowing students and lecturers to quickly refine their content with minimal technical effort. |

**Table 3.9**: Question 3 Interview Analysis

| | |
|---|---|
| **Question 3** | Which AI tools have you used before? Can you provide some examples? |
| **Answer** | Grammarly |
| **Analysis** | The lecturer's experience with Grammarly indicates familiarity with AI-driven writing support tools that focus on grammar correction, clarity, and readability. This insight reinforces the potential usefulness of developing a similar AI-assisted academic system that not only enhances grammar and structure but also provides more specialized academic functions. Her practical experience also highlights that users value simple, automated systems that directly improve written content. |

**Table 3.10**: Question 4 Interview Analysis

| | |
|---|---|
| **Question 4** | What do you think if the system being developed includes features like the AI template shown? Would you agree with having these features? |
| **Answer** | Yes, I agree with the inclusion of these features. |
| **Analysis** | The lecturer's positive response indicates approval of the system's proposed features, especially those that simplify academic work. This supports the project's design direction, confirming that the proposed functions align with user expectations. Her agreement also reflects a recognition of the system's potential benefits in academic productivity, which are critical aspects of institutional AI adoption. |

**Table 3.11**: Question 5 Interview Analysis

| | |
|---|---|
| **Question 5** | If these features are included, what is your overall opinion about the system that will be developed? |
| **Answer** | The system has potential to support students and lecturers in managing their academic work. It can serve as a valuable academic assistant that encourages ethical AI usage within the university environment. |
| **Analysis** | This opinion highlights that the system is not only functional but also educational. It promotes responsible AI integration in academia and encourages proper use of AI tools to complement human learning. The lecturer's view reinforces that the system should maintain academic credibility while offering practical assistance that enhances user productivity and understanding. |

**Table 3.12**: Question 6 Interview Analysis

| | |
|---|---|
| **Question 6** | Do you have any suggestions or opinions for improving the system that we are going to develop? |
| **Answer** | The current features are already good, but it would be helpful if you could explain in more detail how the AI processes information behind the system. This would help users better understand how AI works when generating content. |
| **Analysis** | This feedback emphasizes the importance of transparency in AI operations. The lecturer's suggestion implies that users should have at least a basic understanding of how the AI generates results to build trust and confidence in the system. Incorporating a feature or section that briefly explains the AI's process could improve user engagement, learning value, and credibility. This recommendation will guide future development to ensure that the system not only delivers results but also educates users about responsible and informed AI usage. |

### 3.5.2 Use Case Model

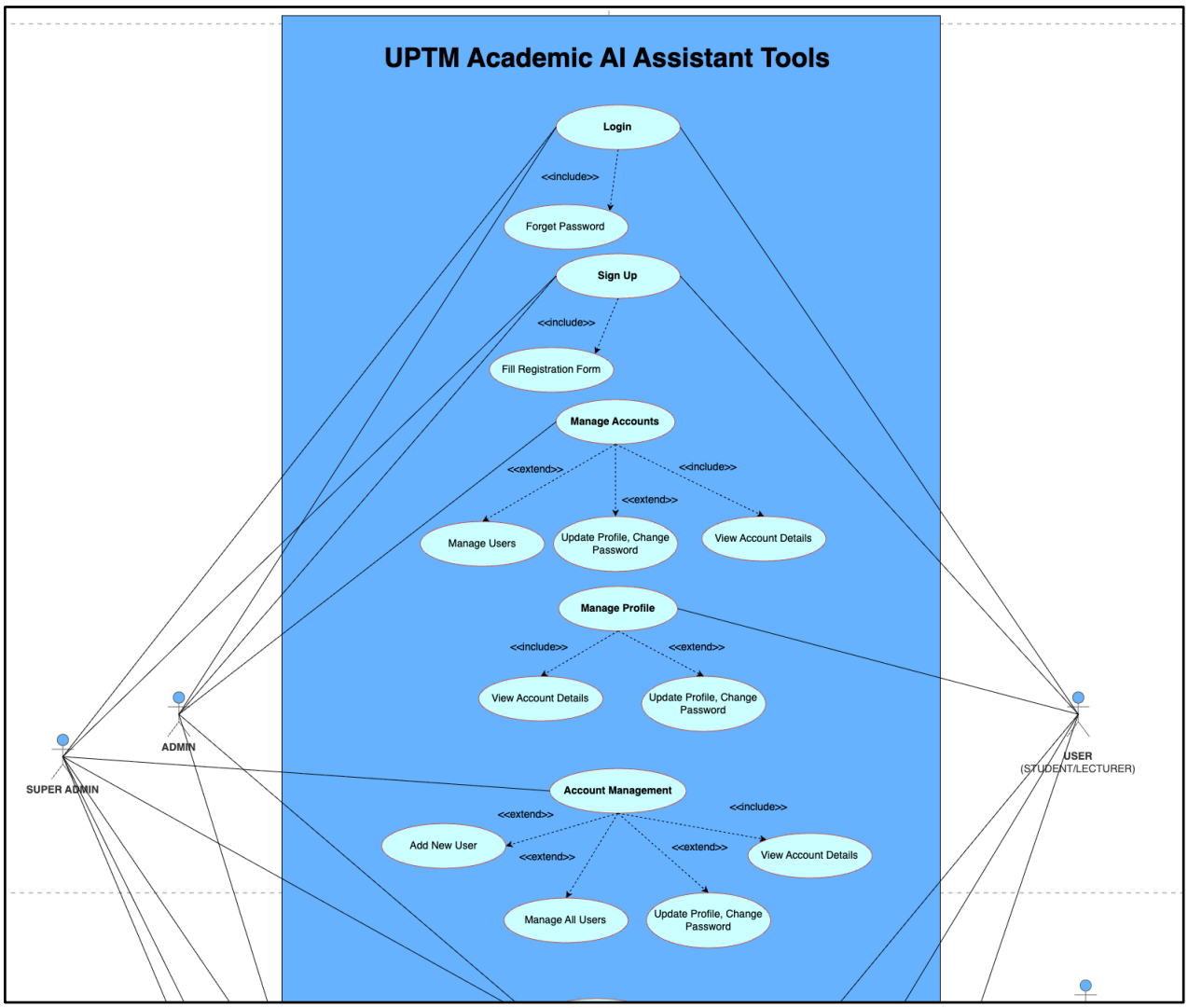



**Figure 3.16:** Use Case Diagram 1.0

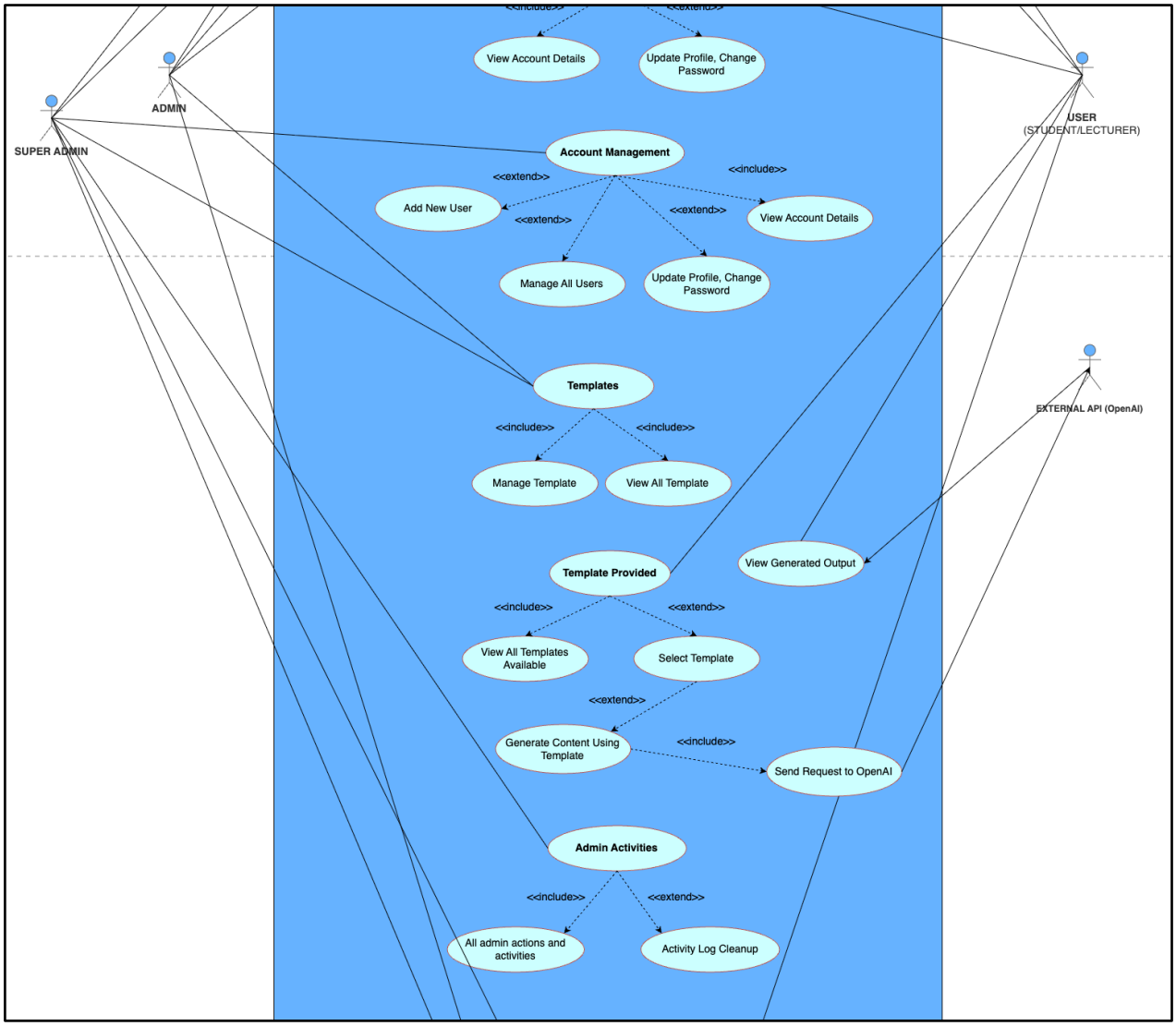



**Figure 3.17:** Use Case Diagram 2.0

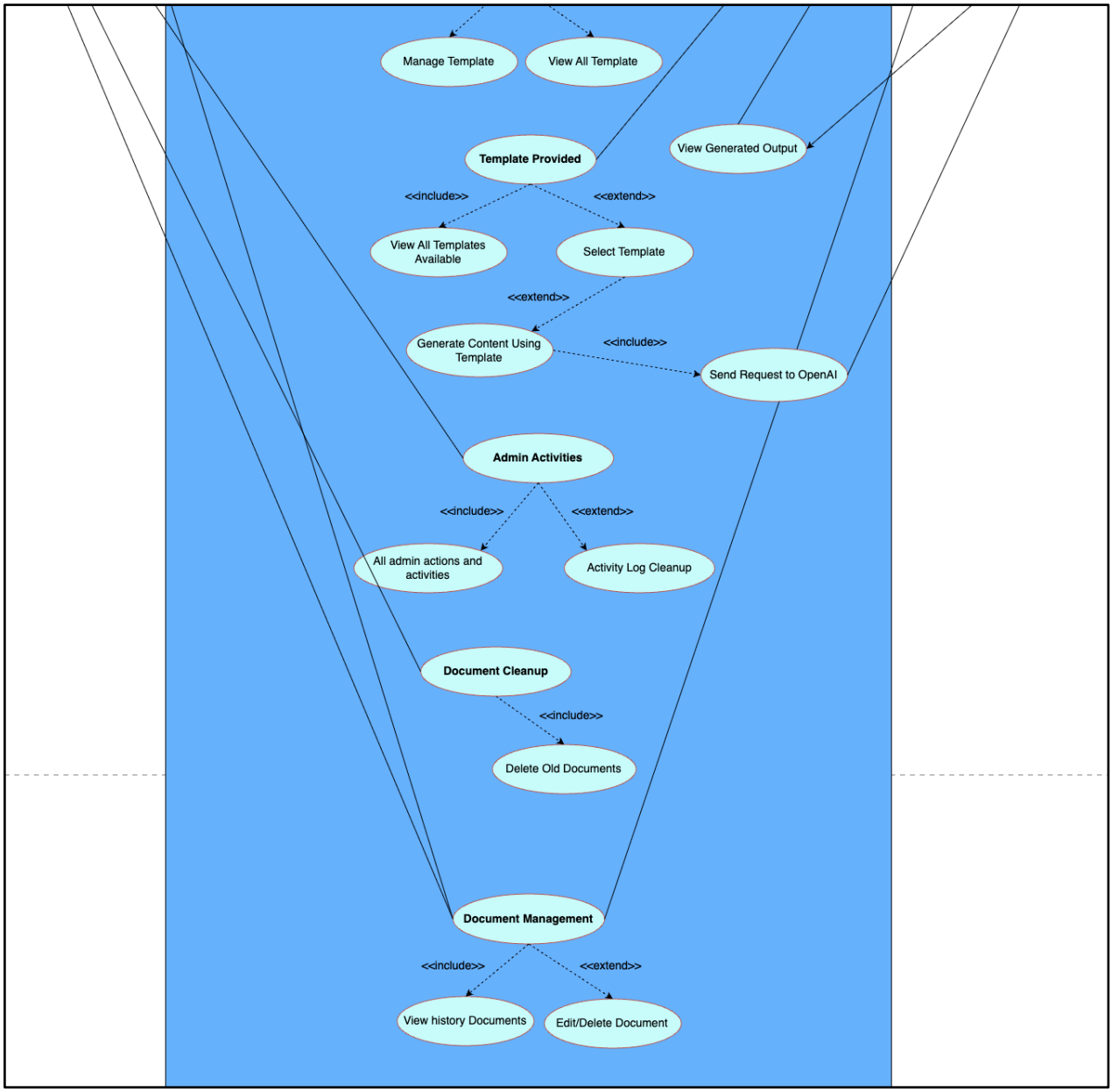



**Figure 3.18:** Use Case Diagram 3.0

The Use Case Model for the UPTM Academic AI Assistant Tools illustrates the system's functional requirements and interactions among three primary actors: Superadmin, Admin, User (Lecturer/Student), and the external OpenAI API. The system functionalities are organized into three main areas: account management, template management and content generation, and document management. All actors are required to authenticate through login or sign-up before accessing the system, with additional support for password recovery to ensure secure access.

Once authenticated, each role is granted different levels of access based on their responsibilities. Users (lecturers and students) can manage their own profiles, change passwords, view available templates, and generate academic content using AI-powered templates. When a user submits input through a selected template, the system sends a request to the OpenAI API to generate academic content, which can then be viewed, edited, or deleted and stored in the user's document history for future reference.

Admins are responsible for managing system content and monitoring usage. In addition to managing their own accounts, admins can manage user accounts, control AI templates by creating, updating, deleting, or adjusting their visibility, and oversee generated documents to ensure academic quality. The Superadmin holds the highest level of authority, with full control over all users, templates, documents, and system logs. Superadmins can add new users, track admin activities, perform document and log cleanup, and manage system-wide data to maintain performance and accountability.

Overall, the use case diagrams demonstrate a structured role-based system where Superadmins manage governance, Admins handle operational control, and Users utilize AI-assisted features for academic content generation through an external AI service.

## 3.5.3 BPNM (Business Process Modelling Notation)

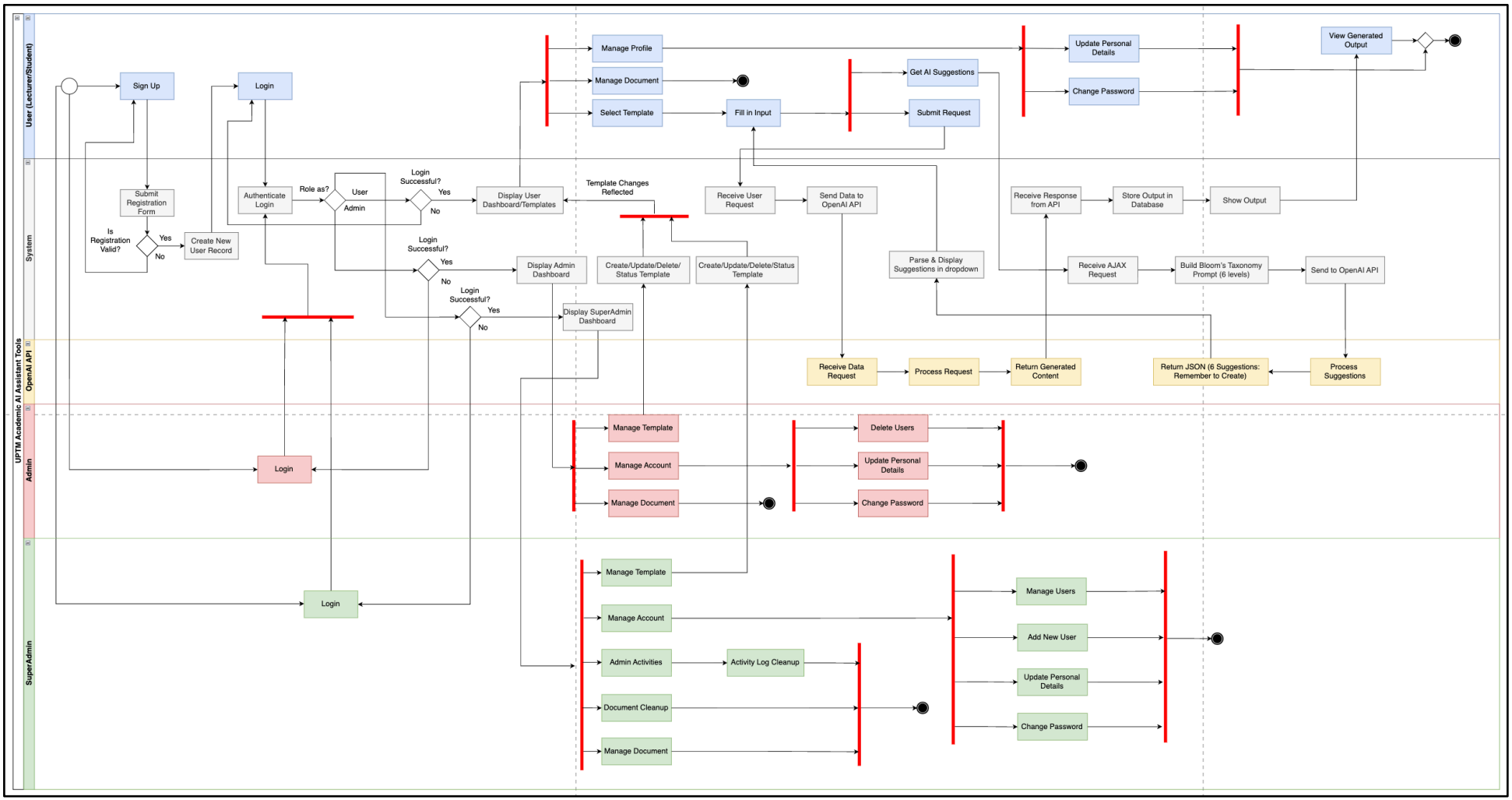



**Figure 3.19:** BPNM (Business Process Modelling Notation)

The swimlane diagram illustrates the process flow of the UPTM Academic AI Assistant Tools by showing the interactions among five main lanes: User, System, OpenAI API, Admin, and Superadmin. The process begins in the User Lane, where a user initiates the system by performing either a sign-up or login action. During registration, the system validates the submitted information and creates a new user record if the data is valid. For login, the system authenticates the credentials and determines the user's role. Based on this role verification, the system directs the user to the appropriate dashboard, either the user dashboard, admin dashboard, or superadmin dashboard, ensuring role-based access control throughout the system.

In the User Lane, the primary workflow focuses on profile management and document generation. A logged-in user may update personal details, change passwords, and access the document management page, which stores a history of previously generated outputs based on selected templates. When generating content, the user selects a template and fills in the required input fields. At this point, the workflow introduces a decision point where the user may either submit the request directly for final content generation or choose the AI suggestion option. If the AI suggestion option is selected, the system processes the input via an AJAX request, constructs an academic prompt based on Bloom's Taxonomy levels, and sends it to the OpenAI API. The OpenAI API generates structured academic suggestions and returns them to the system, which then displays the suggestions to the user for review and selection. The user may refine the input using these suggestions before proceeding to submit the final request.

Once the user submits the final request, the System Lane handles the communication with the OpenAI API by sending the completed input data for content generation. The OpenAI API processes the request and returns the generated academic output. The system then stores the output in the database and presents the result to the user, allowing it to be viewed and managed within the document history. This sequence represents the core functionality of the system, where AI is used to assist users in producing academic content efficiently and systematically.

The Admin Lane represents system-level operational control. Admin users are responsible for managing templates by creating, updating, change status, or deleting them, managing user accounts such as removing inactive users, and maintaining their own account information. These actions ensure that templates remain relevant, users are properly managed, and the system operates securely. The Superadmin lane represents the highest authority within the system, providing full control over admins, users, templates, documents, and system logs.

Superadmins can manage admin accounts, monitor system activities, perform data cleanup, and enforce system governance, ensuring long-term stability, security, and accountability of the platform.

## 3.6 Conclusion

In conclusion, this chapter has explained the overall methodology used in developing the UPTM Academic AI Assistant Tools. The project adopted the Agile methodology, which allowed continuous improvement through iterative phases such as planning, designing, developing, testing, deploying, and reviewing. This approach ensured that user feedback was consistently integrated into each stage, resulting in a system that meets real academic needs.

The requirement analysis identified key user expectations from lecturers and students through questionnaires and interviews, which guided the system's functional and non-functional design. The analysis also produced several system models, such as the Use Case Diagram, and BPMN, to visualize user interactions and data flow.

Overall, this methodology provided a structured yet flexible framework that ensured the system was developed efficiently, user-centered, and aligned with UPTM's academic goals. The next chapter will focus on the system design and implementation, describing how the planned features were transformed into a working prototype.

# 4 DESIGN

## 4.1 Introduction

This section introduces the design phase of the system. The main goal is to translate the requirements gathered in the previous chapter into a clear system structure and interface. The design includes the system architecture, user interface design, and database design, ensuring the system meets both user and functional requirements efficiently.

## 4.2 Interface Design

The interface design focuses on simplicity and ease of use for all user roles, including students, lecturers, administrators, and super administrators. The system's wireframes are structured with a clean and organized layout that allows users to navigate smoothly, select AI templates, input content, and generate academic outputs efficiently. In addition to user-facing screens, dedicated wireframe flows were designed for admin and superadmin roles to support template management, user administration, and system monitoring tasks. Consistent layouts and navigation elements are applied across all pages to ensure a uniform user experience, while responsive design considerations allow the system to function effectively across different device sizes.

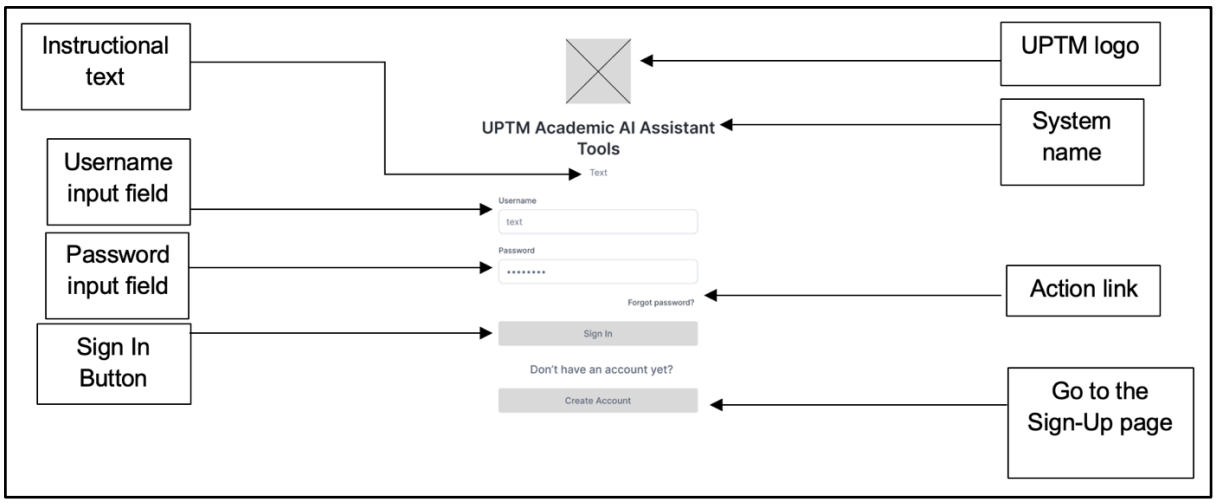



**Figure 4.1:** Login (users/admin/SuperAdmin)

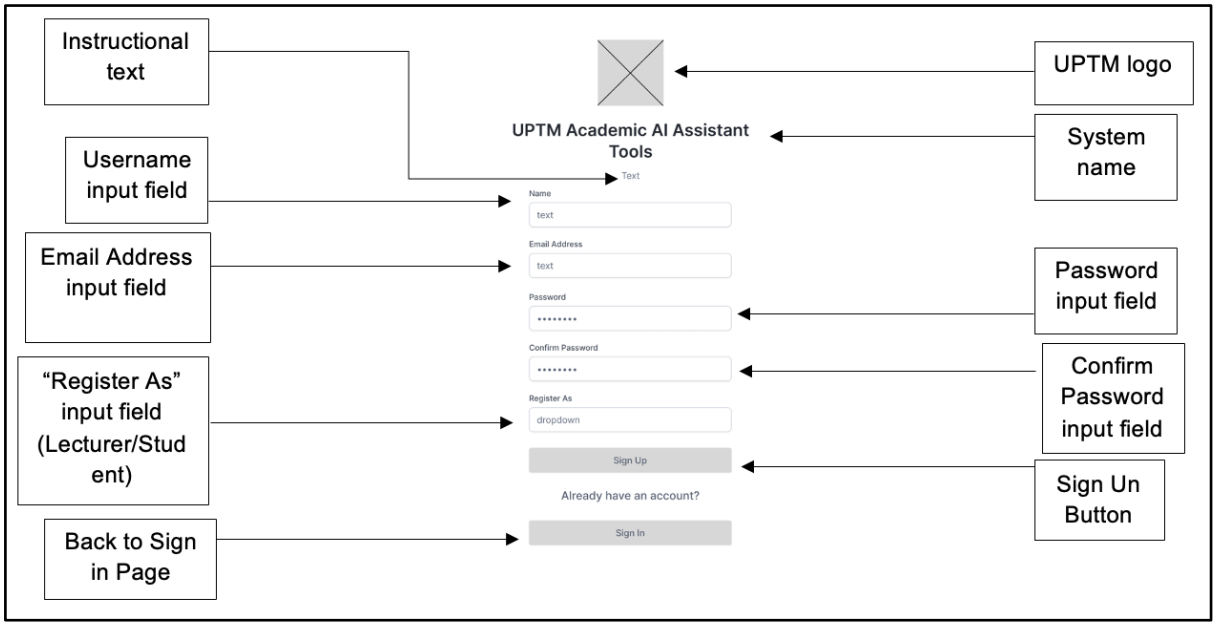



**Figure 4.2:** Sign Up (users)

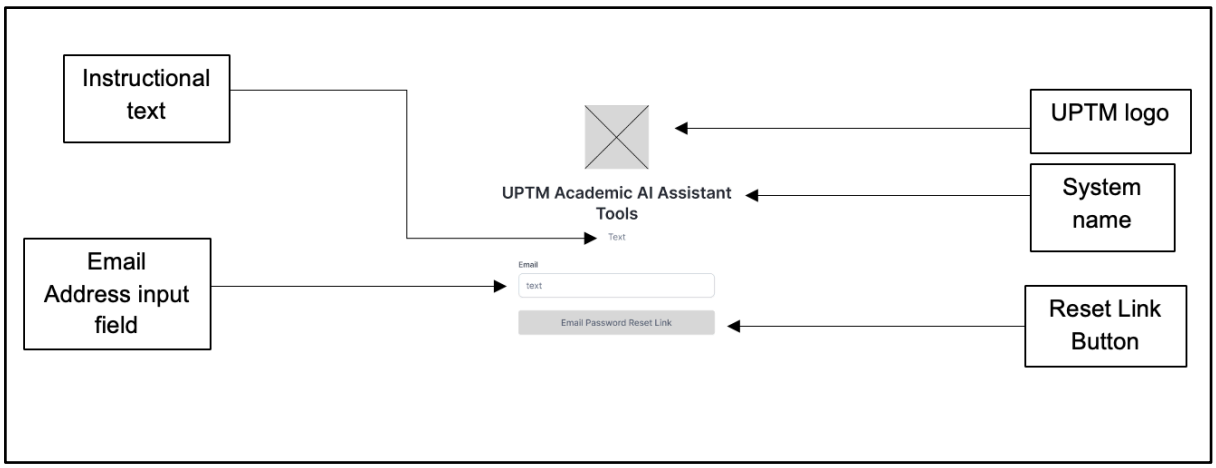



**Figure 4.3:** Forgot Password (users)

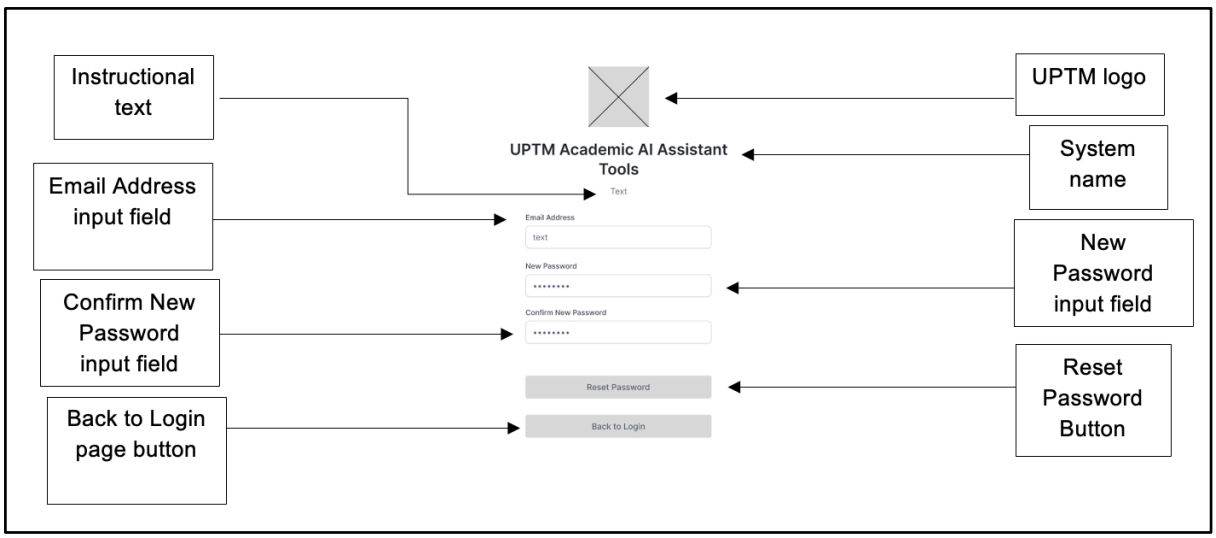



**Figure 4.4:** Reset New Password (users)

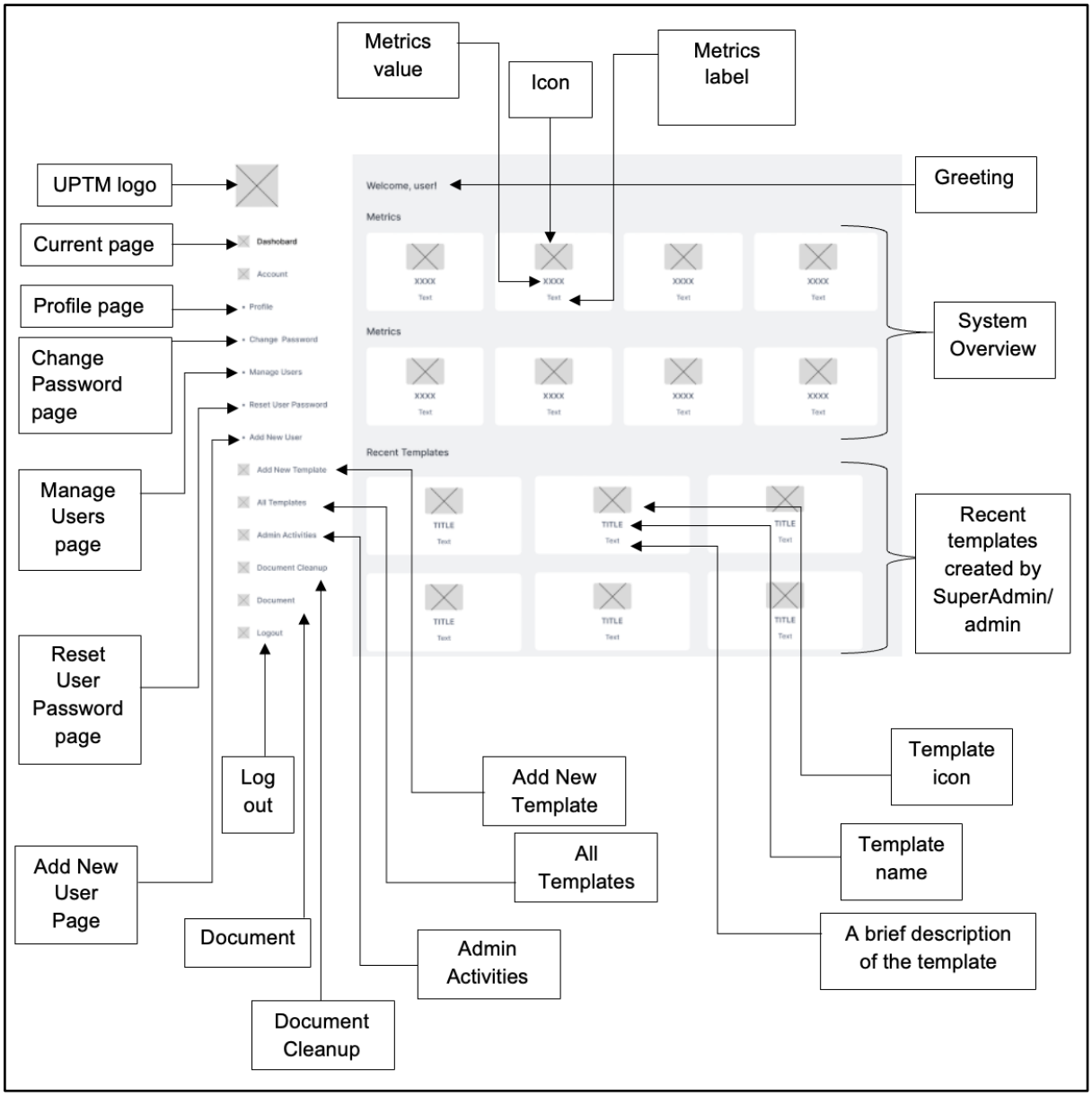



**Figure 4.5:** Dashboard (SuperAdmin)

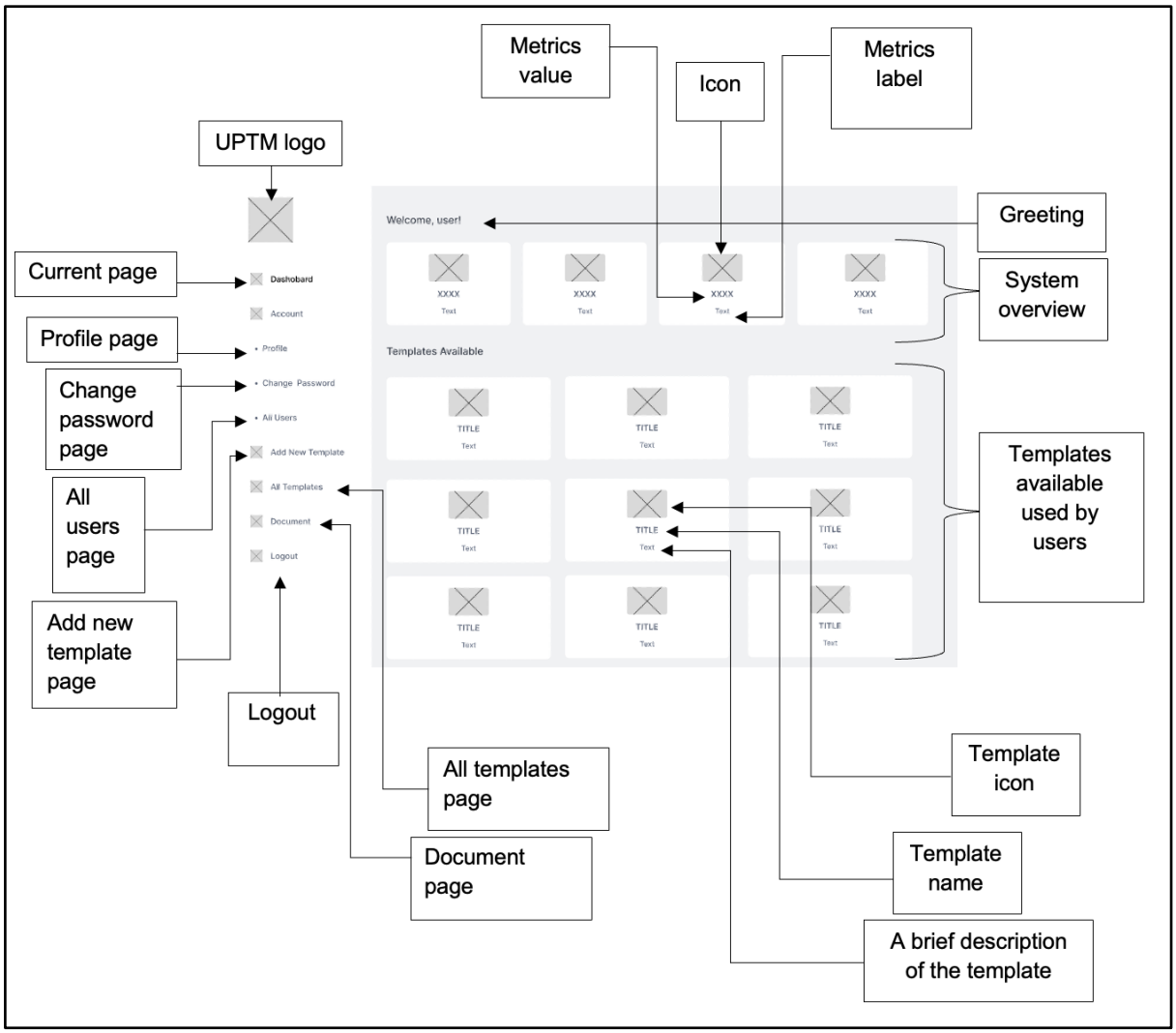



**Figure 4.6:** Dashboard (admin)

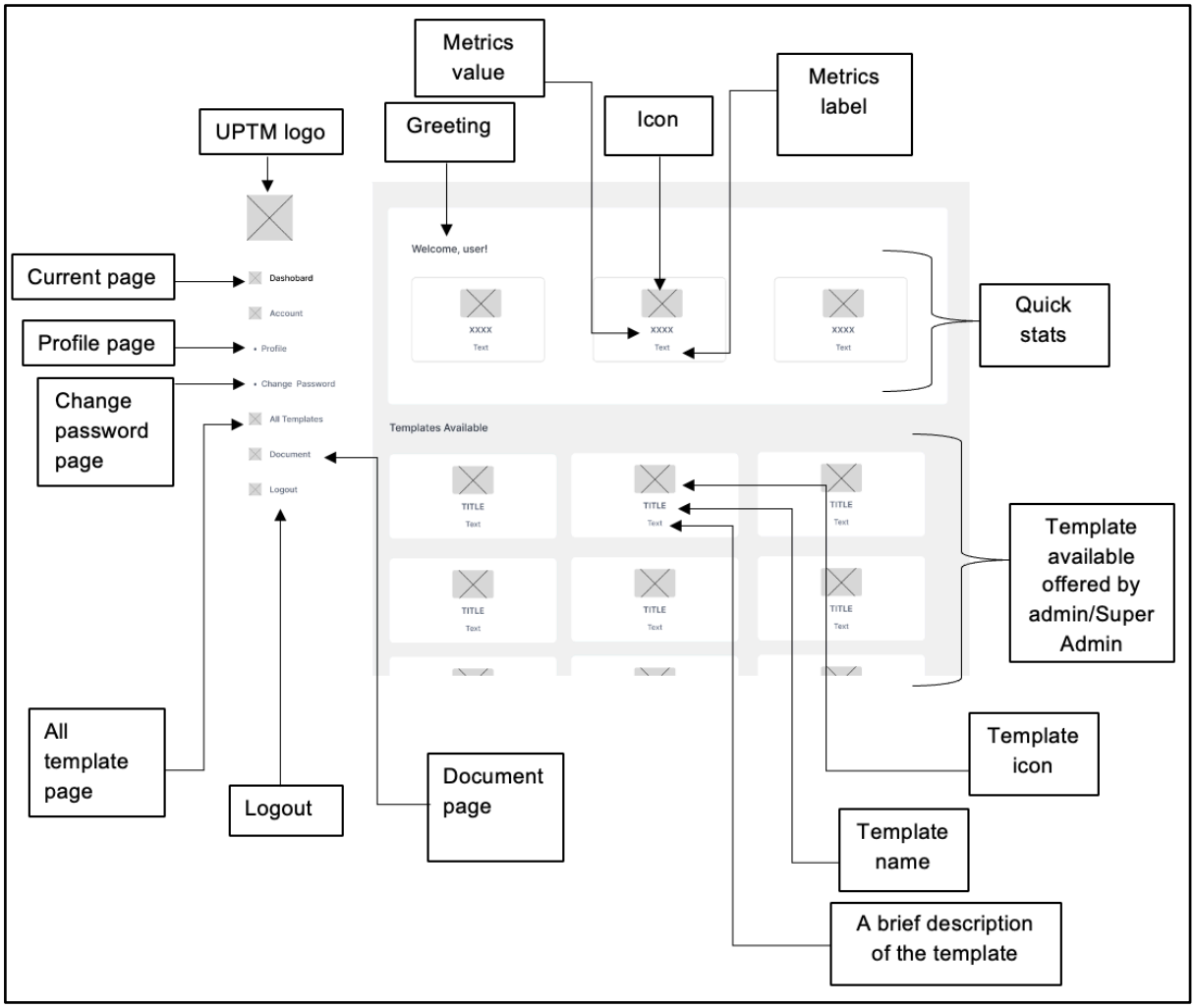



**Figure 4.7:** Dashboard (user)

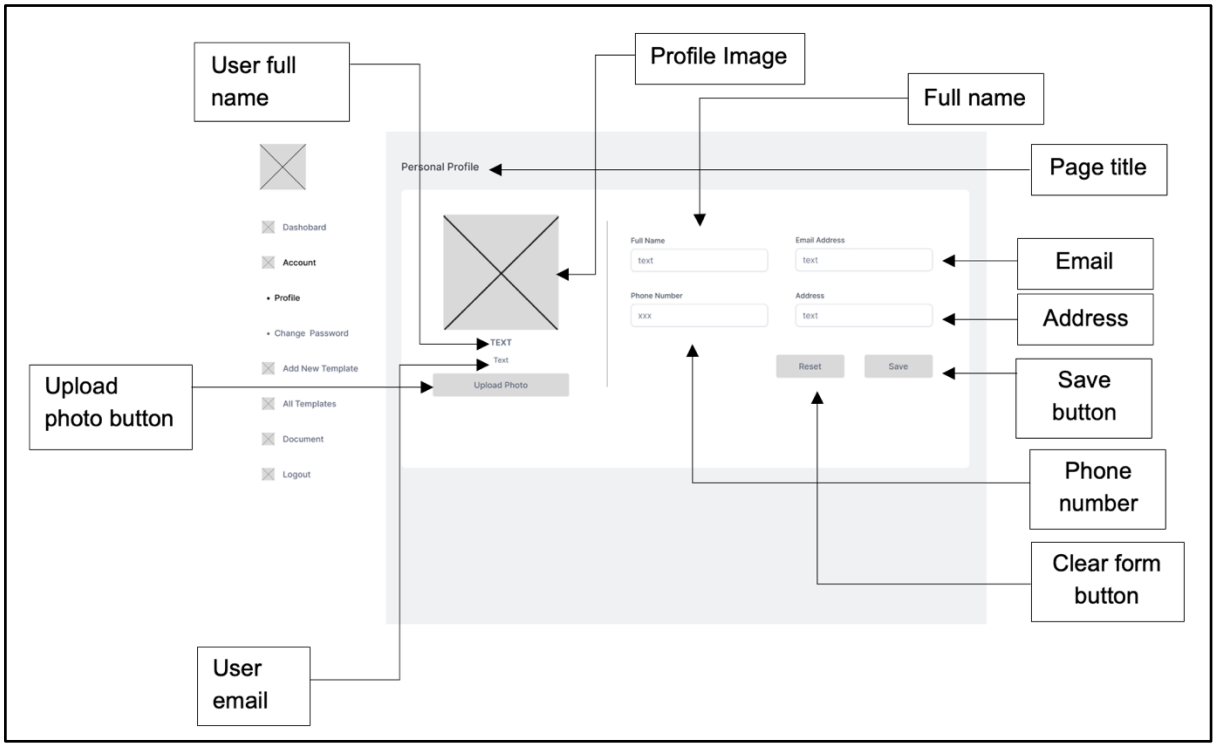



**Figure 4.8:** Profile (users/admin/SuperAdmin)

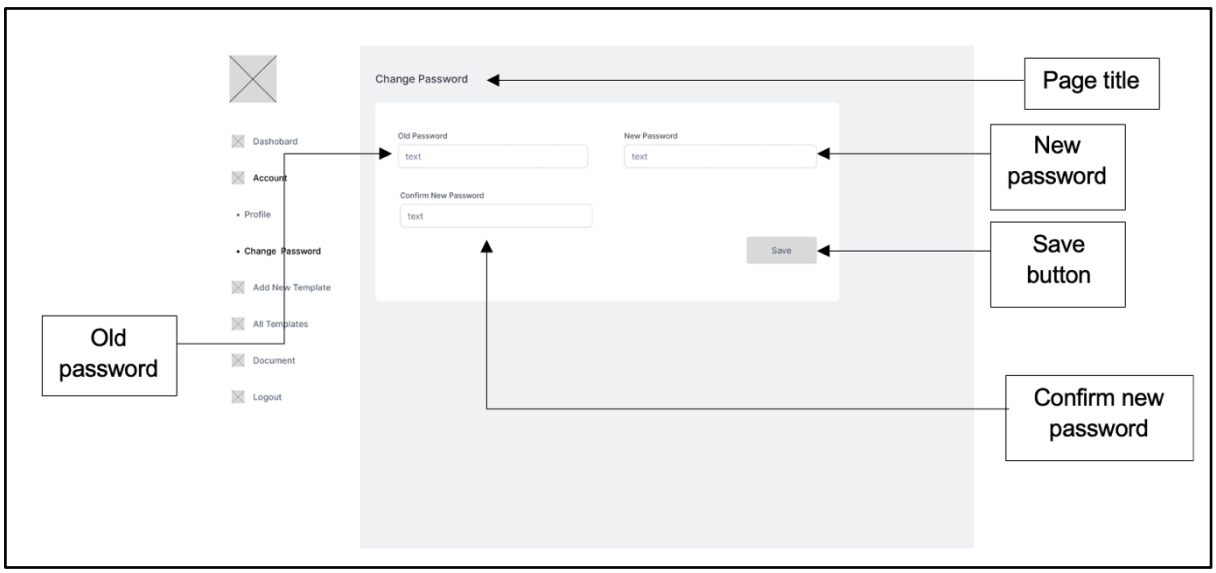



**Figure 4.9:** Change Password (users/admin/SuperAdmin)

**Figure 4.10:** All Users (admin)

**Figure 4.11.:** Manage Users (SuperAdmin)

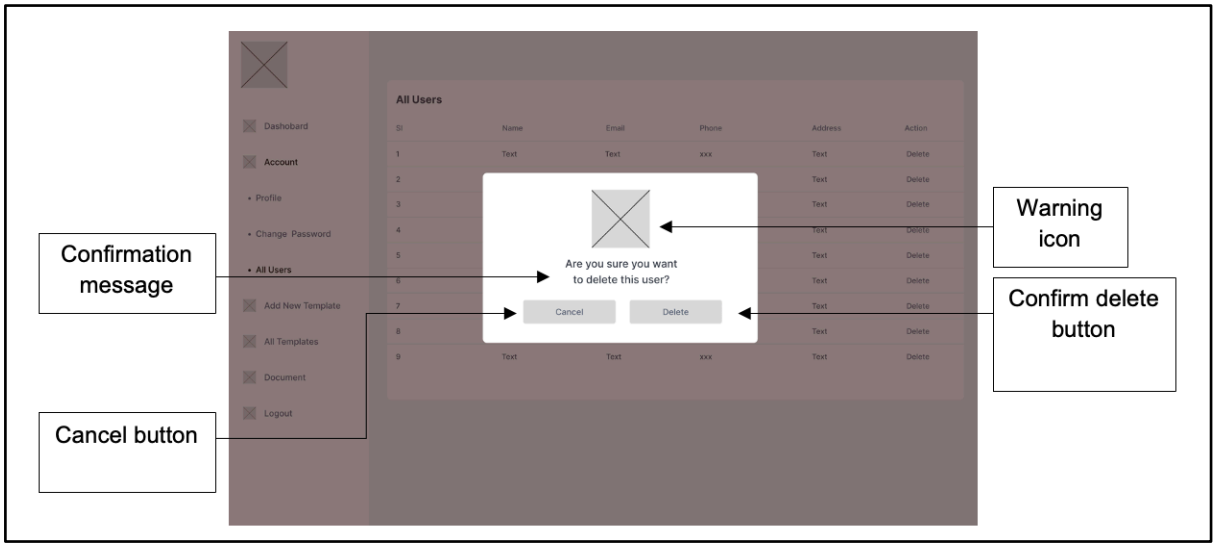



**Figure 4.12:** Delete Confirmation Modal (admin/SuperAdmin)

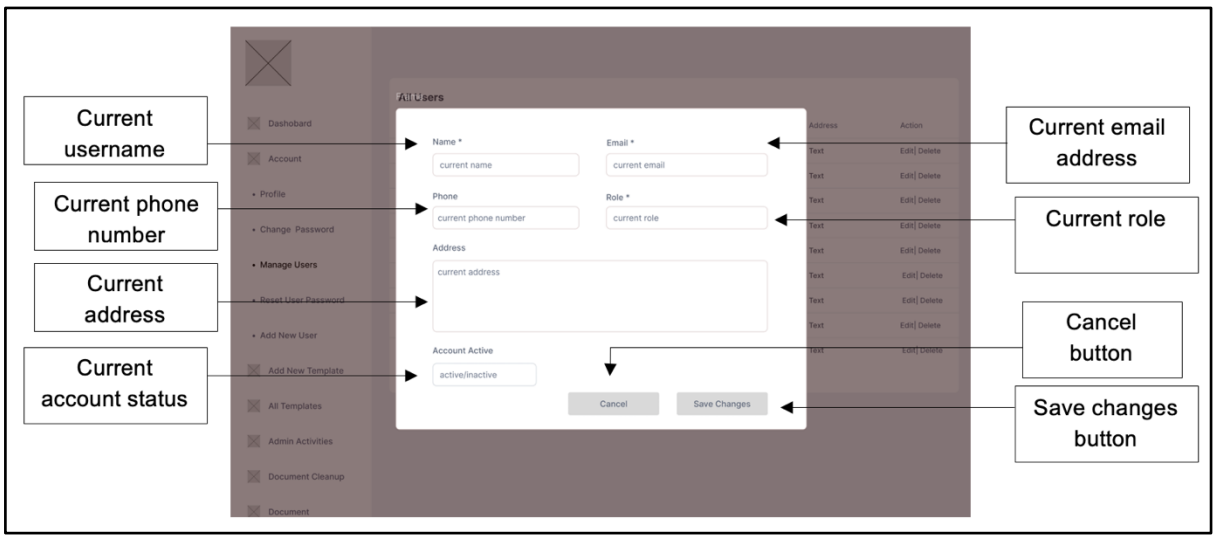



**Figure 4.13:** Edit Modal (SuperAdmin)

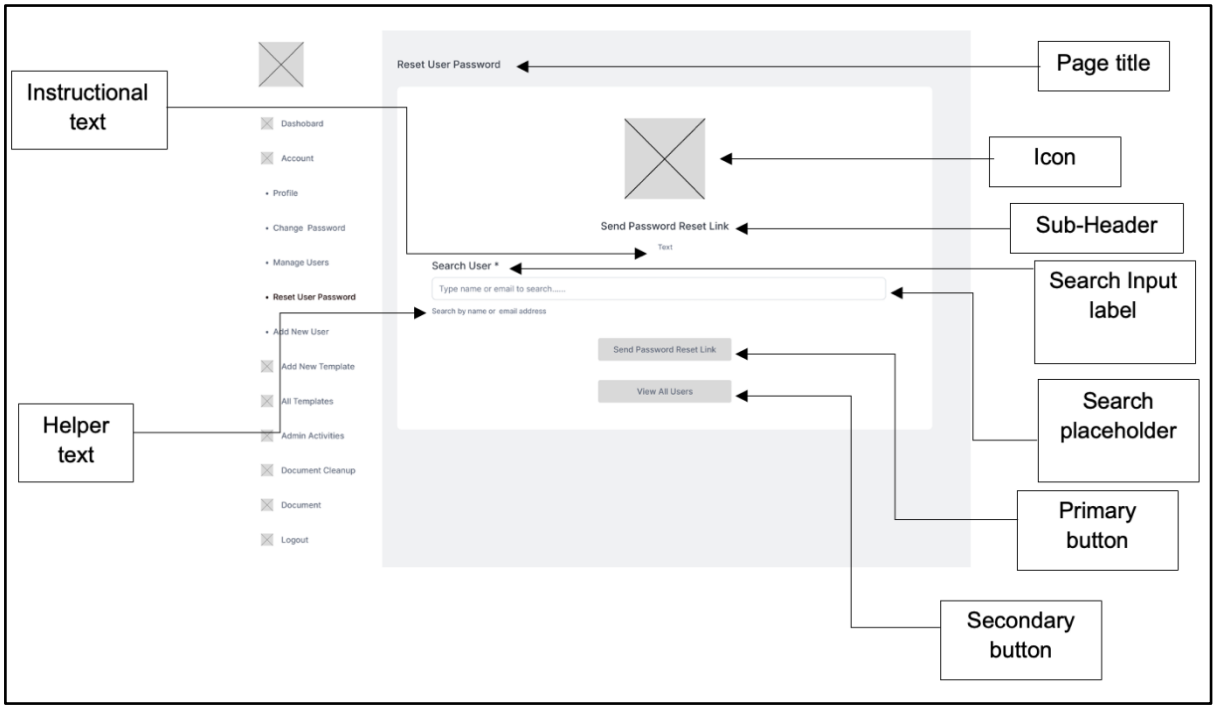



**Figure 4.14:** Reset User Password (SuperAdmin)

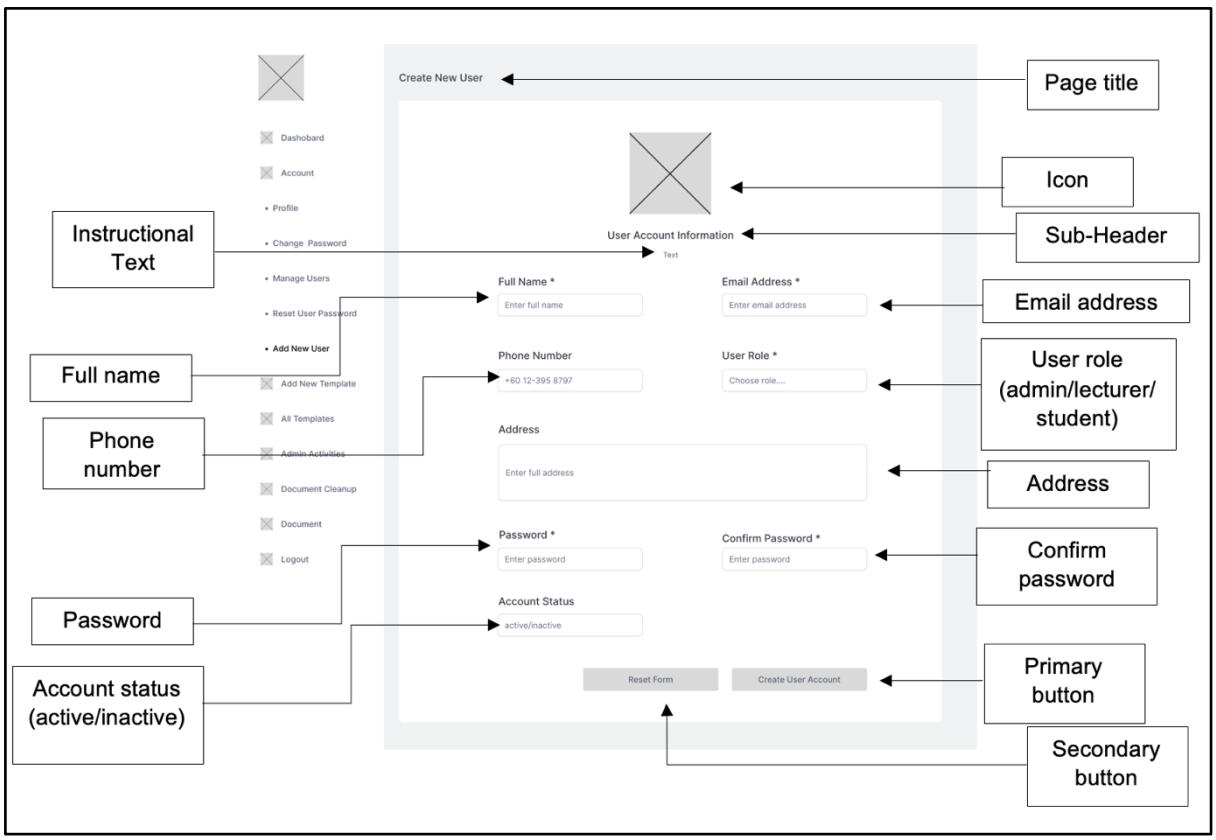



**Figure 4.15:** Create New User (SuperAdmin)

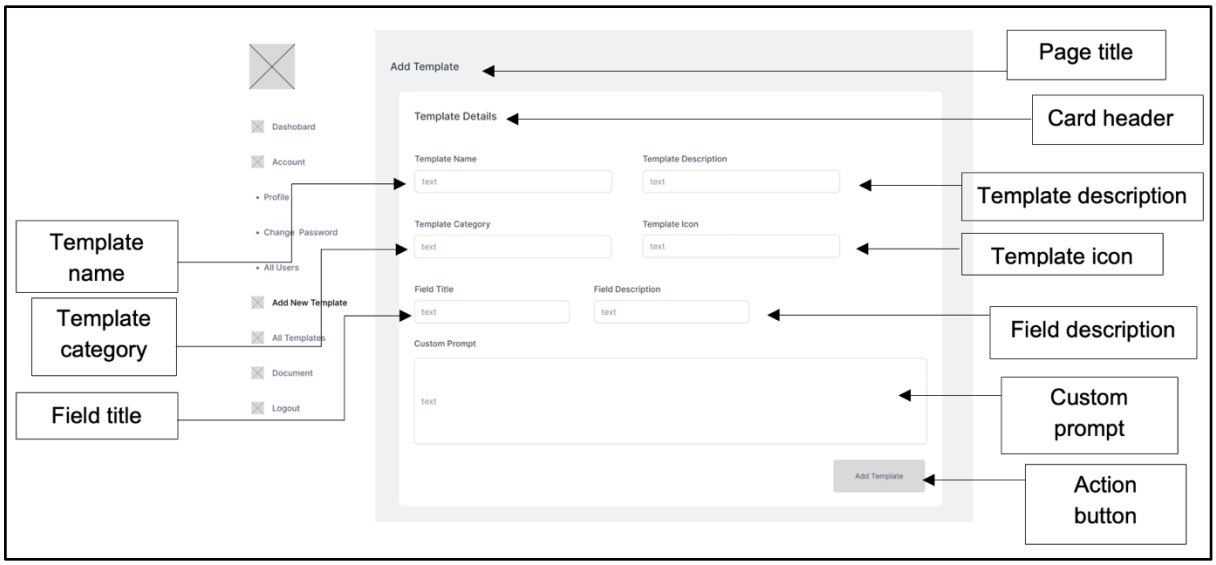



**Figure 4.16:** Add New Template (admin/superadmin)

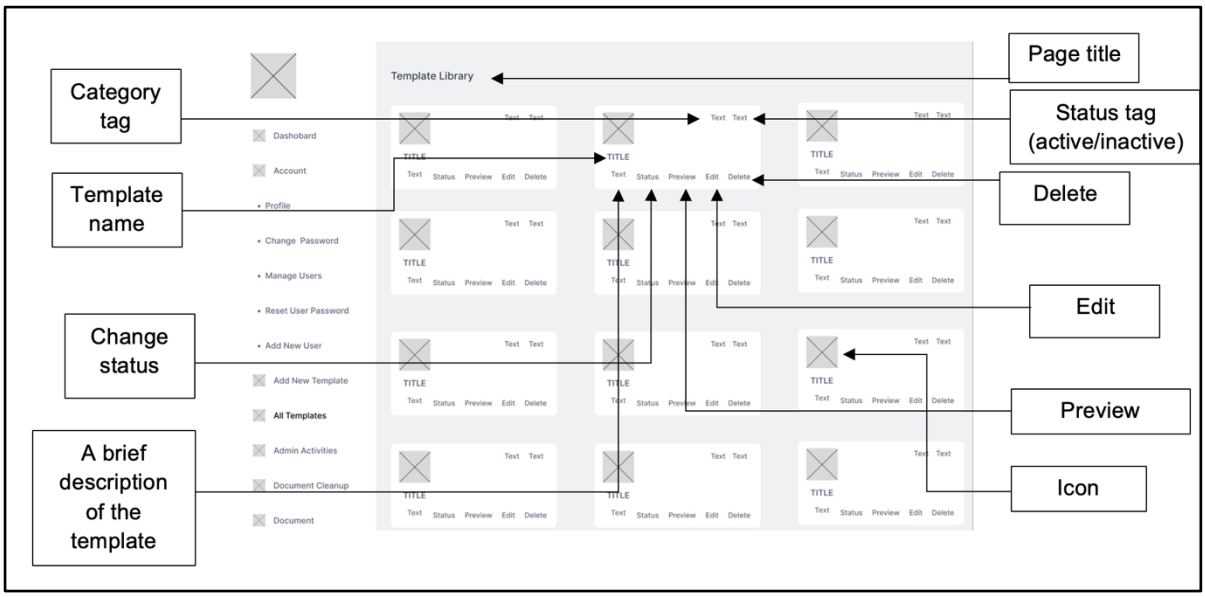



**Figure 4.17:** All Template (SuperAdmin)

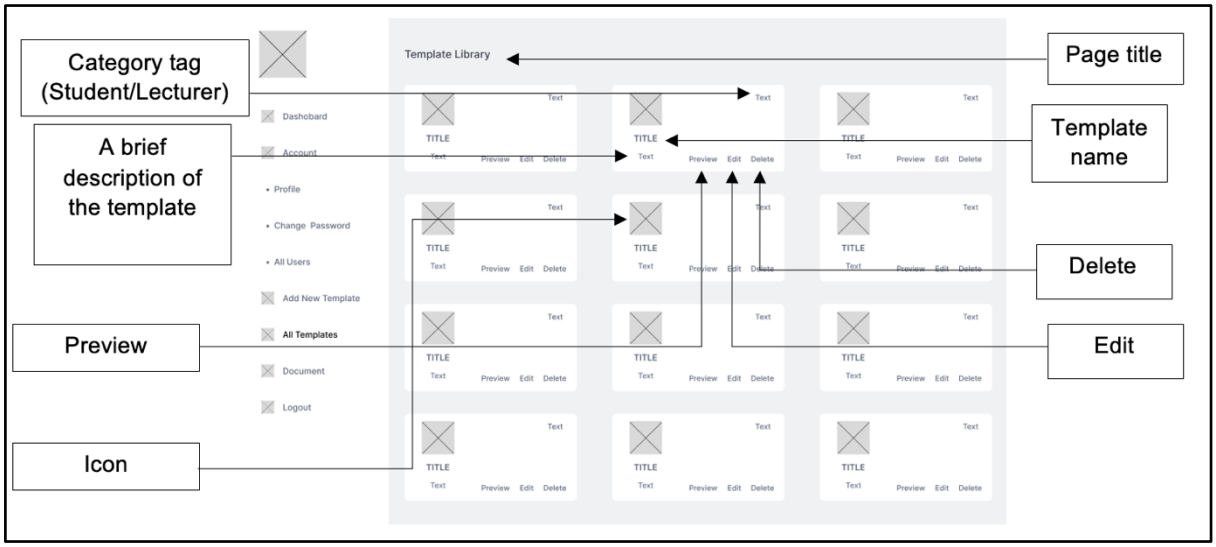



**Figure 4.18:** All Template (admin)

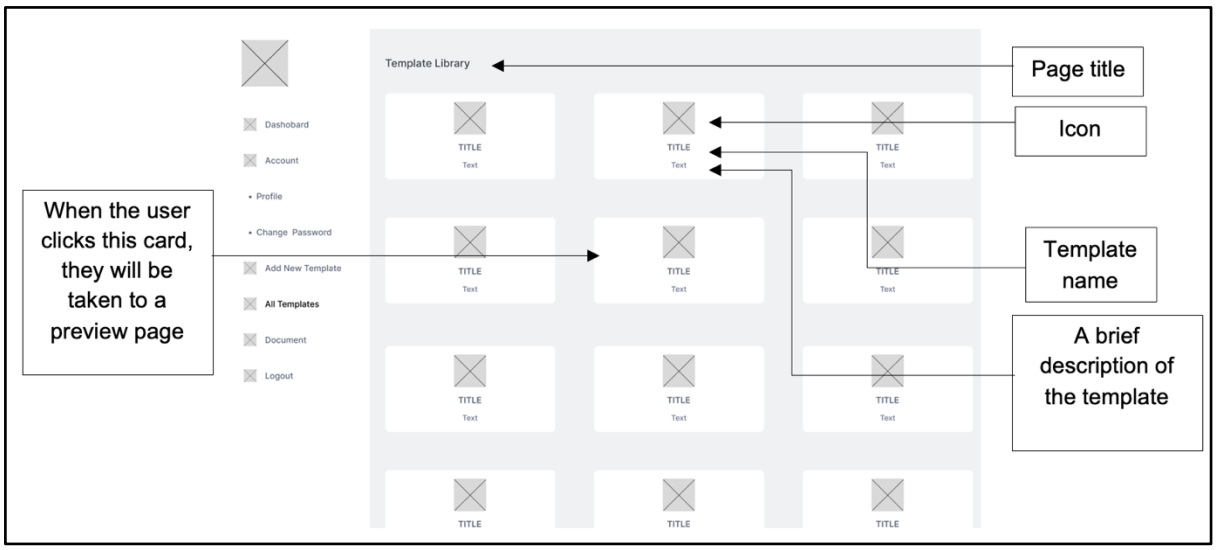



**Figure 4.19:** All Template (users)

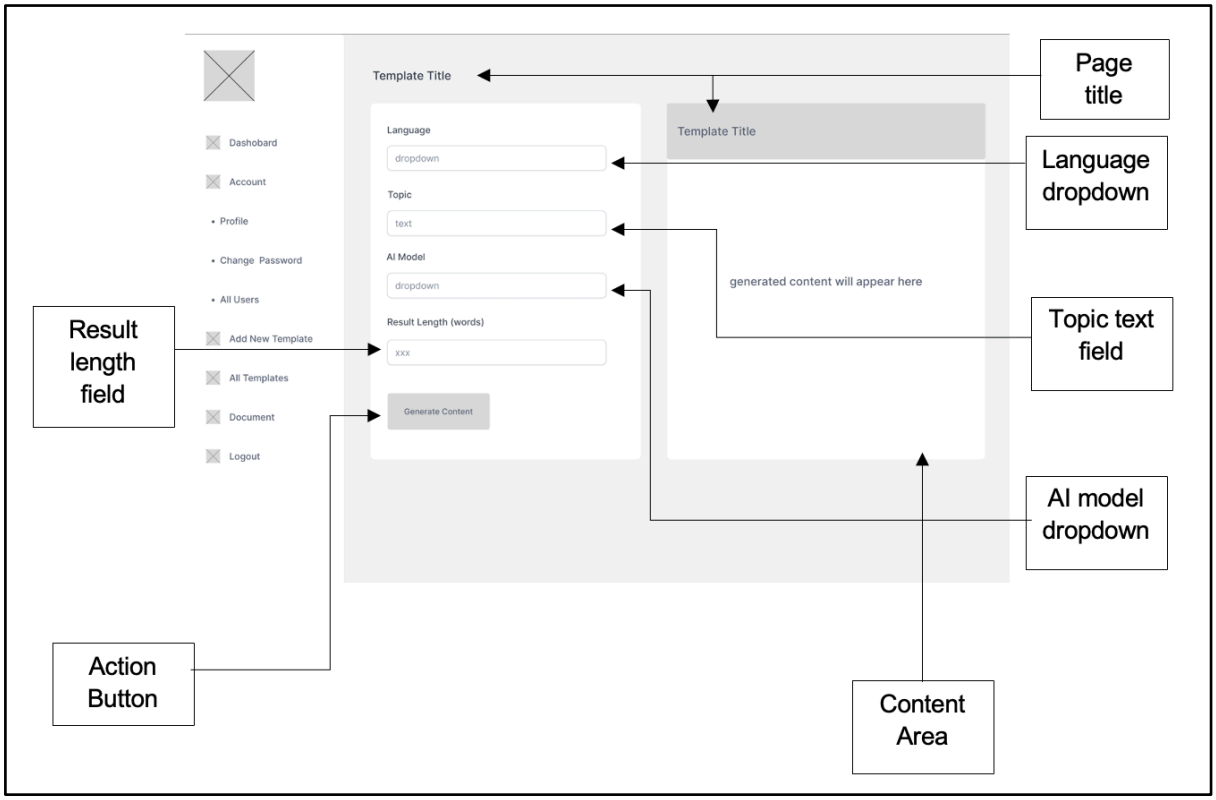



**Figure 4.20:** Preview Template (users/admin/SuperAdmin)

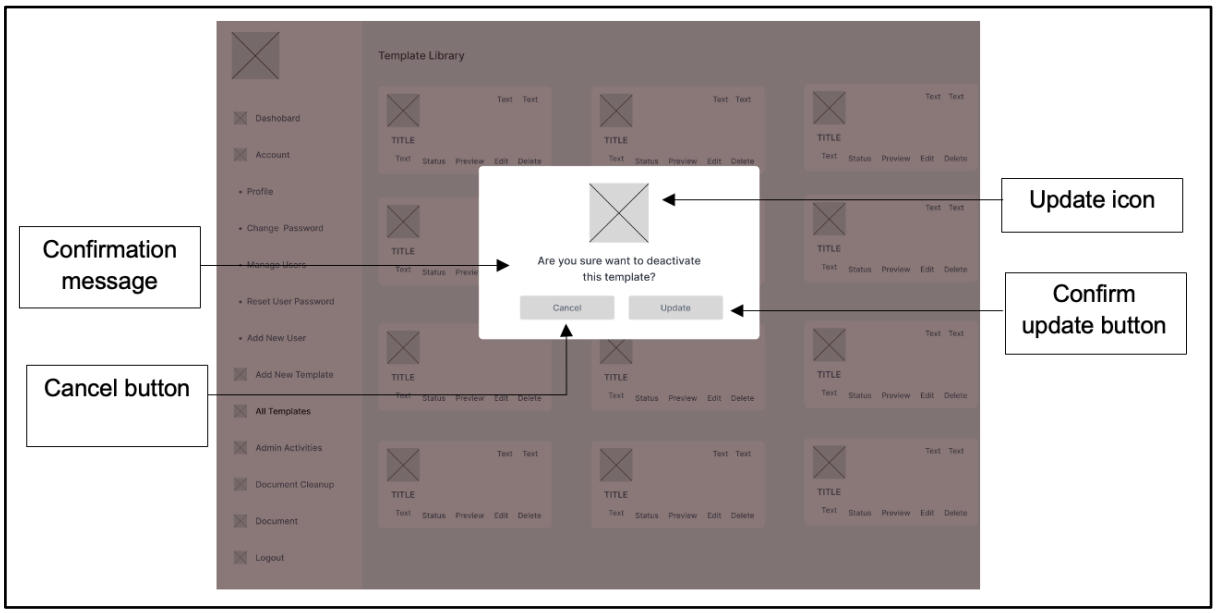



**Figure 4.21:** Change Status Template (SuperAdmin)

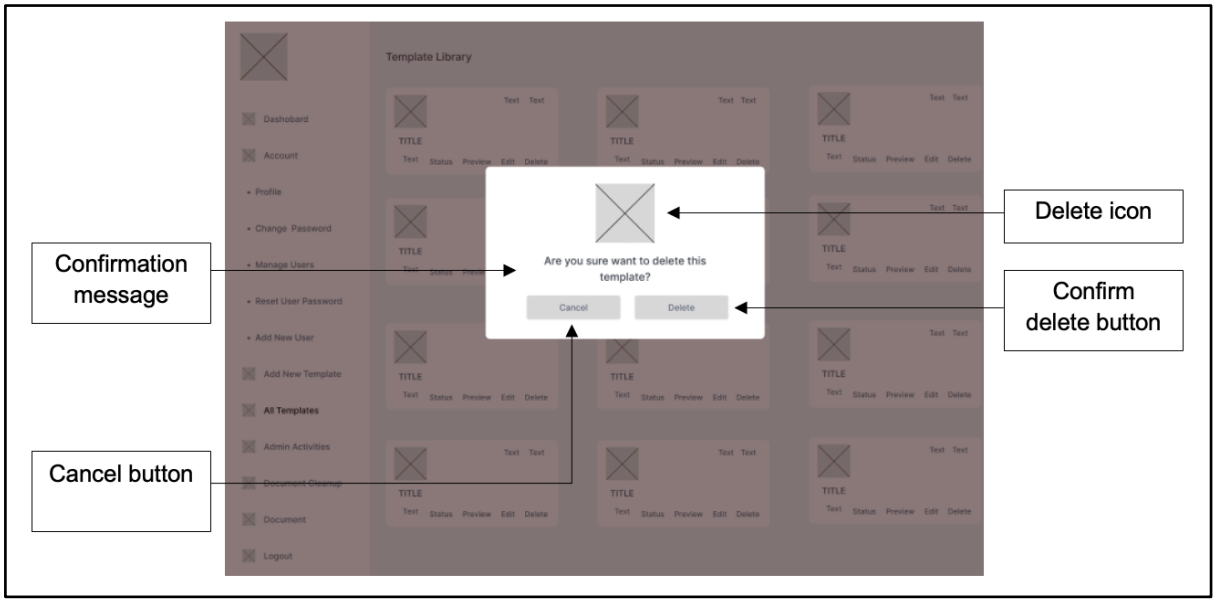



**Figure 4.22:** Delete Template Confirmation Modal

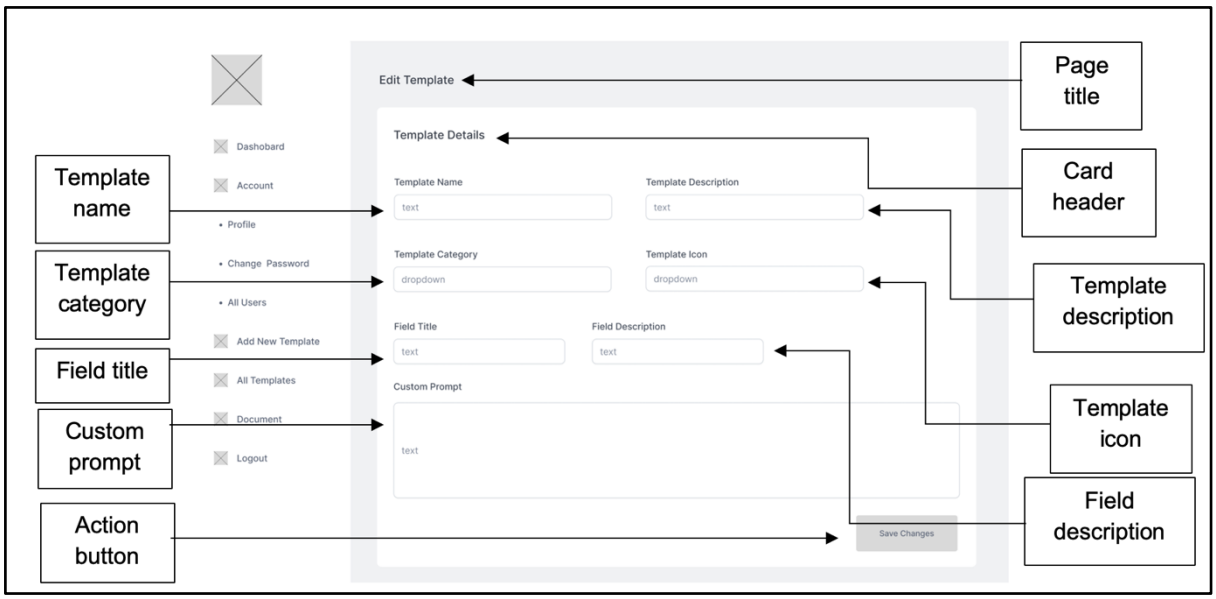



**Figure 4.23:** Edit Template (admin/SuperAdmin)

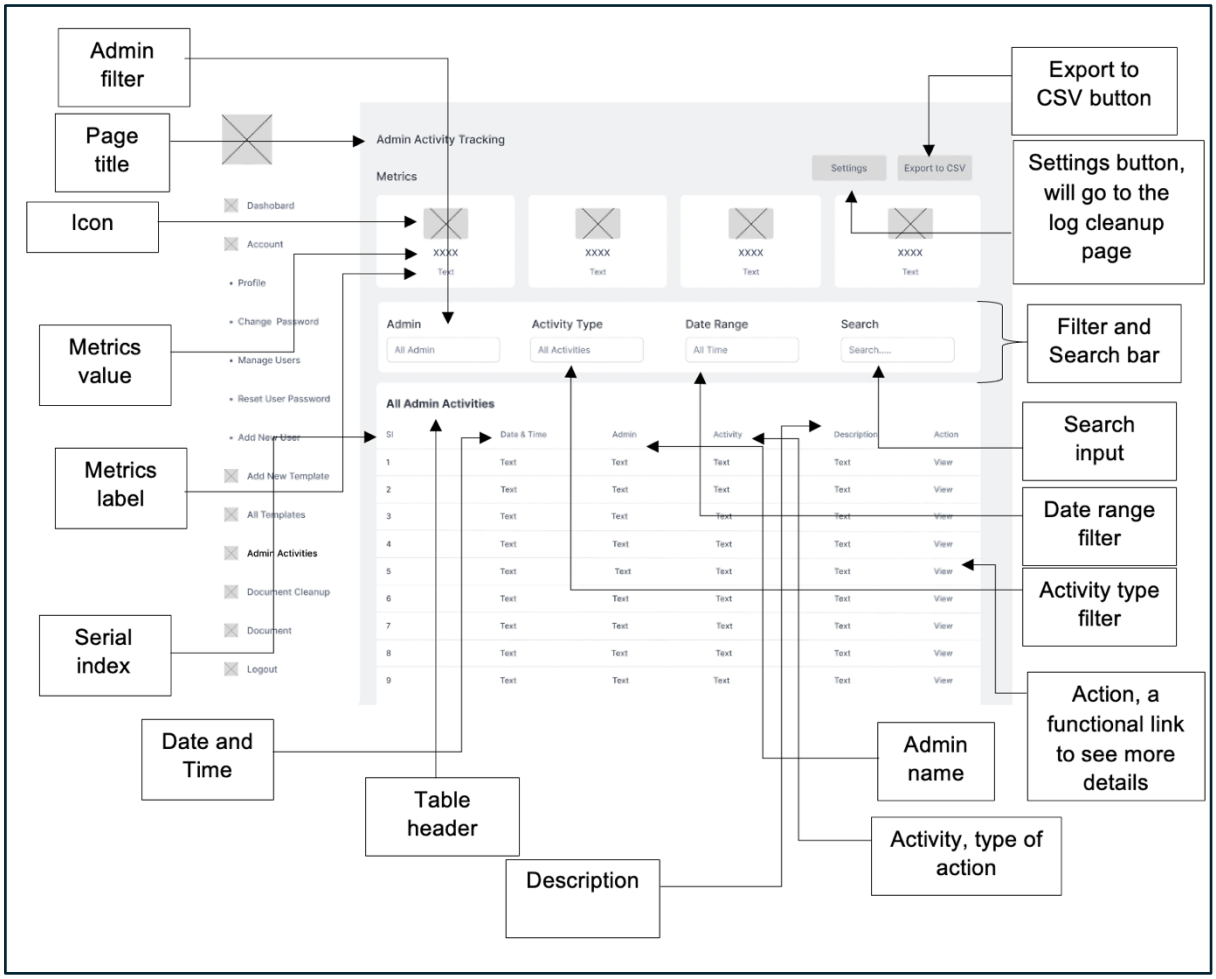



**Figure 4.24:** Admin Activities

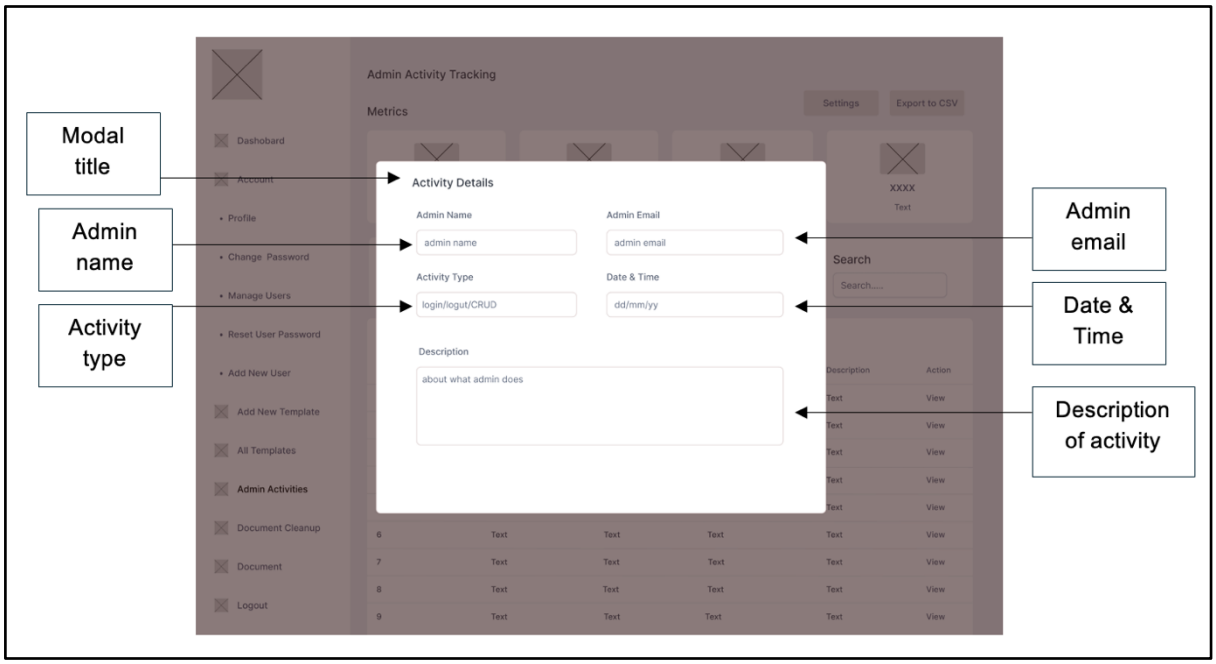



**Figure 4.25:** Activity Details

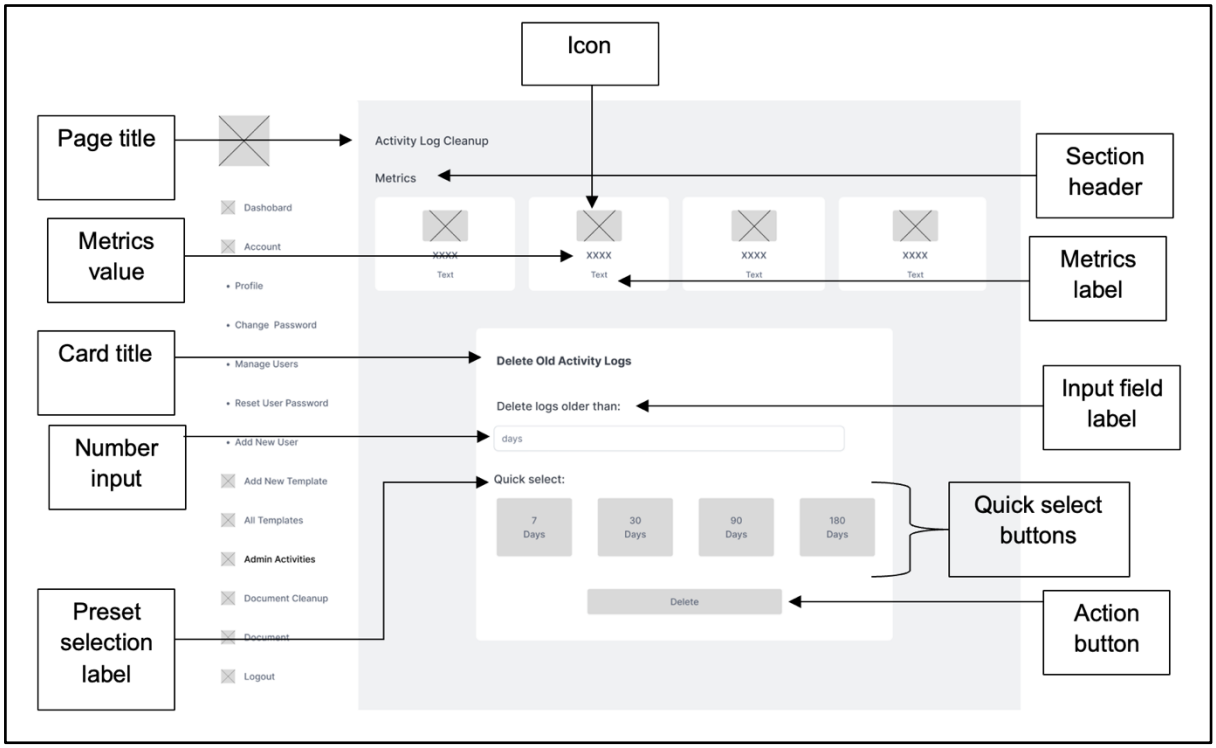



**Figure 4.26**: Activity Log Cleanup

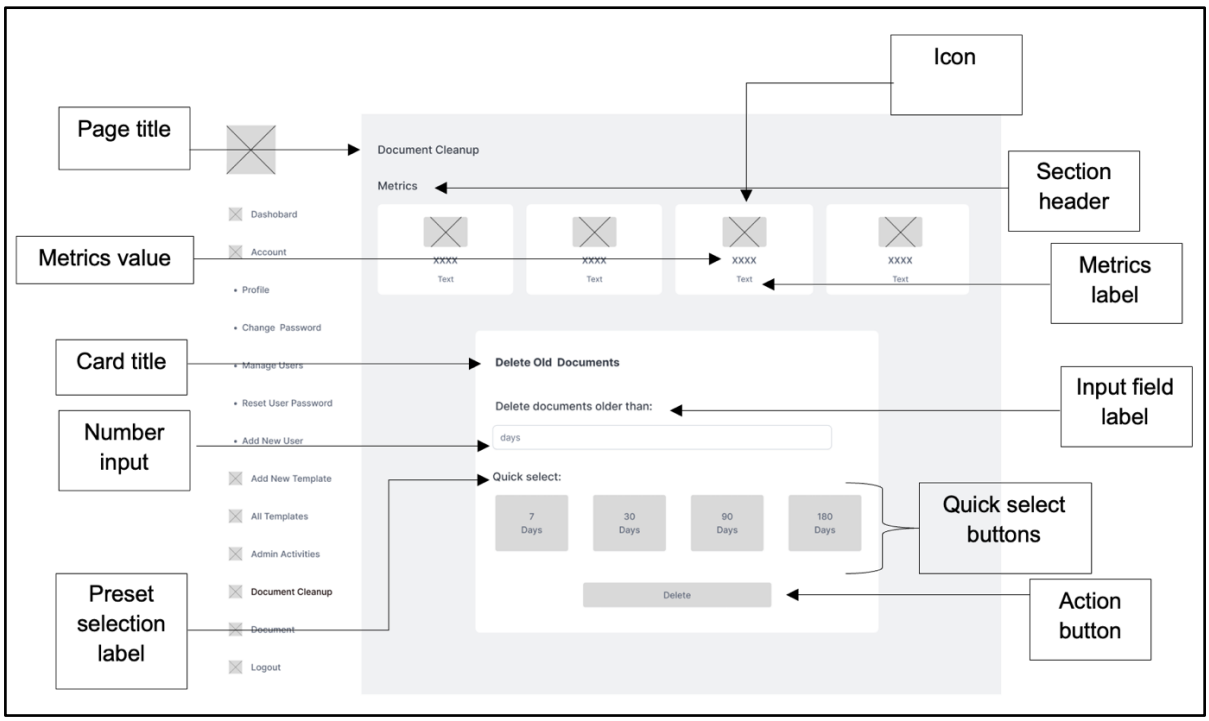



**Figure 4.27:** Document Cleanup

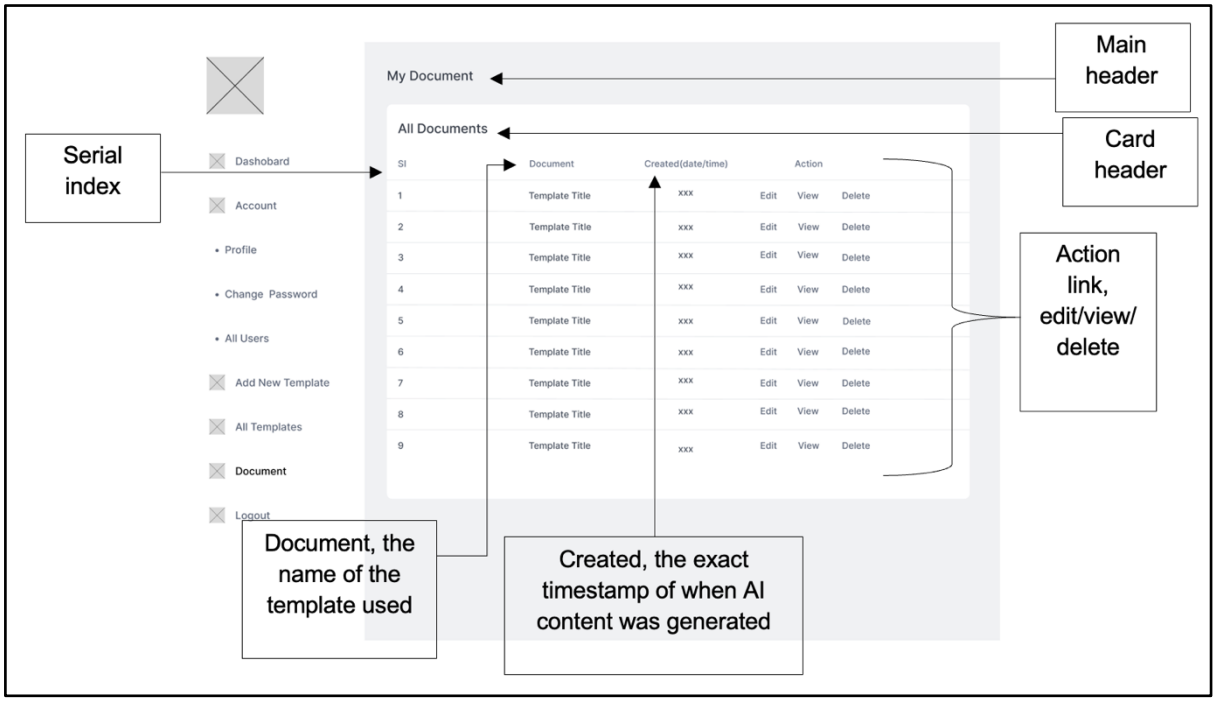



**Figure 4.28:** Document (users/admin/SuperAdmin)

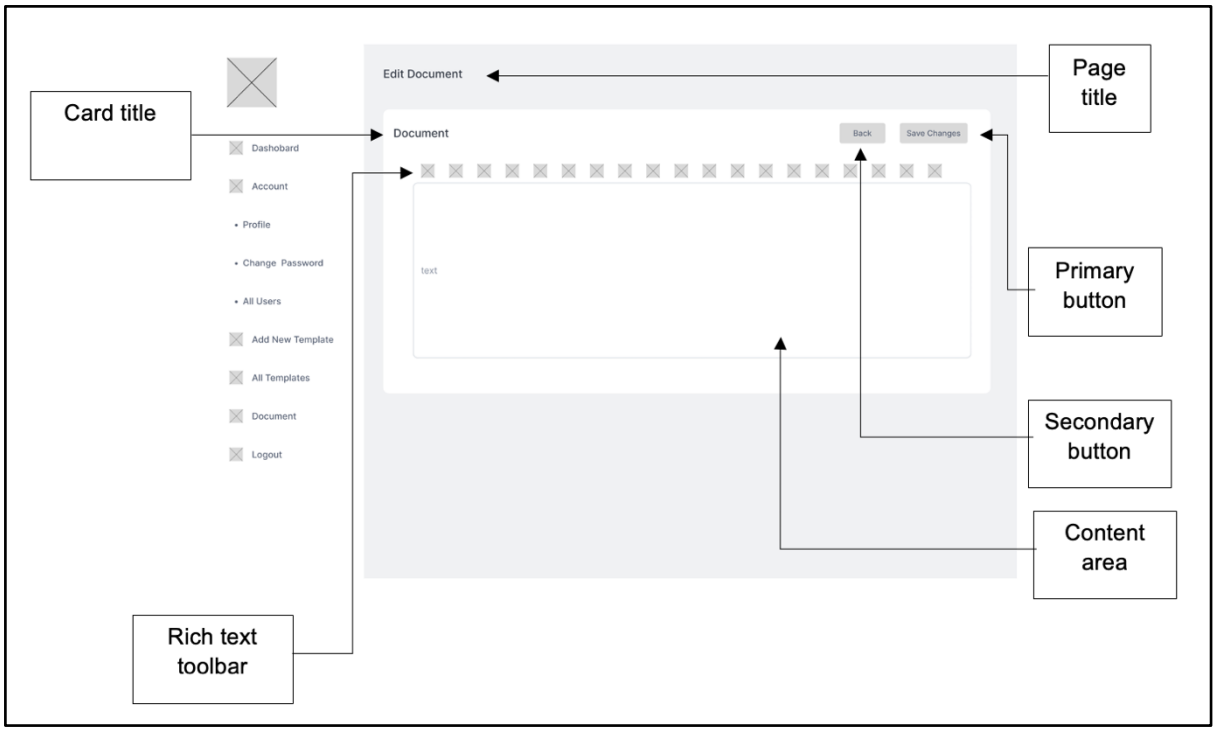



**Figure 4.29:** Edit Document (users/admin/SuperAdmin)

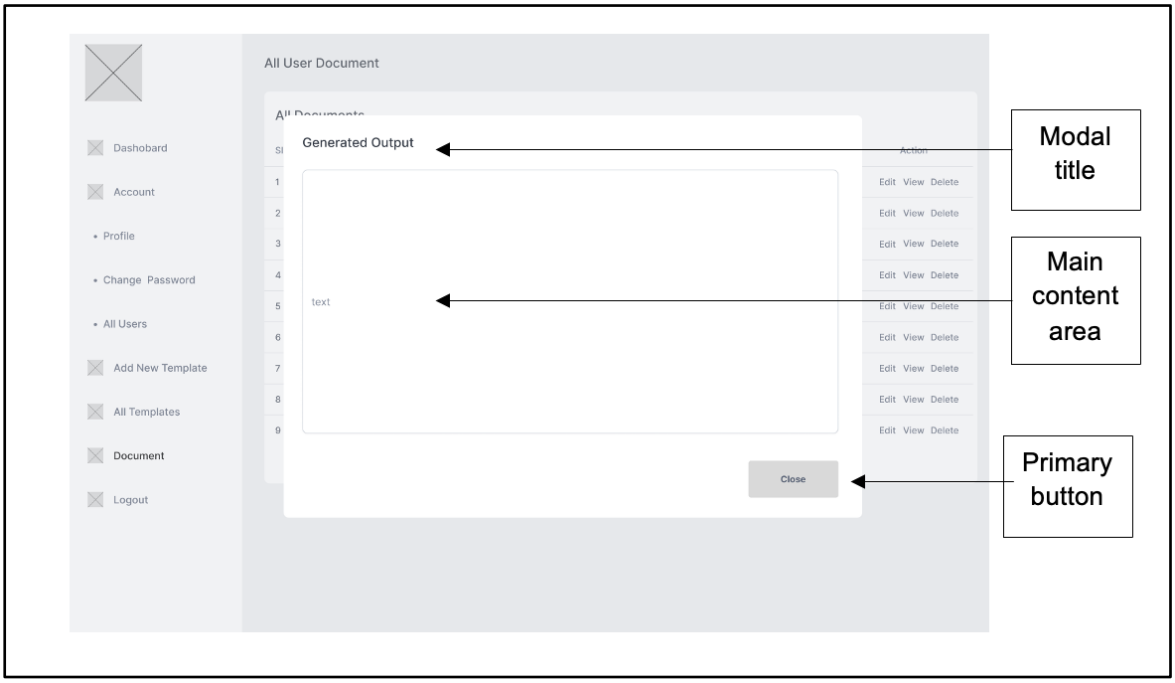



**Figure 4.30:** View Generated Output Modal (users/admin/SuperAdmin)

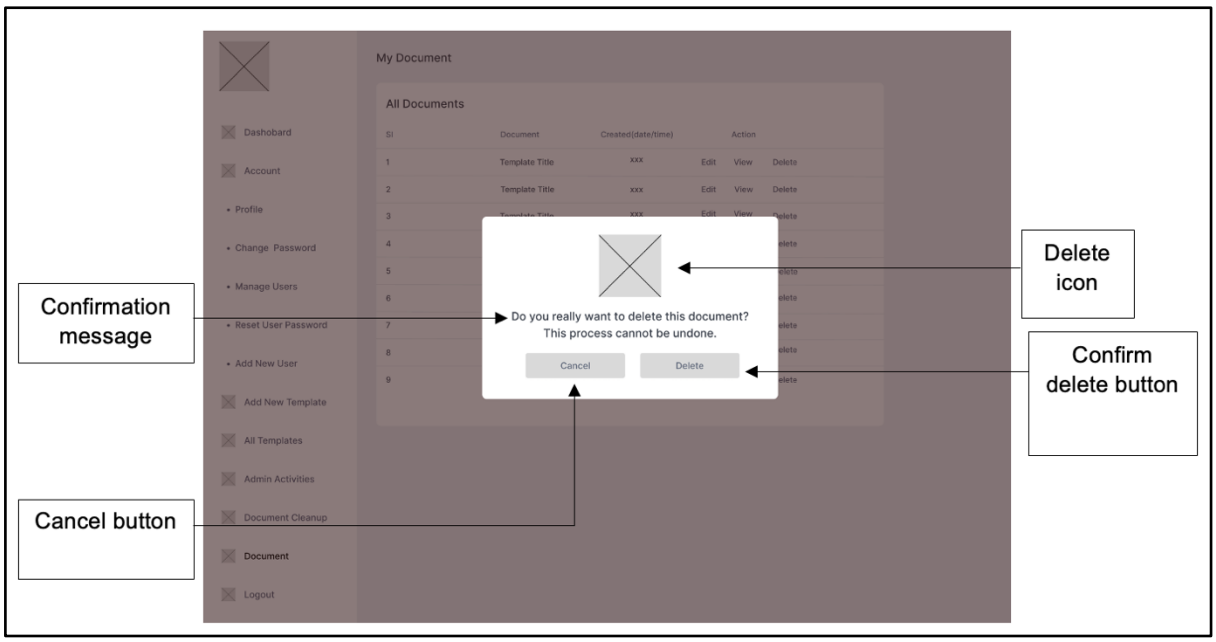



**Figure 4.31:** Delete Document Confirmation Modal

## 4.3 Database Design

The database design defines how all system data is structured and related. It uses relational database principles to maintain data consistency and integrity. The main entities in the system include users, templates, template_input_fields, and generated_contents. Each table is designed with specific relationships to support user activity and content generation features.

### 4.3.1 Data Dictionary

The data dictionary provides detailed information about the database tables, including field names, data types, descriptions, and relationships. It helps developers understand how data is stored, retrieved, and linked across the system.

#### 4.3.1.1 Users

This table stores information about system users, including login credentials, personal details, and usage records. It differentiates between admin and normal users to control system access and features.

**Table 4.1**: Users Data Dictionary

| Field Name | Data Type | Size | Format | Description | Key | Constraint/ Relationship |
|---|---|---|---|---|---|---|
| user_id | BIGINT | 20 | Number | Unique ID for each user | PK | Auto Increment |
| name | VARCHAR | 255 | Text | Full name of the user | - | Required |
| email | VARCHAR | 255 | Email | User's unique email address | - | Unique, Required |
| email_verified_at | TIMESTAMP | - | DateTime | Date and time when email was verified | - | Nullable |
| password | VARCHAR | 255 | Text | Hashed password for authentication | - | Required |

| | | | | | | |
|---|---|---|---|---|---|---|
| photo | VARCHAR | 255 | Text (URL/. Path) | Profile photo file path | - | Nullable |
| phone | VARCHAR | 255 | Text | User's contact number | - | Nullable |
| address | TEXT | - | Text | Home address of the user | - | Nullable |
| role | ENUM | ('superadmin','admin','user') | Enum | Defines user type (Superadmin, Admin, or User) | - | Default 'user' |
| status | VARCHAR | 255 | '1' = Active, '0' = Inactive | Indicates whether the user account is active or disabled | - | Default: '1' |
| remember_token | VARCHAR | 100 | Text | Token for authentication sessions | - | Nullable |
| created_at | TIMESTAMP | - | DateTime | Record creation timestamp | - | Auto generated |
| updated_at | TIMESTAMP | - | DateTime | Record last uodated timestamp | - | Auto generated |

#### 4.3.1.2 Templates

This table contains the predefined AI templates available in the system. Each template includes its title, description, category, and main prompt used for generating AI content.

**Table 4.2**: Templates Data Dictionary

| Field Name | Data Type | Size | Format | Description | Key | Constraint/ Relationship |
|---|---|---|---|---|---|---|
| template_id | BIGINT | 20 | Number | Unique ID for each template | PK | Auto increment |
| title | VARCHAR | 255 | Text | Template title | - | Required |
| description | TEXT | - | Text | Description explaining the template purpose | - | Required |
| category | VARCHAR | 100 | Text | Classifies the template into a specific user role | - | Required |
| icon | VARCHAR | 255 | Text (URL) | Icon image path for the template | - | Nullable |
| prompt | TEXT | - | Text | Main text prompt for content generator | - | Required |
| Is_active | BOOLEAN | 1 | 1 = Active, 0 = Inactive | Determines whether the template is currently available for users | - | Default: 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| created_by | BIGINT | 20 | Number | The ID of the user who created the template | FK | References (user_id) |
| created_at | TIMESTAMP | - | DateTime | Record creation timestamp | - | Auto generated |

#### 4.3.1.3 Template Input Fields

This table defines the input fields for each template, allowing the system to display the required fields dynamically based on the selected template.

**Table 4.3**: Template Input Fields Data Dictionary

| Field Name | Data Type | Size | Format | Description | Key | Constraint/ Relationship |
|---|---|---|---|---|---|---|
| template_input_field_id | BIGINT | 20 | Number | Unique ID for each input field | PK | Auto increment |
| template_id | BIGINT | 20 | Number | The template associated with this input field | FK | Reference emplate (template_id) |
| title | VARCHAR | 255 | Text | Input field title | - | Required |
| description | TEXT | - | Text | Input field description | - | Nullable |
| type | ENUM | (“text”, “textarea”) | Enum | Type of input field | - | Required |
| created_at | TIMESTAMP | - | DateTime | Record creation timestamp | - | Auto generated |

| | | | | | | |
|---|---|---|---|---|---|---|
| updated_at | TIMESTAMP | - | DateTime | Record last updated timestamp | - | Auto generated |

#### 4.3.1.4 Generated Contents

This table records the outputs generated by users using AI templates. It stores the input data, generated output, and word count for tracking and analysis purposes.

**Table 4.4**: Generated Contents Data Dictionary

| Field Name | Data Type | Size | Format | Description | Key | Constraint/ Relationship |
|---|---|---|---|---|---|---|
| generated_contents_id | BIGINT | 20 | Number | Unique ID for each generated content | PK | Auto increment |
| user_id | BIGINT | 20 | Number | The ID of the user who generated the content | FK | Reference users(user_id) |
| template_id | BIGINT | 20 | Number | The ID of the template used | FK | Reference users(template_id) |
| input | JSON | - | JSON | Input data provided by the user | - | Required |
| output | TEXT | - | Text | Generated content output | - | Required |
| created_at | TIMESTAMP | - | DateTime | Record creation timestamp | - | Auto generated |

| | | | | | | |
|---|---|---|---|---|---|---|
| updated_at | TIMESTAMP | - | DateTime | Record last updated timestamp | - | Auto generated |

#### 4.3.1.5 Admin Activities

This table records all activities performed by administrators within the system. It stores the activity type, description, affected entity, and additional metadata to support auditing, monitoring, and security tracking purposes.

**Table 4.5**: Admin Activities Data Dictionary

| Field Name | Data Type | Size | Format | Description | Key | Constraint/ Relationship |
|---|---|---|---|---|---|---|
| admin_activities_id | BIGINT | 20 | Number | Unique identifier for each admin activity log record | PK | Auto increment |
| admin_id | BIGINT | 20 | User ID reference | Identifies the admin user who performed the activity | FK | Reference users(user_id) |
| activity_type | ENUM | - | template_created, template_updated, user_deleted, login, logout. | Categorizes the type of action performed by the admin | - | Not Null |
| activity_description | VARCHAR | 255 | Readable text description | Stores a short explanation of the activity (e.g., “Admin | - | Not Null |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | updated template") | | |
| entity_type | VARCHAR | 255 | template, user, document | Indicates which entity/table was affected by the activity | - | Nullable |
| entity_id | BIGINT | 20 | Numeric entity reference ID | Stores the ID of the affected entity (template_id, user_id, etc.) | - | Nullable |
| metadata | JSON | - | JSON object (old values, new values, changes) | Stores extra details related to the activity for auditing purposes | - | Nullable |
| created_at | TIMESTAMP | - | DateTime | Record creation timestamp | - | Auto generated |
| updated_at | TIMESTAMP | - | DateTime | Record last updated timestamp | - | Auto generated |

### 4.3.2 Data Flow Diagram (DFD)

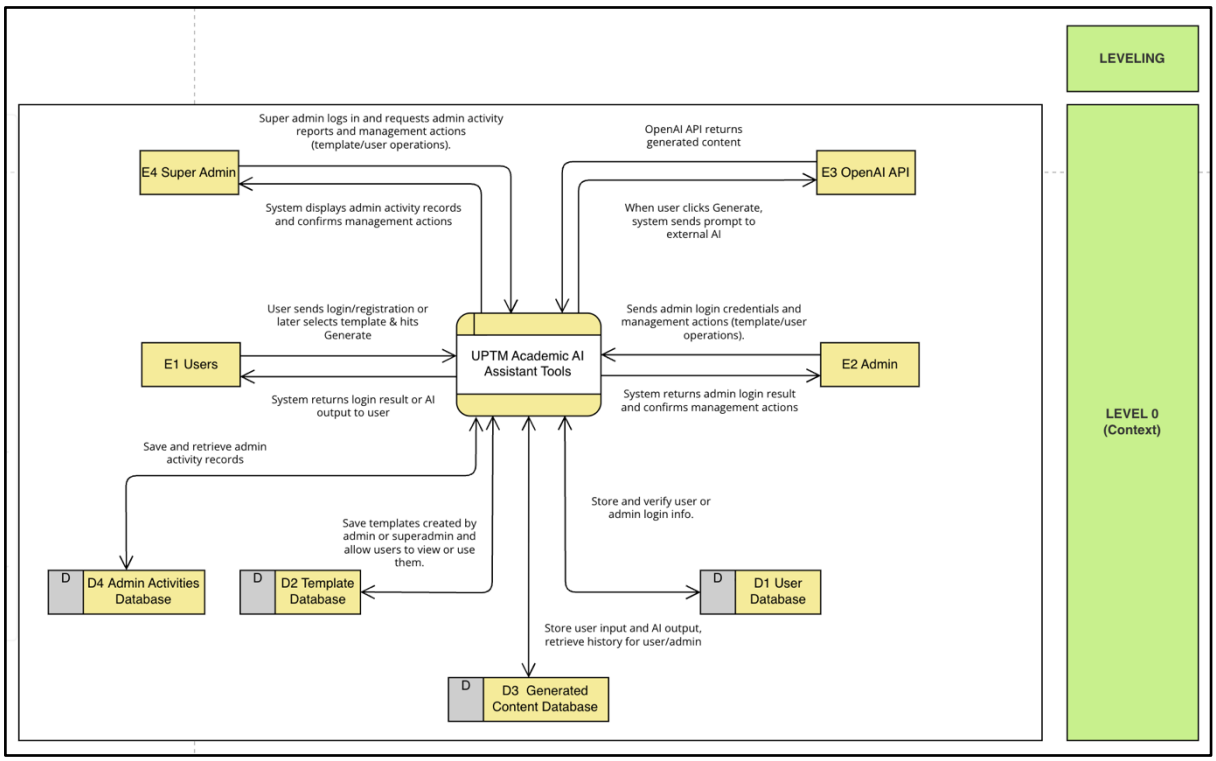



**Figure 4.15:** Data Flow Diagram Level 0

The Data Flow Diagram (DFD) level 0, or context diagram, for the UPTM Academic AI Assistant Tools illustrates the entire system as a single process interacting with external entities and internal data stores. The central process, labeled UPTM Academic AI Assistant Tools, is the sole functional component shown at this level, handling all data passing between four external entities: E1 Users, E2 Admin, E3 OpenAI API, and E4 Super Admin.

The E1 Users provide login/registration details and request content generation, receiving back login confirmations or the generated AI output. The E2 Admin provides credentials and sends management actions for templates or users and receives confirmation of these actions. The E4 Super Admin logs in to request admin activity reports and perform high-level management actions, receiving activity records and confirmation in return. To support its core functionality, the system relies on the E3 OpenAI API to receive prompts and, in turn, provide the processed, generated content back to the system.

The system interacts with four crucial data stores to maintain persistence and functionality. The D1 User Database is essential for storing and verifying all user, admin, and super admin login information during authentication. The D2 Template Database holds the templates

created by the admin, which are necessary for the content generation process. The D3 Generated Content Database is responsible for storing the history of user input and the resulting AI output, allowing users, superadmin and admins to retrieve this historical data. Lastly, the D4 Admin Activities Database is utilized to save and retrieve admin activity records, ensuring all management actions are logged and accessible for super admin reporting.

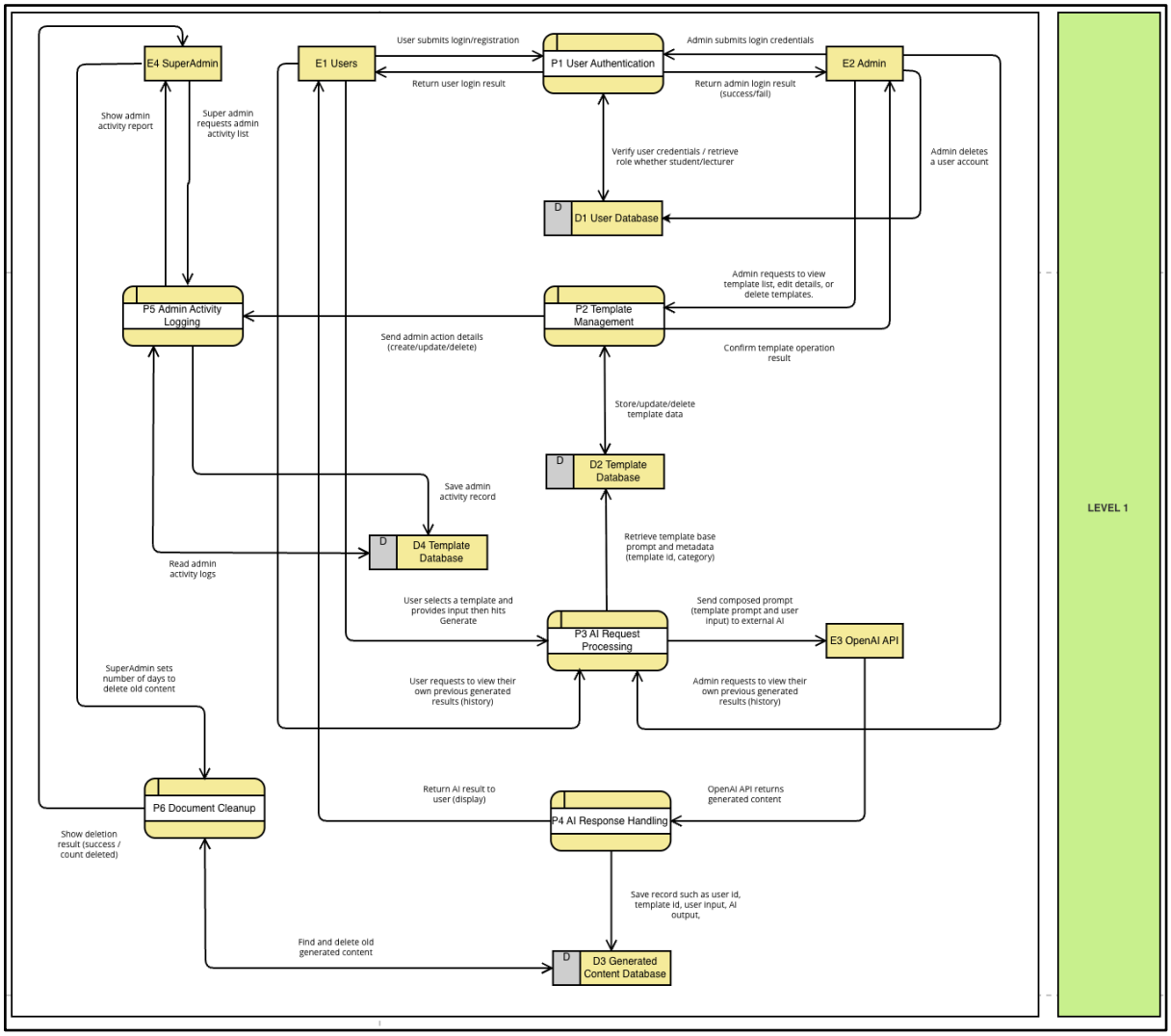



**Figure 4.16:** Data Flow Diagram Level 1

The Data Flow Diagram (DFD) Level 1 provides a detailed breakdown of the internal operations of the UPTM Academic AI Assistant Tools, decomposing the system into six main processes to reflect the updated administrative and maintenance capabilities.

The initial interaction point is P1 User Authentication, which handles all login and registration requests from E1 Users, E2 Admin, and the E4 SuperAdmin. This process verifies credentials and determines the user role whether student or lecturer by accessing the D1 User Database, returning a success or failure result to the respective actor. Additionally, the system allows E2 Admin to perform account maintenance by deleting user accounts directly within the D1 User Database.

Administrative functions and oversight are managed through P2 Template Management and P5 Admin Activity Logging. P2 allows the admin to request to view the template list, edit details, or delete templates within the D2 Template Database, confirming the result back to the admin. Simultaneously, P2 sends these action details to P5, which saves the record into the D4 Admin Activities Database (labeled as D4 Template Database in the diagram). This logging process enables the E4 SuperAdmin to request activity lists and receive comprehensive activity reports, ensuring full accountability of administrative changes.

The core function of the system is encapsulated in P3 AI Request Processing and P4 AI Response Handling. When an E1 User selects a template and provides input, P3 retrieves the template base prompt and metadata from the D2 Template Database, composes the final prompt, and sends it to the E3 OpenAI API. P3 also serves as the retrieval point for history, handling requests from both Users and Admins to view their own previous generated results. Following generation, P4 receives the content from the E3 OpenAI API, saves the complete record (including User ID, Template ID, input, and AI output) into the D3 Generated Content Database, and returns the final AI result to the user for display.

Finally, the system includes P6 Document Cleanup to handle data retention and system optimization. This process is triggered when the E4 SuperAdmin sets a specific number of days for content expiration. P6 then identifies and deletes old records within the D3 Generated Content Database and returns a success status along with the count of deleted documents to the SuperAdmin. This structure highlights a clear separation of concerns, ranging from secure authentication and audited administrative management to a robust AI processing and maintenance workflow.

### 4.3.3 Entity Relational Diagram (ERD)

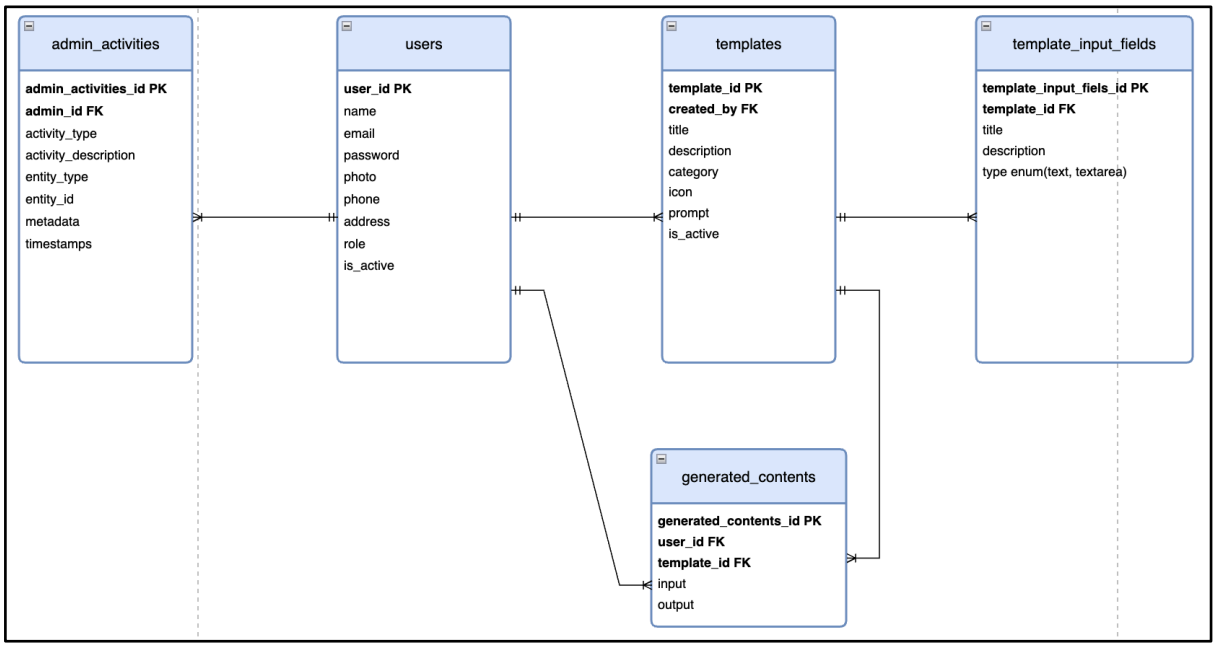



**Figure 4.17:** Entity Relational Diagram

The Entity-Relationship Diagram (ERD) for the UPTM Academic AI Assistant Tools illustrates the database structure and the relationships between the system's key entities. The system consists of five main entities: users, templates, template_input_fields, generated_contents, and admin_activities.

The users entity stores all registered user information and is uniquely identified by user_id (PK). It includes attributes such as name, email, password, phone, address, role, photo, and account status (is_active). This entity serves as the core of the system, as users interact with templates and generate AI-based academic content.

The templates entity represents the AI content generation templates available in the system. Each template is uniquely identified by template_id (PK) and contains important metadata such as title, description, category, icon, prompt, and activation status. Templates are linked to users through the created_by (FK) attribute, establishing a one-to-many relationship, where one admin user can create multiple templates, but each template is created by only one user.

Templates also have a one-to-many relationship with the template_input_fields entity. This table is identified by template_input_fields_id (PK) and includes template_id (FK). It stores the

dynamic form fields required for each template, such as field title, description, and input type (text or textarea). This allows templates to support structured user input in a flexible way.

The generated_contents entity acts as the central recording and history table of the system. It is uniquely identified by generated_contents_id (PK) and connects both users and templates through user_id (FK) and template_id (FK). This establishes a many-to-many interaction between users and templates, mediated through generated contents: a user can generate content using many templates, and a template can be used by many users. The table stores the input provided by the user and the AI-generated output, serving as the system's academic content archive.

Finally, the admin_activities entity supports administrative auditing and monitoring. It is uniquely identified by admin_activities_id (PK) and linked to the users table through admin_id (FK). This table records actions performed by admins, including activity type, description, and the affected entity. The attributes entity_type and entity_id are used to indicate what data or table was involved in a specific activity, ensuring accountability, transparency, and effective system management.

## 4.4 Flow of the System

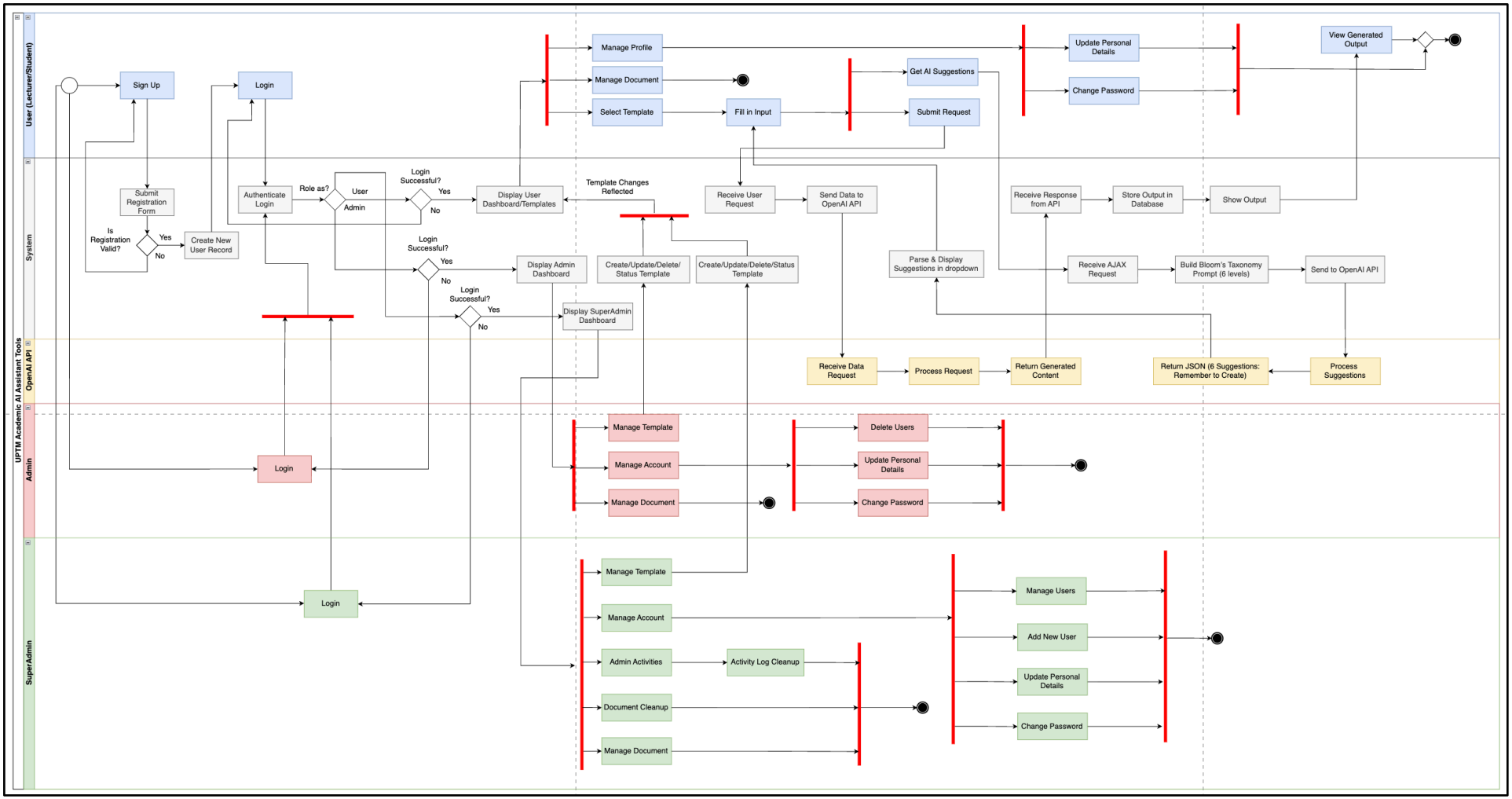



**Figure 4.18:** Flow of the System

The swimlane diagram illustrates the process flow of the UPTM Academic AI Assistant Tools by showing the interactions among five main lanes: User, System, OpenAI API, Admin, and Superadmin. The process begins in the User Lane, where a user initiates the system by performing either a sign-up or login action. During registration, the system validates the submitted information and creates a new user record if the data is valid. For login, the system authenticates the credentials and determines the user's role. Based on this role verification, the system directs the user to the appropriate dashboard, either the user dashboard, admin dashboard, or superadmin dashboard, ensuring role-based access control throughout the system.

In the User Lane, the primary workflow focuses on profile management and document generation. A logged-in user may update personal details, change passwords, and access the document management page, which stores a history of previously generated outputs based on selected templates. When generating content, the user selects a template and fills in the required input fields. At this point, the workflow introduces a decision point where the user may either submit the request directly for final content generation or choose the AI suggestion option. If the AI suggestion option is selected, the system processes the input via an AJAX request, constructs an academic prompt based on Bloom's Taxonomy levels, and sends it to the OpenAI API. The OpenAI API generates structured academic suggestions and returns them to the system, which then displays the suggestions to the user for review and selection. The user may refine the input using these suggestions before proceeding to submit the final request.

Once the user submits the final request, the System Lane handles the communication with the OpenAI API by sending the completed input data for content generation. The OpenAI API processes the request and returns the generated academic output. The system then stores the output in the database and presents the result to the user, allowing it to be viewed and managed within the document history. This sequence represents the core functionality of the system, where AI is used to assist users in producing academic content efficiently and systematically.

The Admin Lane represents system-level operational control. Admin users are responsible for managing templates by creating, updating, change status, or deleting them, managing user accounts such as removing inactive users, and maintaining their own account information. These actions ensure that templates remain relevant, users are properly managed, and the system operates securely. The Superadmin lane represents the highest authority within the system, providing full control over admins, users, templates, documents, and system logs.

Superadmins can manage admin accounts, monitor system activities, perform data cleanup, and enforce system governance, ensuring long-term stability, security, and accountability of the platform.

## 4.5 Implementation

This section describes the process of implementing the UPTM Academic AI Assistant Tools based on the system design and requirements defined in the earlier phases. The implementation phase involves translating the system's architecture, database design, and functional requirements into an operational software solution. It explains the execution environment, development tools used, real user interface design, and significant functions, including security mechanisms integrated within the system. The goal of this phase is to ensure that the system is fully functional, secure, and aligned with user and institutional needs before deployment.

### 4.5.1 Execution Platform

The system was developed and executed on a MacBook Air (M1, 2020) running the macOS operating system. The local development environment was established using MAMP, which serves as an integrated platform providing both the Apache web server and MySQL database. This setup allowed for a smooth and efficient local deployment of the Laravel framework, enabling backend development, routing configuration, and database management without the need for an external hosting service during the initial stages.

The execution environment for the system primarily involved running the web application locally through Google Chrome, which was used to test and verify the interface, performance, and responsiveness of the system across various display sizes. Google Chrome was chosen due to its strong developer tools, which assist in real-time debugging and front-end inspection.

Additionally, the use of Visual Studio Code as the main code editor allowed seamless integration with MAMP and GitHub for version control, ensuring proper project tracking. Overall, the execution platform provided a stable and flexible environment for testing, debugging, and optimizing both the frontend and backend components of the system before potential deployment to a live server or cloud hosting environment.

### 4.5.2 Implementation Tools

This subsection outlines the main tools and technologies used to develop, test, and manage the UPTM Academic AI Assistant Tools. The chosen implementation tools play a crucial role in ensuring efficiency, maintainability, and security throughout the development process. Each tool was selected based on its compatibility with the Laravel framework, ease of integration, and ability to support rapid development within an Agile environment. These tools cover frontend and backend development, version control, testing, and browser compatibility to ensure a smooth and reliable implementation process.

#### 4.5.2.1 Frontend Technologies (HTML, CSS, Bootstrap)

The frontend development of the system utilized HTML, CSS, and Bootstrap to create a visually appealing and responsive interface. HTML served as the backbone of the web structure, defining the layout and content of each page. CSS was used to enhance the visual presentation, improving readability and user experience through consistent typography, spacing, and color themes. Bootstrap, as a responsive front-end framework, enabled quick implementation of pre-styled components such as buttons, forms, and navigation bars. This combination ensured that the system maintained a professional appearance while remaining accessible across various screen sizes, from desktops to mobile devices.

#### 4.5.2.2 Backend Technologies (PHP and Laravel Framework)

The backend was developed using PHP in combination with the Laravel framework to handle business logic and manage database interactions. Laravel's Model-View-Controller (MVC) architecture structured the application into manageable sections, promoting scalability and ease of maintenance. Laravel's built-in features such as authentication, routing, and migration tools helped streamline development, reduce redundancy, and increase security. The use of PHP and Laravel ensured a stable backend capable of efficiently handling multiple system operations and user requests simultaneously.

#### 4.5.2.3 Database and Local Server Environment (MySQL and MAMP)

The system's data management relied on MySQL, a robust relational database management system known for its reliability and speed. MySQL stored essential data such as user accounts, templates, and generated documents in an organized manner. The MAMP (Macintosh, Apache, MySQL, PHP) environment was configured locally to run the system during development. This setup provided a secure and efficient testing platform, allowing the developer to simulate real-world server conditions before deployment.

#### 4.5.2.4 API Integration (OpenAI API)

One of the most innovative aspects of this project was the integration of the OpenAI API. This API powered the system's core feature AI-driven content generation allowing users to create, summarize, and refine academic materials intelligently. The integration of OpenAI enabled the system to process user prompts in real-time, generate meaningful text responses, and enhance user productivity within an academic context. This connection bridged Laravel's backend logic with modern artificial intelligence capabilities, giving the system its unique academic-focused advantage.

#### 4.5.2.5 Development and Testing Tools (Visual Studio Code, GitHub, Postman)

Visual Studio Code (VS Code) served as the main Integrated Development Environment (IDE) due to its flexibility, simplicity, and wide range of extensions that support PHP and Laravel. GitHub was used as the version control platform to manage code history, ensure collaboration readiness, and safeguard the project against data loss. Postman played an essential role in testing the system's API endpoints, such as GET, POST, PUT, and DELETE requests, ensuring smooth and accurate communication between the frontend, backend, and AI API integration.

#### 4.5.2.6 Web Browser (Google Chrome)

Google Chrome was chosen as the main browser for system execution and interface testing. Its developer tools provided real-time debugging for both frontend and backend scripts, allowing the developer to identify layout issues, monitor network activity, and optimize performance. Chrome's responsiveness testing tools also ensured that the design was consistent and adaptive across various screen resolutions.

### 4.5.3 System Interface

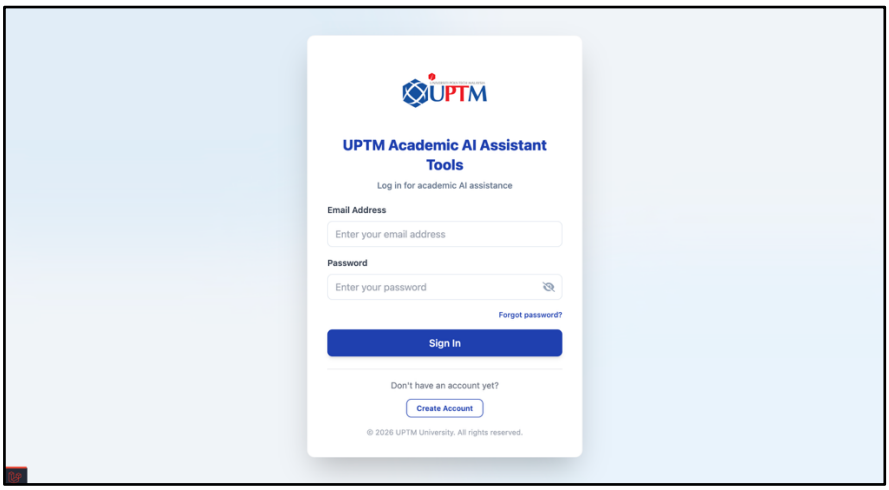



**Figure 4.19:** Login

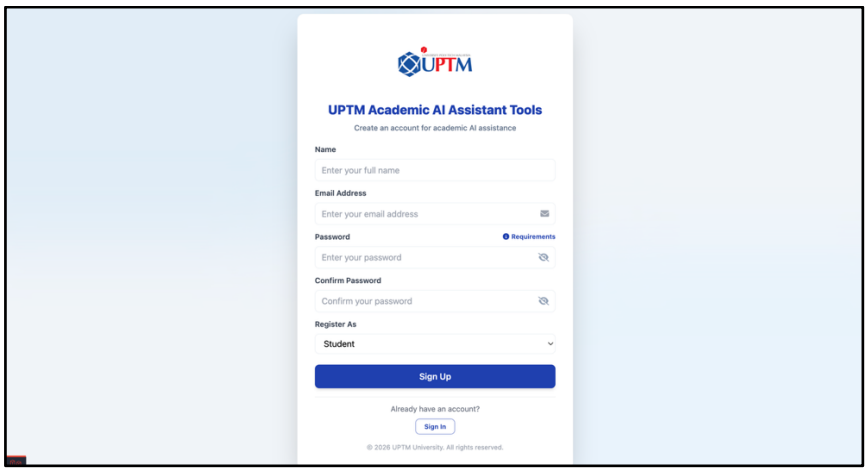



**Figure 4.20:** Sign Up

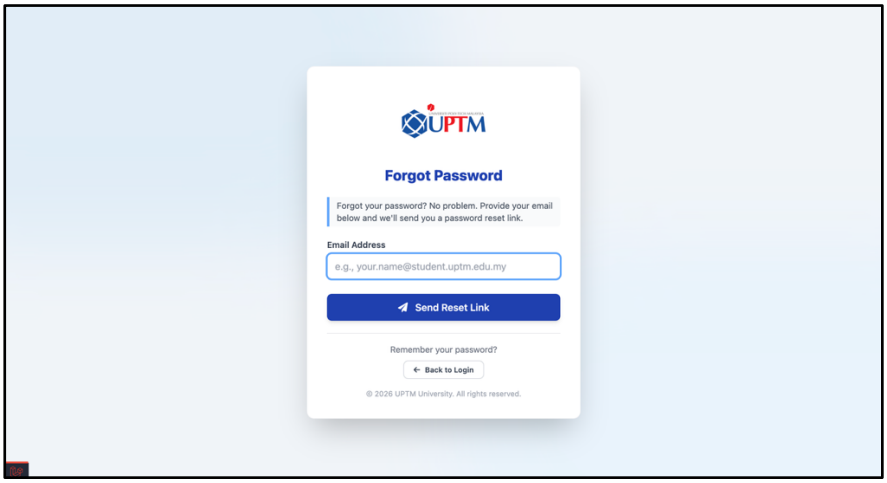



**Figure 4.21:** Forgot Password

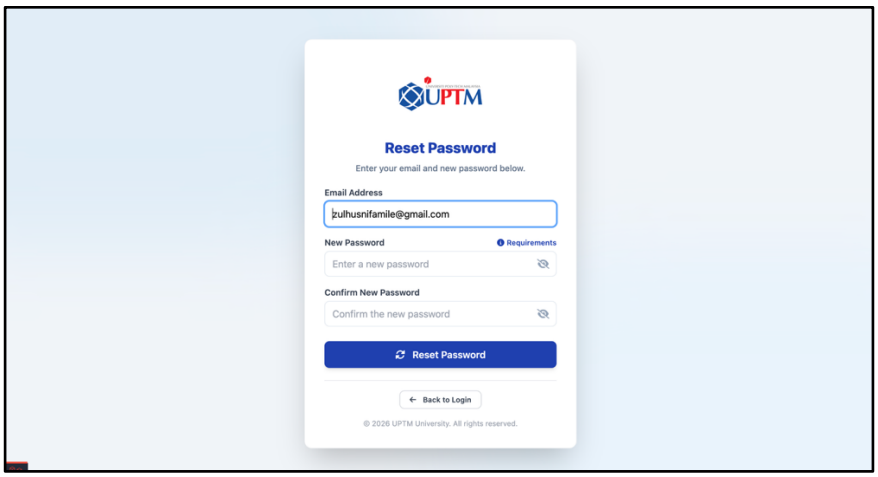



**Figure 4.22:** Reset Password

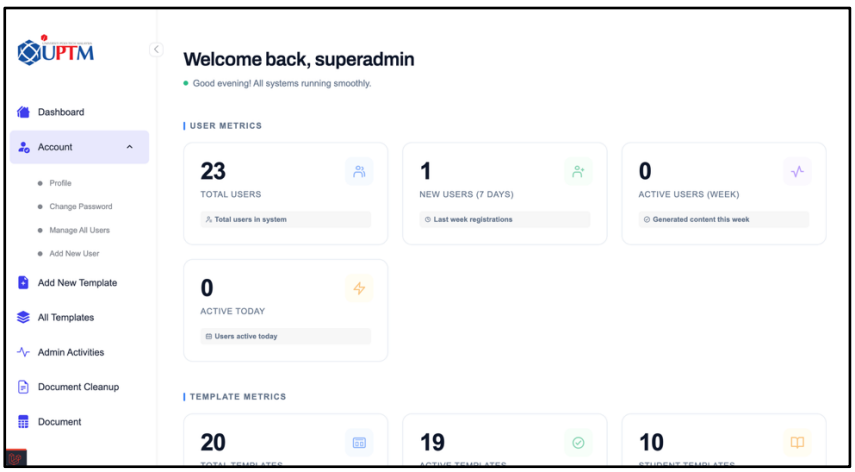



**Figure 4.23:** SuperAdmin Dashboard

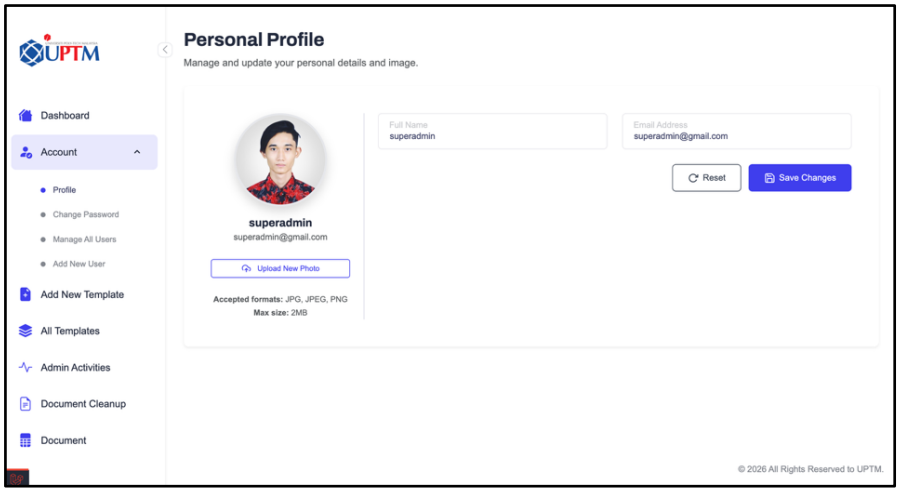



**Figure 4.24:** Personal Profile

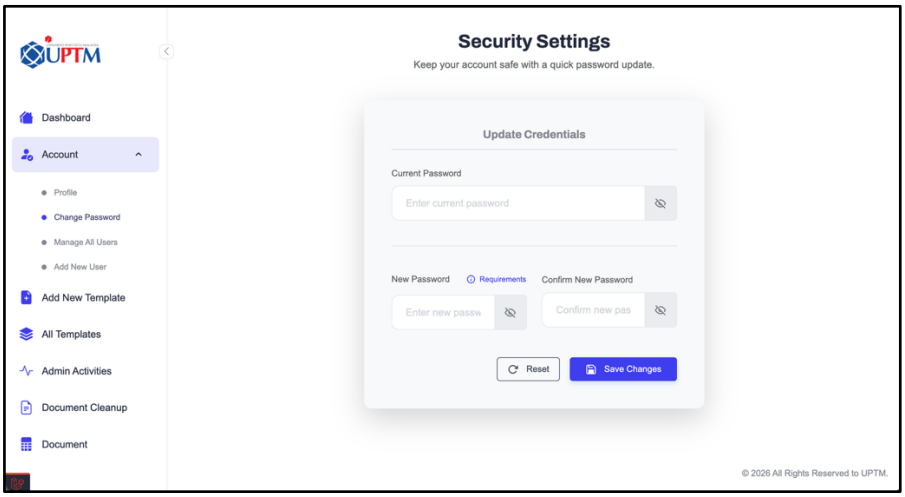



**Figure 4.25:** Change Password

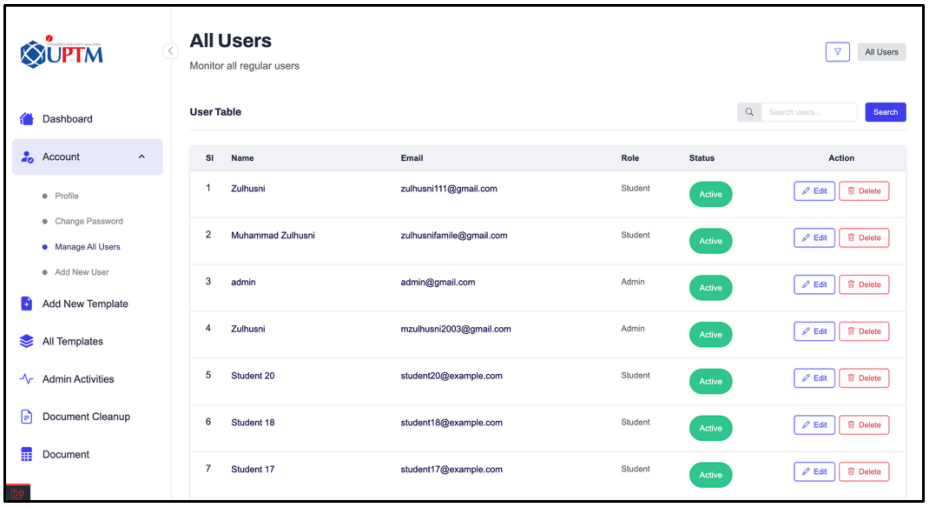



**Figure 4.26:** Manage All Users (SuperAdmin/Admin)

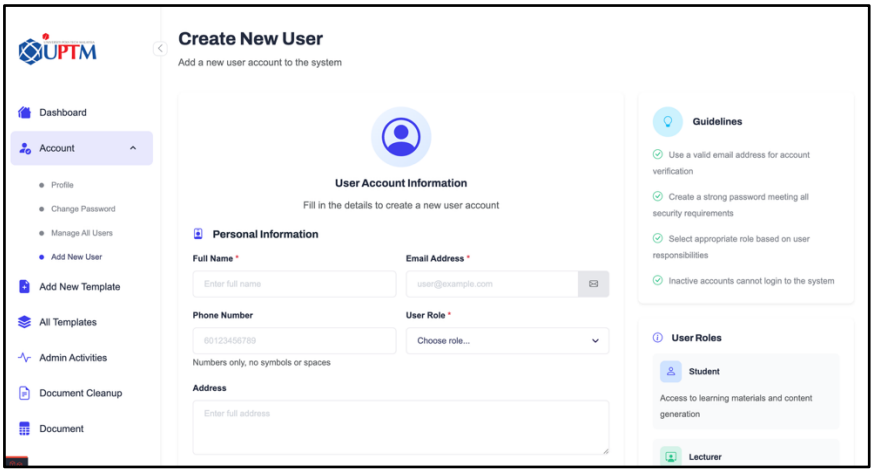



**Figure 4.27:** Create New User (SuperAdmin)

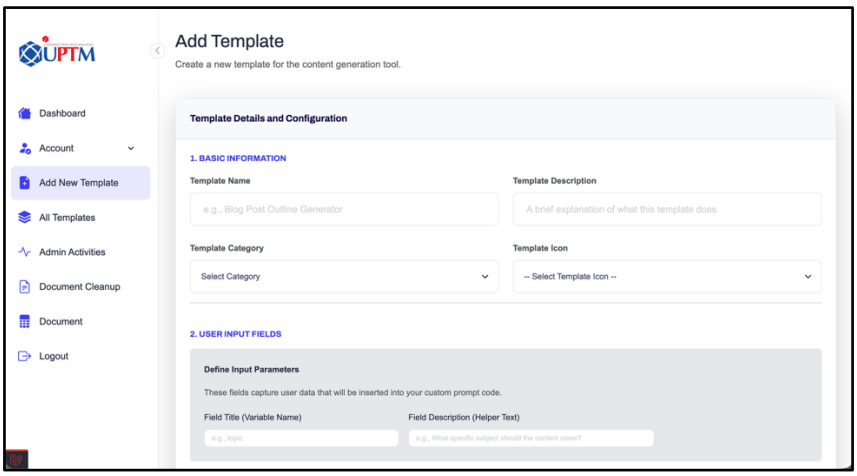



**Figure 4.28:** Add New Template (SuperAdmin/Admin)

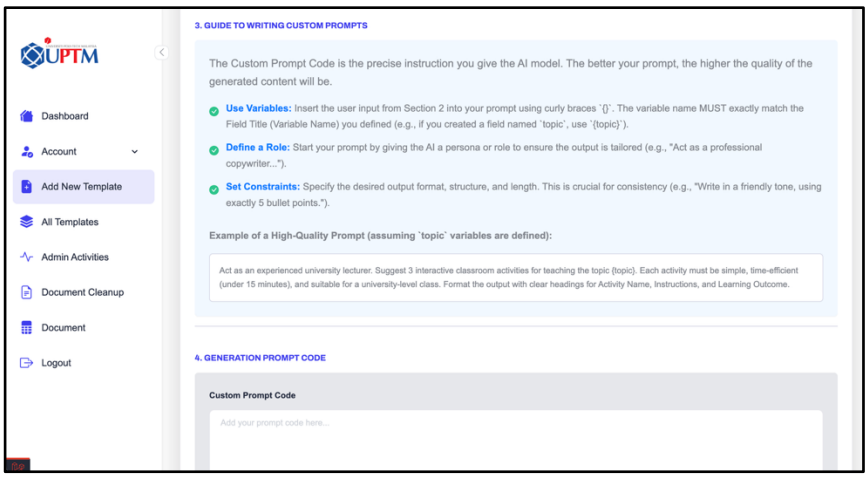



**Figure 4.29:** Add New Template (SuperAdmin/Admin)

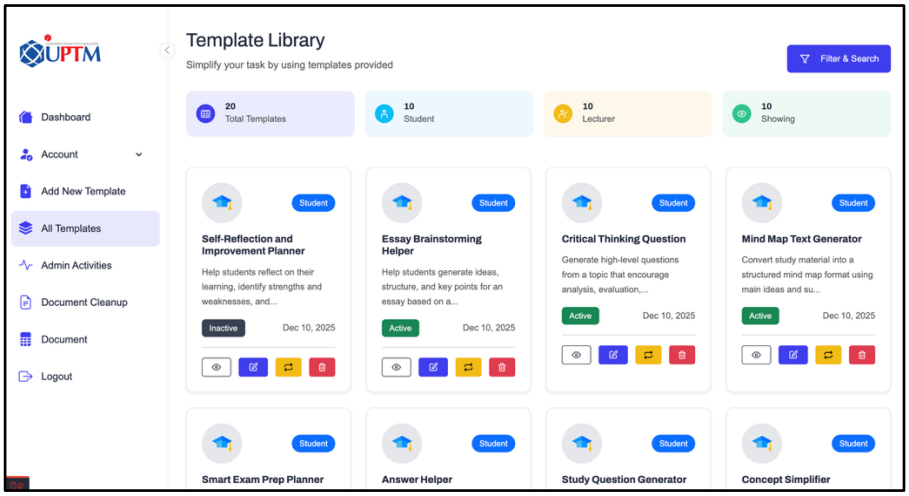



**Figure 4.30:** All Template

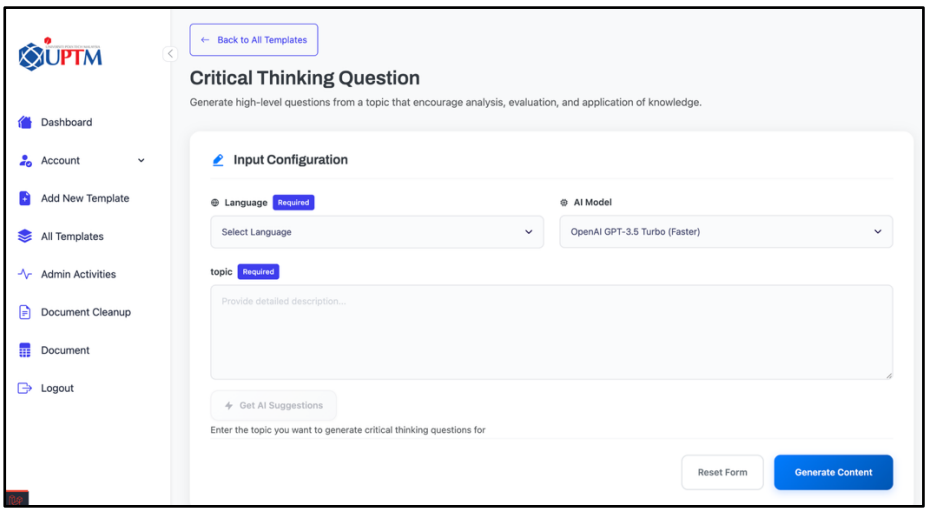



**Figure 4.31:** Template Preview

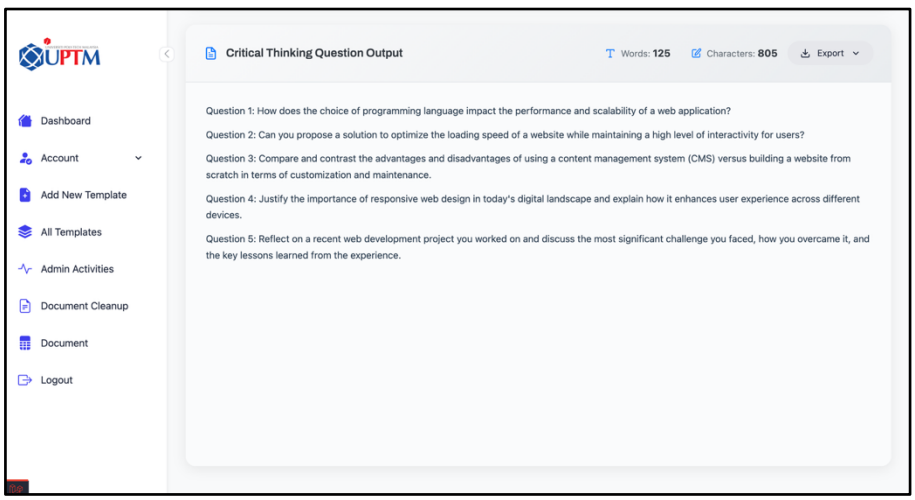



**Figure 4.32:** Template Preview

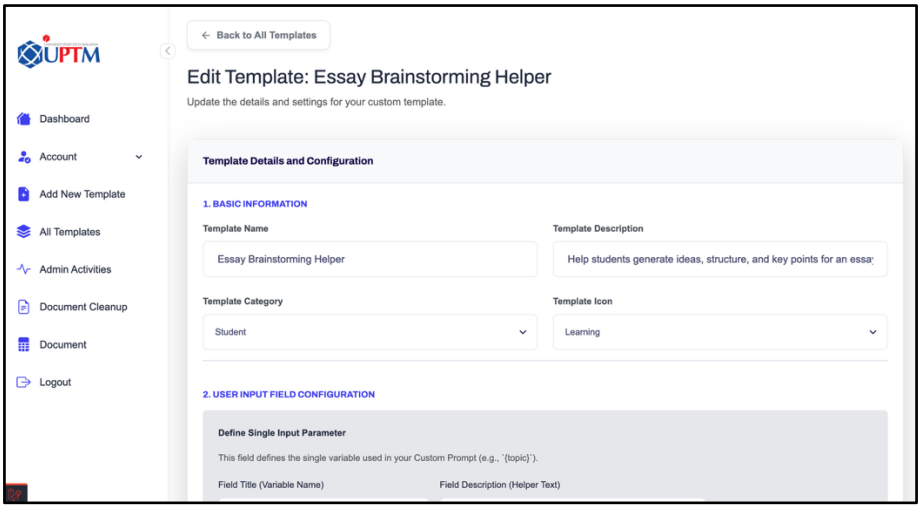



**Figure 4.33:** Edit Template (SuperAdmin/Admin)

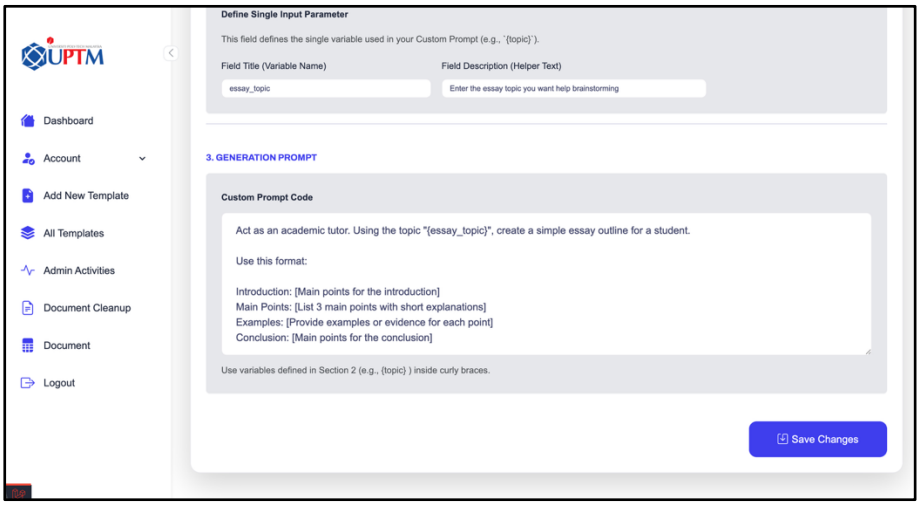



**Figure 4.34:** Edit Template (SuperAdmin/Admin)

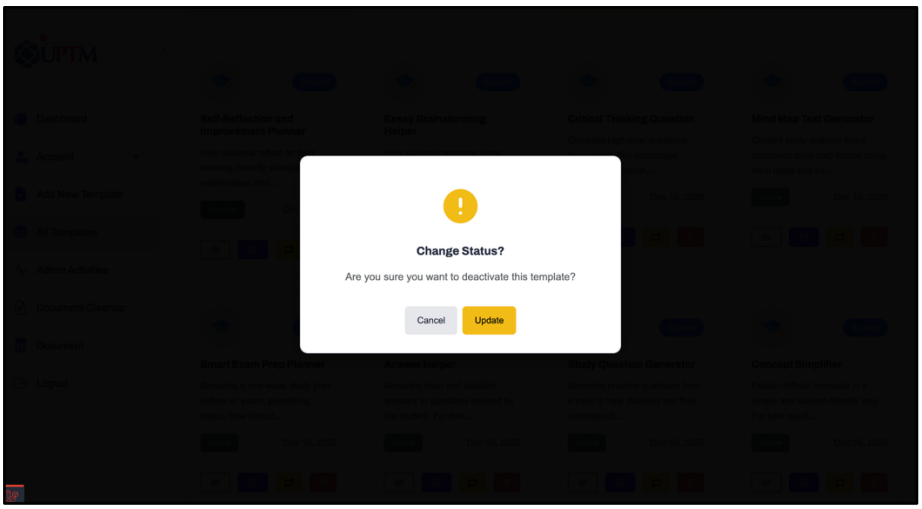



**Figure 4.35:** Change Status (SuperAdmin/Admin)

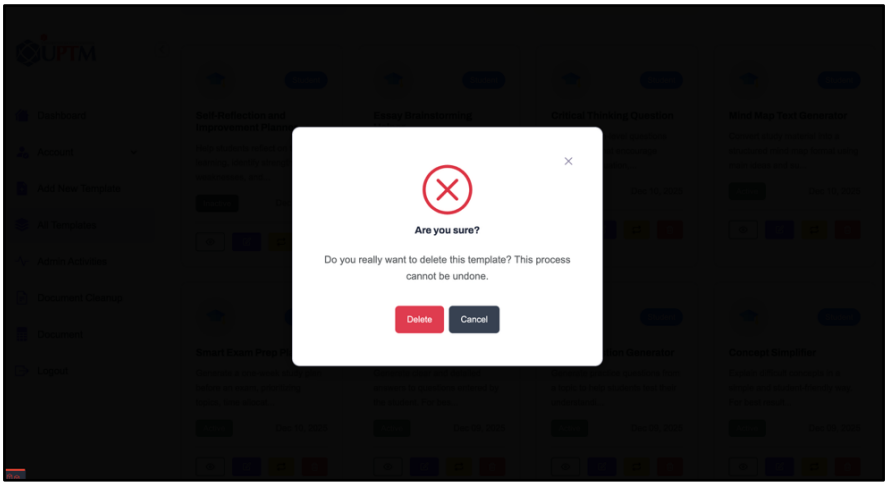



**Figure 4.36:** Delete Template (SuperAdmin/Admin)

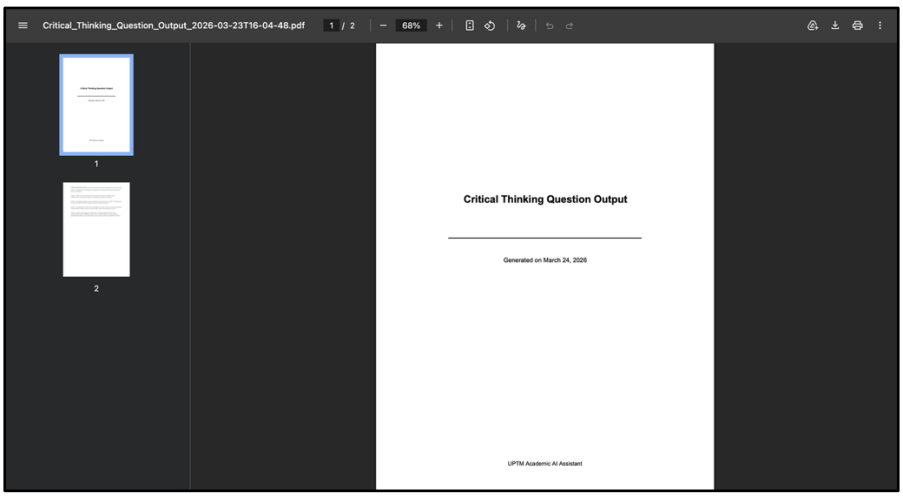



**Figure 4.37:** Download Template

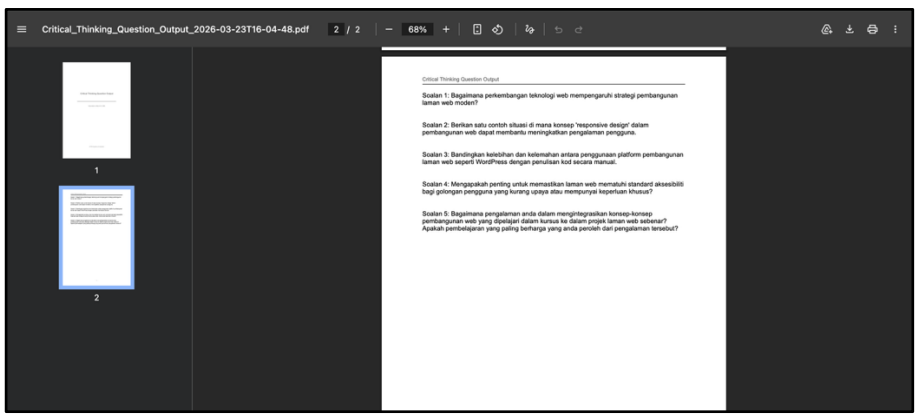



**Figure 4.38:** Download Template

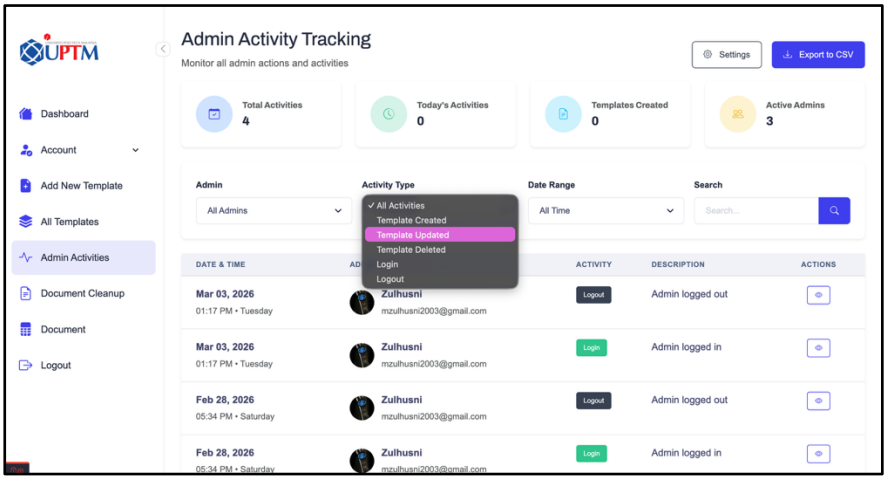



**Figure 4.39:** Admin Activity Tracking (SuperAdmin)

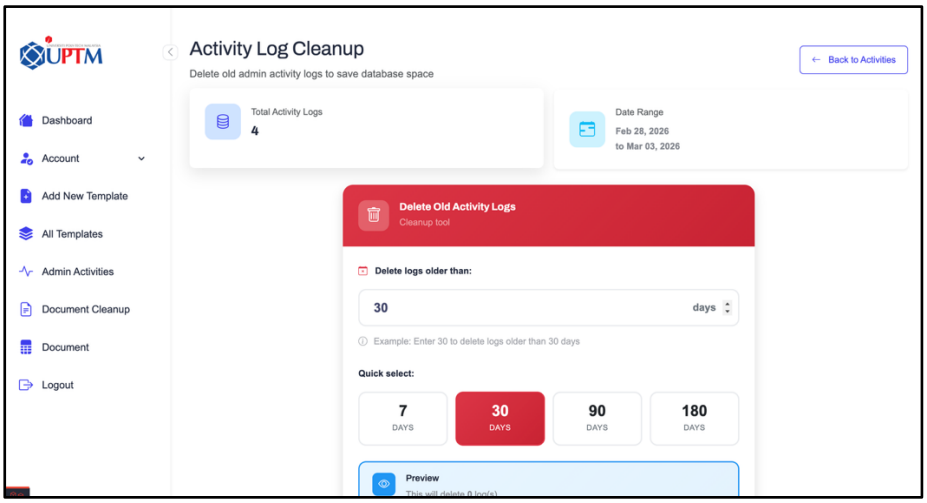



**Figure 4.40:** Log Cleanup (SuperAdmin)

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Date | Time | Day | Admin Name | Activity Type | Description |
| 2 | Mar 03, 2026 | 1:17 PM | Tuesday | Zulhusni | Logout | Admin logged out |
| 3 | Mar 03, 2026 | 1:17 PM | Tuesday | Zulhusni | Login | Admin logged in |
| 4 | Feb 28, 2026 | 5:34 PM | Saturday | Zulhusni | Logout | Admin logged out |
| 5 | Feb 28, 2026 | 5:34 PM | Saturday | Zulhusni | Login | Admin logged in |
| 6 | | | | | | |

**Figure 4.41:** Admin Activity Export to CSV (SuperAdmin)

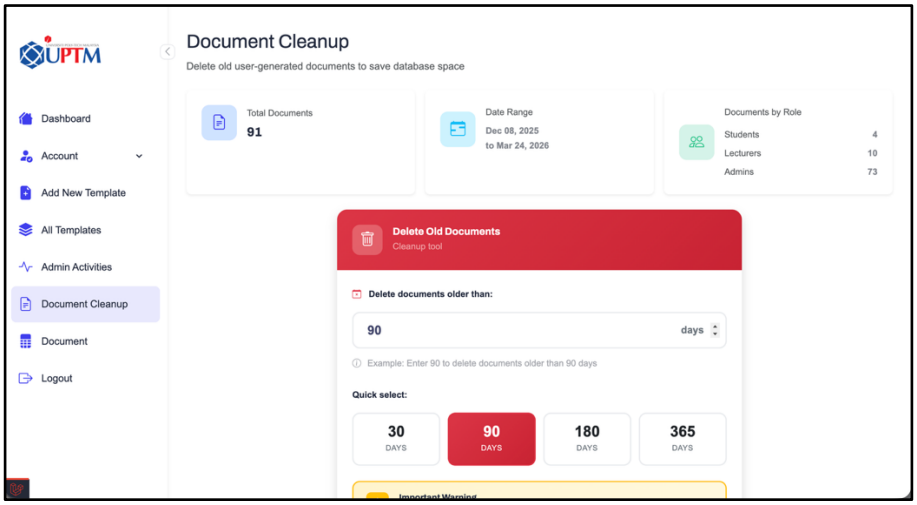



**Figure 4.42:** Document Cleanup (SuperAdmin)

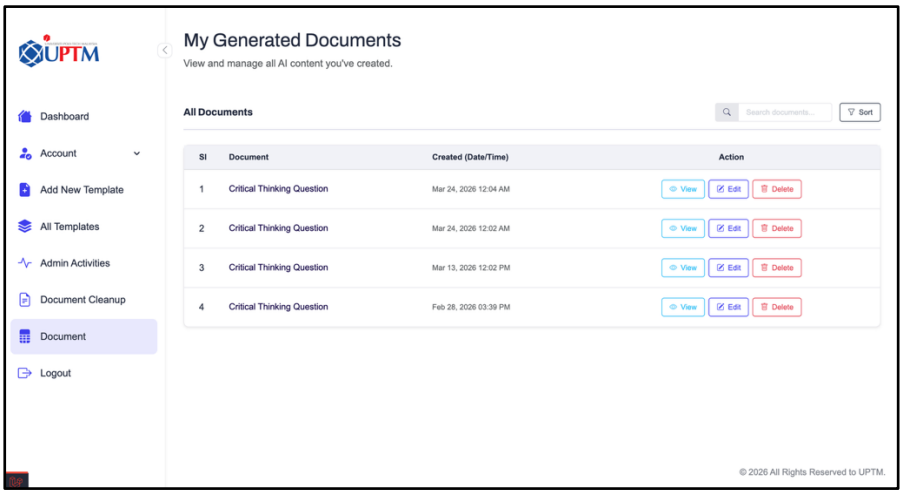



**Figure 4.43:** Documents

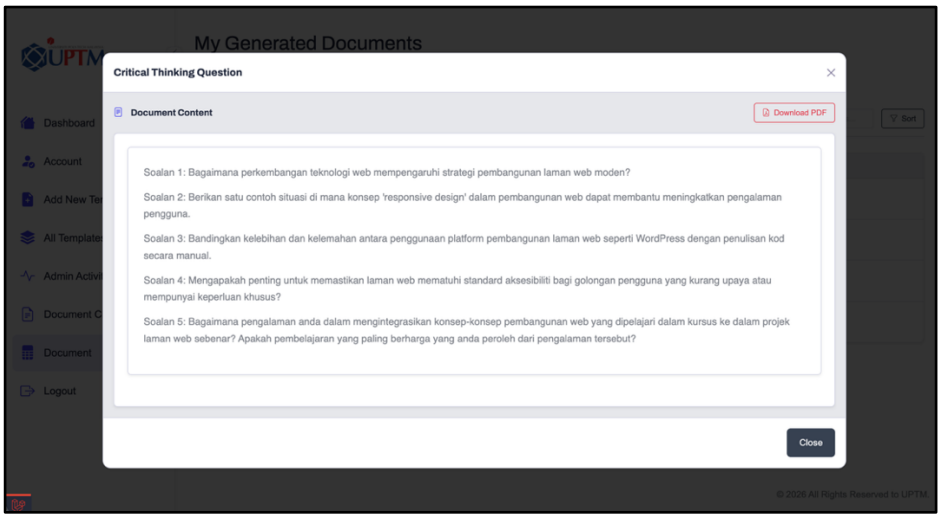



**Figure 4.44:** View Document

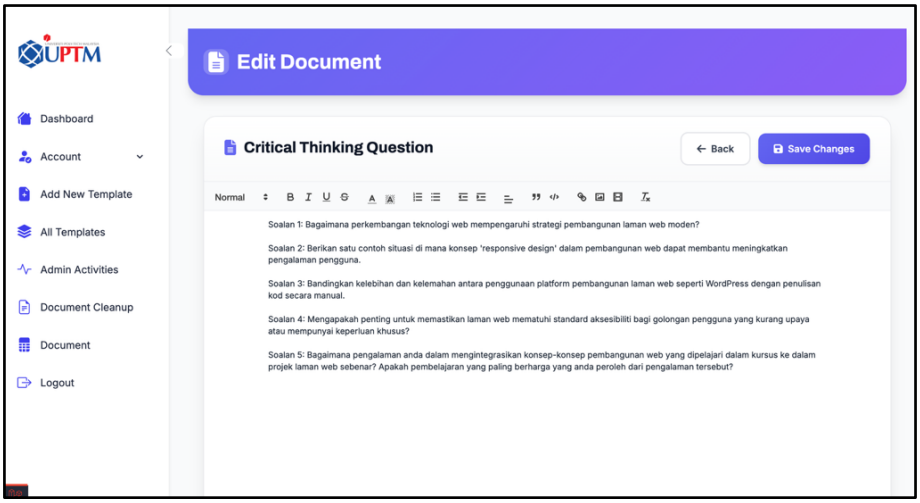



**Figure 4.45:** Edit Document

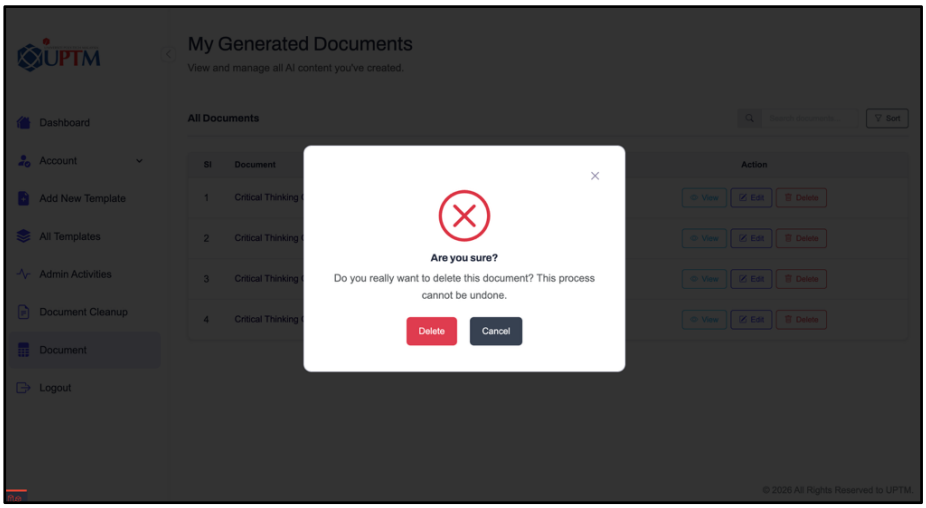



**Figure 4.46:** Delete Document

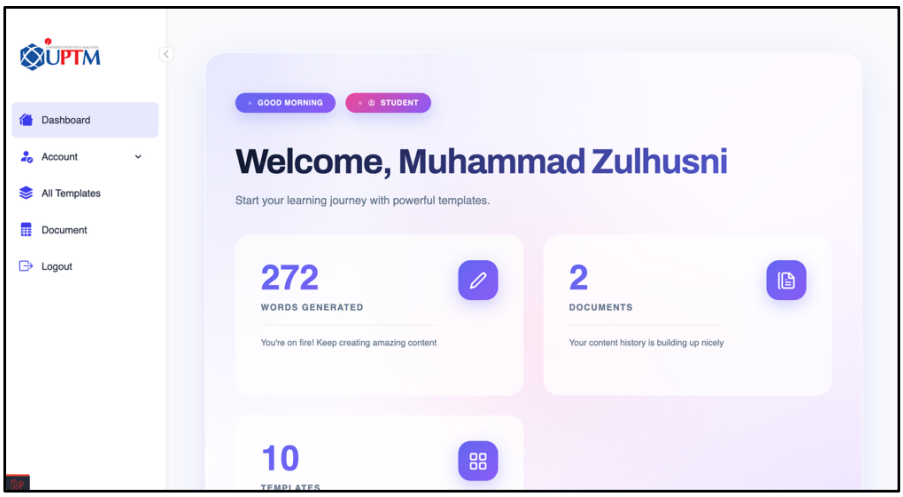



**Figure 4.47:** User Dashboard

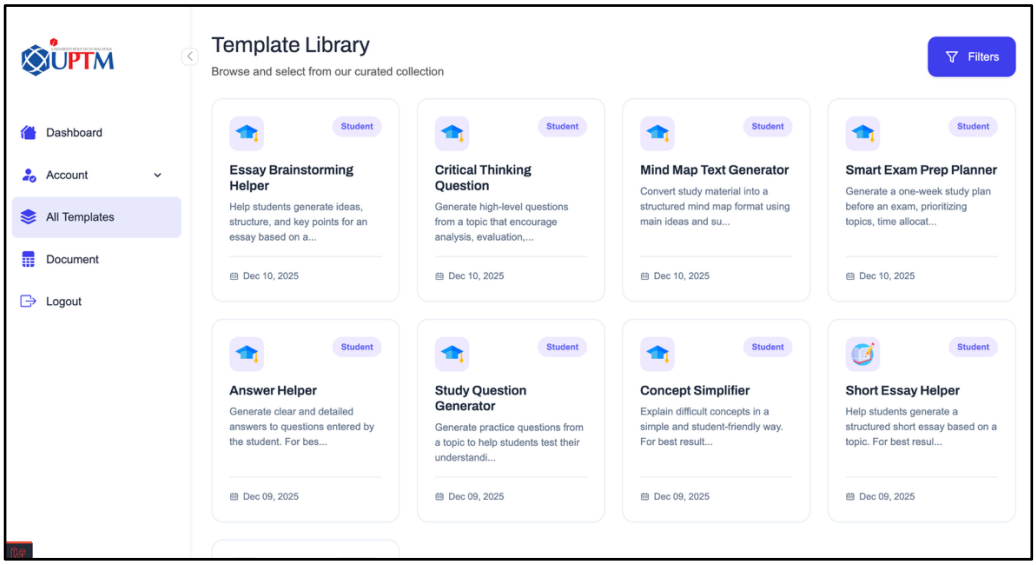



**Figure 4.48:** All Template

### 4.5.3 Significant Functions

Security and core functionality are essential aspects of the UPTM Academic AI Assistant Tools, especially since the system manages user authentication, academic content generation, and institutional data. The system was developed with a strong foundation using Laravel's built-in security mechanisms such as password hashing, CSRF protection, middleware-based route control, and secure session handling. In addition to these features, the system highlights three significant functions that represent its main value: Role-Based Access Control (RBAC) to ensure only authorized roles such as Superadmin, Admin, and Users can access specific modules, the AI Content Generator Workflow which enables template-based academic output generation through structured input submission and OpenAI API processing, and the AI Suggestion Feature that supports users with Bloom's Taxonomy-based academic guidance before final generation. Together, these functions ensure the system operates securely, efficiently, and in a user-friendly manner while maintaining privacy and supporting academic productivity within the university environment.

#### 4.5.3.1 Role-Based Access Control (RBAC)

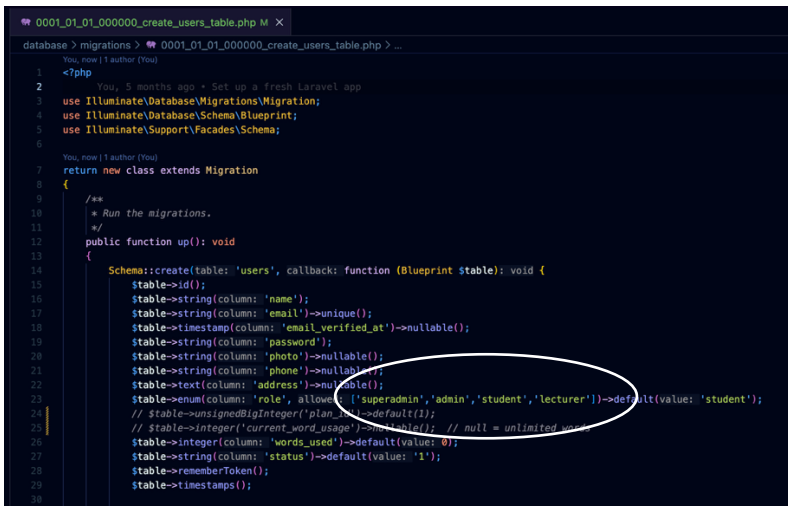



**Figure 4.49:** User role attribute defined in database migration

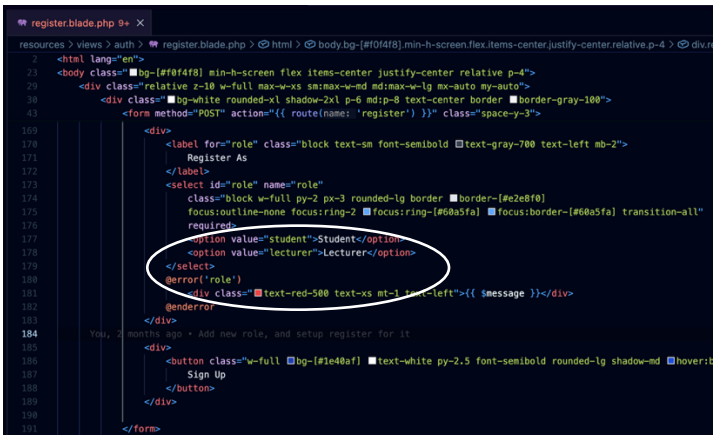



**Figure 4.50:** Role selection option during user registration

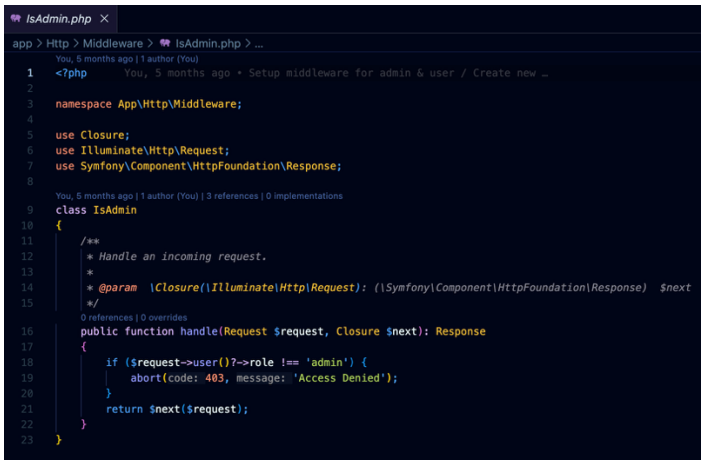



**Figure 4.51:** Admin middleware validating user role before access is granted

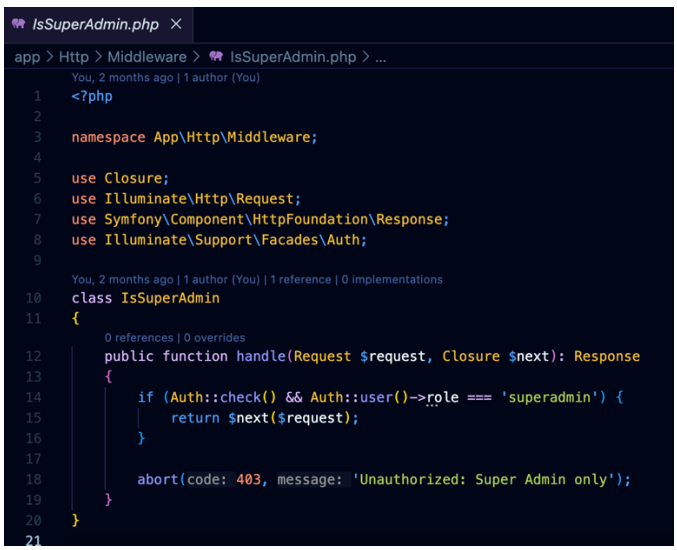



**Figure 4.52:** SuperAdmin middleware validating user role before access is granted

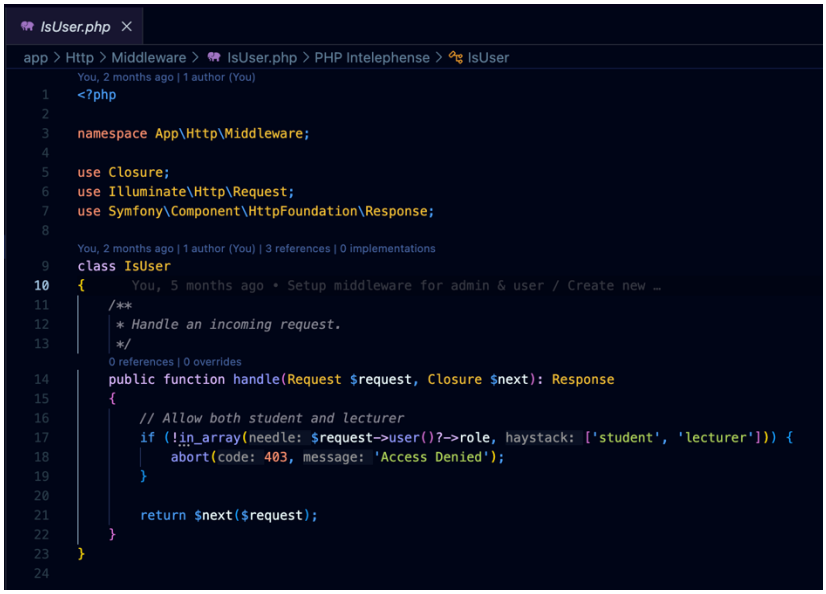



**Figure 4.53:** User middleware validating user role before access is granted

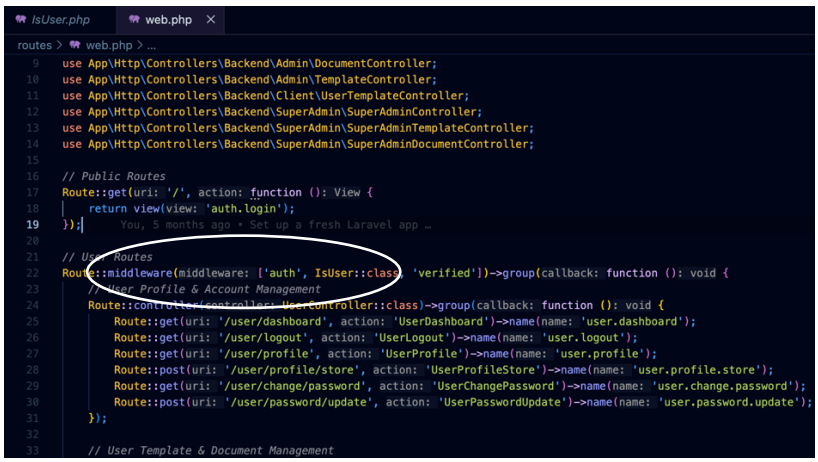



**Figure 4.54:** User routes protected using IsUser middleware

**Figure 4.55:** Admin dashboard routes restricted to admin role

**Figure 4.56:** Superadmin routes restricted to highest-level role

The system implements Role-Based Access Control (RBAC) to ensure that only authorized users can access specific features and functionalities within the UPTM Academic AI Assistant Tools. RBAC plays an important role in maintaining system security by assigning different permissions based on user roles and preventing unauthorized actions, especially in sensitive modules such as template management and user administration.

The RBAC process begins at the database level, where user roles are defined and stored in the users table. The system includes an enum role attribute that supports multiple roles, including superadmin, admin, student, and lecturer. During account registration, users are required to select their role (student or lecturer), ensuring that each account is assigned the correct access level from the beginning. This role information becomes the key reference for controlling access throughout the system.

To enforce role-based restrictions, Laravel middleware is implemented for each role, such as IsUser, IsAdmin, and IsSuperAdmin. These middleware components act as security filters that verify the authenticated user's role before allowing access to protected routes. For example, routes under the admin prefix are restricted to users with the admin role, while routes under the superadmin prefix are accessible only to superadmin users. Similarly, normal users are limited to user-specific routes where they can view templates, generate academic content, and manage their own documents without being able to modify templates or perform administrative actions.

Overall, this separation of access rights strengthens system confidentiality, integrity, and accountability by ensuring that each role can only perform functions relevant to their responsibilities. By combining database role assignment, middleware validation, and route grouping in Laravel, the system effectively ensures secure and structured access control across all dashboards and features.

### 4.5.3.2 AI Content Generator Workflow (Template-Based Generation)

```
<form id="generateForm" action="{{ route(name: 'user.content.generate', parameters: $template->id) }}" method="post">
    @csrf

    <div class="form-grid">
        <!-- Language Selection -->
        <div class="form-group">
            <label for="language" class="form-label">
                <em class="icon ni ni-globe"></em>
                Language
                <span class="badge bg-primary">Required</span>
            </label>
            <select name="language" class="form-select" id="language" required>
                <option value="">Select Language</option>
                <option value="English">English</option>
                <option value="Bahasa Melayu">Bahasa Melayu</option>
            </select>
        </div>

        <!-- AI Model Selection -->
        <div class="form-group">
            <label for="ai_model" class="form-label">
                <em class="icon ni ni-cpu"></em>
                AI Model
            </label>
            <select name="ai_model" class="form-select" id="ai_model"
                <option value="gpt-3.5-turbo" selected>OpenAI GPT-3.5 Turbo (Faster)</option>
                <option value="gpt-4">OpenAI GPT-4 (Higher Quality)</option>
            </select>
        </div>
```

1

2

**Figure 4.57:** Input Configuration Panel

```
    <!-- Action Buttons -->
    <div class="action-buttons">
        <button type="button" class="btn btn-outline-secondary" onclick="document.getElementById('generateForm').reset();">
            Reset Form
        </button>
        <button type="submit" class="btn btn-primary" id="generateBtn">
            <span class="btn-text">
                Generate Content
            </span>
            <span class="btn-loading" style="display: none;">
                <span class="spinner-border spinner-border-sm me-2 mt-1" role="status"></span>
                Generating...
            </span>
        </button>
    </div>
</form>
</div>
```

3

**Figure 4.58:** Generate Content Button

```
0 references | 0 overrides
public function UserContentGenerate(Request $request, $id): JsonResponse
{
    // Step 1: Authenticate user
    $user = Auth::user();
    if (!$user) {
        return response()->json(data: [
            'success' => false,
            'message' => 'User not authenticated.',
        ], status: 401);
    }

    // Validate basic request inputs
    $validatedData = $request->validate(rules: [
        'language' => 'required|string|in:English,Bahasa Melayu',
        'ai_model' => 'required|string|in:gpt-4,gpt-3.5-turbo',
    ]);

    // Step 2: Fetch template and validate dynamic input fields
    $template = Template::with(relations: ['inputFields' => function ($query): void {
        $query->select('id', 'template_id', 'title');
    }])
    ->select(columns: 'id', 'prompt')
    ->findOrFail($id);

    // Build dynamic validation rules
    $dynamicRules = [];
    $fieldNames = [];
    foreach ($template->inputFields as $field) {
        $fieldName = str_replace(search: ' ', replace: '_', subject: $field->title);
        $dynamicRules[$fieldName] = 'required|string|max:1000';
        $fieldNames[] = $fieldName;
    }

    $request->validate(rules: $dynamicRules);
    $inputData = $request->only(keys: $fieldNames);
```

**Figure 4.59:** Step 1 & Step 2

```
// Build dynamic validation rules
$dynamicRules = [];
$fieldNames = [];
foreach ($template->inputFields as $field) {
    $fieldName = str_replace(search: ' ', replace: '_', subject: $field->title);
    $dynamicRules[$fieldName] = 'required|string|max:1000';
    $fieldNames[] = $fieldName;
}

$request->validate(rules: $dynamicRules);
$inputData = $request->only(keys: $fieldNames);

// Step 3: Build AI prompt with input replacements
$replacements = [];
foreach ($inputData as $key => $value) {
    $replacements['{' . $key . '}'] = $value;
    $replacements['{' . str_replace(search: '_', replace: ' ', subject: $key) . '}'] = $value;
}
You, now • Uncommitted changes

$prompt = strtr(string: $template->prompt, replace_pairs: $replacements);

// Build messages without length constraint
$messages = $this->buildLanguageSpecificMessages(validatedData: $validatedData, prompt: $prompt);

// Step 4: Send request to OpenAI API
try {
    // Determine appropriate max_tokens based on model
    $maxTokens = $validatedData['ai_model'] === 'gpt-4' ? 4096 : 3000;

    $response = OpenAI::chat()->create(parameters: [
        'model' => $validatedData['ai_model'],
        'messages' => $messages,
        'max_tokens' => $maxTokens, // Allow full response
```

**Figure 4.60:** Step 3 & Step 4

```
// Step 4: Send request to OpenAI API
try {
    // Determine appropriate max_tokens based on model
    $maxTokens = $validatedData['ai_model'] === 'gpt-4' ? 4096 : 3000;

    $response = OpenAI::chat()->create(parameters: [
        'model' => $validatedData['ai_model'],
        'messages' => $messages,
        'max_tokens' => $maxTokens, // Allow full response
        'temperature' => 0.7,
    ]);

    $output = $response->choices[0]->message->content;

    // Validate output language
    $languageCheck = $this->validateOutputLanguage(output: $output, requestedLanguage: $validatedData['language']);
    if (!$languageCheck['valid']) {
        Log::warning(message: 'Generated content language mismatch', context: [
            'user_id' => $user->id,
            'requested_language' => $validatedData['language'],
            'confidence' => $languageCheck['confidence']
        ]);
    }
```

**Figure 4.61:** Step 4

The core functionality of the system lies in its AI Content Generator, which enables users to produce academic outputs using structured templates instead of manually writing prompts. Each template defines a prompt, dynamic input fields, and validation rules, and the system automatically renders the required form fields based on the selected template, ensuring consistency and ease of use. Users can also select the output language (English or Bahasa Melayu) and the AI model (GPT-3.5 Turbo or GPT-4) to suit their academic needs.

Before submission, the system performs input validation both on the client side and backend. Client-side JavaScript highlights empty fields and temporarily disables the generate button to prevent duplicate submissions, while the Laravel backend further validates all inputs for completeness and security. Once validated, the frontend sends an asynchronous POST request to the Laravel backend, with CSRF protection applied to prevent request forgery.

On the backend, the controller combines the user's input with the template prompt and calls the OpenAI API using the selected AI model to generate content. The API response is returned in JSON format, and the system records the generated content. The frontend then formats and displays the output in the user interface. This workflow demonstrates the system's ability to provide efficient, template-driven academic content generation in a structured, secure, and user-friendly manner.

### 4.5.3.3 AI Suggestion Feature (Bloom's Taxonomy Input Assistance)

```
    <!-- AI Suggestion Button -->
    <div class="ai-suggestion-wrapper mt-3">
        <button type="button"
                class="ai-suggest-btn"
                data-target="{{ $field->title }}"
                disabled>
            <em class="icon ni ni-spark-fill"></em>
            <span class="btn-text">Get AI Suggestions</span>
            <span class="btn-loading" style="display: none;">
                <span class="spinner-border spinner-border-sm" role="status"></span>
            </span>
        </button>
        <div class="suggestions-dropdown"
             id="suggestions-{{ str_replace(search: ' ', replace: '-', subject: $field->title) }}"
             style="display: none;">
        </div>
    </div>
@endif
```

**Figure 4.62:** Get AI Suggestion Button (Frontend)

```
    try {
        // Get CSRF token from meta tag
        const csrfMeta = document.querySelector('meta[name="csrf-token"]');
        if (!csrfMeta) {
            throw new Error('CSRF token not found in page');
        }

        const response = await fetch('{{ route(name: "user.ai.suggestion") }}', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
                'X-CSRF-TOKEN': csrfMeta.getAttribute('content'),
                'Accept': 'application/json',
            },
            body: JSON.stringify({
                field_name: fieldName,
                current_input: currentInput,
                language: language,
                template_context: document.querySelector('.display-6')?.textContent || ''
            })
        });

        if (!response.ok) {
            throw new Error(`HTTP error! status: ${response.status}`);
        }
        You, last month • add AI suggestion ...
```

**Figure 4.63:** Event Trigger (Frontend)

```
0 references | 0 overrides
public function AISuggestion(Request $request): JsonResponse
{
    $user = Auth::user();
    if (!$user) {
        return response()->json(data: [
            'success' => false,
            'message' => 'User not authenticated.',
        ], status: 401);
    }

    // Validate request
    $validated = $request->validate(rules: [
        'field_name' => 'required|string',
        'current_input' => 'required|string|min:3|max:500',
        'language' => 'required|string|in:English,Bahasa Melayu',
        'template_context' => 'nullable|string|max:200'
    ]);
    You, last month • add AI suggestion
    try {
        // Build ACADEMIC-FOCUSED context-aware prompt with Bloom's Taxonomy
        $isMalay = $validated['language'] === 'Bahasa Melayu';
        $userRole = ucfirst(string: $user->role);

        if ($isMalay) {
            $systemPrompt = 'Anda adalah pakar akademik AI yang membantu pelajar dan pensyarah dengan cadangan penulisan akademik berkualiti tinggi dalam Bahasa

            $userPrompt = "Pengguna ({$userRole}): \"{$validated['current_input']}\"\n\n";
            $userPrompt .= "Hasilkan 6 cadangan akademik yang spesifik dan profesional (15-25 patah perkataan setiap satu) berdasarkan 6 tahap Taksonomi Bloom se
            $userPrompt .= "TAHAP 1 - INGAT (REMEMBER): Cadangan yang memfokuskan kepada pengulangan fakta, definisi, atau pengecaman konsep asas.\n";
            $userPrompt .= "TAHAP 2 - FAHAM (UNDERSTAND): Cadangan yang memfokuskan kepada penjelasan maksud, interpretasi, atau rumusan idea.\n";
            $userPrompt .= "TAHAP 3 - APLIKASI (APPLY): Cadangan yang memfokuskan kepada penggunaan teori dalam konteks praktikal atau penyelesaian masalah.\n";
            $userPrompt .= "TAHAP 4 - ANALISIS (ANALYZE): Cadangan yang memfokuskan kepada pemeriksaan mendalam, perbandingan, atau penguraian hubungan antara el
            $userPrompt .= "TAHAP 5 - NILAI (EVALUATE): Cadangan yang memfokuskan kepada penilaian kritikal, justifikasi keputusan, atau pertimbangan kekuatan da
            $userPrompt .= "TAHAP 6 - CIPTA (CREATE): Cadangan yang memfokuskan kepada inovasi, sintesis idea baharu, atau reka bentuk penyelesaian kreatif.\n\n"
```

**Figure 4.64:** Processing & Prompt Engineering (Backend)

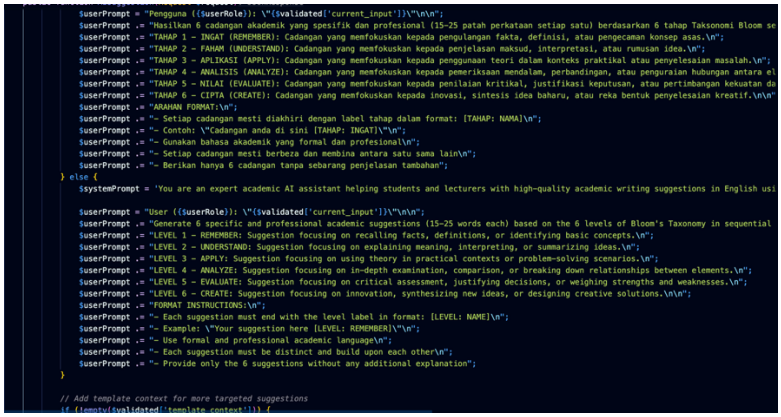


**Figure 4.65:** Processing & Prompt Engineering (Backend)

**Figure 4.66:** Processing & Prompt Engineering (Backend) 2.0

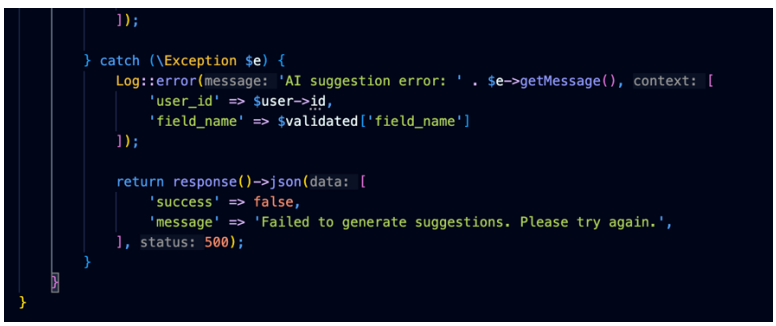


**Figure 4.67:** Processing & Prompt Engineering (Backend) 3.0

```
function applySuggestion(textarea, container, suggestionId, suggestionMap) {
    try {
        const suggestion = suggestionMap.get(suggestionId);

        if (!suggestion) {
            console.error('Suggestion not found in map');
            showError('Failed to apply suggestion');
            return;
        }
            You, last month • add AI suggestion …
        // REPLACE current text with suggestion (don't append)
        textarea.value = suggestion;

        // Hide suggestions after successful application
        container.style.display = 'none';

        // Focus back on textarea and move cursor to end
        textarea.focus();
        textarea.setSelectionRange(textarea.value.length, textarea.value.length);

        // Trigger input event to update button states and character count
        const inputEvent = new Event('input', { bubbles: true });
        textarea.dispatchEvent(inputEvent);

        // Show success feedback
        if (typeof toastr !== 'undefined') {
            toastr.success('Suggestion applied successfully!');
        } else {
            console.log('Suggestion applied:', suggestion);
        }

    } catch (error) {
        console.error('Error applying suggestion:', error);
        showError('Failed to apply suggestion');
    }
}
```

**Figure 4.68:** User Selection & Input Update

In addition to full content generation, the system includes an AI Suggestion feature to help users refine their academic input before submission. This function supports better academic writing by guiding users through structured suggestions aligned with Bloom's Taxonomy levels. The process begins when a user types text into a textarea and clicks the "Get AI Suggestions" button. At this stage, JavaScript sends an AJAX request containing the current input, selected language, and template context to the Laravel backend.

The backend validates the input and constructs a structured academic prompt for the OpenAI API. The system requests six academic suggestions corresponding to Bloom's Taxonomy cognitive levels: Remember, Understand, Apply, Analyze, Evaluate, and Create. Once OpenAI processes the request, the suggestions are returned as a JSON response.

On the frontend, JavaScript dynamically displays these suggestions in a dropdown below the input field. Users can select any suggestion to improve or expand their input before proceeding with final content generation. This feature enhances user experience by providing academic guidance, encouraging deeper thinking, and improving the quality of AI-generated outputs.

## 4.6 Conclusion

The design phase played a crucial role in shaping the overall structure, functionality, and security of the system. Through careful planning and detailed design, the system was developed to ensure a user-friendly interface, reliable performance, and secure data handling. Both frontend and backend designs were integrated using modern technologies such as Laravel, Bootstrap, and MySQL to achieve a responsive and efficient web application.

Furthermore, the implementation of security elements such as authentication, role-based access control, and input validation ensures that the system operates safely and prevents unauthorized access. The design process also emphasized clarity in navigation, consistency in layout, and scalability for future enhancements. Overall, the design phase successfully provided a strong foundation for system implementation, guaranteeing that the final product meets user needs while maintaining high usability and data protection standards.

# 5 FINDING

## 5.1 Introduction

The testing phase is an essential stage in the system development process, ensuring that every feature of the UPTM Academic AI Assistant Tools functions as intended. This phase focuses on identifying and resolving errors, verifying that all modules integrate correctly, and confirming that the system meets the specified user and functional requirements. Through various testing methods, including unit testing, integration testing, system testing, and acceptance testing, the overall performance, security, and usability of the system were evaluated.

The main objective of this phase is to ensure that both users and administrators can interact with the system without encountering functional issues. It also confirms that the connection between the UPTM system and the OpenAI API operates correctly allowing smooth communication, data transfer, and response generation. By systematically conducting different levels of testing, the development ensures that the final product is stable, reliable, and ready for deployment in a real-world environment.

## 5.2 Testing

Testing was conducted to evaluate the functionality, performance, and reliability of the UPTM Academic AI Assistant Tools system. The main goal of this process was to ensure that all system components including authentication, role-based access, profile management, content generation, and integration with the OpenAI API work together seamlessly and meet the intended requirements.

During the testing phase, both functional and non-functional aspects of the system were examined. Functional testing focused on verifying that each feature performs according to its purpose, such as user registration, login, content generation, and template management. Non-functional testing, on the other hand, assessed performance, security, and usability to ensure a smooth user experience across different devices and browsers.

This setup closely simulated a real-world environment where multiple users, including superadmin, admins and regular users, interact with the system concurrently. Throughout testing, different scenarios were executed, including valid and invalid inputs, to observe system behavior and identify potential bugs.

By the end of the testing phase, all critical functions were verified, and the system demonstrated stable performance, accurate role-based operations, and secure data handling. The successful completion of this stage confirmed that the system was ready to proceed to more detailed testing levels such as unit, integration, and system testing.

## 5.3 Unit Testing

Unit testing was conducted to ensure that each individual function within the UPTM Academic AI Assistant Tools system operates correctly before proceeding to integration testing. The testing covered all major modules across the three user roles: SuperAdmin, Admin, and Users (Lecturer/Student). Core components tested include authentication (login and registration), profile and password management, role-based access control (RBAC), user management, template management, AI content generation, AI suggestion functionality, and document management. Each unit was evaluated using both valid and invalid input data to verify correct system behavior, including successful form submissions, validation error handling, secure database transactions, and proper role authorization. This process ensured that every module performs independently according to the defined functional requirements. The following tables summarize the unit testing results for the major system functions.

**Table 5.1**: Unit Testing Results (SuperAdmin)

| Test Case ID | Unit/Function | Description | Input | Output | Expected Result | Status |
|---|---|---|---|---|---|---|
| TC-SA1 | Login | Verify superadmin login functionality | Valid/invalid credentials | Redirect to dashboard or error | Successful login with valid credentials; error with invalid | Pass |
| TC-SA2 | Sign Up | Create superadmin account | Name, email, password | Redirect to login page | Account created and saved to DB | Pass |
| TC-SA3 | Manage Account | Update superadmin personal account | Name, email | Updated info displayed | Profile updated correctly | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| TC-SA4 | Update Profile | Update name/email | New name/email | Display updated info | Profile updated successfully | Pass |
| TC-SA5 | Change Password | Change superadmin password | Old password, new password | Redirect with success message | Password updated securely | Pass |
| TC-SA6 | Manage All Users | View all users, update status, delete account | Select user record | Status updated or user deleted | Changes applied successfully | Pass |
| TC-SA7 | Add New User | Add admin/lecturer/ student account | Name, email, role, password | New account created | User added to system with correct role | Pass |
| TC-SA8 | Templates Management | Create, update, delete, change status of templates | Template title/content /status | Template list updated | Templates created/modified/deleted successfully | Pass |
| TC-SA9 | View All Templates | View templates in system | Click “All Templates” | Template list displayed | All templates visible correctly | Pass |
| TC-SA10 | Admin Activity Tracking | Track admin template activity | View log | Activity history displayed | Admin actions tracked | Pass |
| TC-SA11 | Document Cleanup | Delete old generated documents | Select documents | Documents removed from DB | Cleanup successful | Pass |
| TC-SA12 | Document Management | View history of generated documents | Click document list | Document details displayed | All documents viewable | Pass |
| TC-SA13 | Edit/Delete Document | Edit or remove generated document | Select document | Document updated/deleted | Document changes saved correctly | Pass |

**Table 5.2**: Unit Testing Results (Admin)

| Test Case ID | Unit/Function | Description | Input | Output | Expected Result | Status |
|---|---|---|---|---|---|---|
| TC-A1 | Login | Verify admin login functionality | Valid/invalid credentials | Redirect to dashboard or error | Successful login with valid credentials, error with invalid | Pass |
| TC-A2 | Sign Up | Create admin account | Name, email, password | Redirect to login page | Account created and saved to DB | Pass |
| TC-A3 | Account created and saved to DB | View/delete unauthorized users | Click delete button | User record removed | Users managed successfully | Pass |
| TC-A4 | Users managed successfully | Update name/email/contact | New data | Display updated info | Profile updated correctly | Pass |
| TC-A5 | Change Password | Change admin password | Old password, new password | Redirect with success message | Password updated securely | Pass |
| TC-A6 | Templates Management | Create/update/delete/change visibility | Template data | Template list updated | Template list updated | Pass |
| TC-A7 | View All Templates | View all templates | Click “All Templates” | Template list displayed | All templates visible correctly | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| TC-A8 | Document Management | View document history | Click document list | Document details displayed | All documents viewable | Pass |
| TC-A9 | Edit/Delete Document | Edit or remove generated document | Select document | Document updated/deleted | Document changes saved correctly | Pass |

**Table 5.3**: Unit Testing Results (Users)

| Test Case ID | Unit/Function | Description | Input | Output | Expected Result | Status |
|---|---|---|---|---|---|---|
| TC-U1 | Login | Verify user login functionality | Valid/invalid credentials | Redirect to dashboard or error | Successful login with valid credentials; error with invalid | Pass |
| TC-U2 | Sign Up | Create user account | Name, email, password | Redirect to login page | Account created and saved to DB | Pass |
| TC-U3 | Manage Account | Update personal profile | New name/email | Display updated info | Profile updated correctly | Pass |
| TC-U4 | Change Password | Change user password | Old password, new password | Redirect with success message | Password updated securely | Pass |
| TC-U5 | View Account Details | View personal info | Click “View Profile” | Display user info | Correct account info displayed | Pass |
| TC-U6 | Use Template Provided by Admin | Generate AI-based content | Select template, fill input | AI output displayed | AI content generated and displayed correctly | Pass |
| TC-U7 | Document Management | View history of generated documents | Click document list | Document details displayed | All generated documents viewable | Pass |
| TC-U8 | Edit/Delete Document | Edit or remove generated document | Select document | Document updated/deleted | Changes saved correctly | Pass |
| TC-U9 | AI Suggestions | Get AI suggestions for input | Type input , Click “Get AI Suggestions” | Suggestions dropdown displayed | Suggestions correctly returned for Bloom’s levels | Pass |

## 5.4 Integration Testing

Integration testing was conducted after unit testing to verify that all individual modules within the UPTM Academic AI Assistant Tools system function correctly when integrated. This phase ensures proper communication between the system components, including the frontend interface, backend logic, database operations, and the external OpenAI API integration. The main objective is to confirm that data flows smoothly across different modules, such as user authentication, profile management, AI content generation, template management, and document management, without errors or interruptions.

During the integration testing process, interactions between major system components were evaluated. These components include the Superadmin Module, Admin Module, User Module, Template Management Module, Document Management Module, and the OpenAI API Module. The testing ensures that users can perform end-to-end system operations successfully, such as logging into the system, selecting academic templates, generating content through the OpenAI API, and storing the generated results in the database.

Additionally, integration testing verifies that administrative actions performed by the Superadmin or Admin, such as creating or managing templates, are correctly reflected on the user interface. This ensures consistency and synchronization between administrative functions and user-side operations within the system. The following table summarizes the key integration test cases conducted for the system.

**Table 5.4**: Integration Testing for UPTM Academic AI Assistant Tools

| No | Integration Scenario | Description | Input | Expected Output | Error/Issue Found | Status |
|---|---|---|---|---|---|---|
| 1. | Login to Dashboard Access | Tests the integration between authentication and role-based dashboard routing. | Valid username & password. | User redirected to correct dashboard (SuperAdmin/Admin/User). | None | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2. | Sign Up to Database Record Creation | Checks if user registration form correctly stores new user data into database. | User registration details. | User data saved successfully in users table. | None | Pass |
| 3. | Admin/ SuperAdmin Creates Template to User Views Template | Verifies that templates created by admin appear in user's available template list. | Admin/ SuperAdmin adds new template. | Template visible to users after login. | None | Pass |
| 4. | User Selects Template to Generates Output via OpenAI API | Tests the connection between user input, system request, and AI-generated output retrieval. | Selected template and enter user input. | AI-generated content displayed in output page. | None | Pass |
| 5. | AI Output to Document History Storage | Checks that generated output is automatically saved into user's document history database. | Generated AI content. | Output stored in database and visible under "Generated Contents". | None | Pass |
| 6. | Edit/Delete Document to Database Update | Verifies that user edits or deletions of documents correctly update the | Edited/Deleted document ID. | Document record updated or removed from database. | None | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | database records. | | | | |
| 7. | Change Password to Re-login Verification | Ensures password update module communicates correctly with authentication module. | Old and new password. | User can log in with new password only. | None | Pass |
| 8. | Logout to Session Termination | Confirms session handling between frontend and backend properly terminates user session. | Click logout button. | User redirected to login page, session cleared. | None | Pass |
| 9. | Database to Dashboard Statistics (Admin) | Tests integration between stored data and analytics display for admin dashboard. | Existing user & document data. | Dashboard statistics display accurate data counts. | None | Pass |
| 10. | System to OpenAI API Connection Failure | Tests system behavior when AI API connection fails. | Invalid API key / no internet. | Error message displayed without crashing the system. | None | Pass |

Based on the integration testing results, all system modules were successfully integrated and functioned as expected. No major errors or system failures were encountered during the testing process, indicating that the system components communicate effectively and support the intended system workflow.

## 5.5 System Testing

System testing is the final stage of the testing process where the entire system is evaluated as a whole to ensure that it meets both the functional and non-functional requirements specified during the system design phase. The objective of this testing phase is to verify that all modules within the system, including the Superadmin module, Admin module, and User module, operate correctly when integrated. In addition, this testing phase ensures that the system performs efficiently, securely, and reliably in a real-world environment.

For this project, system testing is divided into two main categories:

1. Functional Testing, which verifies that all system features operate according to the defined system requirements.
2. Non-Functional Testing, which evaluates system quality attributes such as performance, usability, reliability, security, compatibility, and privacy.

Functional Testing (Superadmin)
This section verifies that the Superadmin module, which represents the highest level of authority in the system, operates correctly. The Superadmin is responsible for managing administrators, controlling system templates, and monitoring generated documents within the system.

**Table 5.5**: Functional Testing (Superadmin)

| No | Function | Description | Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|---|
| 1. | Login | Superadmin logs in using valid credentials. | Email & Password | System grants access to Superadmin Dashboard. | As expected | Pass |
| 2. | Sign Up | Superadmin registers a new account in the system. | Registration details | Account successfully created. | As expected | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| 3. | Manage Account | Superadmin manages personal account settings. | Profile information | Account information updated successfully. | As expected | Pass |
| 4. | Update Profile | Superadmin updates personal information such as name and email. | Updated profile data | Profile updated confirmation displayed. | As expected | Pass |
| 5. | Change Password | Superadmin changes account password. | Old and new password | Password successfully updated. | As expected | Pass |
| 6. | Manage All Users | Superadmin views and manages all users including admin, lecturers, and students. | Select user record | User details displayed and manageable. | As expected | Pass |
| 7. | Add New User | Superadmin creates a new user account and assigns a role. | User registration details | New user account successfully added. | As expected | Pass |
| 8. | Templates Management | Superadmin creates, updates, or deletes AI templates. | Template name & content | Template successfully saved or updated. | As expected | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| 9. | View All Templates | Superadmin views all available templates in the system. | Click “All Templates” | Template list displayed correctly. | As expected | Pass |
| 10. | Document Management | Superadmin views all generated document records. | Access document list | Document history displayed correctly. | As expected | Pass |
| 11. | Edit/Delete Document | Superadmin edits or removes generated documents. | Select document | Document successfully updated or deleted. | As expected | Pass |

Functional Testing (Admin)

This section verifies that administrative functions such as user monitoring, template management, and document tracking operate correctly according to system requirements.

**Table 5.6**: Functional Testing (Admin)

| No | Function | Description | Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|---|
| 1. | Login | Admin logs in using valid credentials. | Email & Password | System grants access to Admin Dashboard. | As expected | Pass |
| 2. | Sign Up | Admin registers using correct credentials. | Registration details | Account created successfully. | As expected | Pass |
| 3. | Manage Account | Admin views and manages user accounts. | Select user record | User details displayed, able to delete inactive users. | As expected | Pass |
| 4. | Update Profile | Admin updates name, email, or contact info. | Updated profile data | Profile updated confirmation shown. | As expected | Pass |
| 5. | Change Password | Admin changes or resets password. | Old and new password | Password successfully updated. | As expected | Pass |
| 6. | Templates Management | Admin creates or updates AI templates. | Template name, content | Template saved and visible in list. | As expected | Pass |
| 7. | View All Templates | Admin reviews all AI templates. | Click “All Templates” | Template list displayed. | As expected | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8. | Document Management | Admin views generated document history. | Access document list | Document details displayed. | As expected | Pass |
| 9. | Edit/Delete Document | Admin modifies or removes generated documents. | Select document | Document updated or deleted successfully. | As expected | Pass |

Functional Testing (Users: Lecturer / Student)

This section verifies that system users can access their accounts, utilize AI templates provided by the administrator, generate academic content, and manage their generated documents effectively.

**Table 5.7**: Functional Testing (Users: Lecturer/Student)

| No | Function | Description | Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|---|
| 1. | Login | User logs in using registered credentials. | Email & Password | Redirected to user dashboard. | As expected | Pass |
| 2. | Sign Up | User registers new account within institution. | Registration details | Account created successfully. | As expected | Pass |
| 3. | Manage Account | User updates personal profile. | Profile details | Profile successfully updated. | As expected | Pass |
| 4. | Change Password | User changes password. | Old and new password | Password updated successfully. | As expected | Pass |
| 5. | View Account Details | User views stored personal info. | Click “View Profile” | Profile details displayed. | As expected | Pass |
| 6. | Use Template Provided by Admin | User selects template to generate academic content. | Choose template | AI generates content using template. | As expected | Pass |
| 7. | Document Management | User views list of generated documents. | Access document history | Document list | As expected | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | displayed correctly. | | |
| 8. | Edit/Delete Document | User edits or removes document. | Select document | Document updated or deleted. | As expected | Pass |

Non-Functional Testing

Non-functional testing evaluates the overall system quality in terms of performance, usability, reliability, security, compatibility, and privacy. These tests ensure that the system performs efficiently and provides a positive experience for its users.

**Table 5.8**: Non-Functional Testing

| No | Criteria | Description | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 1. | Performance | The system should process AI-generated requests efficiently and return the output within an acceptable time frame. | AI-generated academic content should be displayed within 5 seconds after submission. | As expected | Pass |
| 2. | Usability | The interface should be user-friendly and intuitive for both students and lecturers. | Users can easily navigate the system without guidance. | As expected | Pass |
| 3. | Reliability | The system should operate continuously without crashes or data loss. | System remains stable during all operations. | As expected | Pass |
| 4. | Security | User accounts and data are protected from unauthorized access. | Secure login and encrypted passwords. | As expected | Pass |
| 5. | Compatibility | The system must work smoothly on major browsers (Chrome, Edge, Firefox). | System runs properly on all tested browsers. | As expected | Pass |
| 6. | Privacy | Sensitive user and information data are | All user-generated data | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | stored securely within the institutional database. No data is shared with third-party services. | remains private and inaccessible externally. | | |

Overall, the system testing results confirm that all functional and non-functional requirements have been successfully satisfied. The Superadmin, Admin, and User modules operate as expected, and the system demonstrates stable performance, strong security mechanisms, and reliable functionality. Based on these results, the UPTM Academic AI Assistant Tools system is considered stable and suitable for deployment in a controlled environment for further evaluation and institutional use.

## 5.6 Acceptance Testing

Acceptance testing is the final stage of the testing process, conducted to ensure that the developed system meets the requirements and expectations of the end users and the client. It focuses on validating the overall functionality, usability, and performance of the system in a real-world environment. The goal is to confirm that the system is ready for deployment and fulfills all objectives outlined during the planning and design phases. In this project, two types of acceptance testing were conducted, Client Acceptance Testing (CAT) and User Acceptance Testing (UAT). The client acceptance test was performed by the system's supervisor (acting as the client), while the user acceptance test involved lecturers and students from UPTM who represented the target end users.

### 5.6.1 Client Acceptance Testing

The Client Acceptance Testing (CAT) was carried out by Madam Nadiah Yusof, AI lecturer at Universiti Poly-Tech Malaysia (UPTM), acting as the client representative. The objective of this test was to ensure that the developed system meets the initial project requirements, aligns with the proposed objectives, and functions according to the agreed specifications.

The client verified critical modules such as authentication (login/register), AI-generated academic tools, admin management panel, and user management. The client also reviewed the user interface design, navigation flow, and security measures such as role-based access control and data privacy.

The overall result showed that the system met all client expectations. Minor design adjustments were suggested for better clarity and user experience. The client approved the system for final submission, confirming that it successfully fulfilled the intended purpose of assisting users in generating academic content securely and efficiently.

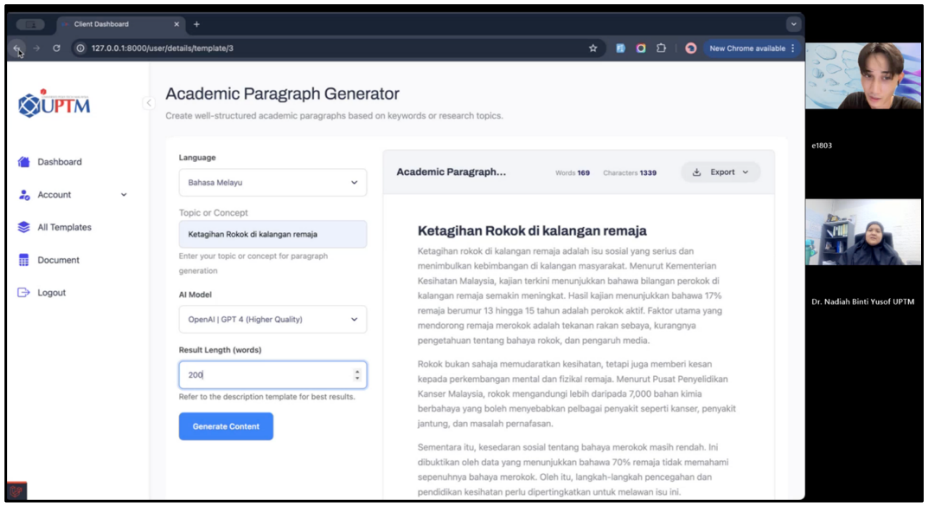



**Figure 5.1:** Client Acceptance Testing with clients

**Table 5.9**: Client Acceptance Testing

| No | Test Criteria | Description | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 1. | Functional Requirements | Verify that all core features (login, AI generation, user management) work correctly. | All functions operate as expected. | As expected | Pass |
| 2. | User Interface | Evaluate layout consistency, button placement, and color scheme. | Professional and consistent UI design. | As expected | Pass |
| 3. | Data Security | Check password hashing and session protection. | User data and sessions are secured. | As expected | Pass |
| 4. | Role Control | Confirm SuperAdmin, | SuperAdmin, Admin and | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | admin and user access levels are separated. | user dashboards work independently. | | |
| 5. | AI Integration | Ensure OpenAI API returns relevants content. | AI-generated responses appear correctly. | As expected | Pass |

### 5.6.2 User Acceptance Testing

The User Acceptance Testing (UAT) involved selected lecturers and students from Universiti Poly-Tech Malaysia (UPTM). The purpose of this test was to evaluate the system's usability, functionality, and overall user satisfaction in performing academic-related tasks using AI tools.

Test participants were asked to perform key activities such as logging in, managing profiles, selecting templates, generating academic content using AI, and viewing saved outputs. Their feedback focused on ease of use, clarity of interface, and speed of response.

The users reported that the system was user-friendly, fast, and effective in generating academic materials such as summaries and explanations. Some users suggested additional features like history search and download options, which may be considered in future improvements.

**Table 5.10**: User Acceptance Testing

| No | Test Criteria | Description | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 1. | Usability | Evaluate the simplicity and intuitiveness of the system. | Easy to navigate and understand. | As expected | Pass |
| 2. | Template Function | Test AI template selection and output generation. | Generates correct content according to user input. | As expected | Pass |
| 3. | Profile Management | Update and change user profile details. | Profile updated successfully. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| 4. | Security | Ensure sessions are protected after logout. | Sessions expire immediately after logout. | As expected | Pass |
| 5. | Satisfaction | Collect overall feedback from users. | Positive response from all testers. | As expected | Pass |

Based on both client and user acceptance testing, the UPTM Academic AI Assistant Tools system has successfully met all the defined functional and non-functional requirements. The client verified that the system operates according to specifications, while users confirmed that it is easy to use, secure, and efficient for generating academic content using AI. Both parties expressed satisfaction with the overall performance and reliability of the system. Therefore, the system is considered fully accepted and ready for deployment within the institutional environment.

# 6 CONCLUSION

## 6.1 Introduction

This chapter provides a comprehensive summary and reflection on the overall progress, development, and outcomes of the UPTM Academic AI Assistant Tools project. It discusses the project schedule, risk management, achievements based on objectives, encountered constraints, future recommendations, and an overall conclusion. The chapter highlights how the system was successfully developed and tested according to its goals and provides critical insights into the challenges faced and lessons learned during the process.

## 6.2 Project Schedule

The development of this project followed a planned schedule using an Agile methodology, which divided the work into several iterative phases. This allowed flexibility in managing time, making continuous improvements, and incorporating feedback during development. Initially, the project was planned to be completed within the semester duration, beginning with requirement analysis and ending with final testing and documentation. Although minor adjustments were necessary particularly during the integration and testing stages due to connection issues with the OpenAI API the overall project was completed successfully within the planned timeframe. The managed time effectively by prioritizing critical components such as authentication, template management, and output generation before focusing on user interface refinements.

## 6.2.1 Work Breakdown Structure

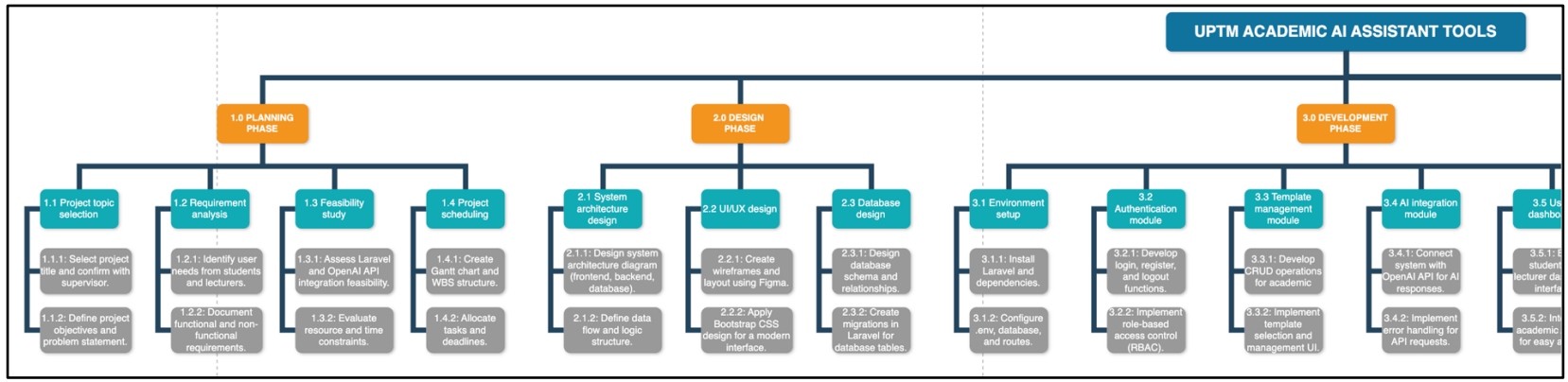



**Figure 6.1:** WBS 1.0

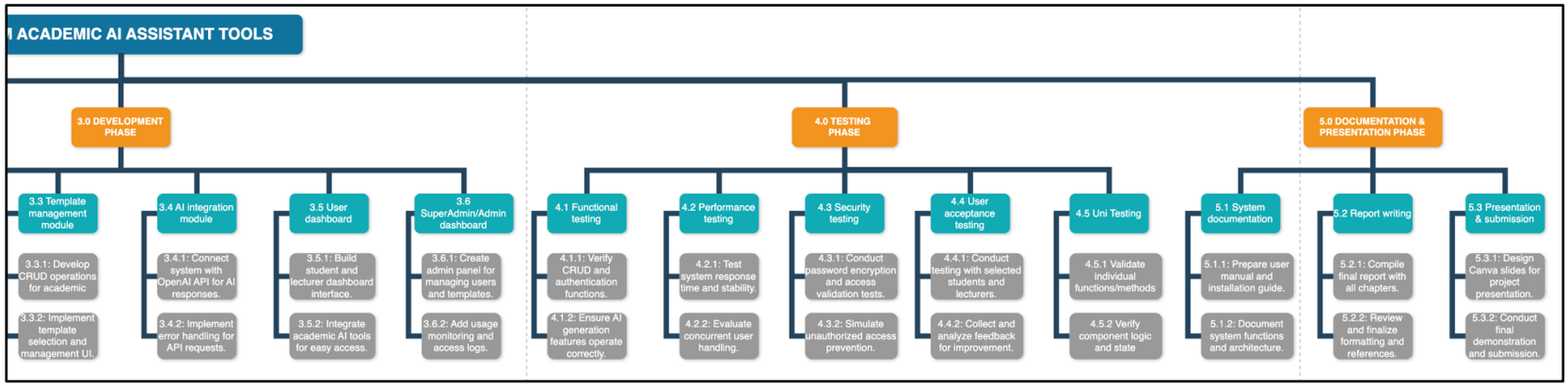



**Figure 6.2**: WBS 2.0

The Work Breakdown Structure (WBS) is an essential project management tool that divides the entire development process of the UPTM Academic AI Assistant Tools into structured and manageable sections. It provides a clear hierarchy of project phases, sub-tasks, and deliverables, ensuring that every stage of the project is well-planned, traceable, and achievable within the given timeframe. This structured approach helped maintain focus, streamline workflow, and ensure that all project objectives were successfully accomplished according to the initial plan.

The project was divided into five major phases, namely Planning, Design, Development, Testing, and Documentation & Presentation. Each phase contained specific sub-tasks and work packages to guide implementation and ensure smooth progress throughout the project lifecycle.

The Planning Phase focused on defining the project foundation. This included selecting the project topic, identifying the problem statement, analyzing user requirements from both students and lecturers, and evaluating the feasibility of integrating Laravel with the OpenAI API. A schedule was then developed using the Gantt Chart and WBS to allocate tasks, set priorities, and ensure that all deliverables were completed on time.

Next, the Design Phase involved creating the system architecture, designing the user interface, and structuring the database. Wireframes and UI layouts were developed using Figma and styled with Bootstrap to ensure a clean and responsive design. The database schema and table relationships were also constructed through Laravel migrations to ensure smooth data handling and scalability.

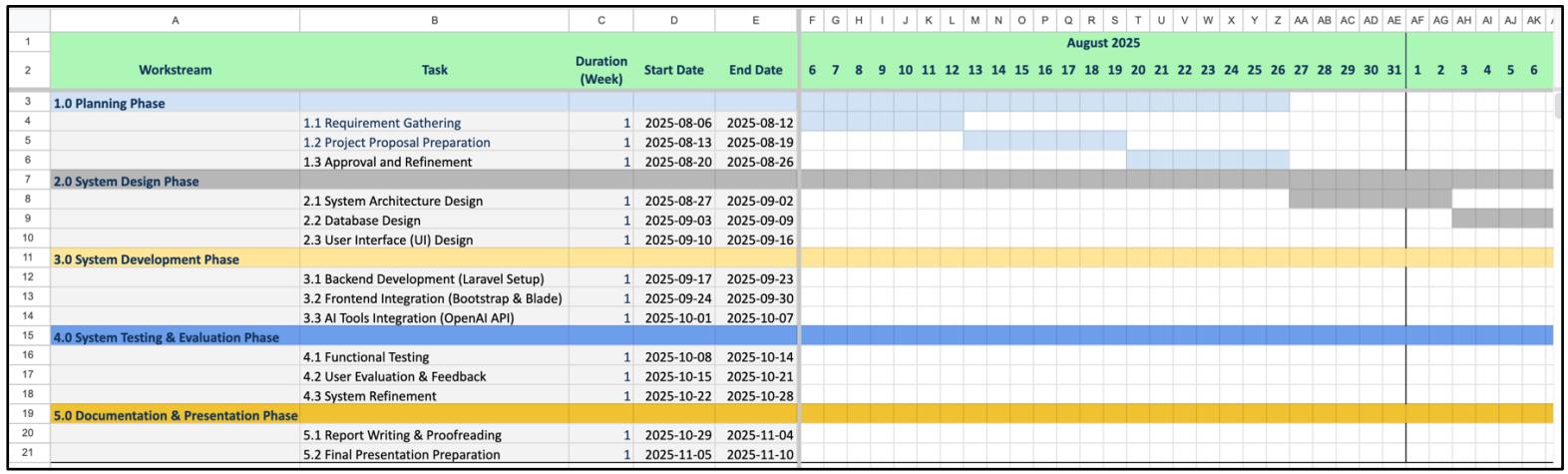
The Development Phase was the most intensive stage of the project. It began with the setup of the Laravel environment and continued with the development of the authentication system, which included login, registration, and role-based access control (RBAC). The academic template management module was then built to handle CRUD operations and allow users to manage different academic AI templates. The AI integration module was also implemented using the OpenAI API to generate automated outputs based on user inputs. Separate dashboards were created for users (students and lecturers) and administrators, enabling role-specific access and management functionalities.

After the development phase, a comprehensive Testing Phase was conducted to ensure all modules functioned correctly and met rigorous performance standards. The process began with Unit Testing, which validated the logic of individual functions and components in isolation to ensure code-level. Following this, Functional Testing verified that core features such as template creation and AI output generation operated as intended within the integrated system. To ensure system reliability and robustness, Security and Performance Testing were executed to assess data protection and system stability under load. Finally, User Acceptance Testing (UAT) was carried out with selected students and lecturers to gather feedback on the system's overall usability and effectiveness in a real-world academic context.

Finally, the Documentation & Presentation Phase focused on preparing the final deliverables. This included producing comprehensive system documentation, user manuals, and the final report that detailed each stage of the development process. A PowerPoint presentation and live system demonstration were also conducted to showcase the project's outcomes and validate its success.

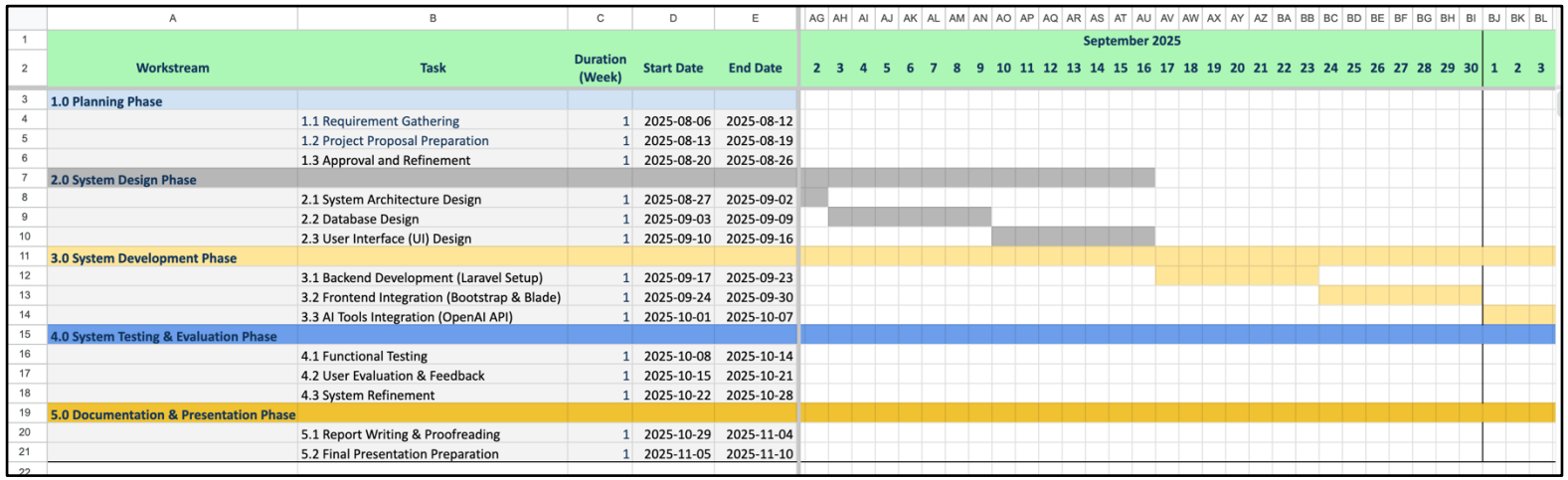
Overall, the Work Breakdown Structure played a vital role in guiding the project's direction and ensuring that all tasks were organized and executed systematically. It provided a clear roadmap from the initial planning to the final implementation, ensuring that each objective was met efficiently within the project's scope and schedule. This structured approach also helped identify dependencies between tasks and maintain consistency throughout the project development process, contributing to the overall success of the UPTM Academic AI Assistant Tools system.

## 6.2.2 Gantt Chart



| Workstream | Task | Duration (Week) | Start Date | End Date |
|---|---|---|---|---|
| 1.0 Planning Phase | | | | |
| | 1.1 Requirement Gathering | 1 | 2025-08-06 | 2025-08-12 |
| | 1.2 Project Proposal Preparation | 1 | 2025-08-13 | 2025-08-19 |
| | 1.3 Approval and Refinement | 1 | 2025-08-20 | 2025-08-26 |
| 2.0 System Design Phase | | | | |
| | 2.1 System Architecture Design | 1 | 2025-08-27 | 2025-09-02 |
| | 2.2 Database Design | 1 | 2025-09-03 | 2025-09-09 |
| | 2.3 User Interface (UI) Design | 1 | 2025-09-10 | 2025-09-16 |
| 3.0 System Development Phase | | | | |
| | 3.1 Backend Development (Laravel Setup) | 1 | 2025-09-17 | 2025-09-23 |
| | 3.2 Frontend Integration (Bootstrap & Blade) | 1 | 2025-09-24 | 2025-09-30 |
| | 3.3 AI Tools Integration (OpenAI API) | 1 | 2025-10-01 | 2025-10-07 |
| 4.0 System Testing & Evaluation Phase | | | | |
| | 4.1 Functional Testing | 1 | 2025-10-08 | 2025-10-14 |
| | 4.2 User Evaluation & Feedback | 1 | 2025-10-15 | 2025-10-21 |
| | 4.3 System Refinement | 1 | 2025-10-22 | 2025-10-28 |
| 5.0 Documentation & Presentation Phase | | | | |
| | 5.1 Report Writing & Proofreading | 1 | 2025-10-29 | 2025-11-04 |
| | 5.2 Final Presentation Preparation | 1 | 2025-11-05 | 2025-11-10 |

**Figure 6.3:** Gant Chart 1.0



| Workstream | Task | Duration (Week) | Start Date | End Date |
|---|---|---|---|---|
| 1.0 Planning Phase | | | | |
| | 1.1 Requirement Gathering | 1 | 2025-08-06 | 2025-08-12 |
| | 1.2 Project Proposal Preparation | 1 | 2025-08-13 | 2025-08-19 |
| | 1.3 Approval and Refinement | 1 | 2025-08-20 | 2025-08-26 |
| 2.0 System Design Phase | | | | |
| | 2.1 System Architecture Design | 1 | 2025-08-27 | 2025-09-02 |
| | 2.2 Database Design | 1 | 2025-09-03 | 2025-09-09 |
| | 2.3 User Interface (UI) Design | 1 | 2025-09-10 | 2025-09-16 |
| 3.0 System Development Phase | | | | |
| | 3.1 Backend Development (Laravel Setup) | 1 | 2025-09-17 | 2025-09-23 |
| | 3.2 Frontend Integration (Bootstrap & Blade) | 1 | 2025-09-24 | 2025-09-30 |
| | 3.3 AI Tools Integration (OpenAI API) | 1 | 2025-10-01 | 2025-10-07 |
| 4.0 System Testing & Evaluation Phase | | | | |
| | 4.1 Functional Testing | 1 | 2025-10-08 | 2025-10-14 |
| | 4.2 User Evaluation & Feedback | 1 | 2025-10-15 | 2025-10-21 |
| | 4.3 System Refinement | 1 | 2025-10-22 | 2025-10-28 |
| 5.0 Documentation & Presentation Phase | | | | |
| | 5.1 Report Writing & Proofreading | 1 | 2025-10-29 | 2025-11-04 |
| | 5.2 Final Presentation Preparation | 1 | 2025-11-05 | 2025-11-10 |

**Figure 6.4:** Gant Chart 2.0

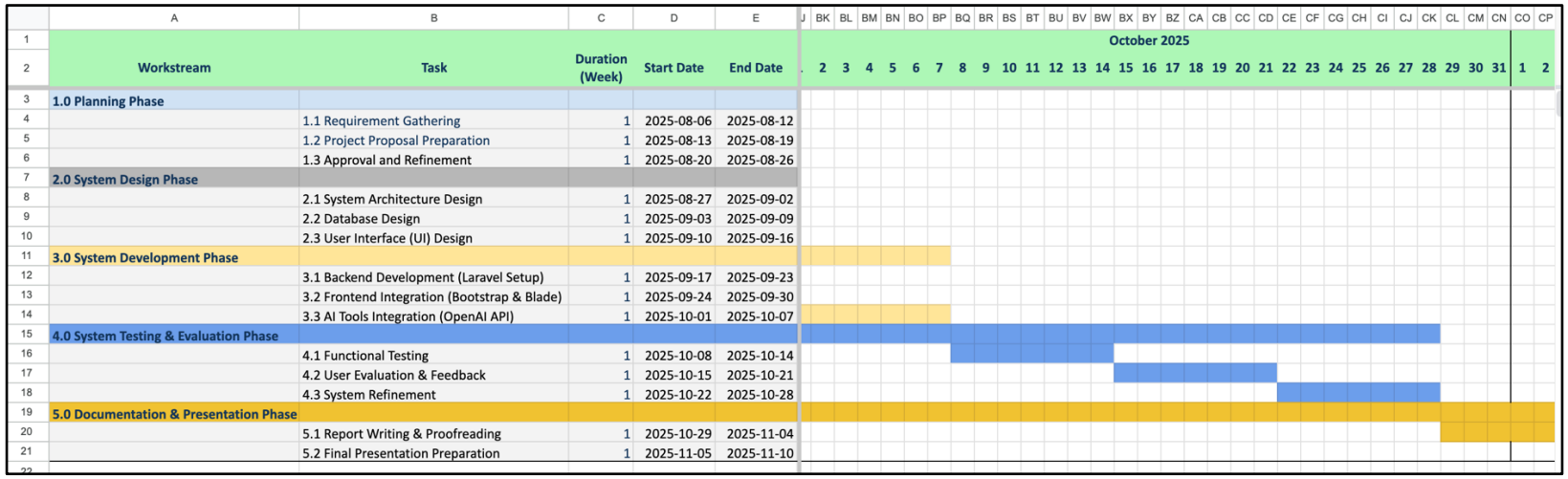



**Figure 6.5:** Gant Chart 3.0

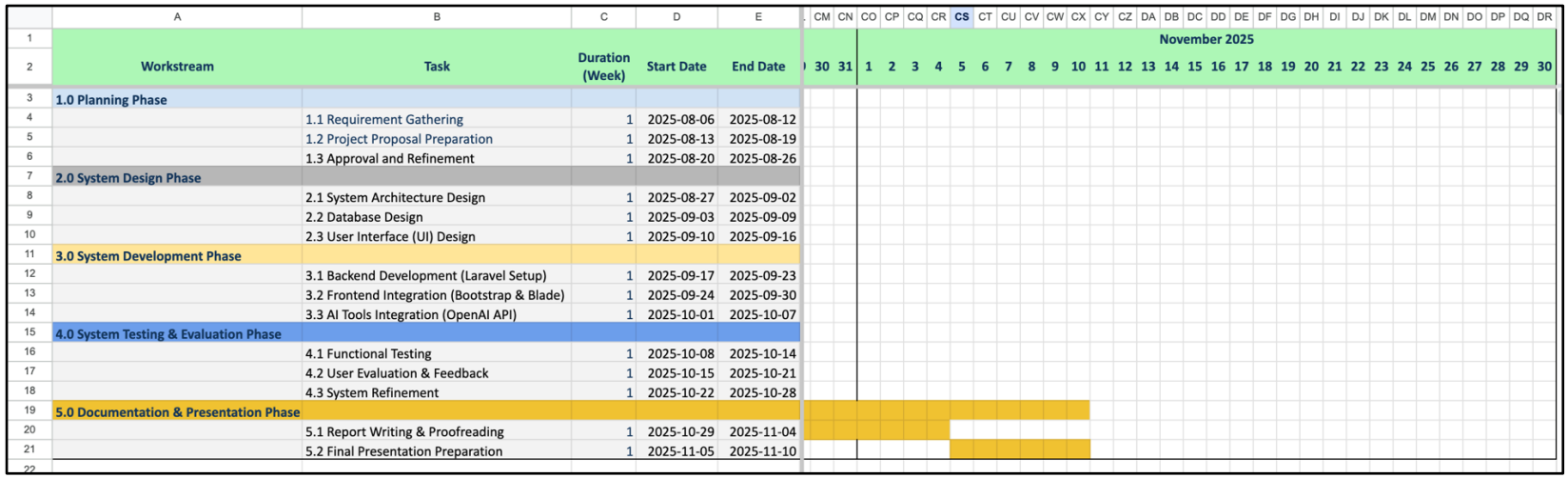



**Figure 6.6:** Gant Chart 4.0

The Gantt Chart outlines the overall project timeline from Week 1 (6 August 2025) to Week 14 (10 November 2025), covering each major phase of development, testing, and presentation. The chart serves as a visual and structured schedule to ensure all project activities are completed systematically and within the academic timeframe.

The project begins with the Planning Phase, which spans the first three weeks. During this stage, tasks such as Requirement Gathering (6–12 August), Project Proposal Preparation (13–19 August), and Approval and Refinement (20–26 August) were conducted. These tasks focused on defining the project's objectives, scope, and deliverables while ensuring alignment with academic requirements.

The System Design Phase follows, lasting from 27 August to 16 September 2025. This phase includes System Architecture Design, Database Design, and User Interface (UI) Design. The main goal during this period was to establish the system's structural blueprint and user interface layout. The team utilized design tools and database modeling techniques to ensure that the system met functional and user experience standards before coding began.

Next, the System Development Phase runs from 17 September to 7 October 2025. This stage is divided into three key development tasks: Backend Development (Laravel Setup), Frontend Integration (Bootstrap + Blade), and AI Tools Integration (OpenAI API). The backend setup involved configuring Laravel, building authentication features, and setting up the database connection. The frontend integration focused on creating responsive layouts and connecting UI components. Finally, the AI integration involved implementing OpenAI-powered academic tools, testing API functionality, and ensuring stable data exchange between the backend and AI services.

The System Testing & Evaluation Phase takes place from 8 October to 28 October 2025. It covers Functional Testing, User Evaluation & Feedback, and System Refinement. During this period, the system underwent rigorous testing to identify and fix bugs. Feedback was collected from both students and lecturers to evaluate usability and system effectiveness. The refinement stage focused on optimizing performance, improving user experience, and ensuring the system met the defined project objectives.

Lastly, the Documentation & Presentation Phase spans from 29 October to 10 November 2025. This final stage includes Report Writing & Proofreading and Final Presentation Preparation. The project report was compiled, refined, and formatted according to institutional

guidelines, while presentation materials were prepared to showcase the system's features, objectives, and outcomes during the final presentation session.

Overall, the Gantt Chart reflects a well-organized and realistic project timeline. Each phase was strategically planned to balance development, testing, and documentation efforts, ensuring the successful completion of the UPTM Academic AI Assistant Tools project within the allocated academic duration.

## 6.3 Risk Management

Risk management played an important role in ensuring the project's success. Several potential risks were identified, analyzed, and mitigated effectively throughout development. The following table outlines the main risks and their corresponding mitigation plans, along with any risks that actually occurred.

**Table 6.1**: Risk Management

| Risk | Description | Impact | Mitigation Strategy | Outcome |
|---|---|---|---|---|
| API Integration Failure | Connection errors during OpenAI API request testing. | High | Conduct API testing in local environment and handle exceptions using error validation. | Occurred once, resolved after debugging. |
| Security Breach Risk | Possible exposure of user data during system access. | High | Use Laravel authentication, encrypted passwords, and store data internally within UPTM. | Prevented successfully. |
| Time Constraints | Limited time for debugging and testing. | Medium | Applied Agile sprint planning and weekly progress reviews. | Managed effectively. |
| User Experience Risk | New users may find it difficult to understand system flow. | Low | Simplified interface using pre-built academic templates. | No major issues reported. |

Through proactive management, all critical risks were handled promptly, allowing the project to remain on track and ensuring stable performance.

## 6.4 Achievement

The UPTM Academic AI Assistant Tools successfully met its objectives as stated in the earlier chapter. The system achieved its goals by providing ready-to-use academic AI features, improving user productivity, and ensuring data security within the university's environment. Below is the detailed evaluation of each project objective and its corresponding proof.

### 6.4.1 To Provide Ready-to-Use Academic AI Tools

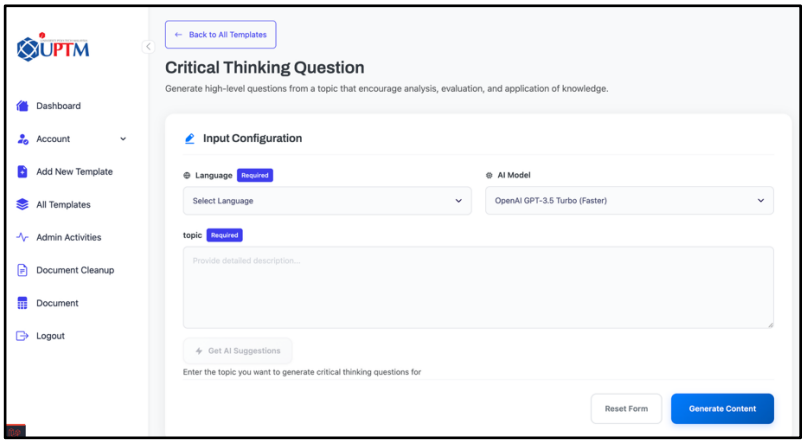



**Figure 6.7:** AI Template

This objective was successfully accomplished through the implementation of multiple academic AI templates. These tools were specifically designed to simplify the AI usage process by removing the need for users to write complex prompts.

Screenshots from the Template Management and User Dashboard modules demonstrate how users can conveniently select a tool, enter their topic, and generate academic outputs with minimal effort. This clearly proves that the system fulfilled its goal of accessibility, user-friendliness, and readiness for immediate use.

### 6.4.2 To Develop an AI System for Productive Academic Tasks

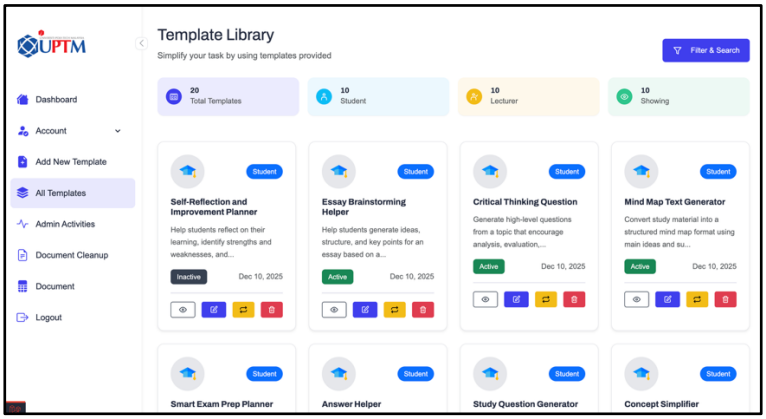



**Figure 6.8:** Student and Lecturer Features

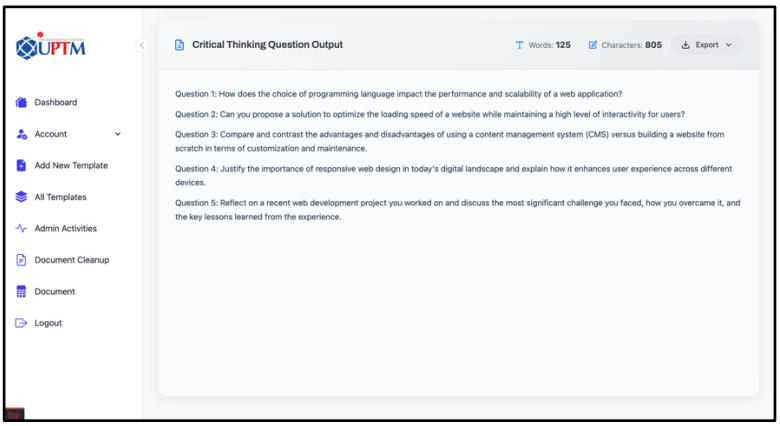



**Figure 6.9:** Student and Lecturer Features

This objective focused on improving productivity and academic efficiency for both students and lecturers. The system allows students to quickly generate academic content such as summaries, comparisons, and concept explanations, while lecturers can utilize the tools to create teaching materials, feedback, and explanations effortlessly.

Performance and functionality testing verified that all AI templates produced relevant and coherent academic outputs. The results and screenshots of sample outputs demonstrate that the system significantly supports faster task completion and enhanced academic productivity.

### 6.4.3 To Provide a Secure and Institution-Friendly AI System

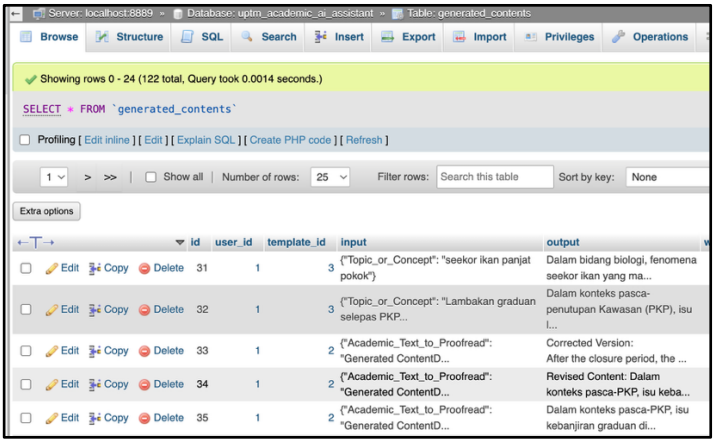



**Figure 6.10:** AI interaction with users database

The final objective was to ensure the system operated within a secure and compliant environment. This goal was fully achieved through several security implementations. All user information, including login credentials and AI interaction with users, is securely stored in UPTM's internal database, ensuring data confidentiality and compliance with institutional policies.

Laravel's built-in authentication and password hashing features were applied to strengthen user security. Furthermore, role-based access control (RBAC) was implemented to differentiate permissions between users and administrators. Security testing confirmed that no unauthorized access or data breaches occurred, indicating that the system successfully achieved a secure, stable, and institution-friendly AI environment.

## 6.5 Constraint and Limitation

During the development of this project, several constraints and limitations were encountered that influenced the scope, progress, and final output of the system. One of the main constraints was time limitation. The project had to be completed within a fixed academic schedule, which restricted the amount of testing and fine-tuning that could be performed. As a result, certain advanced features were postponed to future phases. Despite this, the main core functions, including authentication, AI template management, and secure generation, were successfully implemented.

Another significant constraint was the limited development timeline, which necessitated a strategic prioritization of features. Due to the fixed academic schedule, the time available for complex logic implementation was restricted, meaning that several highly interactive AI features could not be fully developed within this phase. Consequently, certain advanced features were postponed to future phases, particularly those aimed at increasing system automation and real-time user interaction. To mitigate this, the development focused on perfecting the core architecture using Laravel's built-in testing tools, ensuring a stable foundation was ready for these interactive enhancements in future iterations

A further limitation was financial constraints related to the AI integration process. Because the OpenAI API requires a paid subscription to obtain an active API key, the implementation relied on the developer's own budget. This created restrictions on the number of test requests and limited large-scale experimentation with the AI tools. To minimize cost, the system focused on essential functionality and demonstration-level integration instead of extended or high-frequency API calls. Despite the budget limitation, the integration successfully demonstrated the potential of using AI to generate academic outputs effectively and efficiently.

Lastly, limitation encountered during the evaluation phase was the difficulty in obtaining responses from lecturers compared to students for the questionnaire and system testing analysis. While many students actively participated and provided valuable feedback, the response rate from lecturers was relatively low due to their busy academic schedules and limited availability. This imbalance affected the diversity of perspectives in the evaluation results, making it more student-oriented. Despite this limitation, the collected data still provided useful insights into user satisfaction, usability, and the overall functionality of the system, which were sufficient to assess its effectiveness and guide future improvements.

## 6.6 Future Work and Recommendation

Although the UPTM Academic AI Assistant Tools has successfully achieved its core objectives and implemented most of the initially proposed features, there are still several opportunities for future enhancement to further improve the system's flexibility, intelligence, and long-term value.

One of the key future improvements is the integration of multiple AI models and services beyond the current OpenAI API. At present, the system primarily relies on a single AI provider, however, expanding support to include other AI platforms and models such as open-source models or alternative commercial APIs would provide greater flexibility, reduce dependency on a single service, and allow performance comparison between different models. This multi-model integration would also enable the system to select the most suitable AI model based on specific academic tasks, such as content generation, summarization, or question answering.

Another important area for enhancement is the automation of prompt generation for content creation. Currently, administrators or super administrators are required to manually write prompts when creating content generator templates. In future development, this process can be significantly improved by incorporating AI-assisted prompt generation. For example, the system could suggest or automatically generate optimized prompts based on the selected topic, category, or intended output. This would not only reduce manual effort but also ensure higher-quality and more consistent AI-generated results. By leveraging AI to assist in prompt engineering, the system becomes more user-friendly, especially for users who may not be familiar with crafting effective prompts.

In addition, future work can focus on improving the intelligence and adaptability of the system through smarter AI behavior. This includes features such as context-aware responses, adaptive learning from user interactions, and personalized content generation based on user preferences. These enhancements would make the system more responsive to individual user needs and improve the overall user experience.

Lastly, continuous improvement through user feedback and iterative updates should remain a priority. Even though core functionalities have been successfully implemented, regular system evaluation, usability testing, and feature refinement will ensure that the platform remains relevant and effective in supporting academic activities at Universiti Poly-Tech Malaysia (UPTM).

## 6.7 Conclusion

In conclusion, the development of the UPTM Academic AI Assistant Tools project has successfully achieved its intended objectives and demonstrated its potential as an innovative academic support system for Universiti Poly-Tech Malaysia (UPTM). The system was designed and implemented to provide ready-to-use academic AI tools that simplify the learning and teaching process while promoting efficiency, accessibility, and data security.

Throughout the project, various milestones were accomplished, including the integration of OpenAI API for automated content generation, the development of academic-focused templates, and the implementation of secure authentication features through Laravel's framework. Testing results confirmed that the system performed as expected, providing fast, and contextually relevant academic outputs that benefited both students and lecturers.

Despite encountering certain limitations such as restricted testing due to time, budget, and limited lecturer participation in the evaluation process the project successfully delivered a functional prototype that demonstrates the practical use of artificial intelligence in academic environments. These challenges provided valuable lessons in system design, project management, and user evaluation, all of which contribute to the developer's growth and understanding of real-world software implementation.

Looking ahead, the project sets a strong foundation for future development, where enhancements such as full live API integration, cloud deployment, and broader user testing can further elevate the system's functionality and scalability. With continued support and improvement, the UPTM Academic AI Assistant Tools has the potential to become a key digital platform that promotes innovative, efficient, and AI-driven academic practices within UPTM and beyond.

# Appendix A – Requirements Specification Document

This appendix presents the agreed system requirements for the UPTM Academic AI Assistant Tools project. The requirements were established based on the system's objectives, user needs, and development constraints. They are categorized into Functional Requirements, Non-Functional Requirements, System Constraints, and Assumptions & Dependencies.

## A.1 Functional Requirements

The following functional requirements define the core features and operations that the system must perform for both administrators and users (lecturers and students). These requirements ensure that all essential academic AI functions and user management capabilities are available and functioning correctly.

| No | Functional Requirement | Description |
|---|---|---|
| 1. | User Registration and Login | The system shall allow users (Admin, Lecturer, Student) to register and log in using valid credentials. |
| 2. | User Profile Management | Users shall be able to view, update, and manage their personal profiles including name, email, and password. |
| 3. | Admin/SuperAdmin Account Management | The system shall allow the Admin or SuperAdmin to manage user accounts, including viewing, updating, and deleting inactive users. |
| 4. | AI Template Management (Admin/SuperAdmin) | Admin or SuperAdmin shall be able to create, edit, or delete academic AI templates. |
| 5. | AI Content Generation (User) | Users shall be able to select a template provided by the Admin/SuperAdmin and generate AI- |

| | | |
|---|---|---|
| | | generated academic outputs through the OpenAI API. |
| 6. | Document Management | Users shall be able to view, edit, or delete generated documents within their dashboard. |
| 7. | Role-Based Access Control (RBAC) | The system shall enforce different access levels for SuperAdmin, Admins and Users to prevent unauthorized actions. |
| 8. | Logout Function | The system shall allow all users to securely log out from their sessions. |
| 9. | Dashboard Interface | The system shall provide a centralized dashboard for each user role to access their respective functions easily. |
| 10. | Validation and Error Handling | The system shall validate all user inputs and provide error messages for invalid actions such as incorrect login details. |

## A.2 Non-Functional Requirements

The non-functional requirements define the overall quality attributes that ensure the system's usability, security, and performance in the UPTM environment.

| No | Non-Functional Requirement | Description |
|---|---|---|
| 1. | Performance | The system should process AI requests and display results within 5 seconds to maintain user satisfaction. |
| 2. | Usability | The interface must be intuitive and user-friendly, allowing both lecturers and students to use the system without prior training. |
| 3. | Reliability | The system must function continuously without unexpected crashes or data corruption during normal operations. |
| 4. | Security | User credentials and generated data must be protected through Laravel authentication, hashed passwords, and restricted access. |
| 5. | Compatibility | The system must run smoothly on all major browsers, including Google Chrome, Microsoft Edge, and Mozilla Firefox. |
| 6. | Scalability | The system should be designed to support potential expansion to include more templates or users in the future. |
| 7. | Maintainability | The source code should be structured and documented for easy updates and debugging. |

| | | |
|---|---|---|
| 8. | Privacy | Sensitive user data and generated content must not be shared with third parties and should remain stored in UPTM's internal database. |

A.3 System Constraints

Several constraints were identified during development, affecting the design and deployment of the system:

| No | Constraint | Description |
|---|---|---|
| 1. | Time Constraint | The project was required to be completed within two academic semester (28 weeks), limiting extended testing and optimization. |
| 2. | Financial Constraint | The OpenAI API requires a paid subscription, and due to budget limitations, API requests were restricted to minimal testing. |
| 3. | Server Access | Deployment on UPTM's institutional server required administrative approval, which delayed testing on live environments. |
| 4. | Limited Feedback Source | Obtaining responses from lecturers for questionnaire analysis was difficult compared to students due to their busy schedules. |
| 5. | Institutional Data Policy | Data privacy policies prevented the use of third-party penetration testing tools. |

## A.4 Assumptions and Dependencies

The development and successful operation of the UPTM Academic AI Assistant Tools rely on the following assumptions and dependencies:

| No | Assumption / Dependency | Description |
|---|---|---|
| 1. | Internet Connection | Users must have a stable internet connection to enable AI content generation via the OpenAI API. |
| 2. | System Environment | The system depends on PHP, Laravel, MySQL, and Bootstrap being correctly installed and configured. |
| 3. | Browser Compatibility | The system assumes users access it using modern browsers such as Chrome or Edge. |
| 4. | User Cooperation | Users (lecturers and students) are assumed to provide valid data during registration and feedback phases. |
| 5. | Institutional Support | Future scaling or deployment on UPTM's server depends on administrative department approval. |

Appendix A: Questionnaire (List of Questions)

| No | Questions |
|---|---|
| 1. | What is your role in UPTM? |
| 2. | How often do you currently use general AI tools such as ChatGPT or Gemini for academic purposes? |
| 3. | What difficulties do you usually encounter in completing academic tasks without AI tools? |
| 4. | What challenges do you face when using general AI tools? |
| 5. | In your academic work, which area requires the most effort or consumes the most time? |
| 6. | Which AI features would you find most useful and likely to use in your academic work? |
| 7. | Which benefit would motivate you most to use academic AI features regularly? |
| 8. | Which type of guidance would you want most from an academic AI system? |
| 9. | How important is it for the system to provide ready-to-use templates instead of requiring complex prompts? |
| 10. | Do you think it is important that the university ensures the AI system is safe and secure before students use it? |
| 11. | Would you feel more confident using an AI system that stores all academic data securely in UPTM's own database rather than on external platforms? |
| 12. | How important is it for the AI system to protect your research, or academic information from being accessed by unauthorized people? |

| | |
|---|---|
| 13. | How important is it for the AI system to keep a secure history of your past generated documents for your own use only? |

Appendix A: Interview (List of Questions)

| No | Questions |
|---|---|
| 1. | Can you share your general experience or knowledge about Artificial Intelligence (AI)? |
| 2. | What is your understanding of AI assistance tools? |
| 3. | Which AI tools have you used before? Can you provide some examples? |
| 4. | What do you think if the system being developed includes features like the AI template shown? Would you agree with having these features? |
| 5. | If these features are included, what is your overall opinion about the system that will be developed? |
| 6. | Do you have any suggestions or opinions for improving the system that we are going to develop? |

# Appendix B – User Manual

This User Manual provides guidance on how to use the UPTM Academic AI Assistant Tools. It is designed to help users (SuperAdmin/Admin, Lecturer, and Student) understand how to navigate and utilize the system's features effectively. The manual covers essential functions, including account management, AI content generation, and document handling.

The system is a web-based platform that allows users to generate academic documents using AI technology.

It provides two main user roles:

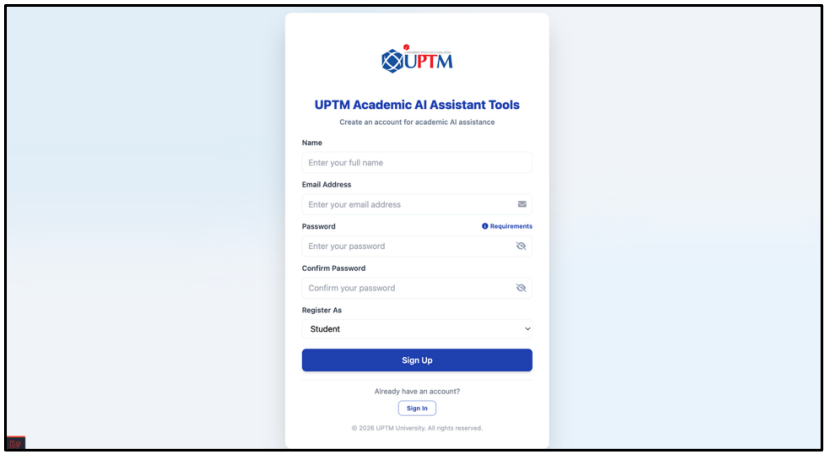
- SuperAdmin/Admin: Manages users, templates, and system content.
- Lecturer/Student: Uses available templates to generate AI-based academic content.

**User Roles and Access:**

| Role | Description | Access Level |
|---|---|---|
| SuperAdmin | Manages user accounts, templates, and generated documents. | Full access |
| Lecturer/Student | Uses templates to generate academic content and manage personal documents. | Limited access |

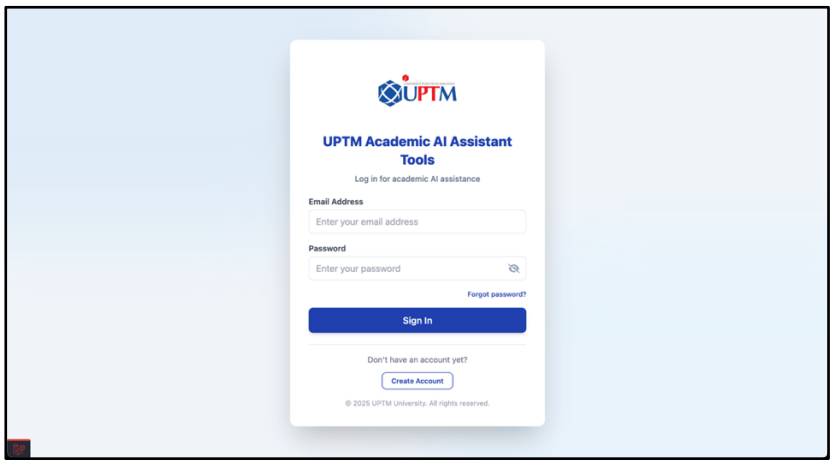
**Login and Registration:**

1) Registration



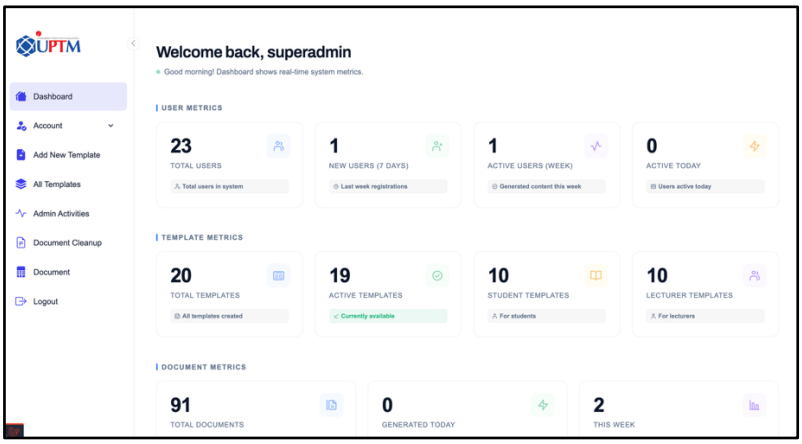

i. Fill in your registration details (Name, Email, Password, Confirm Password, Register As).
ii. Click “Sign Up” to create an account.
iii. A success message will appear confirming your registration and will bring users to the login page.

2) Login

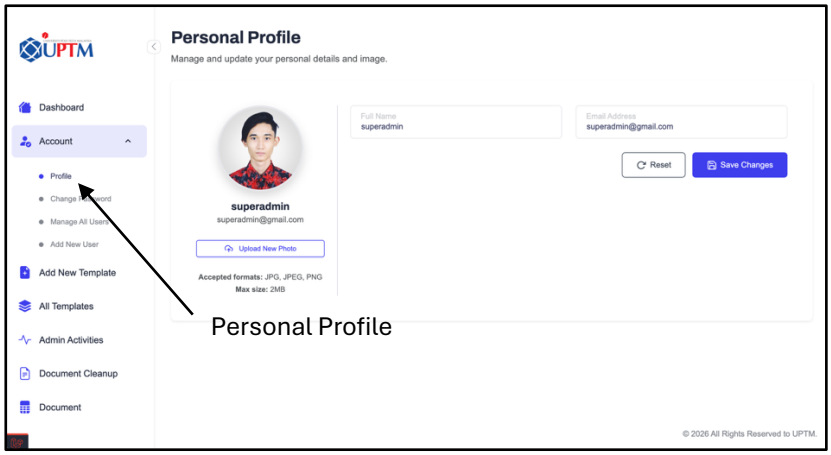


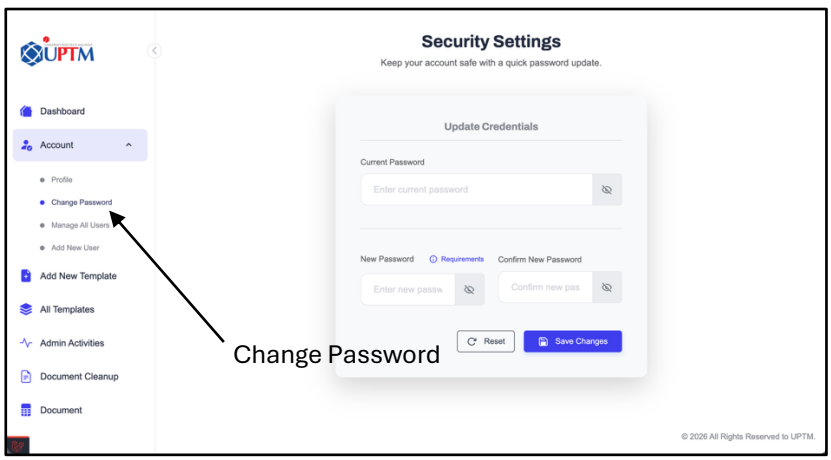

i. Enter your Email and Password.
ii. Click “Login” to access your dashboard.
iii. If the credentials are correct, you will be redirected to your respective dashboard.
- SuperAdmin: Redirected to SuperAdmin Dashboard.
- Admin: Redirected to Admin Dashboard.
- Lecturer/Student: Redirected to User Dashboard.

## SuperAdmin Dashboard Guide:



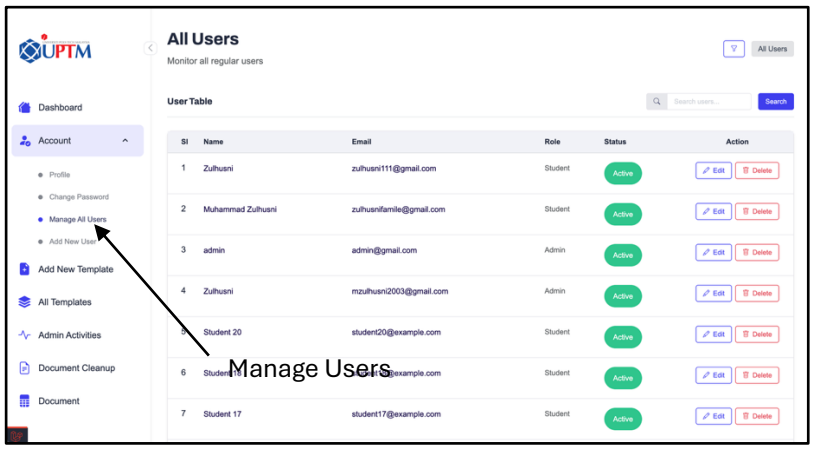

1) Update Profile



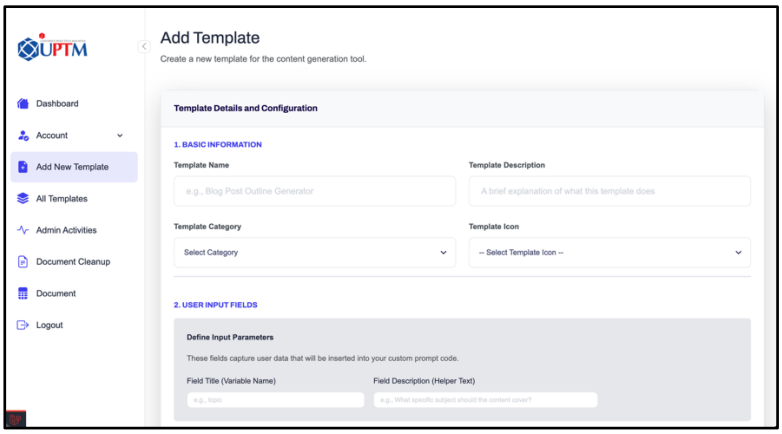

- Go to Personal Profile.
- Edit your personal details (Name, Email).
- Click Save Changes to update information.
- You can also change your password from under account section.

2) Manage User Accounts



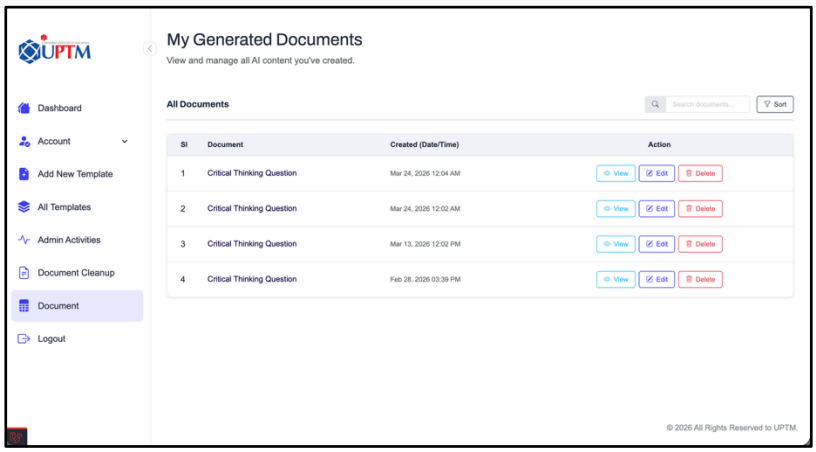

- Navigate to “Manage All Users” in the sidebar.
- View all registered users in a list.
- Click delete button to remove inactive users.

3) Template Management



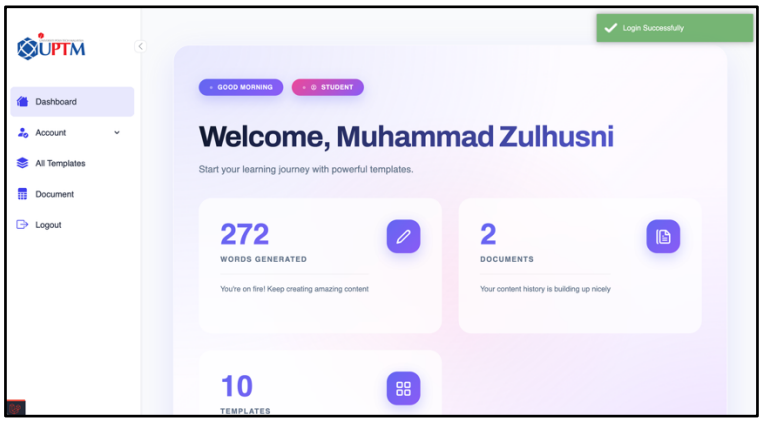

- Click Add New Template to create an AI academic template in the sidebar.
- Enter the template details and custom prompt.
- Save the template, it will appear in the user’s template list.
- You can also Edit or Delete existing templates.

4) Document Management

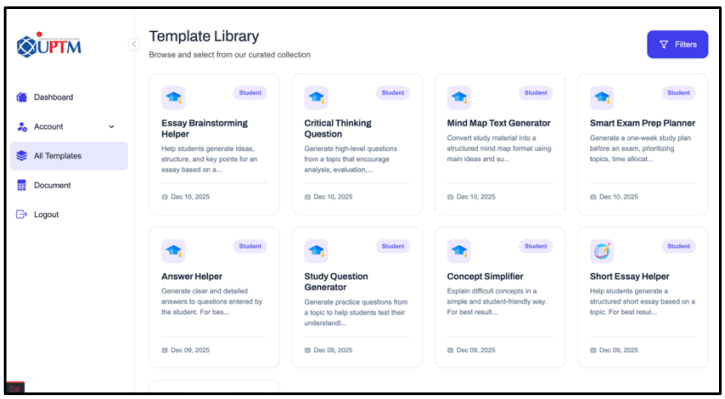


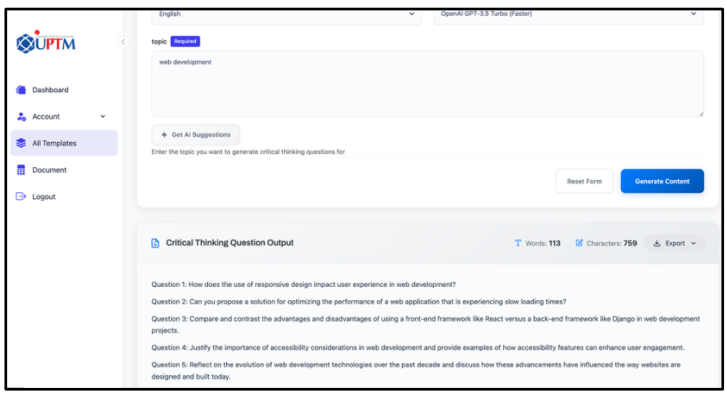

- Access Documents from the sidebar.
- View all AI-generated output (Personal).
- Optionally, delete, edit or download any document for record keeping.

**User Dashboard Guide (Lecturer/Student):**



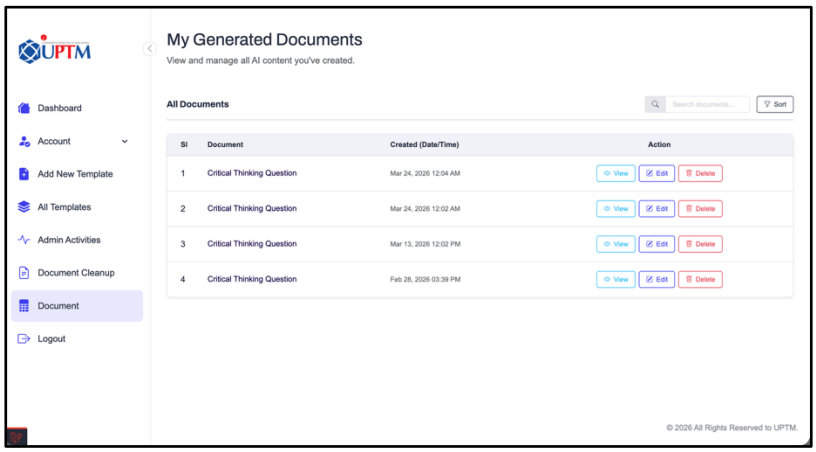

1) Generate Academic Content



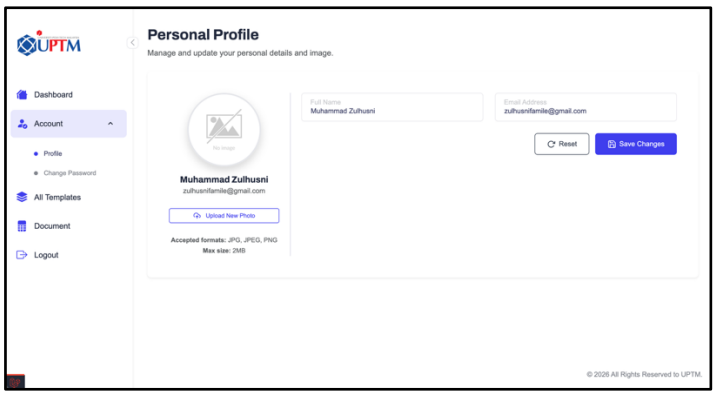

- Go to the Templates section.
- Select a desired AI template.
- Enter your topic.
- Click “Generate”, the system will use the OpenAI API to produce the content.
- The generated content will appear instantly within seconds.

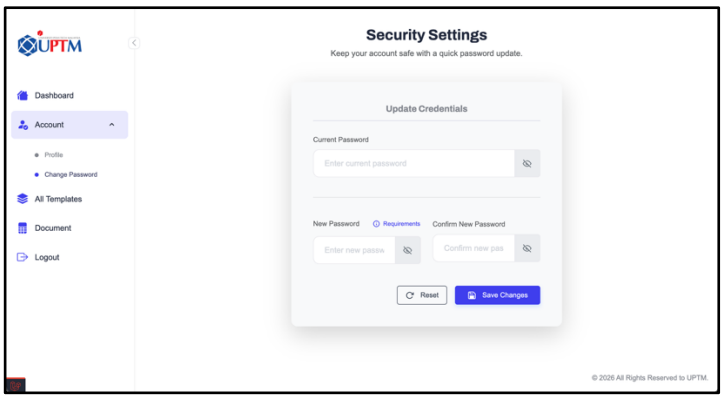
2) Manage Documents




- Navigate to “Document” to view all saved generated output.
- You can Edit, Download, or Delete documents anytime.

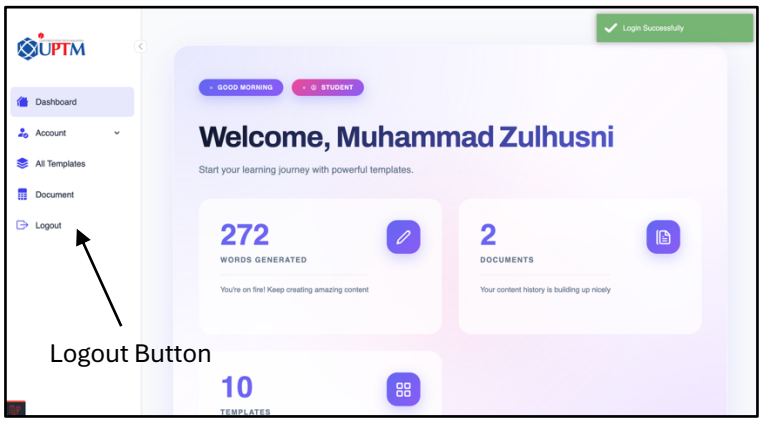
3) Update Profile




- Go to Personal Profile.
- Edit your personal details (Name, Email).
- Click Save Changes to update information.
- You can also change your password from under account section.

**System Logout:**



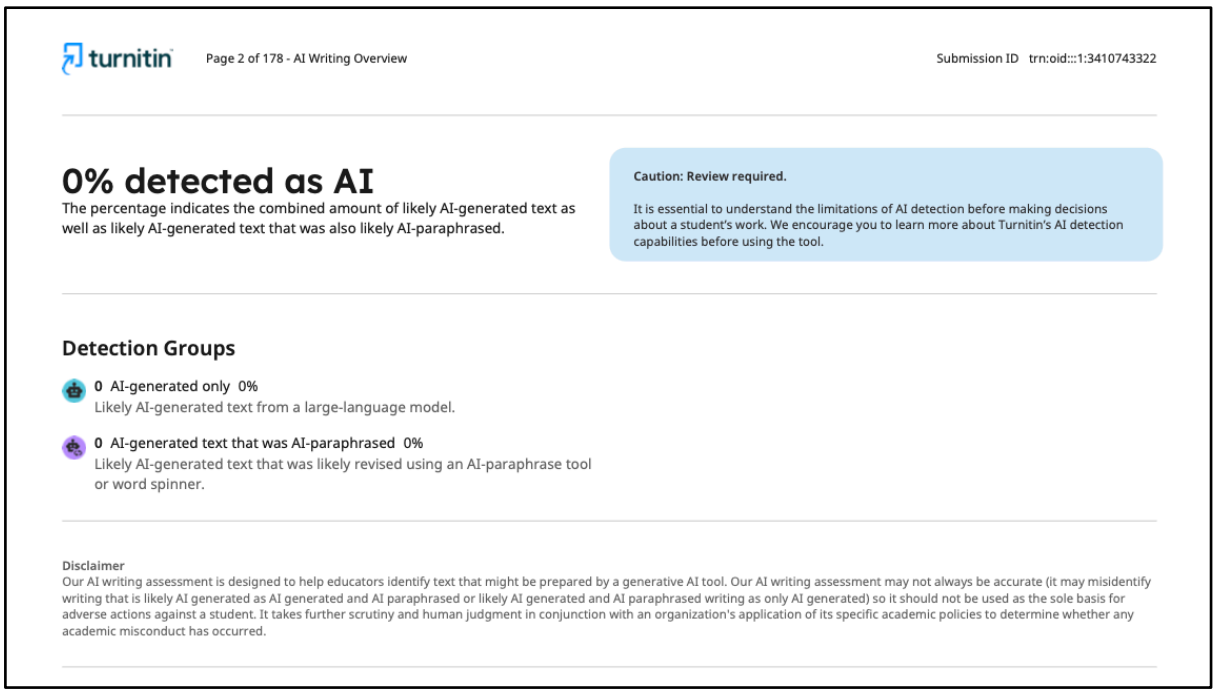

To securely end your session

- Click the Logout button in the sidebar.
- The system will log you out and redirect to the login
- Always log out after each session for security purposes.

**Troubleshooting:**

| Issue | Possible Cause | Solution |
|---|---|---|
| Unable to login | Wrong credentials | Reset password or recheck email input. |
| Unable to register | Email already exists in the system. | Try logging in with your registered email. |
| AI content not generating | API key or network issue | Ensure stable internet. |
| Long delay in content generation | Slow internet connection or high OpenAI API server load. | Wait a few seconds and try again. |
| Generated content not relevant | Input prompt too short or unclear. | Provide a more descriptive topic or keyword for better AI response. |

# Appendix C – Turnitin Result



turnitin Page 2 of 178 - AI Writing Overview Submission ID trn:oid:::1:3410743322

## 0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

**Caution: Review required.**

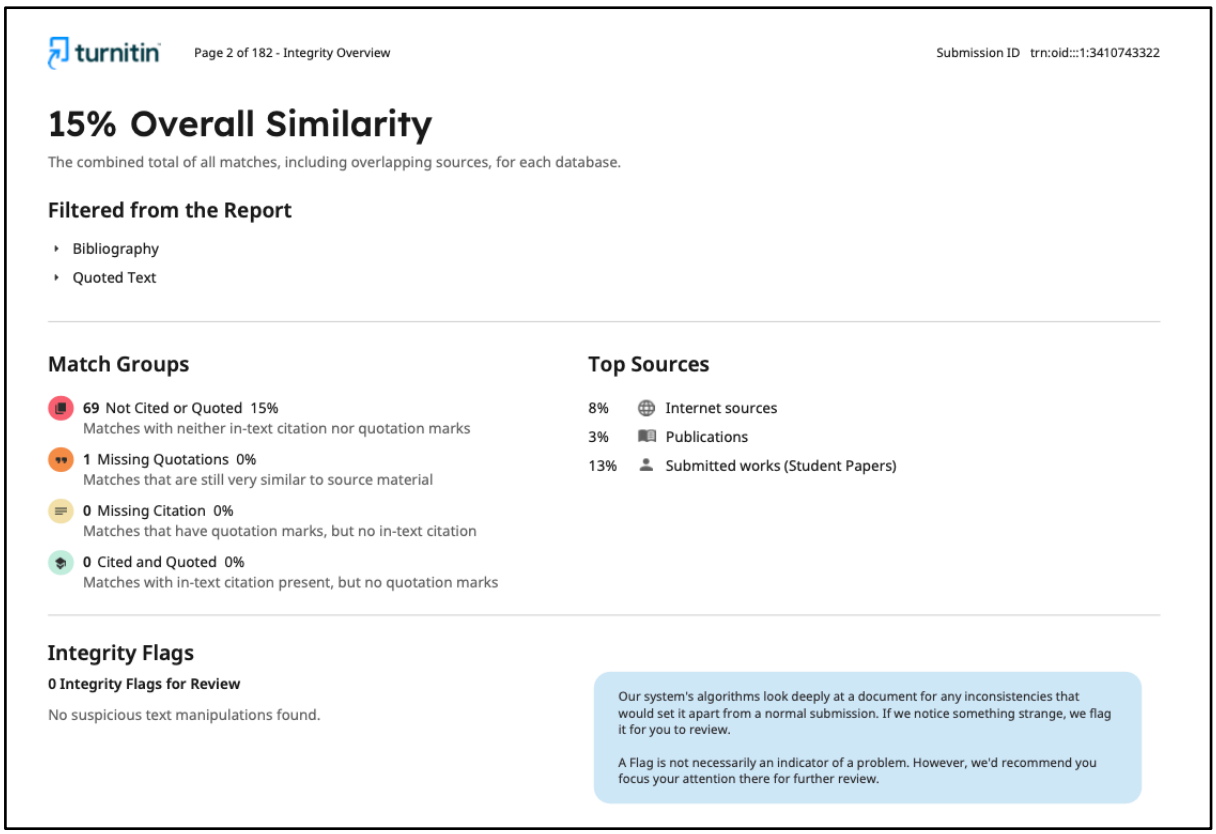
It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

### Detection Groups

**0** AI-generated only 0%
Likely AI-generated text from a large-language model.

**0** AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.



turnitin Page 2 of 182 - Integrity Overview Submission ID trn:oid:::1:3410743322

## 15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

### Filtered from the Report

- Bibliography
- Quoted Text

### Match Groups

**69** Not Cited or Quoted 15%
Matches with neither in-text citation nor quotation marks

**1** Missing Quotations 0%
Matches that are still very similar to source material

**0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation

**0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

### Top Sources

8% Internet sources
3% Publications
13% Submitted works (Student Papers)

### Integrity Flags

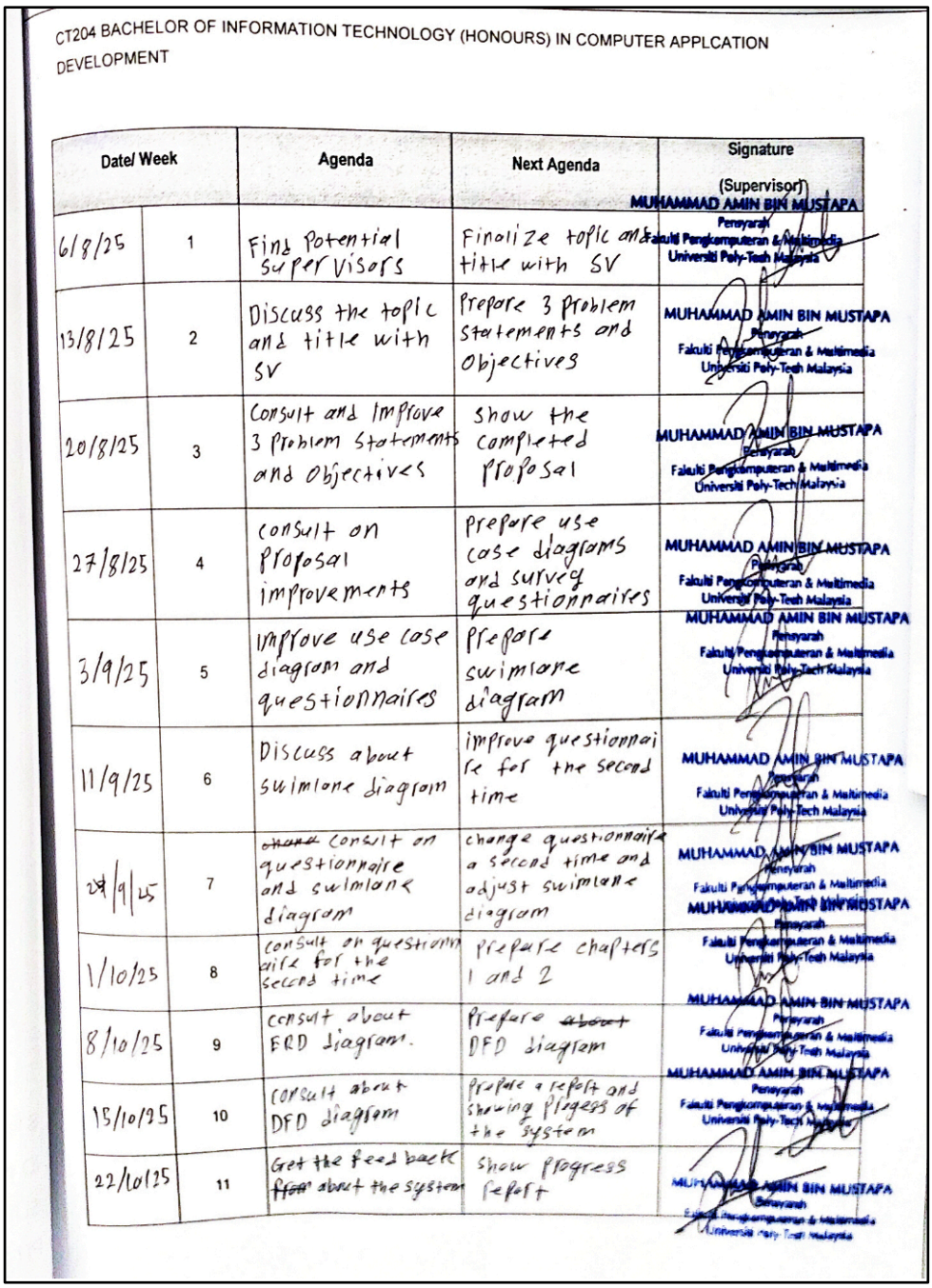
**0 Integrity Flags for Review**

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

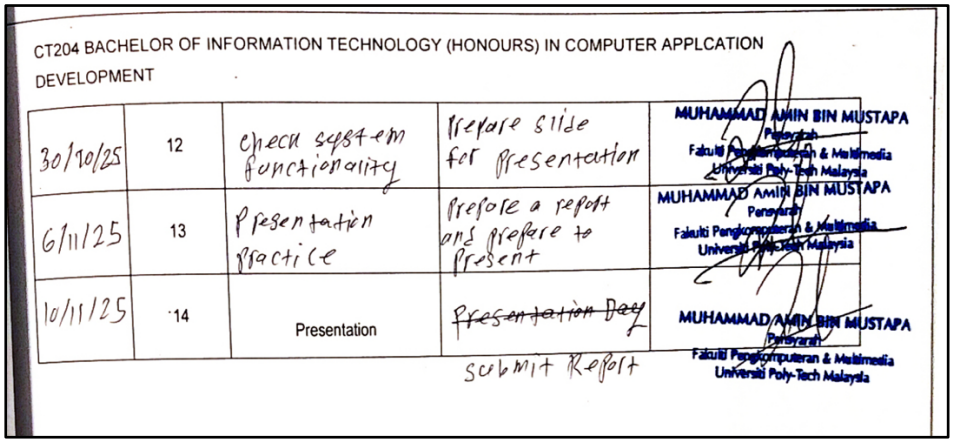
A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

# Appendix D – Log Book



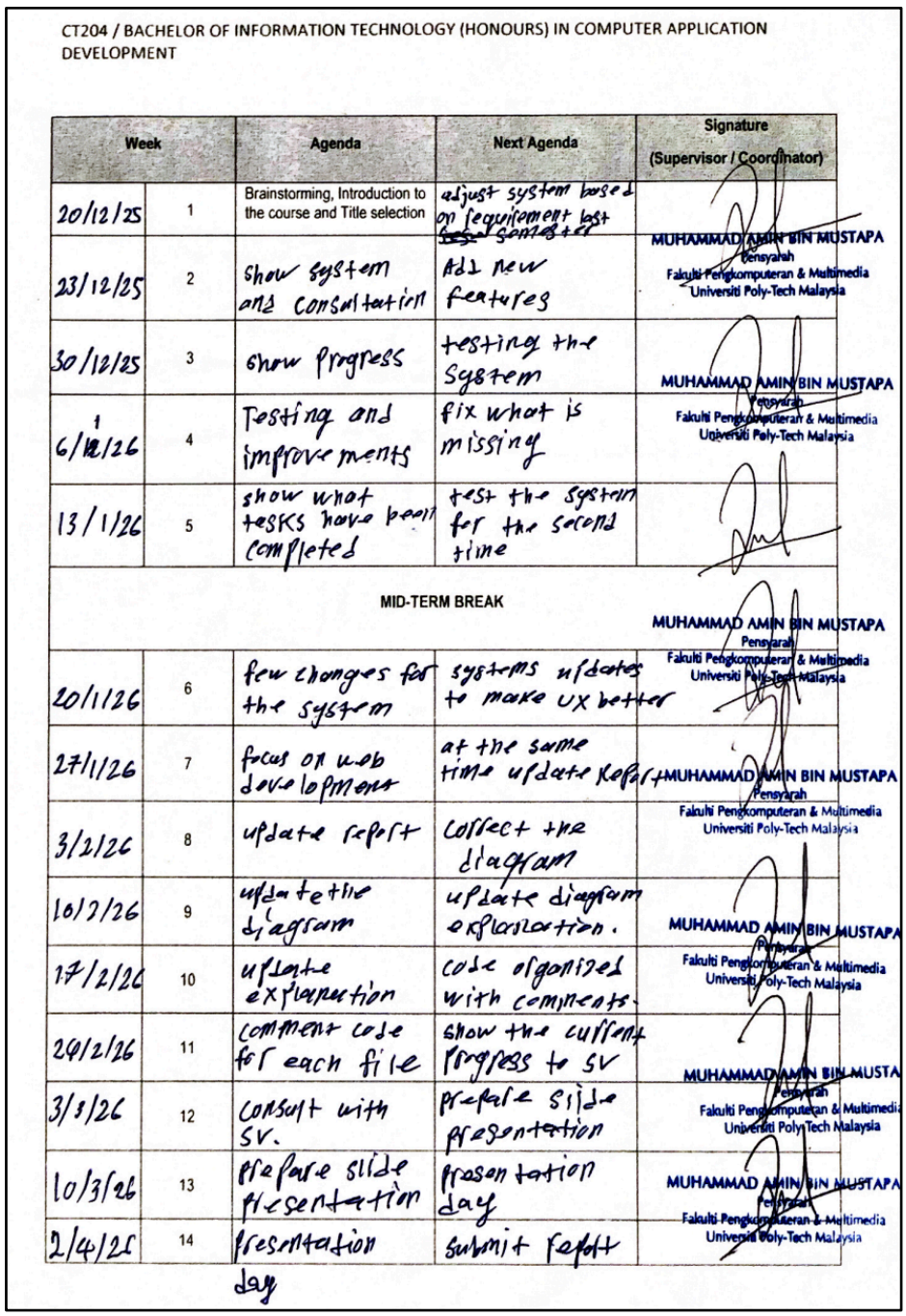
CT204 BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN COMPUTER APPLCATION DEVELOPMENT

| Date/ Week | | Agenda | Next Agenda | Signature (Supervisor) |
|---|---|---|---|---|
| 6/8/25 | 1 | Find Potential supervisors | Finalize topic and title with SV | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 13/8/25 | 2 | Discuss the topic and title with SV | Prepare 3 problem statements and objectives | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 20/8/25 | 3 | Consult and improve 3 problem statements and objectives | show the completed proposal | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 27/8/25 | 4 | consult on proposal improvements | prepare use case diagrams and survey questionnaires | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 3/9/25 | 5 | improve use case diagram and questionnaires | prepare swimlane diagram | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 11/9/25 | 6 | Discuss about swimlane diagram | improve questionnaire for the second time | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 29/9/25 | 7 | ~~about~~ consult on questionnaire and swimlane diagram | change questionnaire a second time and adjust swimlane diagram | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia |
| 1/10/25 | 8 | consult on questionnaire for the second time | prepare chapters 1 and 2 | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 8/10/25 | 9 | consult about ERD diagram. | Prepare ~~about~~ DFD diagram | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 15/10/25 | 10 | consult about DFD diagram | Prepare a report and showing Progess of the system | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 22/10/25 | 11 | Get the feed back ~~from~~ about the system | show progress report | MUHAMMAD AMIN BIN MUSTAPA Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |



CT204 BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN COMPUTER APPLCATION DEVELOPMENT

| | | | | |
|---|---|---|---|---|
| 30/10/25 | 12 | check system functionality | Prepare slide for presentation | MUHAMMAD AMIN BIN MUSTAPA Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 6/11/25 | 13 | Presentation practice | Prepare a report and prepare to present | MUHAMMAD AMIN BIN MUSTAPA Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 10/11/25 | 14 | Presentation | ~~Presentation Day~~ submit Report | MUHAMMAD AMIN BIN MUSTAPA Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |



CT204 / BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN COMPUTER APPLICATION DEVELOPMENT

| Week | | Agenda | Next Agenda | Signature (Supervisor / Coordinator) |
|---|---|---|---|---|
| 20/12/25 | 1 | Brainstorming, Introduction to the course and Title selection | adjust system based on requirement last ~~last~~ semester | |
| 23/12/25 | 2 | show system and consultation | Add new features | MUHAMMAD AMIN BIN MUSTAPA Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 30/12/25 | 3 | show progress | testing the system | |
| 6/12/26 | 4 | Testing and improvements | fix what is missing | MUHAMMAD AMIN BIN MUSTAPA Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 13/1/26 | 5 | show what tasks have been completed | test the system for the second time | |
| MID-TERM BREAK | | | | MUHAMMAD AMIN BIN MUSTAPA Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 20/1/26 | 6 | few changes for the system | systems updates to make UX better | |
| 27/1/26 | 7 | focus on web development | at the same time update report | MUHAMMAD AMIN BIN MUSTAPA Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 3/2/26 | 8 | update report | correct the diagram | |
| 10/2/26 | 9 | update the diagram | update diagram explanation. | MUHAMMAD AMIN BIN MUSTAPA Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 17/2/26 | 10 | update explanation | code organized with comments. | |
| 24/2/26 | 11 | comment code for each file | show the current progress to SV | MUHAMMAD AMIN BIN MUSTA Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 3/3/26 | 12 | consult with SV. | prepare slide presentation | |
| 10/3/26 | 13 | prepare slide presentation | presentation day | MUHAMMAD AMIN BIN MUSTAPA Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia |
| 2/4/25 | 14 | presentation day | submit report | |

# References

Buruk, O. (2023). *Academic writing with GPT-3.5: Reflections on practices, efficacy and transparency*. arXiv. https://arxiv.org/abs/2304.11079

Burwood Blog. (2025). *Gemini education privacy: Everything you need to know*. https://www.burwood.com/blog-archive/google-gemini-ai-for-education-features-privacy-and-security-explained

Chen, B., Zhang, Z., Langrené, N., & Zhu, S. (2025). *Unleashing the potential of prompt engineering for large language models*. *Frontiers in Artificial Intelligence*, 8, 150-162. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12191768/

Devineni, S. K. (2024). *AI in Data Privacy and Security*. *International Journal of Artificial Intelligence & Machine Learning, 3*(1), 35–49. https://iaeme.com/MasterAdmin/Journal_uploads/IJAIML/VOLUME_3_ISSUE_1/IJAIML_03_01_004.pdf

EDUCAUSE. (2023). *Prompt engineering for AI-enhanced teaching and learning* [Webinar]. EDUCAUSE. https://events.educause.edu/member-quicktalks/2023/prompt-engineering-for-ai-enhanced-teaching-and-learning

EDUCAUSE. (2024, October 14). *Mastering the AI Moment in Higher Education: Security and compliance worries*. EDUCAUSE Review**.** https://er.educause.edu/articles/sponsored/2024/10/mastering-the-ai-moment-in-higher-education?

Euronews. (2025, September 30). *Here's what we know about DeepSeek's latest AI offering*. https://www.euronews.com/next/2025/09/30/chinas-deepseek-unveils-new-ai-model-here-is-everything-we-know-about-it

European Data Protection Board. (2025). *Fundamentals of Secure AI Systems with Personal Data*. https://www.edpb.europa.eu/system/files/2025-06/spe-training-on-ai-and-data-protection-technical_en.pdf

FeedbackFruits. (2023). *The opportunities and challenges of AI in higher education*. Retrieved from https://feedbackfruits.com/blog/opportunities-and-challenges-of-ai-in-higher-education?

Google Classroom Features. (2025). *Google Classroom for educators*. Google for Education. https://edu.google.com/workspace-for-education/products/classroom/editions/

Google for Education. (2025). *Privacy and security for teachers & students*. https://edu.google.com/our-values/privacy-security/

Google Research. (2024). *Prompt engineering for generative AI*. Google Developers. https://developers.google.com/machine-learning/resources/prompt-eng

Google Support. (2025). *About Google Classroom*. https://support.google.com/edu/classroom/answer/6020279

Google Support. (2025). *Gemini for Google Workspace FAQ*. https://support.google.com/a/answer/14130944?hl=en

Google Workspace Privacy Notice. (2025). *Google Workspace for Education privacy and compliance*. https://workspace.google.com/intl/en/terms/education_privacy.html

Google Workspace Updates. (2025). *Additional data protection in Gemini*. https://workspaceupdates.googleblog.com/2024/08/additional-data-protection-in-gemini.html

Google Workspace Updates. (2025). *NotebookLM and the Gemini app are now Core Services with enterprise-grade data protection for education customers*. https://workspaceupdates.googleblog.com/2025/04/notebookLM-and-gemini-app-core-services-for-education-customers.html

Google Workspace with Gemini. (2025). *Google Workspace with Gemini – Education*. https://support.google.com/a/answer/13623623?hl=en

Hari, S. (2025, January 30). *What is Agile methodology?* Hackr.io. https://hackr.io/blog/what-is-agile-methodology

Hewing, M., & Leinhos, V. (2024). *The Prompt Canvas: A Literature-Based Practitioner Guide for Creating Effective Prompts in Large Language Models*. arXiv. https://arxiv.org/abs/2412.05127

López-Galisteo, A. J., et al. (2025). *The Creation and Evaluation of an AI Assistant (GPT) for educational contexts*. MDPI. https://www.mdpi.com/2078-2489/16/2/117

MIT Responsible AI for Social Empowerment and Education (RAISE). (2024). *Prompts that support learning: Human-centered design in AI tools*. MIT RAISE. https://raise.mit.edu

Nature. (2025). *How China created AI model DeepSeek and shocked the world*. *Nature*. https://www.nature.com/articles/d41586-025-00259-0

NIST CAISI. (2025, September 30). *Evaluation of DeepSeek AI models finds shortcomings and risks*. U.S. Department of Commerce. https://www.nist.gov/news-events/news/2025/09/caisi-evaluation-deepseek-ai-models-finds-shortcomings-and-risks

OpenAI. (2025). *Business data privacy, security, and compliance*. https://openai.com/business-data

OpenAI. (n.d.). *Introducing ChatGPT Enterprise*. https://openai.com/index/introducing-chatgpt-enterprise/

OpenAI. (n.d.). *Security & Privacy*. https://openai.com/security-and-privacy/

Parambil, M. M. A., et al. (2024). Integrating AI-based and conventional cybersecurity approaches. *ScienceDirect*. https://www.sciencedirect.com/science/article/pii/S2666920X24001309

Reuters. (2025, August 21). *Chinese AI startup DeepSeek releases upgraded model with domestic chip support*. Reuters. https://www.reuters.com/world/china/chinese-ai-startup-deepseek-releases-upgraded-model-with-domestic-chip-support-2025-08-21/

Sajja, R., Sermet, Y., Cikmaz, M., Cwiertny, D., & Demir, I. (2023). *Artificial Intelligence-Enabled Intelligent Assistant for Personalized and Adaptive Learning in Higher Education*. arXiv. https://arxiv.org/abs/2309.10892

SCMP. (2025, September 17). *DeepSeek secrets unveiled: engineers reveal science behind China's viral AI model*. South China Morning Post. https://www.scmp.com/news/china/science/article/3325895/deepseek-secrets-unveiled-engineers-reveal-science-behind-chinas-viral-ai-model

Scribbr. (2024a). *About us*. Scribbr. https://www.scribbr.com/about-us/

Scribbr. (2024b). *How does the Scribbr Plagiarism Checker work?* Scribbr. https://www.scribbr.com/plagiarism/how-do-plagiarism-checkers-work/

Scribbr. (2024c). *Free citation generator*. Scribbr. https://www.scribbr.com/citation/generator/

Scribbr. (2024d). *Grammar checker: Correct your writing mistakes*. Scribbr. https://www.scribbr.com/grammar-checker/

Scribbr. (2024e). *Terms of Use*. https://www.scribbr.com/legal/terms-of-use/

Sohail, S. S., Farhat, F., Himeur, Y., Nadeem, M., Madsen, D. Ø., Singh, Y., Atalla, S., & Mansoor, W. (2023). *Decoding ChatGPT: A taxonomy of existing research, current challenges, and possible future directions*. arXiv. https://arxiv.org/abs/2307.14107

Stanford Institute for Human-Centered Artificial Intelligence (HAI). (2025). *Language Models in the Classroom: Bridging the gap between technology and teaching*. Retrieved from https://hai.stanford.edu/news/language-models-in-the-classroom-bridging-the-gap-between-technology-and-teaching

Stanford Institute for Human-Centered Artificial Intelligence (HAI). (2023). *AI will transform teaching and learning. Let's get it right.* Stanford HAI. https://hai.stanford.edu/news/ai-will-transform-teaching-and-learning-lets-get-it-right

Stanford University. (2024). *Better AI prompts, better results*. Stanford University Information Technology Training. https://uit.stanford.edu/service/techtraining/class/better-ai-prompts-better-results

TechRadar. (2025). *I tried 70+ best AI tools in 2025*. TechRadar. https://www.techradar.com/best/best-ai-tools

USAHS (University of St. Augustine for Health Sciences). (2025). *AI in Higher Education: How AI Is Reshaping Higher Education* [Blog]. Retrieved April 18, 2025, from https://www.usa.edu/blog/ai-in-higher-education-how-ai-is-reshaping-higher-education/

Yan, L., Sha, L., Zhao, L., Li, Y., Martinez-Maldonado, R., Chen, G., Li, X., Jin, Y., & Gašević, D. (2023). Practical and ethical challenges of large language models in education: A systematic scoping review. *arXiv*. https://arxiv.org/abs/2303.13379

Yazdani Motlagh, N., Khajavi, M., Sharifi, A., & Ahmadi, M. (2023). *The impact of artificial intelligence on the evolution of digital education: A comparative study of OpenAI text generation tools including ChatGPT, Bing Chat, Bard, and Ernie*. arXiv. https://arxiv.org/abs/2309.02029

Zhang, C., Zhang, C., Li, C., Qiao, Y., Zheng, S., Dam, S. K., Zhang, M., Kim, J. U., Bae, S.-H., Lee, L.-H., Park, P., & Hong, C. S. (2023). *One small step for generative AI, one giant leap for AGI: A complete survey on ChatGPT in AIGC era*. arXiv. https://arxiv.org/abs/2304.06488

Zipp, G., Gunther, C. B., Kuperman, D., Moore, J., & Rainchuso, L. (2025). Educational use of artificial intelligence in Doctor of Health Sciences programs. *Journal of Innovation in Health Sciences Education, 2*(2). Retrieved from https://soar.usa.edu/jihse/vol2/iss2/4/