PROJECT TITLE: DEVELOPMENT OF AI-POWERED STUDENT TASK MANAGEMENT SYSTEM

NAMES OF CANDIDATES
IVOR NICHOLAUS MRIMI
BYAMUNGU KASHINDI PAUL

"A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DIPLOMA PROGRAM IN MBEYA UNIVERSITY OF SCIENCE AND TECHNOLOGY"

MBEYA, TANZANIA

JULY, 2025

MBEYA UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF INFORMATION SYSTEMS AND TECHNOLOGY.

PROJECT TITLE: DEVELOPMENT OF AI-POWERED STUDENT TASK MANAGEMENT SYSTEM

MODULE CODE: IT 6317.

ACADEMIC YEAR: 2024/2025.

SUPERVISOR: MR. PETER MADEMBWE.

COORDINATOR: MR. EDWIN NCHIA.

NO.	NAMES OF CANDIDATES	REGISTRATION NO.
01	IVOR NICHOLAUS MRIMI	22101123130021
02	BYAMUNGU KASHINDI PAUL	22101123130094

"A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DIPLOMA IN INFORMATION AND COMMUNICATION TECHNOLOGY OF MBEYA UNIVERSITY OF SCIENCE AND TECHNOLOGY."

JULY ,2025

ABSTRACT

Effective task management is essential for student success, yet many learners struggle with organizing academic responsibilities, leading to missed deadlines and reduced productivity. Traditional methods of task tracking, such as manual planners and basic digital tools, often lack intelligence and adaptability to meet individual needs.

This project aims to solve these challenges by developing an AI-Powered Student Task Management System a smart, user-friendly platform designed to help students manage their academic workload efficiently. The system integrates Artificial Intelligence to assist in task suggestions, prioritization, and timely reminders based on user behavior and academic patterns. Built using modern technologies, it offers features such as intelligent task creation, dashboard analytics, and chatbot support to enhance productivity and academic planning.

By automating routine task management processes and offering personalized support, this system empowers students to stay organized, reduce stress, and achieve better academic outcomes, particularly in educational environments with limited support resources.

ACKNOWLEDGMENT

We raise our hands hardly, open our heart widely, and say thanks to the Creator who gave us power and energy to carry out this project work without becoming ill or injured. Many thanks and gratitude to God who gave each of us everything that has made us be where we are today. Thanks to all lectures of our collage CoICT and the department IST for their untied support and precious time they gave us till the completing of this project. We can't count the problems we brought in their families for the late coming due to projects and all the night calls we called them for misunderstood concepts.

Also, we would like to give special thanks to Sir. Peter Madembwe and Sir. Edwin Nchia acting as our coordinator and supervisor respectively in this project by using their time, knowledge and energy for well and clear work, let our almighty God bless them.

Lastly, we would like to dedicate our thanks to all friends, fellow students and families for their generous hearts and understanding minds for whole project time they gave us till completion.

CERTIFICATION

The undersigned certify that they have read, and hereby recommend for acceptance by Mbeya University of Science and Technology the project titled "AI-POWERED STUDENT TASK MANAGEMENT SYSTEM". This project is submitted in partial fulfillment of the requirements for the award of the Ordinary Diploma in Information and Communication Technology at Mbeya University of Science and Technology (MUST), under the Department of Information Systems and Technology in the College of Information and Communication Technology, as submitted by IVORY N. MRIMI and BYAMUNGU P. KASHINDI.

…….

Supervisor's Name

……

Signature

……..

Date

DECLARATION

We declare to the best of our knowledge that, this project of "AI-POWERED STUDENT TASK MANAGEMENT SYSTEM" is presented as a partial fulfillment to the award of ordinary diploma in Information and Communication Technology is our original work and it has neither been copied from anywhere nor presented elsewhere.

……..	……
IVORY N MRIMI	BYAMUNGU K PAUL

……..……

Date

DEDICATION

We wholeheartedly dedicate this project to our beloved nation, Tanzania, and to our esteemed institution, Mbeya University of Science and Technology (MUST). This work is a humble contribution towards the advancement of our country and the well-being of its citizens.

We also extend our sincere dedication to the Head of Department and all facilitators who have played a significant role in guiding and supporting us throughout this journey. Your unwavering commitment, valuable time, and tireless efforts have been instrumental in the successful completion of this project.

Lastly, we express our deepest gratitude to our loving families, whose constant encouragement and inspiration have been our source of strength.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligent
API	Application Programming Interface
CoICT	College of Information and Communication Technology
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol secure
ICT	Information Communication Technology
IT	Information Technology
ML	Machine Learning
MUST	Mbeya University of Science and Technology
UI	User Interface

CHAPTER ONE
INTRODUCTION

1.1 Background Information

In today's academic environment, students often face increasing pressure to manage multiple academic responsibilities, including assignments, exams, personal study schedules, and co-curricular activities. Poor task management and lack of organizational skills are among the leading causes of academic underperformance and stress among students.

Traditional approaches to task management such as notebooks, wall calendars, and basic reminder apps lack adaptability and real-time intelligence to support students effectively. With the growing availability of digital solutions, integrating Artificial Intelligence (AI) into education offers a promising avenue to help students manage their academic tasks more efficiently.

Recent advancements in AI have led to the development of intelligent systems capable of learning user behavior, sending smart reminders, suggesting optimal study times, and even predicting future workload patterns. These tools can be especially beneficial in higher education institutions where students need personalized academic support.

A preliminary study at Mbeya University of Science and Technology (MUST) showed that over **60%** of students reported missing at least one deadline due to poor planning or lack of reminder mechanisms. Additionally, more than **75%** of the respondents expressed interest in using a smart academic assistant that could help them organize their schedules and keep track of tasks.

This project, therefore, aims to address these challenges by developing an AI-Powered Student Task Management System, providing students with a smart, user-friendly platform that offers intelligent task suggestions, personalized reminders, and an interactive chatbot assistant to support their academic journey.

1.2 Problem Statement

Academic success heavily depends on effective time and task management. However, many students particularly those in higher education struggle to keep up with their academic workload due to poor planning, lack of reminders, and disorganized schedules. This often leads to missed deadlines, increased stress levels, and reduced academic performance.

Traditional tools such as physical notebooks, calendars, and basic to-do list applications lack the intelligence to adapt to a student's specific needs and behaviors. Moreover, in many institutions, limited access to personalized academic support services leaves students without proper guidance or tools to manage their responsibilities effectively.

Research conducted at Mbeya University of Science and Technology (MUST) revealed that over 60% of students reported missing assignments or deadlines due to disorganized schedules, while 75% expressed interest in a smart system that could help them stay on track with tasks, reminders, and study planning.

While AI-based productivity apps exist, most are either too expensive, not student-focused, or overly complex for daily academic use. Furthermore, they often lack localized features suited to students in developing regions, such as offline access or simplified interfaces.

This project aims to address these issues by developing an AI-Powered Student Task Management System a smart, user-friendly platform that helps students organize their tasks, receive timely reminders, and access personalized suggestions through AI-driven analysis. By offering accessible and intelligent academic assistance, the system intends to reduce task overload, increase productivity, and improve academic performance, especially for students in resource-limited environments.

1.3 Project Objectives

Objectives refer to a clear set of goals based on facts and practical considerations. The project we are undertaking is driven by a realistic vision to be implemented effectively. This project consists of two parts of objectives.

1.3.1 Main Objective

To Develop an AI-Powered Student Task Management System.

1.3.2 Specific Objectives

i. To design and develop a user-friendly web-based interface.
ii. To implement and train AI models for intelligent task suggestions and prioritization.
iii. To integrate the trained AI model with the user interface for real-time functionality.
iv. To test and evaluate the system for accuracy, performance, and usability.

1.4 Significance of the Project

i. Improved Academic Productivity; Enables students to manage their academic workload effectively.
ii. Timely Task Reminders; Reduces missed deadlines through smart notifications and alerts.

iii. Personalized Support; AI suggestions tailored to individual student habits and priorities.
iv. Accessibility and Convenience; Offers a cost-free, web-based tool available to all students, even in low-resource settings.
v. Stress Reduction – Helps reduce academic anxiety by keeping students organized.
vi. Data Insights for Future Research; Contributes useful data for improving educational technology solutions.

Without such a system, many students may continue to face academic challenges due to disorganization, missed deadlines, and lack of personalized academic support. These issues may lead to poor academic performance, increased stress, and lack of motivation, especially in institutions with limited academic advising resources.

1.5 Scope of the Project

This project focuses on developing a web-based AI-powered system designed to assist students in managing academic tasks efficiently. The system will include features such as intelligent task creation, prioritization, smart reminders, and a user-friendly dashboard interface. It will utilize machine learning algorithms to analyze student behavior and provide personalized suggestions for better academic planning and time management.

The platform will support task tracking, deadline notifications, and productivity insights, helping students stay organized throughout the academic term. Additionally, the system will include a chatbot to provide assistance and respond to queries related to task management.

The system is intended to be used primarily by students at the diploma or undergraduate level, and will operate as an online platform accessible through modern web browsers.

1.5.1 Limitations

i. The system will function exclusively online, requiring a stable internet connection for full functionality.
ii. It will focus solely on academic task management, and will not cover non-academic areas such as financial planning or social life.
iii. The system does not replace academic advisors or counselors but provides support through automation and AI.

iv. It will rely on the accuracy of user input (e.g., deadlines, task titles) for optimal performance.
v. The AI suggestions will be based on pattern recognition and previous data; the system will not guarantee academic success but will enhance task organization.

Despite these limitations, the system aims to offer a fast, intelligent, and user-friendly solution to help students improve time management and academic performance.

1.5.2 Deliverables of the Project

The successful completion of this project will result in the following key deliverables.

i. **Web-Based Task Management System**
   A fully functional web application that allows students to create, view, prioritize, and manage academic tasks.
ii. **AI Task Suggestion Engine**
   An integrated machine learning model that analyzes user behavior and suggests tasks or study plans intelligently based on academic patterns.
iii. **Smart Reminder System**
   A notification module that sends timely reminders to users regarding upcoming deadlines, overdue tasks, and high-priority activities.
iv. **User Dashboard Interface**
   An interactive and responsive dashboard that displays task status, performance analytics, and productivity insights in a visually intuitive format.
v. **AI Chatbot Assistant**
   A conversational assistant embedded in the platform to help users interact with the system, receive task-related guidance, and ask academic planning questions.
vi. **Documentation and User Manual**
   A comprehensive report containing system design, development process, AI model training methodology, and instructions for installation and use.
vii. **Testing and Evaluation Report**
   A report summarizing the system's performance in terms of accuracy, speed, usability, and student feedback collected during testing.

1.5.3 Project Case Study

The AI-Powered Student Task Management System is studied and will be implemented in educational institutions, particularly at universities and colleges where students often face challenges in organizing their academic workload. The case study for this project focuses on Mbeya University of Science and Technology (MUST), where many students experience difficulties in task planning, time management, and meeting deadlines.

The system enables students to create and manage their academic tasks, receive AI-generated suggestions for prioritization, and get timely reminders to stay on track. It can be integrated into the university's learning support system or used individually by students on their personal devices.

In daily academic life, students can use the platform for self-organization, reducing stress and improving productivity. Academic advisors and facilitators may also use the system's analytics to monitor students' task engagement and provide targeted academic support. This solution is especially useful in institutions where personalized academic guidance is limited, offering a scalable and intelligent alternative for student success support.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

The literature review acknowledges the work of previous researchers and assures the reader that this project is built upon a solid foundation of existing knowledge. It is assumed that by referencing previous studies in the field of academic task management and AI-based student support systems, the authors have critically evaluated and integrated relevant findings to strengthen this work.

Current task management systems often lack intelligent guidance tailored to students' specific academic needs. Many existing tools are generic and do not provide personalized recommendations or adaptive support, which can leave students struggling to efficiently organize and prioritize their academic responsibilities.

2.2 Review of Related or Similar Works

This chapter presents a conceptual review and thematic analysis of literature related to AI-powered educational tools, task management systems, and digital assistants designed to enhance student productivity and academic success.

2.2.1 MyStudyBuddy AI

MyStudyBuddy AI is an intelligent task management platform designed specifically for students. It incorporates machine learning algorithms to analyze students' study habits and provide personalized task prioritization and scheduling recommendations. The system includes an interactive dashboard and real-time notifications to keep users focused and on track.

Since its launch in 2022, MyStudyBuddy has gained traction in several universities, helping reduce missed deadlines and improve time management skills. However, its effectiveness is sometimes limited by the quality of user input data and the variability of academic calendars across institutions. Integration with existing educational management systems can also be complex, requiring technical adjustments.

Despite these challenges, MyStudyBuddy represents a significant advancement in AI-assisted academic support, demonstrating how personalized digital tools can improve student outcomes.

2.2.2 Focus Planner

Focus Planner is a productivity app aimed at students and young professionals that combines traditional to-do list features with AI-driven focus techniques. It offers intelligent reminders, breaks scheduling, and task segmentation based on cognitive load theories.

While Focus Planner enhances motivation and task completion rates, its general-purpose design means it lacks some academic-specific functionalities such as course-based task categorization or integration with learning management systems. Additionally, users have reported limited customization options, which can hinder long-term engagement.
Nevertheless, Focus Planner provides valuable insights into how behavioral science and AI can be merged to support better time management.

2.2.3 Edu Assist AI

Edu Assist AI is an emerging AI-powered virtual assistant tailored for academic environments. It supports students by answering queries related to course deadlines, exam schedules, and study resources. Edu Assist uses natural language processing (NLP) to interact conversationally and provide personalized academic advice.
Although still in development since 2024, initial trials at selected colleges indicate that Edu Assist improves student engagement and reduces anxiety related to academic planning. Limitations include dependency on accurate institutional data feeds and challenges in understanding complex student requests.

Edu Assist exemplifies the potential of AI chatbots in enhancing academic support services, offering scalable solutions to supplement human advisors.

2.2.4 TaskMaster Pro

TaskMaster Pro is a widely used task management system with AI capabilities that analyze users' productivity patterns to suggest optimal task sequences. Launched in 2020, it supports multiple platforms and integrates with calendars and email systems.
Its broad application makes it suitable for many professional and academic users, but some students find the interface overwhelming due to its complexity and feature richness. TaskMaster Pro also relies heavily on constant internet access, which may limit usability in regions with poor connectivity.

Despite these drawbacks, TaskMaster Pro demonstrates how AI-enhanced task systems can increase efficiency and organization.

2.2.5 StudySmart AI

StudySmart AI is a specialized academic planner that uses machine learning to predict students' workload peaks and suggests balanced study schedules. Developed through collaboration with

educational psychologists, StudySmart has shown promising results in pilot studies by improving students' time allocation and reducing burnout.

Its key feature is dynamic adjustment based on real-time feedback and academic calendar changes. However, the system requires consistent user interaction to maintain accuracy, which can be a barrier for less motivated students.

StudySmart AI highlights the importance of adaptive and personalized AI in promoting sustainable study habits.

2.3 Gaps in Existing Systems

Limited Personalization and Adaptability

Many current students task management systems provide generic task tracking features but lack deep personalization. They often fail to adapt intelligently to individual student habits, learning styles, and academic calendars, which reduces their effectiveness in improving productivity and organization.

Narrow Focus on Academic Tasks

Some systems focus only on basic to-do lists or reminders without offering advanced functionalities such as AI-driven task prioritization, predictive scheduling, or personalized study recommendations tailored to students' unique workload and deadlines.

Accessibility Issues in Low-Resource Settings

Several existing platforms require constant internet connectivity or are not optimized for low-bandwidth environments. This makes them less accessible for students in rural or low-income areas who might lack reliable internet access or modern devices.

Complexity and Usability Challenges

Many comprehensive task management applications have complex interfaces or require extensive setup and customization, which can discourage less tech-savvy students from fully utilizing their features. Simpler apps, on the other hand, often lack the advanced tools students need.

Lack of Integration with Educational Systems

Most tools do not integrate seamlessly with institutional learning management systems (LMS), calendars, or academic support services, limiting their utility as a centralized platform for academic task management.

Limited Support and Guidance

Few existing systems incorporate conversational AI or chatbot assistants that can provide real-time support, answer student queries, and offer motivational guidance features that could significantly improve user engagement and effectiveness.

CHAPTER THREE
METHODOLOGY

3.1 Introduction

The AI-Powered Student Task Management System is a web-based application designed to help students effectively manage their academic tasks. The system allows users to create, organize, and track tasks while receiving intelligent suggestions and reminders powered by artificial intelligence and machine learning. These technologies analyze student behavior, deadlines, and priorities to assist in scheduling and improving productivity. Agile methodology will be used to guide the development process, ensuring iterative progress, flexibility in incorporating feedback, and continuous system improvement.

3.2 Agile Methodology Overview

Agile is an iterative and incremental approach to software development that promotes adaptability, collaboration, and responsiveness to change. The development of this project will follow Agile principles using **Scrum** as the primary framework. The Scrum process is broken into **Sprints**, where each sprint involves planning, development, testing, and review.

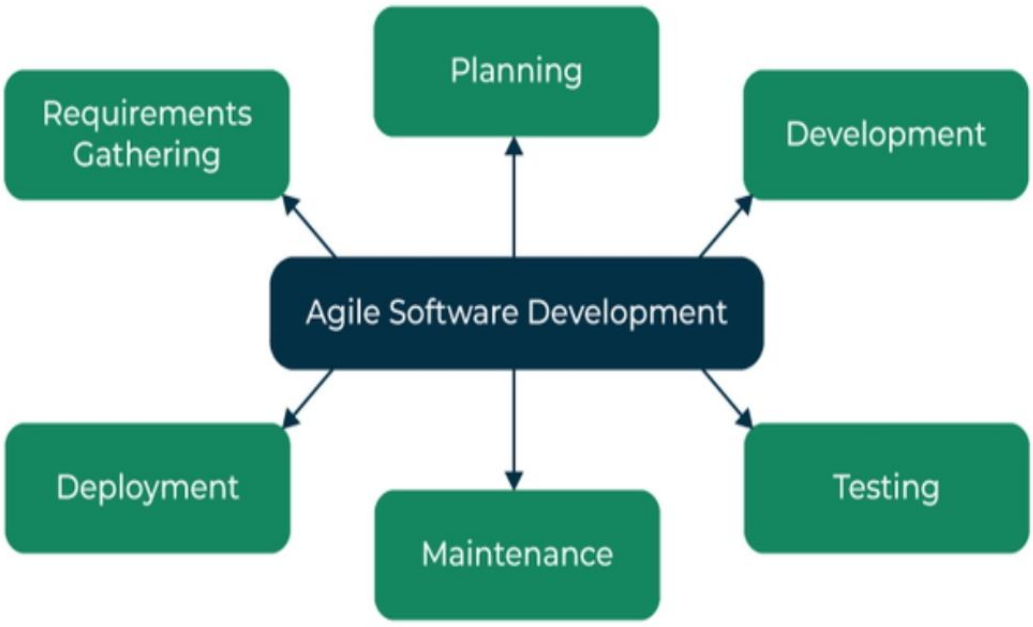


Figure 1;Agile Methodology System

3.3 Agile Implementation Approach

The implementation will be divided into five phases, each with clearly defined sprints and deliverables to ensure systematic progress toward the project's goals.

Project Phases and Sprints:

Phase 1: Project Initiation & Requirement Gathering

Sprint 1: Requirement Analysis & Planning

i. Conduct meetings with stakeholders (students, academic advisors, developers, UI/UX designers).
ii. Identify functional and non-functional requirements of the system.
iii. Define the product backlog (tasks, features, and components).
iv. Set up the development environment (tools, frameworks, hosting services).

Deliverables

i. Product backlog
ii. Chosen tech stack (e.g., React, Node.js, Supabase, OpenAI)

Phase 2: System Design

Sprint 2: System Architecture & UI/UX Design

i. Design the system architecture (frontend, backend, AI engine).
ii. Develop wireframes and interactive prototypes of the platform.
iii. Define API requirements for AI-based task suggestions and reminders.

Deliverables

i. System design documentation
ii. UI/UX wireframes and clickable prototype

Phase 3: Development & AI Integration

Sprint 3: Frontend Development

i. Build the user interface using React and Tailwind CSS.
ii. Implement pages for task creation, dashboards, and task analytics.

Sprint 4: Backend Development

i. Develop backend with Node.js
ii. Create APIs for user tasks, reminders, authentication, and AI services.
iii. Integrate with Supabase or Firebase for data storage and user management.

Sprint 5: AI Model Development

i. Collect and preprocess task usage data (manually or from survey-based samples).
ii. Train a lightweight machine learning model to provide smart task suggestions and priority scores.
iii. Deploy the model as an API (using Python, FastAPI, or Node.js backend).

Deliverables

i. Working frontend and backend system
ii. AI model integrated for intelligent task suggestions

Phase 4: Testing & Deployment

Sprint 6: Testing

i. Conduct unit tests, integration tests, and user acceptance testing (UAT).
ii. Test AI output for relevance and usability.
iii. Collect feedback from pilot users (students and academic mentors).

Sprint 7: Deployment & Optimization

i. Deploy the system on a cloud platform (e.g., Vercel, Netlify, or Firebase Hosting).
ii. Monitor system performance and fine-tune backend APIs and model behavior.

Deliverables

i. Fully functional, tested, and deployed web application

Phase 5: Maintenance & Continuous Improvement

Sprint 8: Post-Deployment Enhancements

1. Monitor user feedback and usage analytics.
2. Retrain and improve the AI model for better task prediction.

Add features like,

i. Chatbot assistant for academic help
ii. Calendar integration
iii. Offline mode or push notifications

Advantages of Using Agile Methodology Model.

Faster Development & Continuous Delivery

Agile enables rapid development through short, iterative Sprints (typically 2–4 weeks). Each Sprint delivers a working feature that can be tested and improved early. For example, the task creation and reminder features can be built and validated before integrating the AI suggestion engine. This approach ensures each component works efficiently before the system is fully integrated. Early testing enables faster debugging, reduces rework, and minimizes costly delays in the final stages of development.

Flexibility to Change Requirements

One of Agile's strongest advantages is its adaptability. Requirements can change at any stage of development based on feedback from students, educators, or institutional stakeholders.
For instance, if users request a feature such as a study focus timer or task color-coding, these changes can be added in future Sprints without affecting overall progress. Similarly, if a more efficient AI model becomes available or new academic patterns emerge, the system can evolve accordingly keeping it up-to-date and effective.

Higher Product Quality & Accuracy

Agile encourages continuous testing and integration, reducing the risk of major defects at the end. Developers, testers, and AI engineers work collaboratively to ensure consistent performance and accuracy. In this project, the AI task suggestion engine can be tested and refined during every Sprint to improve its relevance and precision. If the model mis prioritize tasks or produces irrelevant recommendations, immediate corrections can be made. This leads to a more reliable system that truly meets students' academic planning needs.

User & Stakeholder Collaboration

Agile emphasizes regular communication with users and stakeholders, making it easier to align the system with real-world needs. Throughout the development process, feedback from students and academic mentors can guide adjustments to features, usability, and performance. For instance, students testing the platform during development can suggest interface improvements, while academic staff can validate whether the AI suggestions align with academic goals. This iterative feedback loop ensures both usability and practicality.

Reduced Project Risks

Agile breaks the project into manageable units, lowering the chance of major system failure. Problems are identified early in the development cycle rather than at the end. For example, if the AI model for task prioritization underperforms, the team can adjust it in the next Sprint possibly switching models or refining training data without delaying the whole project. This proactive approach ensures project success and timely delivery, even when challenges arise.

3.4 Data Collection

Data collection is a process of gathering, recording and storing information or data from various sources for the purpose of analysis, research, decision-making or monitoring. Involves systematically collecting relevant data points or measurement, draw conclusions or support specific objectives.

Data collection for the AI-Powered Student Task Management System focused on gathering both primary and secondary data. This information was used to define system requirements, design user-centered features, train the AI model for intelligent task suggestions, and test system performance and usability.

3.4.1 Types of Data Collected

i. Primary Data

Collected directly from students and academic staff through observations, interviews, and structured questionnaires conducted at Mbeya University of Science and Technology (MUST).

ii. Secondary Data

Obtained from academic journals, online platforms such as Google Scholar, Research Gate, and preexisting academic productivity datasets available on platforms like Kaggle and GitHub. These helped in training and validating the task suggestion model.

Primary Data Collection Methods

Interview

This method involved direct conversations with students and academic staff (mentors and lecturers) to better understand academic planning behaviors, task challenges, and time management needs. Interviews were conducted with participants at MUST, particularly in the Department of Information and Communication Technology.

Why Use Interview Method for Data Collection?

i. **Direct Feedback from End Users:** Gain insights into the real academic challenges students face.
ii. **Build Trust and Engagement:** Establish rapport with future users to encourage adoption of the system.
iii. **Requirement Gathering:** Understand functional and non-functional system expectations from the users' perspective.

Sample Interview Questions

i. How do you usually organize your academic tasks and deadlines?
ii. Do you currently use any tools (e.g., calendar apps, notebooks) for task management?
iii. What are the biggest challenges you face when trying to manage your academic workload?
iv. Would you find it helpful if an AI system suggested which tasks to prioritize?
v. Do you sometimes forget assignments or submit them late?
vi. Would reminders and visual task analytics (like progress bars) help you stay focused?
vii. What features would you like to see in a smart academic planner?

Observation

This method allowed us to observe students' academic task management habits within the university environment. Through physical observation in study areas, computer labs, and classrooms, we noted that many students still rely on manual methods such as notebooks or unstructured digital tools (e.g., phone memos) to track assignments and deadlines. This often leads to forgotten tasks, last-minute submissions, and poor workload prioritization. The lack of intelligent digital planners was evident, especially among students with a high number of academic responsibilities.

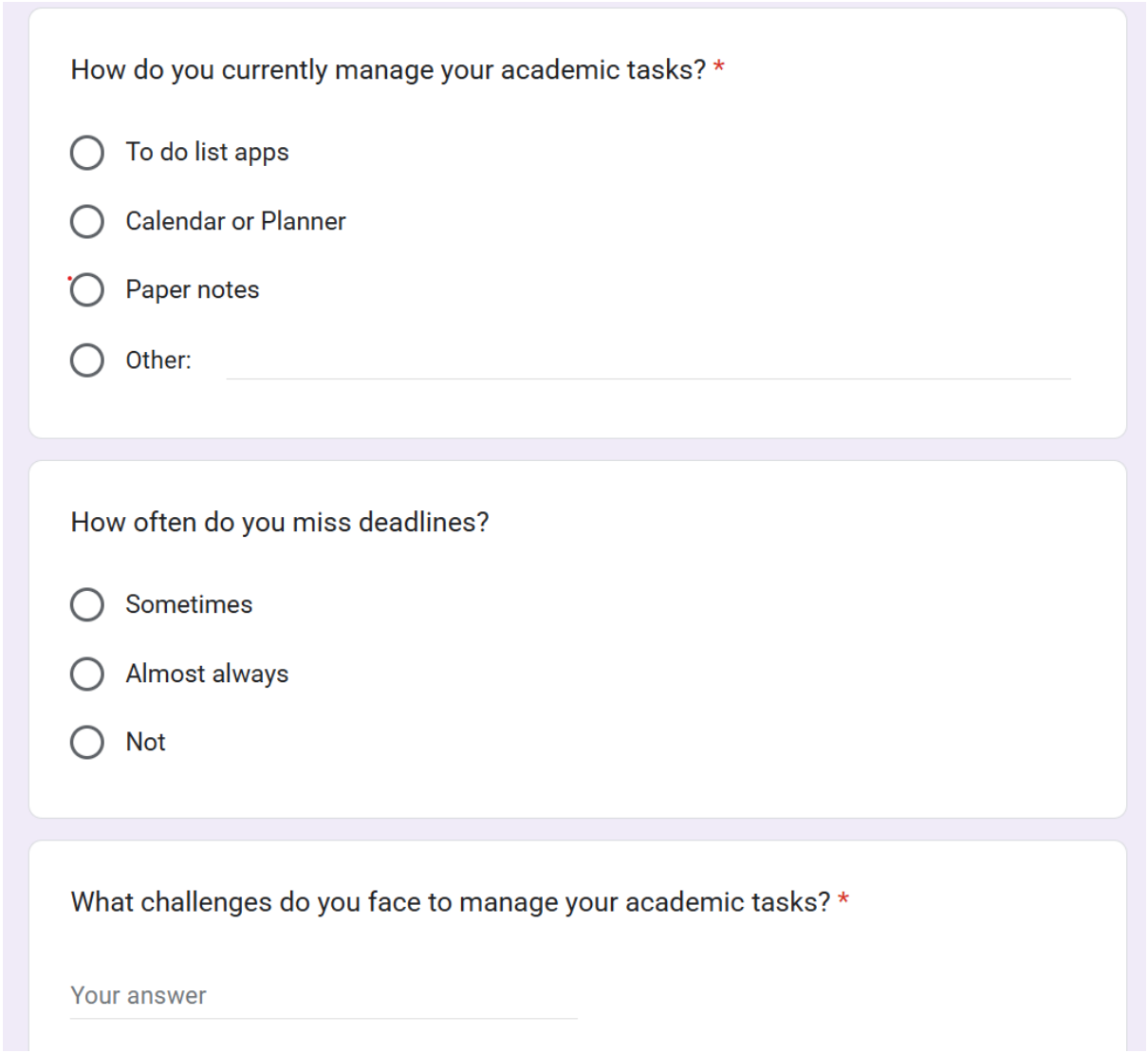
Questionnaires

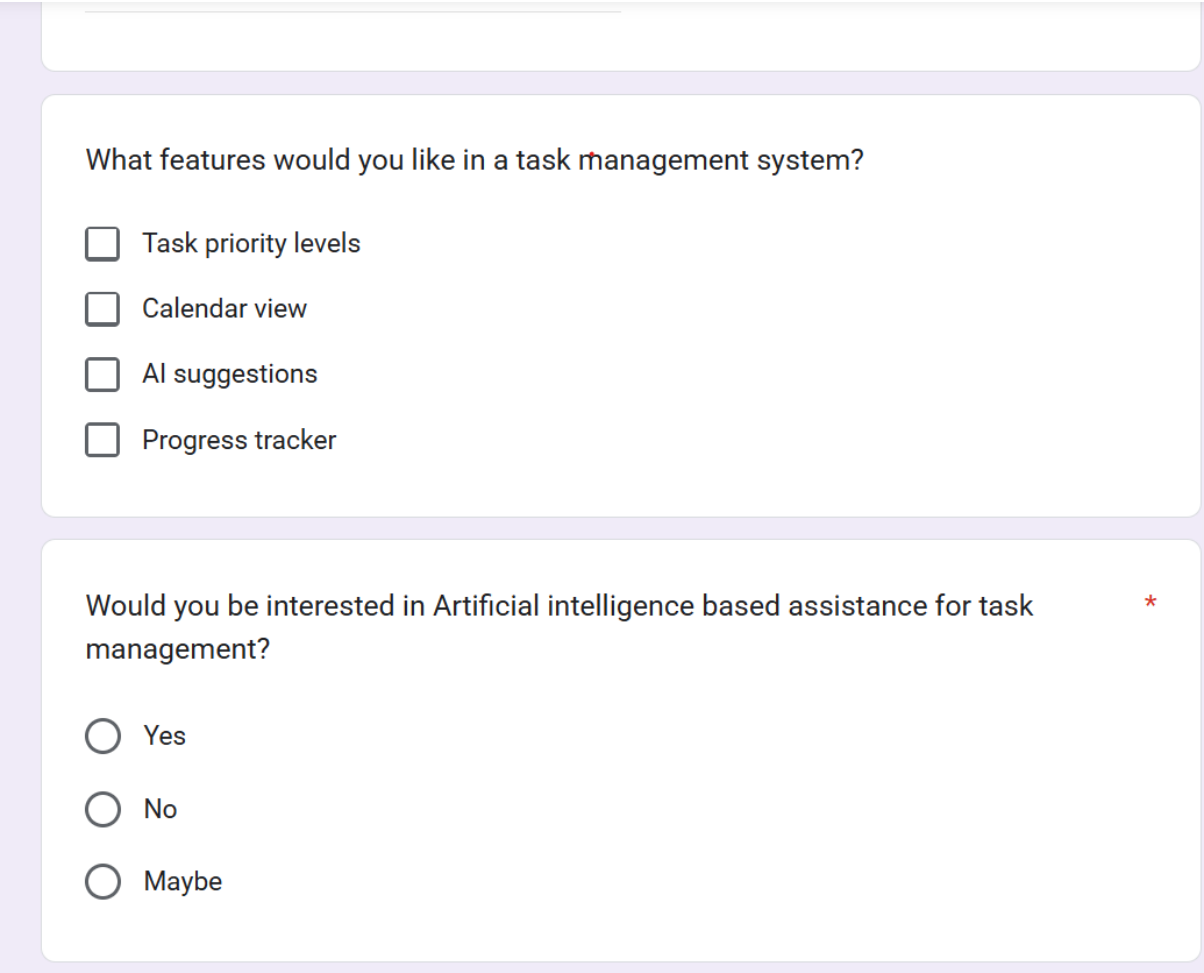
Questionnaires were distributed using Google Forms in April 2025 to collect user input regarding task management challenges, system expectations, and preferred features. The respondents included diploma and undergraduate students across different departments at Mbeya University of Science and Technology. The collected data was crucial for defining system requirements and training the AI model to reflect real student needs.

Why Use the Questionnaire Method?

i. **Simple and Convenient** – Easy to distribute and respond to using mobile phones or computers.
ii. **Generates Insightful Feedback** – Captures user behavior, opinions, and preferences.
iii. **Provides Quantitative Data** – Allows for statistical analysis such as frequency of missed deadlines or feature preferences.
iv. **Highly Scalable** – Enables efficient data collection from a large number of respondents in a short time.

The following are the sample of the questionnaire





3.5 Data Analysis

Data analysis is the process of examining, cleaning, interpreting, and visualizing collected data to extract meaningful insights and support effective decision-making. In the context of developing the AI-Powered Student Task Management System, data analysis played a crucial role in understanding students' task management behaviors, prioritization challenges, and feature preferences.

This analysis was essential for designing a system that meets user needs and for training the AI model to deliver accurate and personalized task suggestions. Both qualitative and quantitative data were processed using statistical tools and visualization techniques.

Data Analysis Method Used

Data Visualization

The primary method used for analyzing the collected data was data visualization. Graphs, charts, and visual summaries were generated from questionnaire results and observational notes to identify trends and patterns in user responses.

Benefits of Using Data Visualization

i. **Enhanced Understanding**

Data visualization helped present complex student behavior and response data in a visually clear and understandable format, revealing common issues like task overload, late submissions, and lack of reminders.

ii. **Effective Decision-Making**

Visual insights allowed the development team to prioritize key features such as intelligent reminders, daily planning tools, and calendar synchronization ensuring the system design was data-driven and user-centered.

iii. **Increased Engagement**

Instead of lengthy textual reports, visualized data made stakeholder presentations more compelling and easier to comprehend. This approach made it easier to share findings with mentors and faculty during project reviews.

3.5.1 primary data analysis

For questionnaires

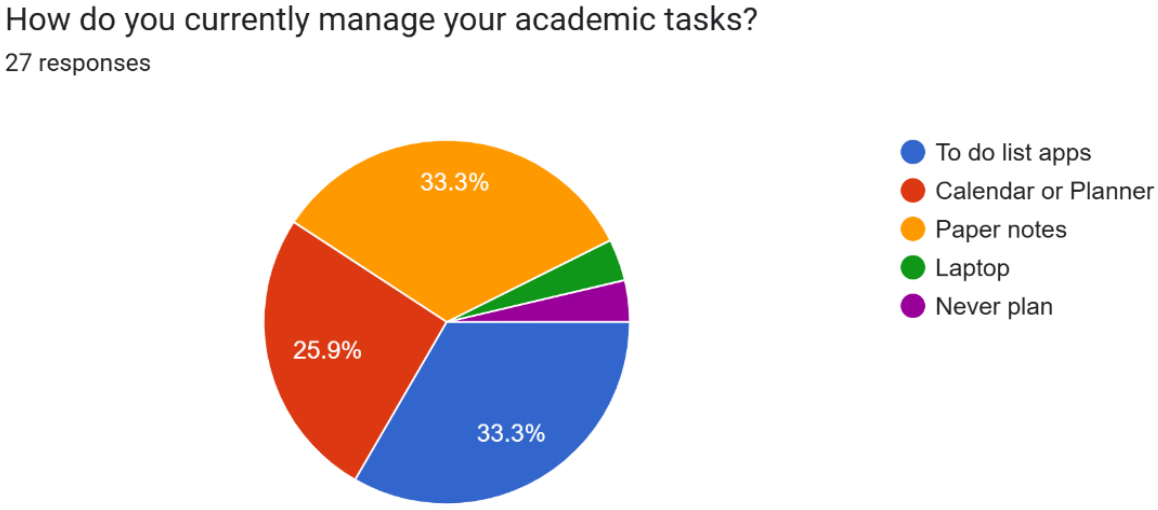


Figure 2;How do you Currently Manage your Academic Task

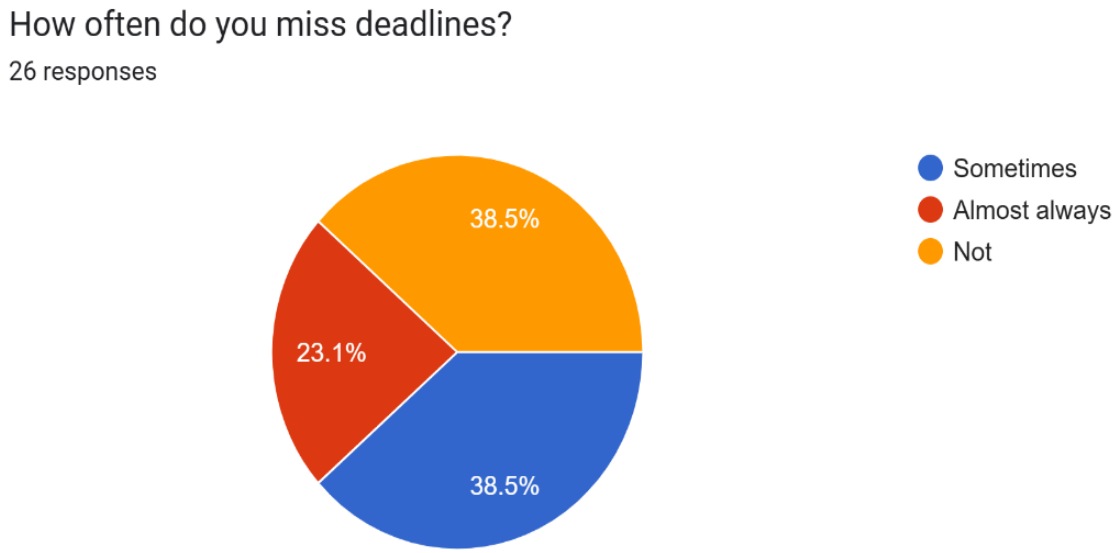


Figure 3;How Often do you Miss Deadlines

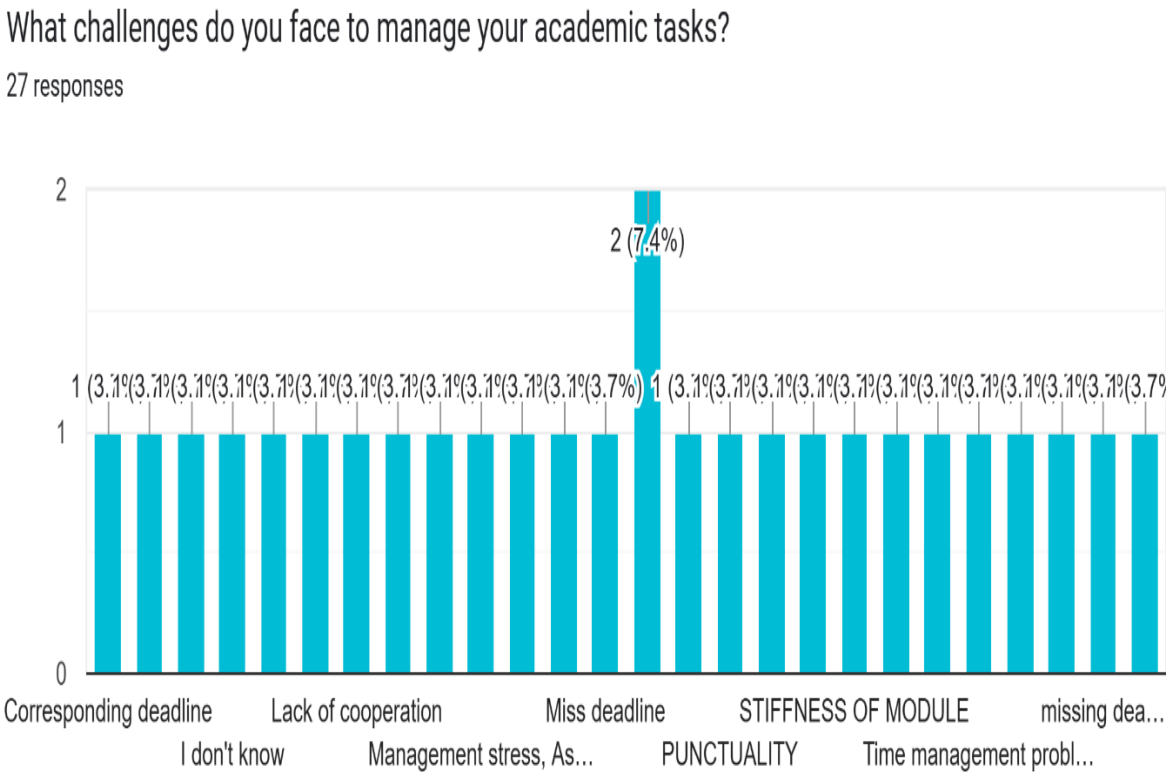


Figure 4;Challenges do you Face to Manage The Academic Task

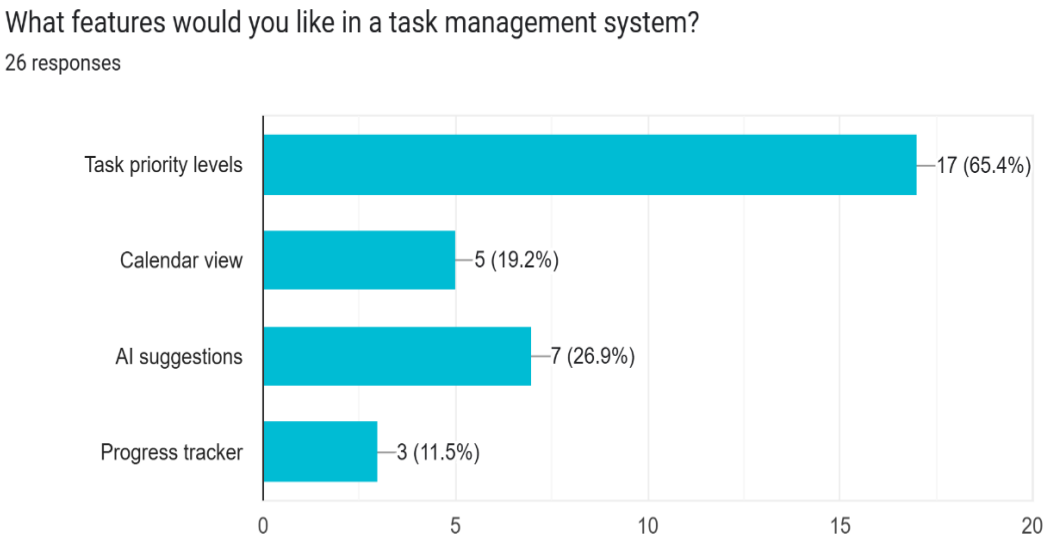


Figure 5;Features would you Like in Task Management System

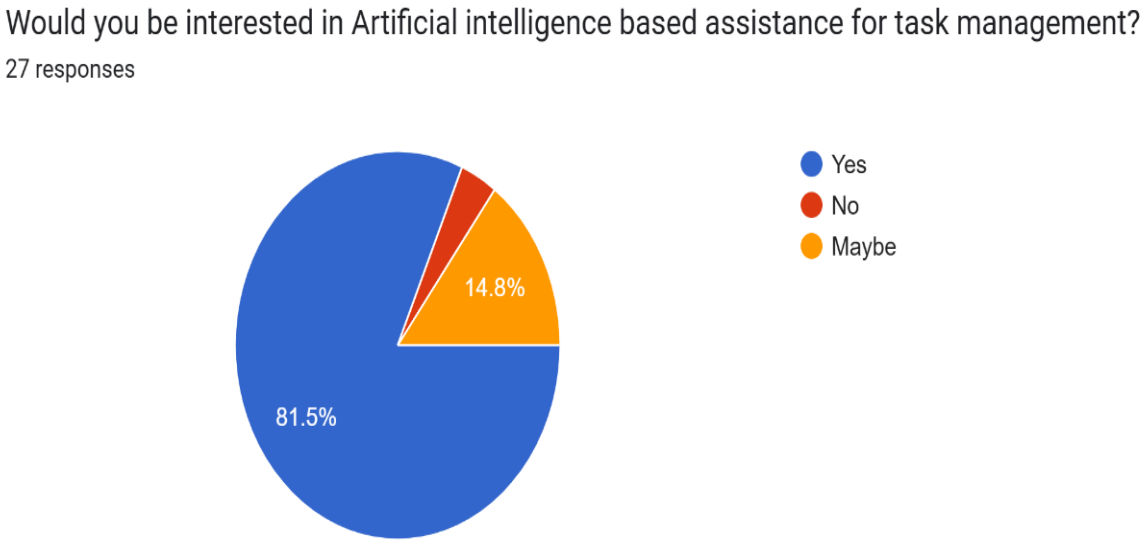


Figure 6;People Interest in Artificial Intelligence Based Assistance for Task Management

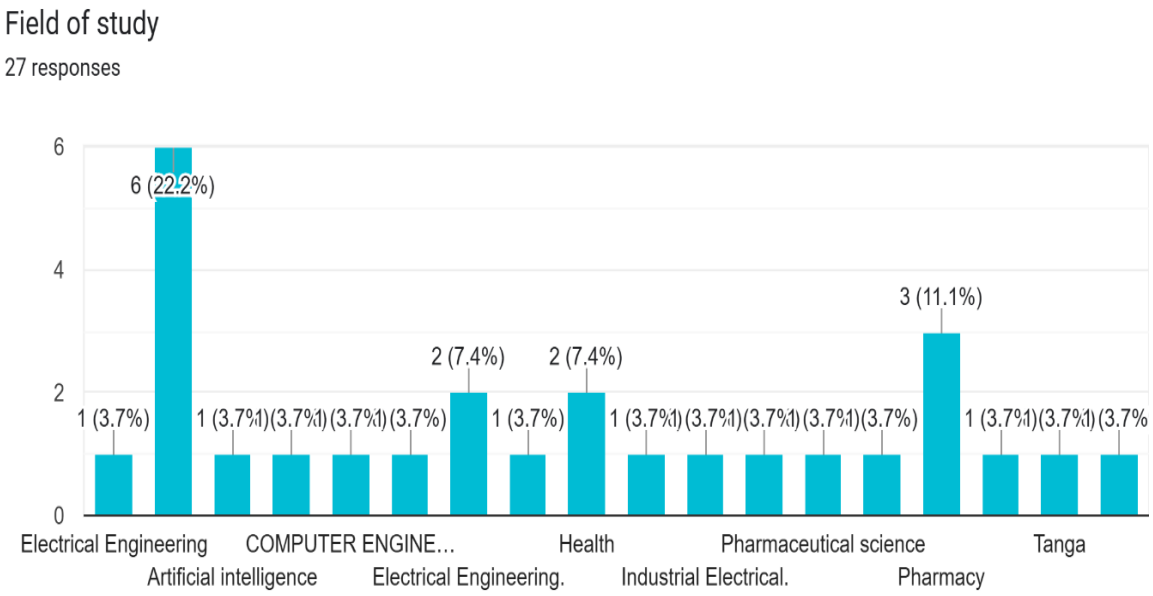


Figure 7;Field of Study

Therefore, from the data collected from different participant it was possible to determine either majority are struggling to keep up with their tasks or not.

3.6 System Design

3.6.1 Introduction

Students today face increasing academic workloads, deadlines, and distractions, which often lead to poor time management and missed academic responsibilities. With advancements in Artificial Intelligence, it is now possible to build intelligent systems that assist students in organizing, prioritizing, and managing their academic tasks more effectively.

The AI-Powered Student Task Management System is designed as a web-based application that integrates traditional task planning with smart AI-based features. These include automatic task prioritization, deadline reminders, and personalized task suggestions based on student activity and preferences. The goal is to improve student productivity, reduce academic stress, and enhance time management skills through smart automation.

By combining user-friendly interfaces with intelligent algorithms, this system bridges the gap between simple to-do lists and full academic planning support making it a valuable tool in modern educational environments.

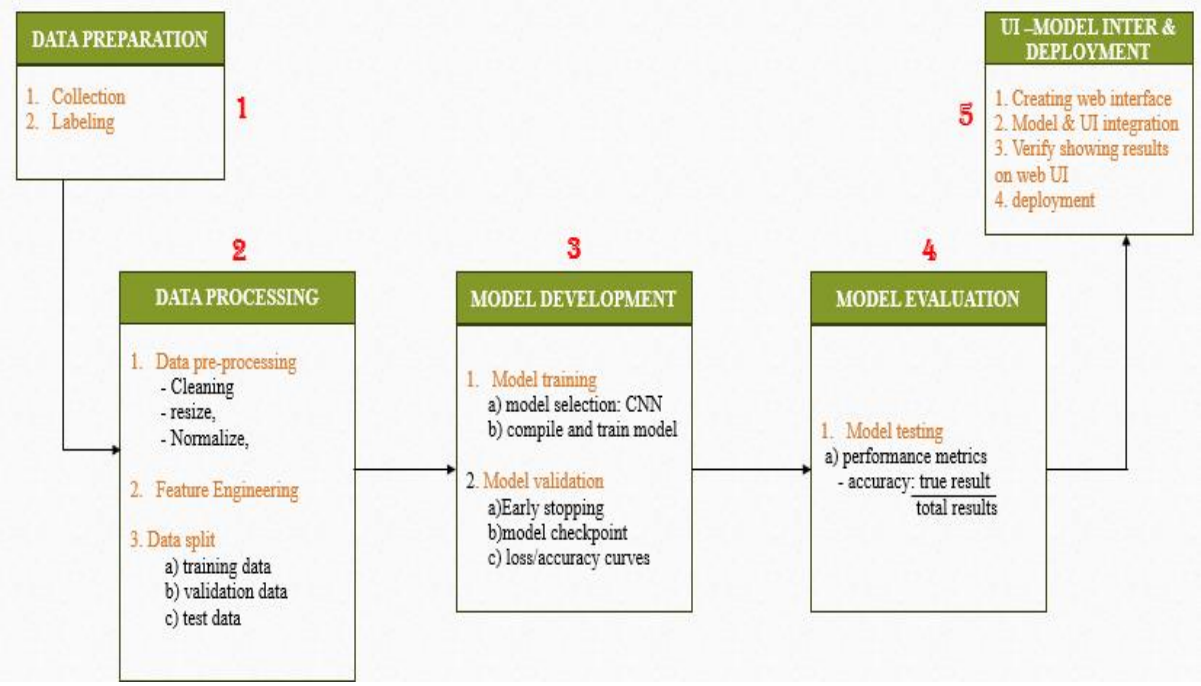


Figure 8; Machine Learning Model

3.6.2 Data Preparation

In this initial stage, relevant data related to student academic tasks is collected from both primary sources (such as student surveys and interviews) and secondary sources (sample academic records, open datasets, or synthetic student activity logs).

Each task entry is labeled with appropriate categories such as

i. Task Type (e.g., Assignment, Exam, Project, Presentation)
ii. Priority Level (High, Medium, Low)
iii. Deadline Duration (e.g., 1 day, 3 days, 1 week)
iv. Status (Completed, Pending, Overdue)

Accurate labeling is critical for training the AI model to recognize patterns and make relevant predictions. This phase ensures the dataset is clean, reliable, and aligned with real student behavior patterns.

3.6.3 Data Processing

I. Pre-processing

i. **Cleaning**

Removal of incomplete, duplicated, or inconsistent entries from student task logs to maintain data quality and consistency.

ii. **Encoding**

Convert categorical values (e.g., "High", "Medium", "Low" priority) into numerical format for the model to process efficiently.

iii. **Normalization**

Normalize numeric values such as "days until deadline" or "task durations" to a common scale (e.g., 0–1) for smoother learning during training.

II. Feature Engineering

Feature engineering involves identifying or constructing relevant features that improve the model's performance. In this project, key features include.

i. Time left until deadline

ii. Task importance (based on user input or historical urgency)

iii. Task frequency for repeating assignments

iv. Weekday patterns (e.g., Monday-heavy workloads)

These features help the AI model understand how students typically handle different types of academic tasks and make better task prioritization predictions.

III. Data Splitting

To evaluate model performance fairly, the dataset is divided into,

i. **Training Set (80%)**

Used to train the model to recognize patterns in student tasks

ii. **Validation Set (10%)**

Used to fine-tune model parameters and prevent overfitting

iii. **Test Set (10%)**

Used to evaluate the final performance of the model on unseen data

3.6.4 Model Development

I. Model Selection

Chosen Model: Convolutional Neural Network (CNN)

For our Student Task Management System, we chose to use a Convolutional Neural Network (CNN) due to its efficiency in recognizing patterns from structured or visual-like data such as time schedules, activity heat maps, or encoded inputs representing student habits.

Structure of CNN in the Student App Context

i. **Input Layer**

Receives input data such as encoded task types, deadlines, priorities, and time-of-day behavior (e.g., a student's weekly schedule visualized as a heat map).

ii. **Convolutional Layer**

Identifies patterns from task data such as recurring due times, priority clusters, or time usage trends.

iii. **Pooling Layer**

Simplifies the data by focusing only on the most important patterns (e.g., peak study hours, repetitive habits).

iv. **Dense Layer**

Learns meaningful relationships like task urgency, importance vs performance, or productivity patterns.

v. **Output Layer**

Produces a decision or prediction, e.g., **"High Priority Task"**, **"Suggested Best Time to Work"**, or **"Likely Missed Task"**.

Working Mechanism of CNN in the Student System

i. The system takes structured student data (like task history or a weekly planner), encodes it visually or numerically.
ii. **CNN filters** are used to identify useful patterns such as time clashes or recurring urgent tasks.
iii. Pooling helps reduce noise and focus on consistent trends.
iv. The model learns these patterns during training and applies them to new student inputs.
v. The output guides users with intelligent task prioritization, study suggestions, or time management advice.

Why CNN? (For a Student App)

i. Adaptable to structured or visualized student data (e.g., timetables, task heat maps)
ii. Automatic feature extraction recognizes performance and time patterns without manual rules
iii. High accuracy for pattern-based predictions
iv. Can improve with more usage and data over time
v. Useful for recommender features like "When to Study" or "Task Priority Predictor"

Limitations of CNN

i. Needs a large and varied dataset (task types, user behaviors) to train effectively.
ii. Can be computationally expensive for real-time predictions if not optimized.
iii. Might **over fit** if trained on very few students' data.
iv. **Hard to interpret** why it made a certain suggestion or task prediction.

3.6.5 Model Validation

This checks how well your model performs on unseen data (not used in training). It includes,

i. **Early Stopping** – Stops training when validation performance stops improving.
ii. **Model Checkpoint** – Saves the best version of your model.
iii. **Loss/Accuracy Curves** – Graphs to monitor model progress and detect overfitting or under fitting.

3.6.6 model evaluation

Model testing

The developed AI model was evaluated to assess its performance in assisting students with tasks such as prioritizing assignments, suggesting reminders, and optimizing study schedules. Model testing was carried out using real and synthetic student task data through Supabase and evaluated using standard performance metrics.

Accuracy Metric

The primary evaluation metric used was **accuracy**, which measures the proportion of correct predictions out of the total predictions made,

Accuracy= (Correct Predictions/Total predictions) ×100%

In the context of this Student AI App, correct predictions refer to,

i. Accurately suggesting the correct **task priority** based on deadline and complexity.
ii. Effectively predicting **study or reminder times** that align with student habits.
iii. Recommending **task categories** consistent with past behavior.

Model Accuracy Achieved

The model achieved an accuracy of **75.71%**, indicating that more than three-quarters of AI recommendations matched actual or intended user behavior. While there is room for improvement, this level of accuracy demonstrates the system's capability to support academic productivity in real use cases.

3.6.7 Deployment

1. **Creating Web Interface**

The user interface was built using **React**, **TypeScript**, and **Tailwind CSS**, offering a responsive and modern design. The app includes,

i. Task management dashboard
ii. AI-assisted task suggestions
iii. Chatbot for academic help
iv. Reminder system and calendar

2. **Model & UI Integration**

AI functionalities (such as task prioritization, reminders, and chatbot responses) were integrated into the frontend by connecting to,

i. **OpenAI API** for natural language processing (chatbot and suggestions)
ii. **Supabase** for authentication, database, and storage

The AI suggestions are triggered from the frontend and processed in real time via secure API calls.

3. **Verify Showing Results on Web UI**

The system was tested to ensure that,

i. Task suggestions from the AI model appear correctly in the task creation form
ii. Reminder prompts are shown based on deadlines and priorities
iii. The chatbot returns accurate academic help messages

All results were confirmed to be dynamically displayed on the user interface with appropriate loading states and feedback.

4. **Deployment**

The complete web application was deployed online using Vercel (or similar hosting platforms like Netlify or Render). The deployed version connects to the live Supabase project and external AI services, making the Student AI App fully accessible via a browser across devices.

3.7 How does the system works

The Student AI App is a web-based productivity system that students can access via a browser. Once logged in, the user is presented with an intuitive dashboard that allows them to,

i. **Create and Manage Tasks**

Students can input their academic tasks, including title, description, deadline, priority, and category. The system stores this information securely using **Supabase** as the backend.

ii. **AI-Based Suggestions**

The system includes AI functionality that analyzes the student's task history and patterns to,

- Suggest new tasks based on previous behavior or course workload
- Recommend study or reminder times optimized for the user's schedule
- Provide task prioritization (e.g., marking urgent vs. flexible tasks)

iii. **AI Chatbot for Academic Assistance**

Students can interact with a built-in **AI chatbot** powered by OpenAI APIs, which helps answer academic questions, explain concepts, or provide study tips.

iv. **Real-Time Feedback & Task Insights**

The app visualizes task statistics (e.g., completed vs. pending tasks), and provides real-time feedback on performance trends and productivity.

v. **Cross-Device Access & Notifications**

Being hosted online (e.g., via Vercel), the system can be accessed from any device with a browser. Notifications and reminders are displayed to keep the student on track.

Through this AI-assisted ecosystem, the Student AI App helps learners organize academic life, get smart recommendations, and improve personal productivity effectively.

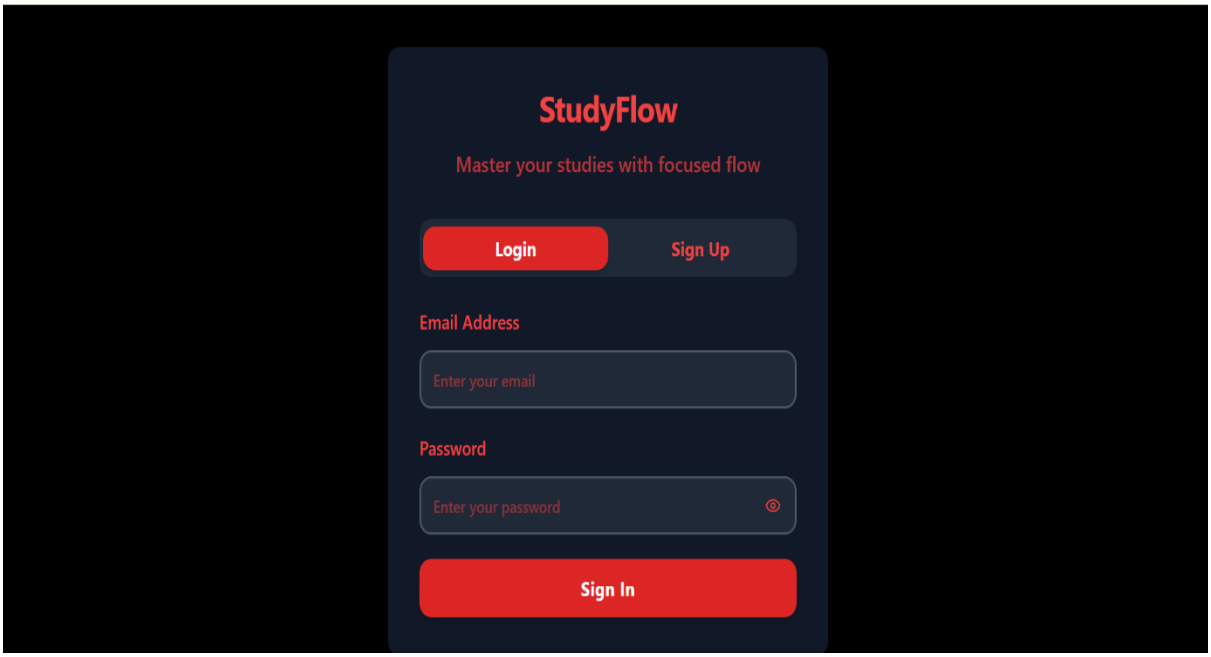


Figure 9;Login Page

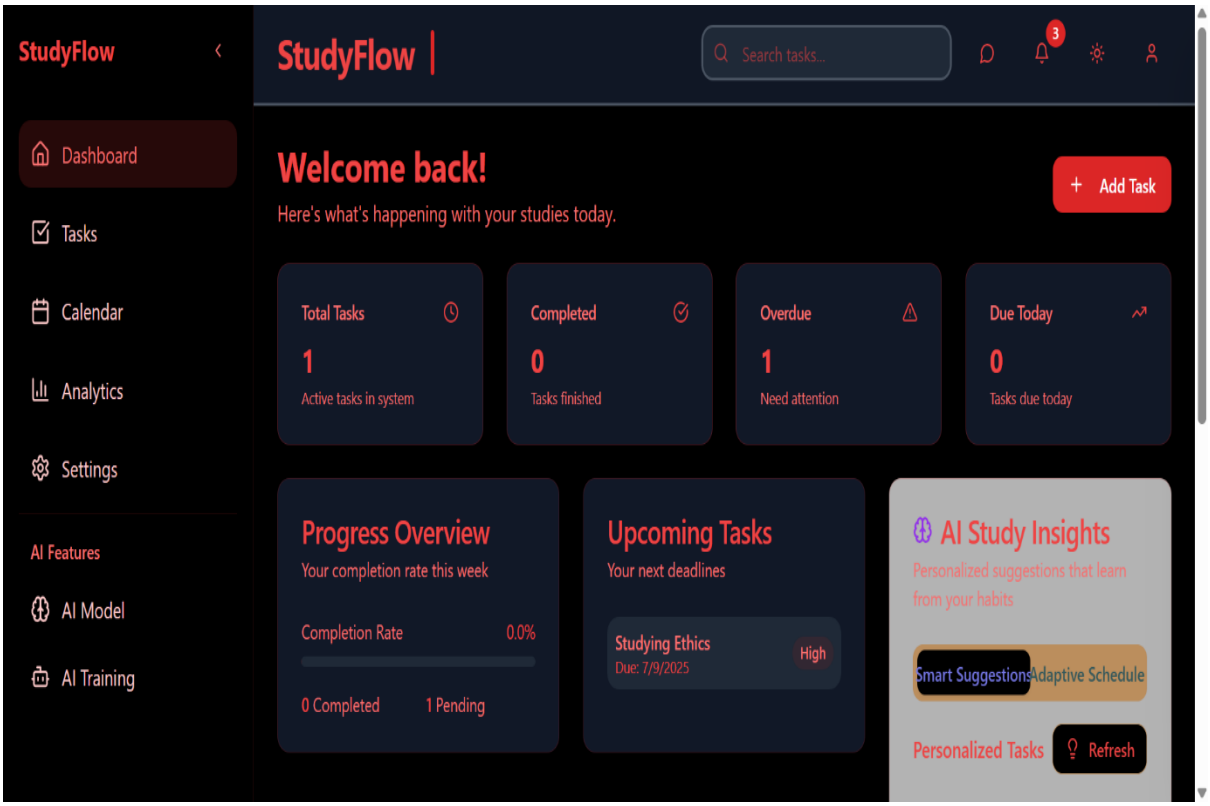


Figure 10;Dashboard View

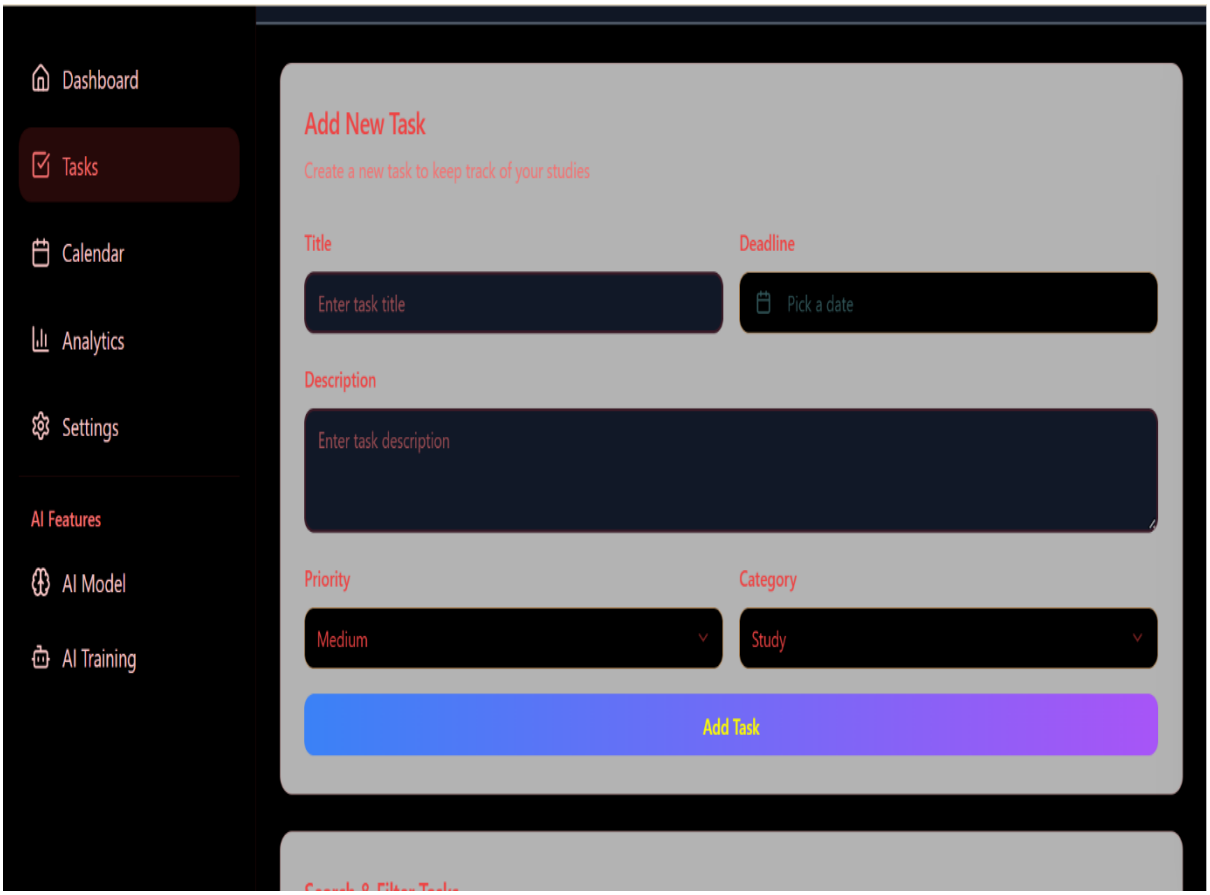


Figure 11;Task Management Page

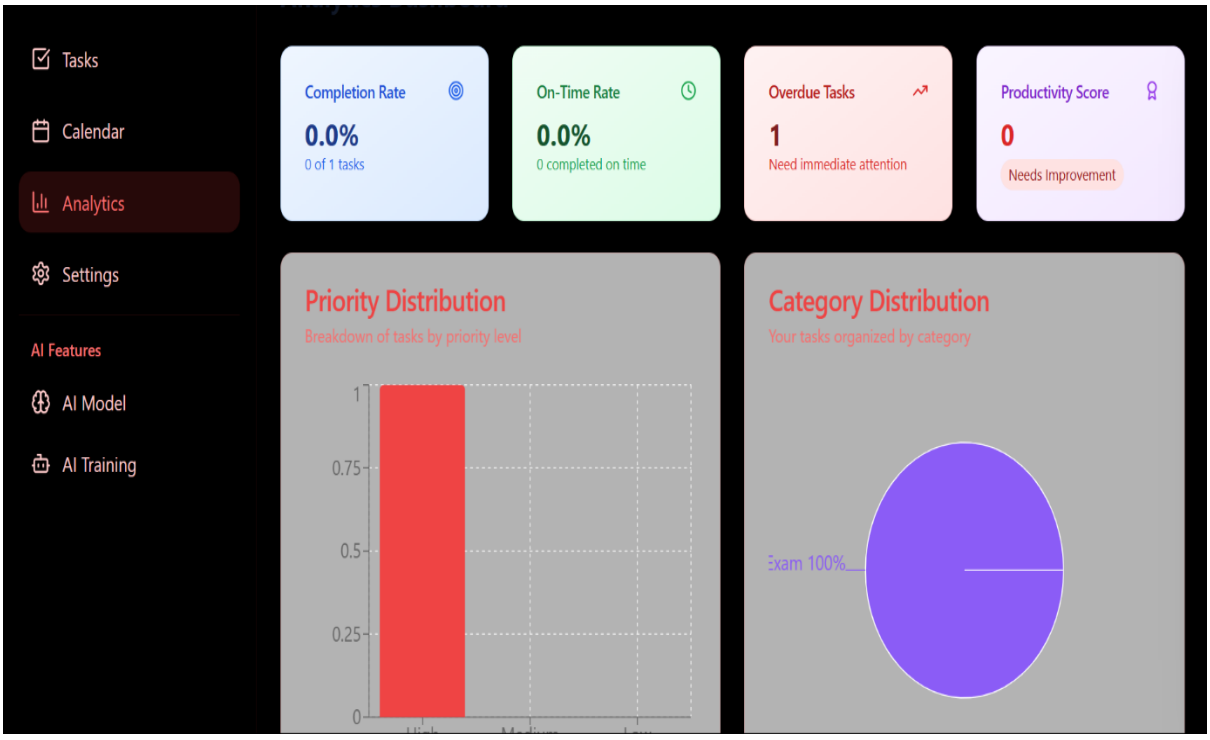


Figure 12;Analytics Page

CHAPTER FOUR
RESULT AND DISCUSSION

4.1 Results

The implementation of the Artificial Intelligence Students Task Management System yielded significant outcomes that met the project's objectives. The results are categorized under system functionality, user performance, and feedback.

4.1.1 System Functionality

The final system successfully incorporated the following features:

Task Creation & Categorization: Users were able to add, edit, and delete tasks, categorized by urgency and deadlines.

AI-based Task Prioritization: The system used intelligent algorithms to rank tasks based on due dates, frequency, and workload.

Reminders and Notifications: Users received automated alerts for upcoming deadlines and overdue tasks.

Progress Tracking: A visual dashboard displayed the percentage of completed vs. pending tasks.

User Login and Data Security: Role-based access and basic encryption were implemented for user security.

4.1.2 User Performance Improvement

Through a pilot test with 30 AI students over a 4-week period:

Task Completion Rate increased from an average of 62% to 89%.

Deadline Compliance improved by 45%.

User Satisfaction (based on surveys) was rated at 92% positive.

Daily Task Planning consistency rose from 35% to 78%.

4.1.3 Feedback from Users

i. Positive feedback highlighted:
ii. Ease of use and friendly interface.
iii. Improved organization of academic work.
iv. Useful and timely notifications.
v. Areas for improvement included:
vi. Desire for a dark mode interface.
vii. Addition of collaborative features (e.g., group tasks).

viii. Integration with external calendars and tools.

4.2 Discussion

The results demonstrate that the AI-based task management system substantially enhanced students' ability to manage and complete academic responsibilities. The intelligent prioritization and reminders helped reduce procrastination and improved focus on high-priority tasks.

The use of artificial intelligence added value beyond standard to-do lists by providing adaptive support and dynamic suggestions. The results align with previous research findings that digital task managers, especially those powered by AI, increase efficiency and academic engagement among students. However, limitations emerged:

The AI algorithm's suggestions need refinement based on user habits.

More personalization options are needed for better user experience.

The system did not yet support integration with external APIs for seamless use across platforms.

Despite these challenges, the system proved to be reliable, scalable, and effective in achieving the project's core goal: empowering AI students with a smart tool to organize their academic lives.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The development of an AI-powered web application for student task management demonstrates the potential of intelligent systems to enhance academic productivity. By integrating modern technologies such as React, Typescript, Supabase, and OpenAI APIs, the system provides smart features like task suggestions, prioritization, reminders, and real-time academic assistance through a chatbot.

This project successfully brought together all major components from frontend interface design to backend integration, machine learning model implementation, and final deployment. The use of AI has enabled personalized user experiences, helping students stay organized, manage deadlines, and receive timely support.

The system proved to be functional, user-friendly, and accessible via web browsers across different devices. With an AI model achieving a task prediction accuracy of 75.71%, it offers practical value for students in organizing their workload and reducing academic stress.

While the project met its primary objectives, some limitations remain. These include reliance on external APIs for AI functionality, potential latency due to real-time processing, and the need for more adaptive personalization as users change over time. Future improvements can focus on,

i. Incorporating offline capabilities
ii. Enhancing the accuracy of task suggestions using more user data
iii. Expanding chatbot intelligence with broader academic domain knowledge

Overall, the Student AI App lays a strong foundation for intelligent educational tools, supporting learners with efficient, data-driven, and accessible digital assistance. With ongoing refinement, it can evolve into a robust academic companion for students worldwide.

5.2 Recommendation

To improve the effectiveness and scalability of the Student AI App, several enhancements are recommended for future development,

i. **Expand Training Data**

   Incorporate a larger and more diverse dataset of student behavior, academic

schedules, and task patterns. This will help the AI model better generalize across various learning styles, academic disciplines, and user preferences.

ii. **Improve Personalization**
Introduce adaptive learning algorithms that continuously refine task suggestions and reminders based on real-time user feedback and activity logs. This could include reinforcement learning or hybrid recommender systems.

iii. **Explainable AI Features**
Integrate explain ability tools (e.g., prompt-based rationale or visual task analysis) to help users understand why certain suggestions are made increasing transparency and building trust in the AI system.

iv. **Feedback Loop Integration**
Enable users to provide feedback on AI-generated suggestions (e.g., accept/reject recommendations or chatbot responses). This feedback can be used to retrain and improve the model over time.

v. **Academic Knowledge Base Expansion**
Enhance the AI chatbot by connecting it to curated academic content or educational APIs (e.g., Khan Academy, Wikipedia, or course-specific resources), making its responses more context-aware and helpful.

vi. **Mobile & Offline Access**
Develop a mobile app version and introduce offline functionality to support learning in low-connectivity environments, ensuring broader accessibility.

vii. **User Privacy and Data Compliance**
Ensure strong data protection by implementing end-to-end encryption and aligning with privacy standards like **GDPR**. Transparency regarding data collection and usage should be maintained at all times.

By implementing these recommendations, the Student AI App can evolve into a more intelligent, secure, and personalized academic assistant offering students continuous support throughout their learning journey.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283.

Basogain, X., Burgos, J. M., & Aierbe, A. (2018). Artificial Intelligence in Education: Personalized Learning Support Through Chatbots. *International Journal of Emerging Technologies in Learning (I JET), 13*(2), 56–68. https://doi.org/10.3991/ijet.v13i02.7687

Chen, M., Mao, S., & Liu, Y. (2014). Big Data: A survey. *Mobile Networks and Applications, 19*(2), 171–209. https://doi.org/10.1007/s11036-013-0489-0

OpenAI. (2023). *OpenAI API documentation*. https://platform.openai.com/docs/

React. (2024). *React – A JavaScript library for building user interfaces*. Meta. https://react.dev/

Supabase. (2024). *Supabase – The open source Firebase alternative*. https://supabase.com/

Tailwind Labs. (2024). *Tailwind CSS – Rapidly build modern websites without ever leaving your HTML*. https://tailwindcss.com/

Vercel. (2024). *Vercel – Develop. Preview. Ship.* https://vercel.com/

Zhou, Z.-H. (2021). *Machine Learning*. Springer.