DESIGN AND IMPLEMENTATION OF AN INTELLIGENT ACADEMIC SUPPORT AND PROGRESS TRACKING INFORMATION SYSTEM

BY

MBADIWE CHINEMEREM
GOU/U22/CSC/893

DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY, GODFREY OKOYE UNIVERSITY, ENUGU STATE.

SUPERVISOR: ENGR. NJOKU CAMILLUS

JUNE, 2026

DESIGN AND IMPLEMENTATION OF AN INTELLIGENT ACADEMIC SUPPORT AND PROGRESS TRACKING INFORMATION SYSTEM

BY

MBADIWE CHINEMEREM
GOU/U22/CSC/893

DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY, GODFREY OKOYE UNIVERSITY, ENUGU STATE. **IN PARTIAL FULFILMENT OF THE AWARD OF BACHELOR OF SCIENCE (B.SC), DEGREE IN COMPUTER SCIENCE.**

SUPERVISOR: ENGR. NJOKU CAMILLUS

JUNE, 2026

APPROVAL PAGE

This research, titled "**DESIGN AND IMPLEMENTATION OF AN INTELLIGENT ACADEMIC SUPPORT AND PROGRESS TRACKING INFORMATION SYSTEM**," has been assessed and approved by the Department of Computer Science Committees in the Faculty of Computing and Information Technology, Godfrey Okoye University, Enugu, Nigeria.

Engr. Njoku Camillus	**..................................**	**..................................**
Supervisor	Signature	Date
External Examiner	**..................................**	**..................................**
	Signature	Date
Dr. Frank Okebanama	**..................................**	**..................................**
HOD, Department of Computer Science	Signature	Date
Prof. I. A. Ajah	**..................................**	**..................................**
Dean, Faculty of Computing And Information Technology.	Signature	Date

CERTIFICATION

I certify that I completed the research included therein and that it was primarily based on my observations and investigation. Consequently, I will fully accept the University's decision if I am found to have cheated on the assignment.

Mbadiwe Chinemerem
GOU/U22/CSC/893

DEDICATION

This project is dedicated to God almighty who has made this possible, and everyone who has contributed positively to my education.

ACKNOWLEDGMENTS

I appreciate my supervisor, Mr. Njoku Camillus for supporting me in the development of this project, The Dean, Prof. I. A. Ajah, and HOD, Dr. Frank Okebanama for their dedication to the success of their students, and all the lecturers of the department who have contributed positively to my education.

I appreciate my parents, Dr. and Dr. Mrs. Okezie Mbadiwe for supporting and sponsoring me throughout my education journey, and the team at BrainCapital Artificial Intelligence for giving me a wonderful internship experience and ultimately giving me the idea for this project.

ABSTRACT

This project provides a solution to the common problem of periodical unavailability and lack of personalization in the academic system, especially in under-resourced areas. It does this by developing an intelligent academic support system that is always available. In many education systems, learning platforms sometimes lack personalized guidance for each individual student, because of this, the systems fail to identify struggling students until it is too late for teachers to provide meaningful help. The students may have already failed an exam before they are identified as “struggling”. To solve this problem, the project uses an approach where a machine learning model observes the activities of students on the learning platform and is trained to identify those who are lagging behind or at risk of failing very early in the semester. It uses a Random Forest machine learning model to observe and monitor the student’s activities closely. Using the information, the system alerts teachers early if a student is at risk of failing their courses. Furthermore, the platform includes an AI powered tutor to assist learners at all times. The AI tutor is powered by Google's Gemini AI, and a technology called “Retrieval-Augmented Generation” (RAG) is used to fine tune the AI tutor so it gives responses based on the learning materials of the school. The primary purpose of the AI tutor is to provide students with continuous, 24/7 access to academic support. The project’s software system was built using Flutter for the frontend and Flask framework for the backend. The choice of development framework helps create a simple and beautiful user interface and a backend that will run smoothly on the web (because Flask framework is very effective for web development). After development, the system was thoroughly tested to verify that its performance is good and that the system is reliable. The evaluation results were very promising for the progress tracking system. This solution showed the accuracy of 90.0%, 89.3% precision and 83.3% F1 Score. Such figures prove that the system is able to detect lagging students effectively and give almost no false alerts. The tests of the separate components of the system including the training sessions and live classroom confirmed all use cases to be applicable. The testing procedure also included the synchronization between the teacher's and student's dashboards ensuring that the updates on the latter would automatically update the former. Furthermore, the testing confirmed that the AI tutor consistently delivers reliable answers because the “RAG” technology ensures the answers are given based on the school’s materials. The system transforms academic support from a reactive system into a proactive process. However, future improvements are recommended to better support areas with slow internet connections.

Table of Contents

Title Page	i.
Approval	ii
Certification	iii
Dedication	iv
Acknowledgement	v
Abstract	vi
Table of Contents	vii
List of Tables	x
List of Figures	xi
Chapter One: Introduction	**12**
1.1 Background of the Study	12
1.2 Statement of the Problem	13
1.3 Aim and Objectives of the Study	14
1.4 Significance of the Study	14
1.5 Scope of the Study	15
1.6 Operational Definition of Terms	15
Chapter Two: Literature Review	**17**
2.1 Introduction	17
2.2 Theoretical Background	17
2.2.1 Conceptual Frameworks for Intelligent Academic Support	17
2.2.2 Methodologies and Techniques	20
2.3 Implementation Limitations	22
2.4 Summary of Literature Review	25
2.5 Research Gap	28
Chapter Three: System Analysis and Design	**29**
3.0 Introduction	29
3.1 System Analysis	29
3.1.1 Analysis of the Existing System	29

3.1.2 Limitations of the Existing System 32
3.1.3 Analysis of the Proposed System 33
3.2 System Requirements Specification (SRS) 34
3.2.1 Functional Requirements 34
3.2.2 Non-Functional Requirements 34
3.3 System Design Methodology 35
3.3.1 Justification for Methodology 35
3.3.2 Method of Data Collection 36
3.4 System Modeling Using UML 36
3.4.1 Use Case Diagram 37
3.4.2 Activity Diagram 39
3.4.3 Class Diagram 40
3.5 System Design 41
3.5.1 Input Design 41
3.5.2 Output Design 43
3.5.3 Database Design 45
3.5.4 System Architecture Design 51

Chapter Four: System Implementation 53
4.0 Introduction 53
4.1 Development Environment and Tools 53
4.1.1 Development Platform and Hardware Requirements 53
4.1.2 Programming Languages and Frameworks 53
4.1.3 Backend Libraries and Dependencies 54
4.2 Dataset Description and Preprocessing 55
4.3 Model Architecture and Training 55
4.4 System Implementation and Modules 55
4.5 Testing and Evaluation 56
4.5.1 Evaluation Metrics 56
4.5.2 System Testing 57
4.5.3 User Interface Testing and Walkthrough 58
4.6 Results Presentation and Analysis 58
4.6.1 Output Samples and Interface Screens 68
4.6.2 Discussion of Results and System Performance 61

Chapter Five: Summary, Recommendations, and Conclusion **62**
5.1 Introduction 62
5.2 Summary of the Study 62
5.3 Conclusion 62
5.4 Recommendation 63
References **64**
Appendices **66**

List of Tables

Table 1.0 Literature Review Summary **Page 24**

Table 2.0 Database Design Tables **Page 46**

Table 3.0 Sample Dataset **Page 67**

List of Figures

Figures	Page No.
1.0 J. Sule's Waterfall Model	**30**
2.0 Use Case Diagram	**37**
3.1 Student Activity Diagram	**39**
3.2 Teacher Activity Diagram	**40**
4.0 Class Diagram	**40**
5.0 Entity-Relationship Diagram	**46**
5.1 First Normal Form	**48**
5.2 Second Normal Form	**48**
5.3 Third Normal Form	**49**
6.0 System Architecture Diagram	**51**
7.0 Confusion Matrix of the Early Warning System	**55**
8.0 Chat Screen Showing Conversation Between the AI Tutor and a Student	**57**
9.0 Teacher Dashboard Showing Student Risk Status	**58**
10.0 Student Dashboard Showing an AI generated Practice Assessment	**58**
11.0 Student Dashboard Showing AI generated Study Plan	**59**
12.0 Teacher's Dashboard Showing a Live Class Link	**59**

CHAPTER ONE
INTRODUCTION

1.1: Background of Study

Educators sometimes struggle to give personalized and timely academic feedback that meets student's needs. This is as a result of various factors which include increasing class sizes, growing student diversity, workload and time constraints to the level of knowledge and training of the educator and to the limitations of existing digital learning systems which include the absence of real time interaction, perceived lack of personalization by the students and the complexity of "intelligent" integration in these systems (A.A. Aldino et al., 2025).

This challenge experienced by educators has a significant impact on the performance of students by influencing their motivation, self-efficacy and academic growth (A.A. Aldino et al., 2025). Digital learning continues to handle these problems through various innovations that have so far been able to overcome geographical and time limitations and also expand access to high-quality education by providing students in remote locations with high-quality educational resources that might otherwise be unavailable in their local institutions. This has led to the transitioning of global education from traditional models to student-centered digital environments particularly over the last decade (Saufi S., 2025).

Academic assistance in the past was provided by means of physical libraries, face-to-face tutorial sessions, and traditional marking procedures. Such systems perform efficiently in small-scale environments but do not fit well into larger contexts or poorer educational institutions. The majority of learning management systems (LMS), such as Google Classroom and Canvas LMS, provide solutions for large-scale courses by means of hosting all course materials; however, they remain static and unable to perceive the difficulties encountered by learners. As a result, the problem emerges when there is plenty of information, but no assistance (Sirohi et al., 2025).

One major challenge that exists in contemporary education is delayed detection of difficulties experienced by learners. In many cases, the challenge faced by the learner comes to light only after he/she has performed poorly in an assessment test or at the end of the academic semester.

There are no means left for addressing the problem by that time. Feedback delay causes students to become disinterested in the subjects as a result of not being able to see any positive changes or getting stuck in a large and cluttered syllabus (Kurniawan et al., 2021).

Emergence of Artificial Intelligence (AI) and Data Analytics has come up with a remedy for these issues. Intelligent systems are able to observe the patterns in the behavior of learners such as login frequency, time spent on activities and performance in quizzes and then use them to predict the future (Balogun et al., 2024). Through the use of Adaptive Feedback Mechanisms, such systems have the capability of acting as online tutors who are available round the clock, thereby providing personalized assistance which teachers in poorly resourced areas may not be able to give (Zhan et al., 2025).

The project is titled "Design and Development of an Intelligent Information System for Student Academic Assistance and Monitoring." The intention of this project is to bridge the gap between the problems faced by students and the corresponding interventions by the school. Through the use of intelligent algorithms that help in tracking progress and giving instant feedback, this project intends to turn learning into a proactive rather than reactive process. This will help ensure that no student falls through the cracks due to lack of intervention at the right time.

1.2 Statement of Problem

Students often struggle to get timely and personalized academic help, especially in under-resourced environments. Limited access to tutors, unavailability of interactive learning platforms, and lack of adaptive feedback are common challenges. This leads to disengagement and lower academic performance (Kurniawan et al., 2021), (Sirohi et al., 2025).

Current Situation and Challenges

i. Lack of 24/7 access to academic support.

ii. Limited personalized feedback in current learning environments.

iii. Students have varying skill levels but receive uniform instruction.

iv. Manual tutoring systems are labor intensive and not scalable.

v. Technological infrastructure is minimal or inconsistent in many regions.

1.3 Aim and Objectives

The aim of this project is to design and implement an intelligent academic support system that proactively monitors student progress. In other words, the project aims at designing a Learning Management System (LMS) that is capable of providing personalized academic support and also tracking the student's activity and progress to ensure all students are carried along.

The objectives of the project are to:

i. Review existing researches and systems on intelligent academic support.
ii. Build a platform that can support all theoretical learning activities.
iii. Integrate a repository for learning materials into the platform.
iv. Integrate a generative AI model for providing academic support.
v. Integrate a predictive model for tracking and identifying underperforming students early.
vi. Integrate modules for practice exams and quizzes.

1.4 Significance of the Study

Firstly, the major significance of this study lies in the ability of the proposed approach to address the ongoing problems of conventional educational programs in resource-scarce situations, which include the inability to provide personalized education and periodic academic assistance (Zhan et al., 2025). The significance of the introduced system can be highlighted from the following three aspects:

i. **Student Benefit:** The platform delivers a structured and convenient online classroom environment that provides access to learning resources continuously. More critically, the use of LLMs enables the provision of a smart and round-the-clock artificial intelligence tutor. This makes it possible for the students to obtain personalized assistance in academics without delays (Sirohi et al., 2025).
ii. **Educator and Institution Benefit:** Through the use of this initiative, educators get access to a preventative early warning system that allows the use of machine learning technology to predict academic performance based on students' interaction history (Henriquez-Nunez et al., 2023). This makes it possible to detect at-risk learners based on the behavioral traits observed.

iii. **Contribution to Academic Knowledge:** Academically, this study provides practical, evidence-based insights into the successful application of artificial intelligence and predictive modeling within learning settings. It contributes valuable functional frameworks to the growing literature on intelligent learning systems.

1.5 Scope of the Study

The project focuses on the design and implementation of an intelligent learning management system targeted at secondary and tertiary institutions. The system will cover modules for online and hybrid (physical but making use of online systems) lectures, an integrated AI powered class assistant and tutor, modules for practice tests and quizzes as well as for the actual tests and quizzes. It will have repositories containing course materials and past examination questions to aid students. The system will also offer these same features to the teachers helping them manage and track their student’s performance and to enhance and refresh their knowledge on the subjects they teach.

1.6 Definition of Terms

i. Learning Management System (LMS): A learning technology system that delivers digital content, referred to as “content objects”, to learners within a client-server environment (IEEE Standard 1484.11.1-2022).

ii. Academic Support Systems: They are structured programs, resources and interventions provided by educational institutions specifically designed to help students succeed academically, address learning challenges and promote retention and graduation (Study.com).

iii. Artificial Intelligence (AI): This is a technology that allows machines to perform tasks that would otherwise require human intelligence to perform.

iv. Intelligent System: An artificial system that operates as an agent, perceiving its environment and exhibiting rational behavior to achieve specific goals (IEEE).

v. Adaptive Feedback: A dynamic form of communication in educational and intelligent systems where the timing, content, and delivery method of feedback are continuously adjusted based on a learner's specific actions, needs, and performance data (tchop.io).

vi. Generative AI: This is a type of artificial intelligence that can create completely new content like text, images, videos based on rules observed from massive amounts of data which they were trained with.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

The current chapter reviews existing literature on intelligent academic assistance and student performance tracking through use of computer-based systems. This will give an insight into the concepts and methods used in previous works on design of such systems, and their various limitations in implementation. This chapter seeks to show what has been done previously and highlight gaps that have not been fulfilled.

2.2 Theoretical Background

Earlier research studies adopted various theoretical frameworks in order to provide intelligent academic assistance. The aim of this chapter is to conduct a review of the theories that were employed in the earlier research studies as well as their methodologies and technologies.

2.2.1 Conceptual Frameworks for Intelligent Academic Support

For intelligent systems to be effective in fostering students' growth and development, they should be based on well-known learning theories. Below are some of the theories for providing intelligent academic assistance in previous projects and researches.

i. **The Perspective of Digital Transformation of Student-Centred Education:** The conceptual framework of the project "Design and Implementation of a Learning Management System" by J. Sule (2025), is focused on transformation of education from out-of-date manual practices to contemporary digital ones. In particular, the concept views LMS not as a mere storage folder for files but as an interactive "digital classroom" where students and teachers can communicate instantly. The main idea of the project is addressing the problem of delays in communication and lack of accessibility of materials due to the fact that everything will take place on a single website.

   The crucial aspect of the proposed conceptual framework is making the learning process "student-centred", which implies designing of a system taking into consideration needs and requirements of computer science students who are quite different from the majority of school students. As compared to school websites, the system under discussion includes special technical features such as built-in coding environment and places where collaborative works of students can take place, which is similar to real professional IT environments. Besides, the system will be able to automatically track students' progress and give feedback thus fostering motivation of learners and speeding up learning process for the instructor.

ii. **Ecological Perspective:** In the study “Generative artificial intelligence as an enabler of student feedback engagement: a framework” by Y. Zhan et al. (2025), the conceptual framework used is based on an ecological perspective, whereby student engagement is perceived as an outcome of the interaction between the learner and their environment. In particular, the framework examines the interaction between the feedback environment provided by Generative AI (GenAI), including affordances such as instant access and constrains such as risk of bias (bias being the risk that AI models generate biased, distorting and/or discriminatory content, images, or code due to learning based on unrepresentative, or biased data. (www.SAP.com)), and the student’s feedback literacy in relation to that environment. The core idea in this study is that engagement enabled by GenAI requires the presence of a positive alignment in which students have the required abilities (such as prompt engineering and evaluative judgement).

In doing so, the authors present the “Cyclical Self-Regulation Feedback Model” which helps the student go through the three processes of feedback namely; elicitation, processing, and enactment. This framework organizes the student’s interaction with the AI technology in such a way that the entire feedback process is divided into three regulatory processes; Feedback Forethought, where the students create goals for the prompts to be made, Feedback Control where they evaluate the outputs based on accuracy, and Feedback Retrospect, where they assess their strategy in using the technology.

iii. **Adaptive Perspective** In the research project entitled "Adaptive Model for the Selection of Resources and Activities in a Virtual Learning Environment" carried out by Y. Henriquez-Nunez et al. (2023), 10 the conceptual framework aims at breaking free from the traditional approach of "a one size fits all" education by designing a system that looks at each student individually. The conceptual framework is made up of various types of information that are collected on individual students. These include the student’s personality, whether he/she is an introvert or extrovert, the way he/she learns best, whether visually or verbally, and his/her previous knowledge level based on test results.

Having set the profile, the system makes use of the intelligent “inference engine” to provide automated filtering and recommendations on how best to match the most appropriate learning material to this particular student. For example, a "visual" and "introverted" learner can be recommended visual materials such as diagrams and puzzles, while a "verbal" and "active" learner will be recommended textual discussion groups or team games on the same subject matter. Such a framework ensures that the learning process, which involves reading and performing certain activities, is tailored to the needs and preferences of the learner.

iv. **Socio-Technical Perspective:** In the project “Deep Learning-Based Early Warning Systems for Identifying At-Risk Students in Higher Education: A Socio-Technical Framework for Precision and Intervention” by M. Balogun et al. (2024), The conceptual framework is based on the assumption that there is an observable trajectory of academic performance of students in terms of which student success can be traced through time. Rather than focusing only on the final score of a student for the whole academic year, the proposed system relies on the "smart timeline" method (specifically, using a Bi-LSTM model), which allows following the behavioral pattern of a student, his/her activity in logging into his/her account or accessing the learning resources. Based on analysis of this pattern, "at-risk" students can be detected as early as the fifth week of the semester, when a potential failure of a student becomes apparent.

Apart from being used to generate data, the framework adopts the socio-technical approach, meaning that it links up the prediction of the computer to actual human assistance. It goes further than just identifying the at-risk student; the process should lead to the intervention by the teacher or advisor and provision of appropriate assistance to the particular student. By merging accurate technology with proper academic assistance, the framework strives to shift from mere identification of potentially struggling students to an effective prevention of dropout or failure.

v. **Knowledge-Augmented Interaction Perspective:** The article "From Static Platforms to Intelligent Assistants: Reviewing LMS Challenges and the Role of Retrieval Augmented Generation" by T. Sirohi et al. (2025) presents the concept of changing traditional learning management systems from static document warehouses to intelligent assistants. Students usually download some documents and have to wait for an answer to their queries from the lecturer. The new framework suggests the use of the Retrieval-Augmented Generation technology, which plays the role of a mediator between the student and course material. It is not a simple warehouse for documents; it is capable of reading them and can offer students assistance in understanding difficult concepts 24 hours a day.

In order to achieve this, the framework applies a two-stage approach, namely, retrieval and explanation. The first stage includes the retrieval of relevant passages from the actual course textbooks, slides, and other materials that are given by the lecturer. Next, the framework applies artificial intelligence to create an explanation that is consistent with the retrieved content. In this way, the framework can provide accurate information to the user. By providing such assistance in real time, the framework will decrease the load on the lecturers and ensure that no students are stuck with their questions unanswered.

2.2.2 Methodologies and Technologies

i. **Waterfall Model, Web2, PHP and SQL:** In developing his LMS, J. Sule (2025) adopts the Waterfall Model, a development process that involves following a set process from requirements gathering to analyzing, design, implementation, and testing of the system in a specific order. Data was collected during the development process by using observation and interviews to determine the needs of both the learners and lecturers. A three-tier architecture was used for the development of the system to separate presentation layer from the logic layer and the data management layer. These include HTML, CSS, and Javascript for the presentation layer, PHP/Node.js for the logic layer, and MySQL/PostgreSQL for the data layer.

ii. **Cyclical Framework, Generative Artificial Intelligence:** Y. Zhan et al. (2025) used a research method which includes four essential stages such as the search for literature, analysis of different theories, adoption of an "ecological" viewpoint (how do students interact with the environment), and the improvement of generated concepts by means of expert feedback. Instead of performing an experiment or creating a real-world system, the authors relied on logical reasoning in order to create a new cyclical framework, which shows students how to successfully get and utilize feedback. The main technology described is Generative Artificial Intelligence (GenAI), which makes use of Large Language Models (LLMs) – supercomputers trained on large databases of text data to give instant human-like feedback to students in the form of AI chatbots.

iii. **Case-Study Methodology, Inference Engine, Standard Web Technologies:** Y. Henriquez-Nunez et al. (2023) used the case study method to develop and test an adaptive learning system known as ALPY among 73 engineering students using the MOODLE platform. In order to provide a personalized experience, the system uses data from online personality tests and learning style tests taken by students and analyzes this data using an “inference engine” to create a unique digital profile for every user. All of these features have been developed using common web technologies such as HTML, CSS, and PHP, enabling the system to recommend suitable reading material and exercises for every user.

iv. **Deep Learning, Synthetic Data Generation, Generative Adversarial Networks (GANs):** The authors M. Balogun et al. (2024) utilized an advanced research methodology based on synthetic data generation with deep learning to design a "early warning system" for students. In order to address privacy concerns associated with the project, the authors applied technology known as Generative Adversarial Networks (GANs), which can generate a realistic dataset of fake students' activity data reflecting actual classroom activities. Then, an artificial intelligence algorithm called Bidirectional Long Short-Term Memory (Bi-LSTM) neural network was deployed to analyze students' behavior and identify potentially struggling learners as early as during the fifth week of the semester. In addition to this sophisticated algorithm, the authors applied the methods of Explainable AI (XAI) like SHAP to provide easy-to-understand justifications of the AI decision-making process.

v. **Systematic Literature Review, Retrieval-Augmented Generation (RAG):** In their study T. Sirohi et al. (2025), use the methodology of a Systematic Literature Review, which implies the analysis of the already available literature on the subject in order to explore the ways of combining AI technologies and LMS systems. Based on current trends and limitations of these digital platforms, the authors try to estimate how their newly suggested solution may work in practice. The core technology used in this case is "Retrieval-Augmented Generation (RAG)", where an intelligent AI technology, i.e. Large Language Model (LLM), is connected to actual documentation of the university (course manuals, policies, etc.). For this purpose, the system uses the Vector Databases (FAISS, Pinecone, etc.) and Semantic Search, enabling it to find the most appropriate information for the given question based on its semantic meaning.

2.3 Implementation Limitations

However, despite numerous advantages of the actual implementation of these systems, there have been various weaknesses noticed. Some of the disadvantages that have been noted during past implementations are stated below.

J. Sule (2025): J. Sule (2025) noted several limitations when he completed the implementation of his Learning Management System; these include:

i. **No Artificial Intelligence Support:** The system does not currently include advanced features like AI-based tutoring or personalized learning assistants.
ii. **Lack of Immersive Technologies:** There is no integration of virtual reality (VR) or augmented reality (AR) tools for more interactive learning experiences.
iii. **Third-Party Limited Connections:** The program is not programmed to be used in conjunction with any other third party or educational software.
iv. **Specificity to the Department:** Although it serves its purpose effectively, it is only meant to be used in an academic environment and for Computer Science majors in particular.

Y. Zhan et al. (2025): Although the authors suggested an innovative approach to the use of Artificial Intelligence in feedback process, they identified several limitations as well as dangers related to its implementation:

i. **Misleading information**: In some cases, the system might have provided biased and misleading information which could be confusing for the learners.
ii. **Ethical considerations:** The possibility of using technology for plagiarism purposes instead of studying remained persistent.
iii. **Dependence from Technology:** The risk existed for students to develop dependence from the technology which could have an impact on the development of critical thinking skills.
iv. **Lack of literacy:** The effectiveness of the framework required the presence of "feedback literacy" of a student; otherwise, the system did not work effectively.

Y. Henriquez-Nunez et al. (2023): From the ALPY research outcomes, a number of important constraints have been identified by researchers in the following ways as a result of implementation of the adaptive pathway approach among the engineering students:

i. **Lack of Variety in Test Subjects:** This case study only took into consideration an introductory system engineering class of one particular university, implying that the study is only applicable for limited set of students and subjects.
ii. **Small Set of Learning Styles:** Even though the system has used both learning styles and personality for its adaptation, however, the diversity of ways through which different students absorb information for different subjects has not yet been considered by the system.
iii. **Necessity of Including More Data:** Currently, the model takes into account certain variables, such as the previous knowledge of students and their personalities, whereas it has been noted that more contextual information is also needed.
iv. **Necessity of Verification of Results Over Time:** As the results of this study have been analyzed at the end of this case study, therefore, there is a need to prove the long term effects on student retention and completion of degree program.M.

Balogun et al. (2024): Based on the findings of the researchers and the design of the project, the following limitations have been highlighted after the end of the project:

i. **Use of Synthesized Data**: In the study, synthesized data was mainly produced by artificial intelligence (GANs) as an attempt to simulate the students' behaviors. Even though the data used in the process is anonymous, it is worth noting that the validity of the model is still to be tested on different university datasets.

ii. **"Black Box" Model Limitation:** Deep learning models such as Bi-LSTM model are inherently opaque. That is why it becomes rather difficult for instructors to know what led to the flagging of the student's account. That is why the researchers recommended to integrate Explainable Artificial Intelligence (XAI).

iii. **Ethical and Biased Risks:** It has been indicated by the researchers that if the training data is historically biased, the AI might cause systemic discrimination towards marginalized and vulnerable student groups.

iv. **Necessity of Human Intervention:** The project highlights that while accurate predictions are critical, it is one of the main limitations of the process because the system will not work without proper institutions and human staff.

T. Sirohi et al. (2025): Limitations and Risks of Implementation of a RAG-Based System in a Learning Management System:

i. **Risk of “Hallucinations”:** Despite the fact that the system operates based on the university’s documents, sometimes the AI generator still generates information which is incorrect and is not included in the original text.

ii. **Dependence on the Quality of Documents:** The quality of the AI answers depends fully on the way university documents (e.g., policies and handbook) are written and structured; hence, in case the university documents are outdated and poorly drafted, the AI provides bad pieces of advice.

iii. **Technical Issues and Barriers:** The high-speed internet connection is a must for optimal operation of the system; besides, some problems such as server load and system latency may interfere in the learning process, especially in regions with inadequate infrastructure.

iv. **Technical Competence**: It is important for the system to be used efficiently by students and teachers in order to have good technical knowledge.

2.4 Summary of Literature Review

Author	Journal/Source	Key Findings	Limitations
J. Sule (2025)	Bachelor of Science (BSc.), Nasarawa State University, Keffi	the project implements a student-centered digital classroom that replaces manual education methods with an interactive, specialized platform featuring built-in technical tools and automated tracking to enhance real-time communication and personalized learning.	No AI Support, Lack of Immersive Technologies, Limited Third-Party Connectivity, Departmental Specificity.
Y. Zhan et al. (2025)	Higher Education Research & Development 2025, VOL. 44, NO. 5, 1289–1304	GenAI effectively enables student engagement only when a student's technical literacy aligns with the AI's capabilities, a process optimized through a cyclical model of planning, monitoring, and reflecting on feedback.	Risk of Misinformation, Ethical Concerns, Dependency Issues, Literacy Gaps.
Y. Henriquez-Nunez et al. (2023)	Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 1, pages 256-263	the project implements an adaptive learning framework that uses an intelligent inference engine to match personalized student profiles, based on personality, learning style, and prior knowledge, with the most effective customized resources and activities.	Limited Diversity in Study Participants, Narrow Range of Learning Styles, Need for Broader Data Integration, Validation of Long-term Effects.
M. Balogun et al. (2024)	Research Gate, presented at University of Ibadan	the project implements a socio-technical early warning system that uses advanced deep learning to track real-time behavioral patterns, allowing for the precise identification and proactive human intervention of at-risk students by the fifth week of the semester.	Reliance on Simulated Data, Black Box Problem, Ethical and Bias Risks, Need for Human Intervention.

T. Sirohi et al. (2025)	2025 International Conference on Future Technologies (ICFT)	the project implements a Knowledge-Augmented Interaction framework using Retrieval-Augmented Generation (RAG) to transform static platforms into intelligent assistants that provide instant, contextually accurate support by automatically retrieving and explaining relevant sections of course materials.	Risk of “Hallucination”, Dependency on Document Quality, Technical Performance Hurdles, Need for Technical Literacy.
G. Lampropoulos (2025)	Applied Sciences, MDPI	his paper carried out a systematic literature review through the PRISMA method on how intelligent tutoring system is being incorporated in augmented reality and virtual reality. It revealed that such an approach ensures that there are immersive, interactive, and personalized educational sessions for students which improve their motivation and increase their academic results.	Some of the difficulties discussed include issues such as data security problems, biases in algorithms, ethics concerns, hardware constraints, and expensive development costs.
Y. Wang (2025)	Scientific Reports	A mixed AI system which includes decision trees, random forest, support vector machine, and artificial neural network is used for the purpose of monitoring the academic progress and predicting at risk students. In the case of the UCI data set, the accuracy rate of the hybrid system is 98.8%, which is much higher than the individual base system.	The system is still sensitive to data insufficiency and imbalance. Actual implementation will be constrained by the organization’s preparedness, cost, legal and ethical issues.
A. A. Soomro et al. (2025)	The Critical Review of Social Sciences Studies	Based on the survey carried out among 1,200 students, this study reveals that the use of the artificial intelligence advising system brings confidence, satisfaction, and increase in	The survey technique employed by this study is subject to bias by the students and causality cannot be determined through

		chances of retention of students. In addition, these systems are extremely effective in assisting students to link up their classes to their future careers.	this technique. In addition, there are issues related to the privacy of the data collected and the need for human advisors.
Chunyan Duan et al. (2024)	ICIEAI 2024, ACM	Intelligent tutoring systems use big data, natural language processing, and real-time interactive feedback to create personalized learning plans. The systems dynamically adjust teaching materials and provide immediate help when students face difficulties in their studies.	Challenges include technological complexity and the need for a deeper understanding to provide true personalized services. Future research must address data storage privacy, the fair distribution of resources, and compatibility with existing teaching methods
Haifang Li & Zhandong Liu (2025)	Electronics, MDPI	The researchers created an intelligent educational system using large language models and SageGNN-based graph embeddings. The system analyzes students' campus card consumption and academic scores to find learning risks early and generate personal behavior reports.	The system depends heavily on campus card data, which can create bias if students do not use their cards often. The model is built on one university's systems, limiting its general use, and the AI might misinterpret some behaviors.
Irina Shpolianskaya & Tatyana Seredkina (2020)	BRAIN. Broad Research in Artificial Intelligence and Neuroscience	The study presents a personalized online learning support system that uses ontologies and fuzzy logic to evaluate e-learning courses. The system acts as a recommender, matching the semantic descriptions of online resources with a student's profile to build an individual learning path.	Building a learning path is highly complex because it requires multi-parametric analysis of course outcomes, learning styles, and student knowledge. The system relies on accurate ontology models to link user needs with the educational materials.

Abeer Assiri et al. (2020)	International Journal of Advanced Computer Science and Applications (IJACSA)	This systematic review found that most electronic academic advising systems are single-task systems focused on course selection. The study concludes that multi-task intelligent systems using artificial intelligence are the best way to reduce advisor workloads and provide accurate, customized advice.	Major challenges in this field include handling special-case students, reducing errors caused by bad input data, and preventing information overload. Advising systems also require constant updates to keep rules and guidelines accurate.

Table 1.0 Literature Review Summary.

2.5 Research Gap

We have now seen that previous researchers have been able to develop systems which are able to support traditional teaching and learning methods with a student-centered digital classroom (J. Sule, 2025), predict when a student is in danger of failing (M. Balogun et al.,2024), provide 24/7 academic support for students using generative AI (T. Sirohi et al., 2025), and provide adaptive learning paths that match student's profiles (Y. Henriquez-Nunez et al., 2025). However, there isn't a known unified system that implements these solutions to provide a proactive and personalized academic support that's always available.

M. Balogun's early warning system has been able to predict if a student is going to fail, but they stop there, a human teacher is then required to step in. There's room for the addition of a system that provides 24/7 tutoring to help flagged students catch up. Y. Henriquez's project was limited to a few learning styles and narrowed to the software engineering field, a system that covers a wider range of learning styles across many more fields solves the identified problem for more individuals. Studies by T. Sirohi and Y. Zhan showed that using AI in education could lead to "hallucinations" (AI giving wrong answers or giving sources that don't exist). A system that uses Retrieval Augmented Generation (RAG) tied directly to the school's course materials will ensure the AI assistant gives factual, subject specific answers without making things up.

Previous developers have built individual puzzle pieces, the gap is that no known system has successfully combined the pieces into a single intelligent academic support system.

CHAPTER THREE
SYSTEM ANALYSIS AND DESIGN

3.0 Introduction

In this chapter, we would be analyzing existing academic support systems as well as the new system we would be proposing, our focus will be on the system requirements, the design methodology and the visual models for describing the system behavior.

3.1 System Analysis

We have previously understood that academic support systems and learning management systems (LMS) are structured programs, resources and interventions provided by educational institutions specifically designed to help students succeed academically, address learning challenges and promote retention and graduation (Study.com), in this section we would be analyzing how J. Sule's LMS and M. Balogun's Early warning system were able to achieve those functions, we would analyze the limitations of the systems they were able to develop and highlight which of the limitations the proposed system will be attempting to solve.

3.1.1 Analysis of Existing Systems

i. **Learning Management System by J. Sule (2025):** The system was designed to solve the problem of the "scattered" method of managing classes and learning materials across different platforms like WhatsApp, emails and google classrooms observed in his research. The system attempts to solve the problem by putting everything in one easy-to-use place.

   System Requirements: The system has two types of requirements:

   a. **Functional Requirements:** This system requires secure user login, registration in the courses, and access to study materials. Additionally, it should allow the users to submit their assignments, real-time chatting, notifications, and grade checking.

b. **Non-Functional Requirements:** This system should be secure, efficient and scalable for handling large number of users. Furthermore, it should be user-friendly and compatible with various web browsers.

Designing Approach: In the design phase, the project follows the waterfall approach where tasks are carried out in sequence such as requirement gathering, systems analysis, systems design, implementation, and testing.

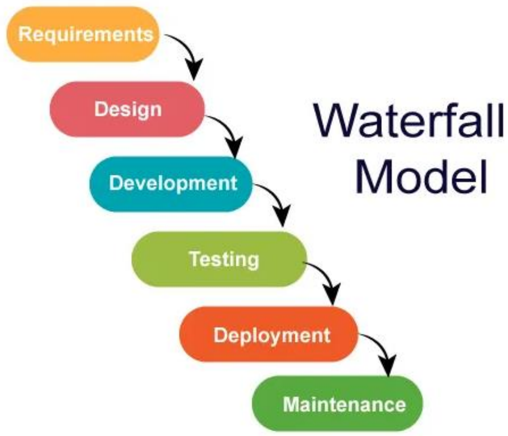

Figure 1.0 J. Sule's Waterfall Model

The system is designed using three basic components:

a. A user interface for people to see and click (using HTML, CSS, and JavaScript).
b. A server side that does the background thinking (using PHP or Node.js).
c. A database to store all the information safely (using MySQL or PostgreSQL).

System Algorithm Description: The algorithm below describes the logical flow a user (student or instructor) follows while interacting with the LMS platform. This interaction focuses on accessing learning content, submitting assignments, and reviewing progress. Figure 3.3 presents the corresponding flowchart.

i. **Start**: The user initiates interaction with the LMS by launching the system.

ii. **Login**: The system prompts the user to log in using secure authentication (role-based for students or instructors).

iii. **Access Dashboard**: Upon successful login, the user is redirected to a personalized dashboard.

iv. **Select Feature**: The user chooses an activity such as accessing lecture content, submitting code assignments, joining a project group, or reviewing grades.

v. **Perform Task**:

   a. For students: View/download lecture materials, write and submit code, or participate in group projects.

   b. For instructors: Upload content, create coding tasks, or review student submissions.

vi. **Real-Time Feedback or Updates**:

   a. If code is submitted, the system evaluates it and displays immediate feedback.

   b. For content interactions, learning progress is updated automatically.

vii. **Receive Notification**: The system sends a confirmation or notification regarding task completion, deadlines, or feedback.

viii. **Log Interaction**: The system records the session activities in the database for tracking and analytics.

ix. **End**: The session concludes when the user logs out or exits the platform.

This algorithm serves as the foundation for modelling various system interactions in subsequent UML diagrams, such as use case diagrams, activity diagrams, and sequence diagrams, which are detailed in the following sections.

ii. **Early Warning System by M. Balogun et al. (2024):** The system was designed to provide a solution to students dropping out or failing because traditional systems are not able to catch the problem early. It attempts to solve the problem by using deep learning algorithms to predict students who are likely to drop out as early as the fifth week of a semester. The predictions it makes are based on things like how often the student logs into the system or how often they click on the learning materials.

System Requirements: To work properly, the system requires the following:

a. **Data Requirements:** It needs data to be constantly collected in real time through a Learning Management System (LMS) such as time spent by the student on doing a task and the speed at which they turn in their assignments.
b. **Explainability (Explainable AI):** The algorithm needs to use techniques such as SHAP for transparency purposes to help teachers understand why a certain student is flagged as "at risk".
c. **Fairness and Privacy:** The model needs to maintain the privacy of the student and not unfairly penalize any particular group of students who may have been discriminated against in the past.

Design Methodology: The researchers applied the following methodology for designing and testing the system:

a. **Data Generation:** Since actual data from the students can be confidential, they generated a synthetic dataset with the help of an artificial intelligence method known as a GAN which contains 10,000 data of students.
b. **Feature Engineering:** The static factors were mixed with the dynamic factors in order to train the system. For the purpose of making training consistent, the SMOTE technique was applied to generate balanced data.
c. **Experimenting and Comparative Analysis:** The researchers designed the main system by applying a sequential aware deep learning method known as a Bi-LSTM model. They compared it with the XGBoost model. There was strong focus on the Recall metric.

3.1.2 Limitations of Existing Systems

i. **Learning Management System by J. Sule (2025):**

   a. **Narrow Focus:** The system is developed especially for the computer science department. It is not made to be a versatile tool for use by all students at the university.

b. **Absence of Innovative Technology:** The Learning Management System uses simple technologies that can be found in regular digital classrooms and does not incorporate any new technologies such as AI tutors, VR technology, and third-party software integration tools. Being deprived of these technologies, the system cannot provide any intelligent tutoring and cannot interact dynamically with students who have certain disabilities.

ii. **Early Warning System by M. Balogun et al. (2024):**

a. **Application of Fake Data:** In order to avoid exposing students' data to any security risks, the team conducted their research using computer-synthesized data. As such, the system still requires testing with actual university data to establish that it performs ideally.
b. **Discrimination Risk:** If the data utilized for training the system contains inherent discrimination bias in any way, it can cause AI to make discriminatory predictions that will impact marginalized student groups adversely.
c. **Dependency on Human Interventions:** While the system can notify the school about the risk to the student, it does nothing in terms of actually helping the student out because it is fully dependent on the school being prepared to do that through human interventions.

3.1.3 Analysis of the Proposed System

The proposed system aims to provide a comprehensive platform where all students can easily access course materials and get academic support anytime they need it. It not only aims at detecting when a student is in danger of failing, but at providing the student with customized learning paths and the necessary support to help them catch up. The specific objectives are to:

i. Provide easy access to necessary learning materials.
ii. Provide always available, personalized and contextual intelligent support to students, based on their school's materials. This solves the lack of intelligent support limitation observed in J. Sule's Learning Management System.
iii. Proactively track student's progress and flag those who are in danger of failing early.

iv. Offer necessary support to flagged students so they can catch up, solving the reliance on human action limitation observed in M. Balogun et al.'s Early Warning System.

3.2 System Requirement Specification

In this section, we would be analyzing what the system should be able to do (Functional requirements) and how it should behave (non-functional requirements).

3.2.1 Functional Requirements

i. **Course material storage:** The system should have a repository to store course materials and past examination questions for easy access.
ii. **AI Tutor:** The system should have an integrated AI-powered class assistant that can answer subject specific questions interactively at any time.
iii. **Progress Tracking:** The system should proactively monitor student activities and automatically flag students who are underperforming early in the semester.
iv. **Customization of learning content:** The platform should have the capability of adjusting itself to suit each individual user's level of expertise according to his or her interaction history.
v. **Practice quizzes, tests, and exams:** The platform should have features to enable students to take practice quizzes, tests, and real exams.
vi. **Teacher's dashboard:** The platform should have features that give teachers the ability to manage their classes and track the performance of their students.
vii. **Lecture Support:** The system should include modules to host online and hybrid lectures.

3.2.2 Non-Functional Requirements

i. **Availability:** The academic support, especially the digital tutor must be highly available 24 hours a day, 7 days a week.
ii. **Usability:** The platform must feature an organized classroom structure and a user-friendly interface to ensure it is easy for both students and teachers to use.

iii. **Scalability:** The system must be able to handle a large volume of students smoothly, solving the problem of high student-to-teacher ratios in under-resourced schools.
iv. **Timeliness:** The feedback generated by the system should be instant and should be able to identify any weak student in time before they fail in their exam.
v. **Contextual Accuracy:** The intelligent help generated by the system should be contextually accurate according to the curriculum of the particular school.

3.3 System Design methodology

In this project, two methodologies have been applied to address both of the components of the system i.e. the software application as well as the machine learning model. In the case of the development of software and overall system design, the methodology used is the Agile Methodology while for the development of the predictive model, CRISP-DM methodology has been used.

3.3.1 Justification for Methodology

The following design methodologies were chosen for this project due to the following advantages:

i. **Agile Methodology:** The choice of this approach was made against rigid design methodologies like the Waterfall model, which were used in previous software products. The Agile methodology is very flexible and allows constant testing and improving of the project. Due to the fact that this project incorporates the use of the Artificial Intelligence (AI) assistant, which should be thoroughly tested to prevent the creation of false information (known as hallucinations), this approach will allow the testing of the system using actual data from the course material. In addition, this approach allows developing the project module by module and testing if such modules as course repository, testing modules, and teacher dashboards function properly.
ii. **CRISP-DM:** This methodology is widely used in the industry as an ideal process for the development of projects related to data mining and machine learning. It perfectly suits the needs of the early warning system and provides

guidance through all stages of the project: understanding the problem of the poor academic performance of the students, preparation of the necessary data, training the algorithm to detect patterns in the behaviour of at-risk students, evaluation of its accuracy and deployment into the teacher's dashboard.

3.3.2 Method of Data Collection

For this project, data was collected through two major ways to make sure that the system functions and operates correctly:

i. **Dataset Download for Machine Learning**: To develop the predictive model of the system, a dataset with the logs of the interactions of the students with the system and their behaviour, test results was downloaded from Kaggle.com. Real-world data helps the model to detect real behavioural patterns without the limitations of computer generated data, which was the issue in previous research.

ii. **Secondary Data Collection:** To understand the features of the Learning Management System, the data was collected by analysing literature, academic articles, and already developed academic support systems.

3.4 System Modeling

In this part, we will use the Unified Modeling Language (UML) to represent how our system works and its internal structure.

3.4.1 Use Case Diagram

This diagram represents the relationship between the actors (users) and the functionalities of the system (use case).

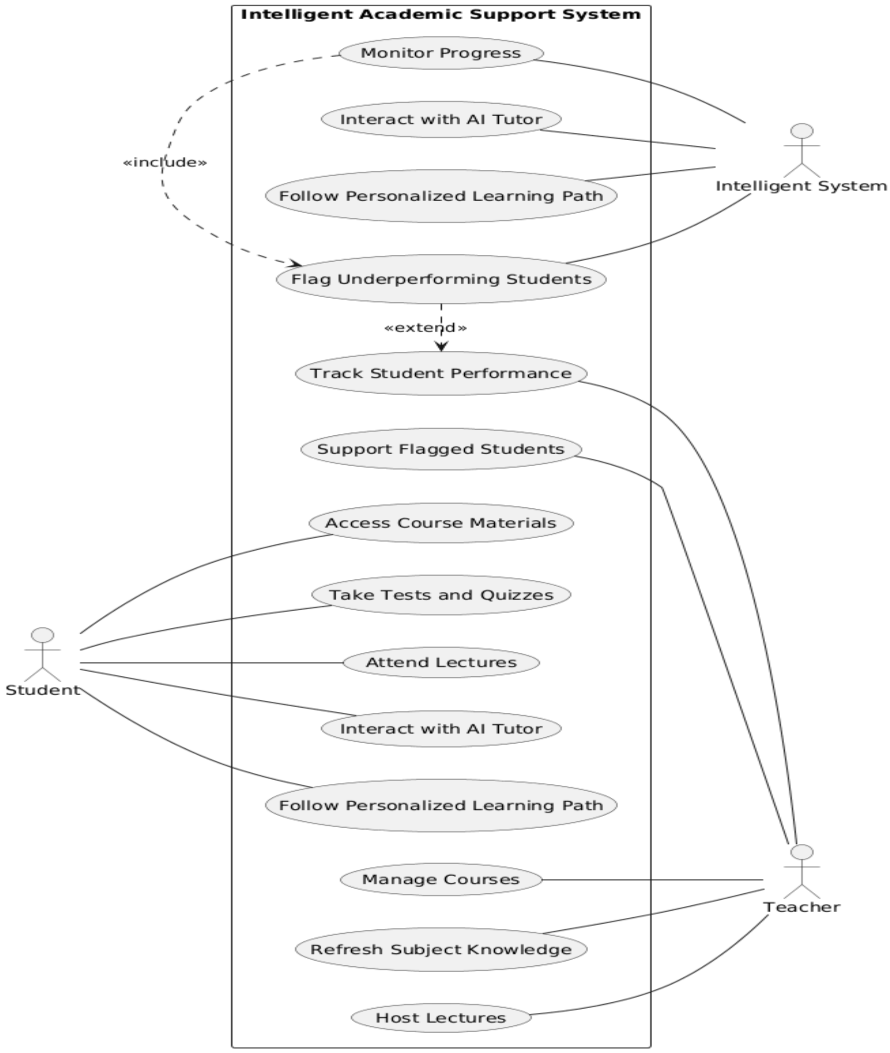

Fig 2.0 Use Case Diagram

Actors:

i. **Student**: A user who accesses learning materials, takes assessments, and receives personalized AI support.

ii. **Teacher**: An instructor who manages course content, tracks student progress, and provides extra help to struggling learners.

iii. **Intelligent System (AI)**: An automated background actor that acts as a digital tutor, monitors user activities, and flags underperforming students.

i. Students:

a. **Retrieve Learning Materials:** The learners will be able to access past examination papers and learning materials from the system's repository.
b. **Interact with an AI Tutor:** The learners will be able to interact with an AI tutor for answers to specific subject matters.
c. **Take Tests and Quizzes**: Students can take practice tests, practice quizzes, and actual exams directly on the platform.
d. **Follow Personalized Learning Paths**: Students receive customized learning routes that adapt to their specific skill levels based on their past interactions.
e. **Attend Lectures**: Students can join online or hybrid classes using the system modules.

ii. Teachers:

a. **Manage Courses**: Teachers can use the system tools to organize and control their class structure.
b. **Track Student Performance**: Teachers can view data to monitor how well their students are doing.
c. **Support Flagged Students**: Teachers can step in to offer necessary help to students who the system has marked as underperforming.
d. **Host Lectures**: Teachers can conduct online and hybrid lecture sessions for their classes.
e. **Refresh Knowledge**: Teachers can use the system features to enhance and refresh their own knowledge on the subjects they teach.

iii. Intelligent System:

a. **Monitor Progress**: The system proactively tracks student activities in the background.
b. **Flag Underperforming Students**: The predictive model automatically alerts teachers early in the semester if a student is in danger of failing.

3.4.2 Activity Diagram

The activity diagrams depict the flow of activities in the core processes of the system. In the proposed system, a user can either be a student or a teacher, the diagrams below show the flow of activities when a student logs in (Fig 3.1), and when a teacher logs in (fig 3.2).

Student Activity Diagram:

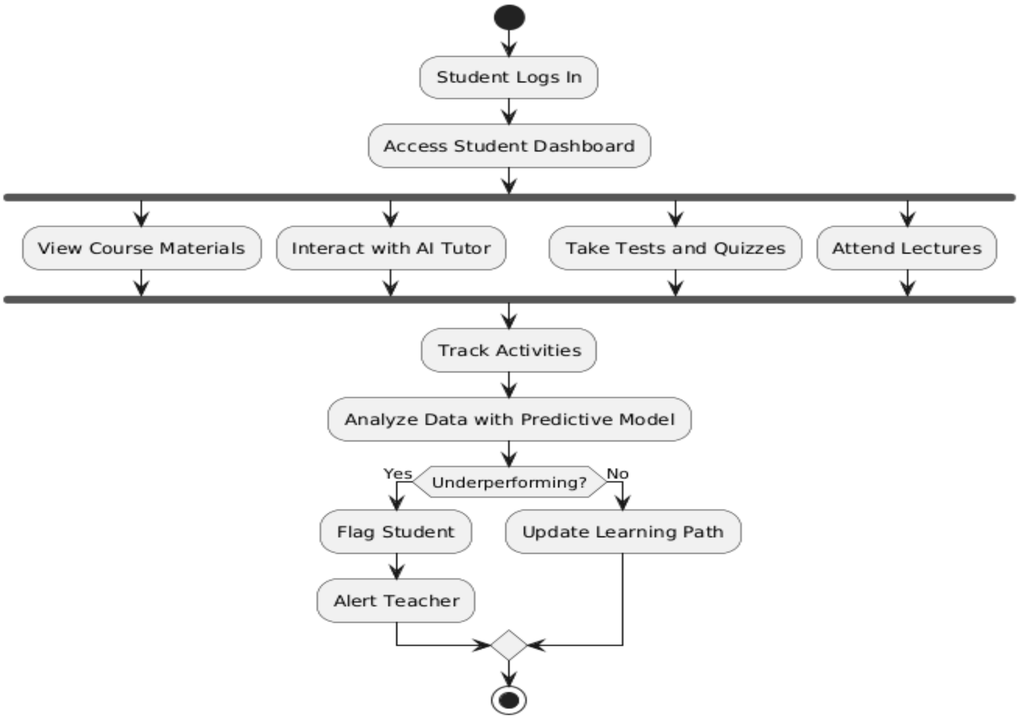

Fig 3.1

Student Activity Diagram

Teacher Activity Diagram:

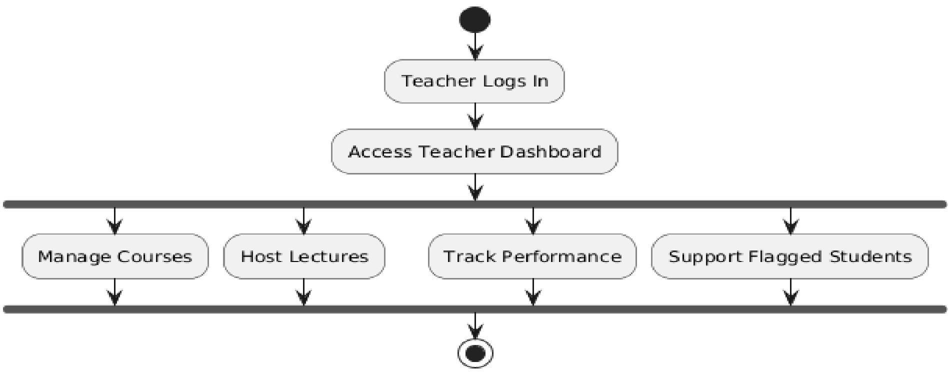

Fig 3.2 Teacher Activity Diagram

3.4.3 Class Diagram

The class diagram illustrates the classes and the relationships between components in the system.

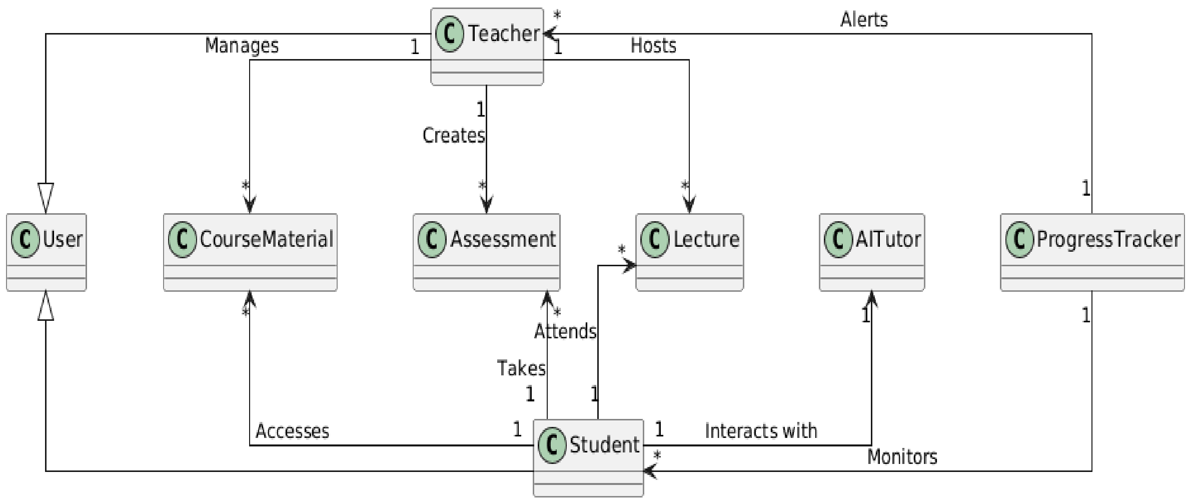

Fig 4.0 Class Diagram

i. **User (Base Class):** It is the base class which encompasses all users using the system. It has common attributes such as ID, Name, Email, and Password.

ii. **Student (Child Class):** Derived class from the user class. It caters to the student functionality such as accessing course material, assessment, lecture attendance, and communication with the AI tutor.

iii. **Teacher (Child Class):** Inherits from User. It handles teacher-specific functions like managing courses, hosting lectures, and supporting flagged students.

iv. **Course Material:** Represents the repository items like past examination questions and lecture notes.

v. **Assessment:** This refers to practice tests, practice quizzes, and examinations.

vi. **AITutor:** This refers to the class assistant system which uses artificial intelligence to answer subject-related questions.

vii. **ProgressTracker:** This refers to the prediction system which tracks the performance of students and identifies poor performing students.

viii. **Lecture:** Represents the online and hybrid lecture sessions.

3.5 System Design

In this section, the analysis of the system will be converted into a design.

3.5.1 Input Design

The input design will be based on how data will be collected in the system. The input design will ensure the correctness, safety, and convenience of the submitted information. The below tables present the critical input screens, the formats of collecting data, and the validation techniques that will be applied to it.

1. User Registration and Login Screen

i. **Description:** This screen is the first one that the users encounter in the system. Users can register new accounts or login into existing ones to have access to their respective dashboards.

ii. **Data Capture Formats:**
   a. Email address (Text box).
   b. Password (Masked text box).
   c. User Role (drop-down menu with options "Student" or "Teacher").

iii. **Validation Techniques:**
   a. **Format:** Checks whether the email is in the right format (contains "@" symbol and the name of the domain, e.g. edu.com).
   b. **Presence Check:** Will not allow the user to submit an empty form.
   c. **Length:** Checks whether the password is not less than 8 characters long.

2. AI Tutor Chat Interface

i. **Description:** In this screen, the users will have the possibility to type in subject-related questions and communicate with the Gemini AI digital tutor.

ii. **Data Capture Formats:** Chat input area (multi-line text box).

iii. **Validation Techniques:**

a. **Character Limit:** Will restrict the user from typing more than the set number of characters (e.g. 500) per each message to avoid system overload.
b. **Sanitization:** Removes any potentially harmful symbols and computer code automatically if the user tried to put them into the chat interface.
c. **Empty Submission:** Will disable the "send" button if the text box is empty.

3. Course Material Upload Form

i. **Description:** This form is used by teachers to upload their lecture notes, readings, and previous examination questions into the system repository.
ii. **Data Capture Formats:**
  a. Document Title (Text box).
  b. Course Code (Text box).
  c. File attachment (File upload button).
iii. **Validation Techniques:**
  a. **Specific File Formats Acceptance:** This form accepts only certain file formats such as .pdf, .docx, or .pptx
  b. **File Size Validation:** Files exceeding certain size, say 10 Megabytes, are not accepted due to the storage limitation.
  c. **Presence check:** It ensures that the document title and course code have been provided by the teacher before accepting the file upload.

4. Assessment and Quiz Form

i. **Description:** The screen through which students take their assessment test or exam.
ii. **Data Capture Formats:**
  a. Multiple Choice Answers (Radio buttons).
  b. Multiple Select Answers (Checkboxes).
  c. Short Answer (Text boxes).
iii. **Validation Techniques:**
  a. **Warning message about incomplete answers:** If a student attempts to click "submit" without answering all questions, a warning message pops up.
  b. **Automatic Submission after Time Expires:** Background timer submits the answer after expiry of the allotted time to the student.

5. Lecture Scheduling Form

i. **Description:** A screen through which teachers can schedule their future lectures either online or blended.

ii. **Data Capture Formats:**
   a. Lecture Topic (Text box).
   b. Date (Calendar picker tool).
   c. Time (Clock picker tool).
   d. Meeting Link (Text box).

iii. **Validation Techniques:**

   a. **Limitation of date selection:** Teacher cannot select dates and times that have passed.
   b. **Checking URL format of the meeting link entered:** The system checks that the entered meeting link has a proper format beginning with http:// or https://.

3.5.2 Output Design

Output design involves the presentation of the information after processing to the users. The aim of output design is to provide clear, understandable and actionable data to be presented through reports, alert messages and dashboards to the students and teachers. The following are some of the output screens and reports provided by the system.

1. Students' Progress and Result Report

i. **Description:** Personalized dashboard for every student that contains details about the performance in different subjects.

ii. **Information displayed:**
   a. The total marks gained for the quizzes, tests and actual examinations.
   b. The percentage of the course content that has been covered.
   c. The recommended topics for study in the personalized learning process.

iii. **Output format:** Visual dashboard that can also be downloaded into a report card in pdf format.

2. AI Tutor Response Display

i. **Description:** The output panel where the computerized teacher provides answers to the question asked.

ii. **Information Displayed:**

   a. Explanation of subject related concepts in text form.

   b. Answers to complex questions in steps.

   c. References to the exact lecture notes or subject matter in the repository database.

iii. **Output format:** Chat-like display with distinct bubbles for the messages and answer messages.

3. Teacher Class Performance Dashboard

i. **Description**: General overview screen for teachers to track the performance of the class as a whole.

ii. **Data Displayed:**

   a. The average marks obtained by the entire class.

   b. List of submitted and marked assessments.

   c. Attendance details of the class in case of online/hybrid lecture mode.

   d. No. of downloads/visits of course content.

iii. **Output Format:** Tables and bar graphs shown on teacher's screen. Also, there is an option to download the data in form of a spreadsheet file (Excel / CSV)

4. Early Warning Alert Report (Output of Predictive Model)

i. **Description:** Most important output for teachers. Generated by the machine learning model to identify at-risk students.

ii. **Data Displayed:**

   a. List of "Flagged Students."

   b. The reason why they were flagged (e.g. "Missed 3 quizzes", "Low engagement with AI tutor", "Scored less than 40% in recent tests")

   c. Risk Level (e.g. High Risk, Medium Risk).

iii. **Output Format:** Alerts on teacher's main dashboard and automated email alerts sent to the teacher directly.

5. Course Material and Lecture Schedule View

i. **Description:** Screen showing the list of learning resources and lectures scheduled by the teacher.
ii. **Data Displayed:**
    a. Organized and categorized list of uploaded lecture notes, old questions, readings etc.
    b. Calendar view showing the date, time and join links for future online/hybrid lectures.
iii. **Output Format:** List view and calendar view on screen.

3.5.3 Database Design

The database schema depicts how the information will be stored, managed, and linked together. The type of database schema to be employed in this project will be a relational database schema because it can depict the linkages between different entities.

The diagram below presents the ERD with tables, fields, and linkages through the primary keys (PKs) and foreign keys (FKs).

Entity-Relationship Diagram (ERD)

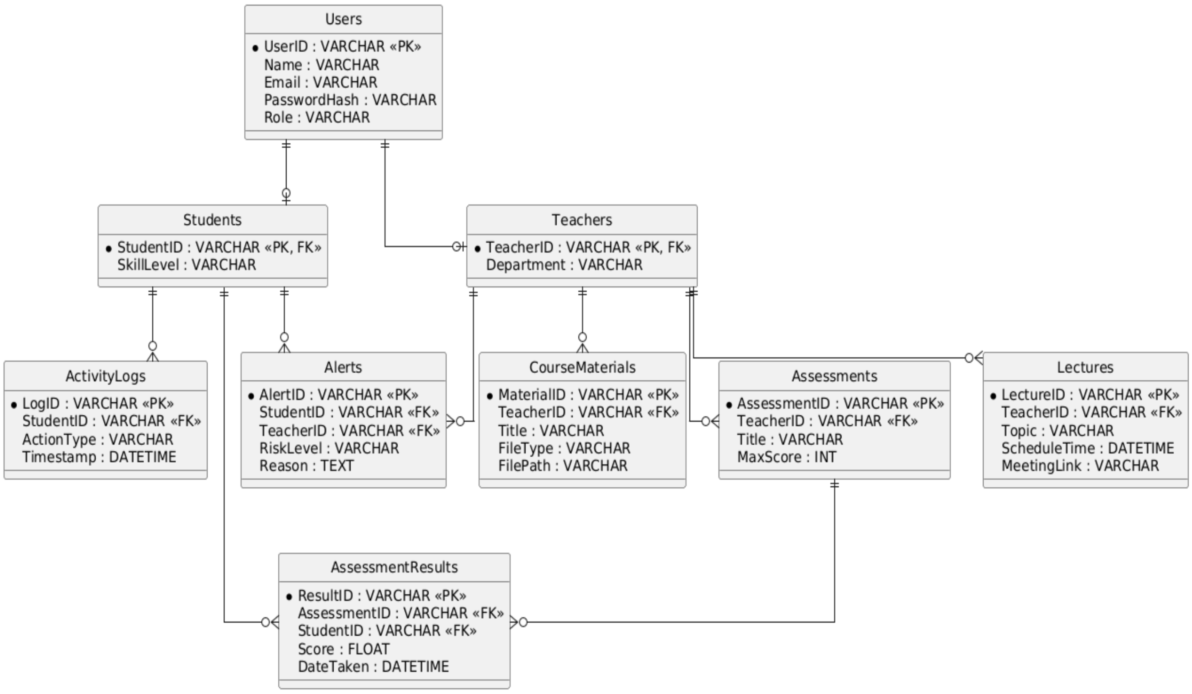

Fig 5.0 Entity-Relationship Diagram

Relational Schema Details:

The database utilizes the following tables in managing information:

i. Users Table:

Column Name	Data Type	Constraints	Description
UserID	VARCHAR (20)	Primary Key	Unique identifier for the user.
Name	VARCHAR (40)		Full name of the user.
Email	VARCHAR (40)		Email address of the user.
PasswordHash	VARCHAR (30)		Encrypted password.
Role	VARCHAR (20)		Defines if the user is a Student, Teacher, etc.

ii. Students Table:

Column Name	Data Type	Constraints	Description

StudentID	VARCHAR (20)	Primary Key, Foreign Key	References Users.UserID.
SkillLevel	VARCHAR (20)		The student's current skill or proficiency level.

iii. Teachers Table:

Column Name	Data Type	Constraints	Description
TeacherID	VARCHAR (20)	Primary Key, Foreign Key	References Users.UserID.
Department	VARCHAR (20)		The department the teacher belongs to.

iv. ActivityLogs:

Column Name	Data Type	Constraints	Description
LogID	VARCHAR (20)	Primary Key	Unique identifier for the log entry.
StudentID	VARCHAR (20)	Foreign Key	References Students.StudentID.
ActionType	VARCHAR (20)		The type of action performed (e.g., login, view).
Timestamp	DATETIME		The date and time the action occurred.

v. Alerts:

Column Name	Data Type	Constraints	Description
AlertID	VARCHAR (20)	Primary Key	Unique identifier for the alert.
StudentID	VARCHAR (20)	Foreign Key	References Students.StudentID.
TeacherID	VARCHAR (20)	Foreign Key	References Teachers.TeacherID.
RiskLevel	VARCHAR (6)		Severity of the alert (e.g., Low, Medium, High).
Reason	TEXT (60)		Detailed explanation for the alert.

vi. CourseMaterials:

Column Name	Data Type	Constraints	Description
MaterialID	VARCHAR (20)	Primary Key	Unique identifier for the material.
TeacherID	VARCHAR (20)	Foreign Key	References Teachers.TeacherID.
Title	VARCHAR (20)		Title of the course material.
FileType	VARCHAR (20)		Type of the file (e.g., PDF, DOCX).

FilePath	VARCHAR (20)		Storage path or URL for the file.

vii. Assessments:

Column Name	Data Type	Constraints	Description
AssessmentID	VARCHAR (20)	Primary Key	Unique identifier for the assessment.
TeacherID	VARCHAR (20)	Foreign Key	References Teachers.TeacherID.
Title	VARCHAR (40)		Title or name of the assessment.
MaxScore	INT		The maximum possible score.

viii. Lectures:

Column Name	Data Type	Constraints	Description
LectureID	VARCHAR (20)	Primary Key	Unique identifier for the lecture.
TeacherID	VARCHAR (20)	Foreign Key	References Teachers.TeacherID.
Topic	VARCHAR (40)		Topic of the lecture.
ScheduleTime	DATETIME		Scheduled date and time for the lecture.
MeetingLink	VARCHAR (40)		URL for the virtual meeting or location details.

ix. AssessmentResults:

Column Name	Data Type	Constraints	Description
ResultID	VARCHAR (20)	Primary Key	Unique identifier for the result record.
AssessmentID	VARCHAR (20)	Foreign Key	References Assessments.AssessmentID.
StudentID	VARCHAR (20)	Foreign Key	References Students.StudentID.
Score	FLOAT		The score achieved by the student.
DateTaken	DATETIME		The date and time the assessment was completed.

Table 2.0 Database design tables

Database Normalization Process: In order to ensure that data is logically stored without duplication, the technique of database normalization was implemented. The following process illustrates the way in which the student assessment data was organized from its initial disorganized state into the optimum state.

Unnormalized Form (UNF): Initially, all data for a student's test result is grouped in one single record: StudentID, StudentName, SkillLevel, AssessmentID, AssessmentTitle, MaxScore, TeacherID, TeacherDepartment, Score, DateTaken.

First Normal Form (1NF): A table is in 1NF if every column holds only one piece of data and the table has a primary key. A single row is uniquely identified by combining the StudentID and the AssessmentID.

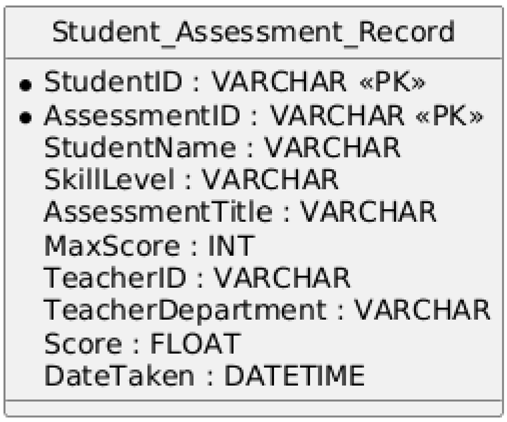

Fig 5.1 First Normal Form

Second Normal Form (2NF): A table is in 2NF if it is in 1NF and has no partial dependencies. Columns that only depend on the StudentID are moved to a students table. Columns that only depend on the AssessmentID are moved to an Assessments table. The Score and DateTaken depend on both, so they stay in a Results table.

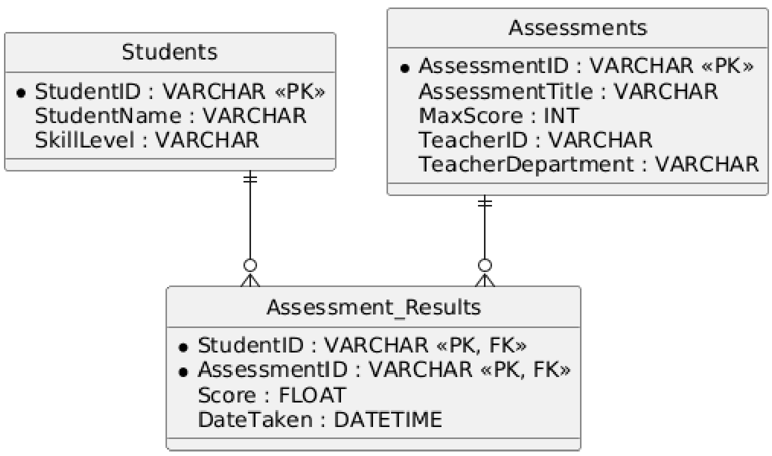

Fig 5.2 Second Normal Form

Third Normal Form (3NF): A table is in 3NF if it is in 2NF and has no transitive dependencies. A non-key column cannot depend on another non-key column. TeacherDepartment depends on TeacherID, not AssessmentID. A new Teachers table is created.

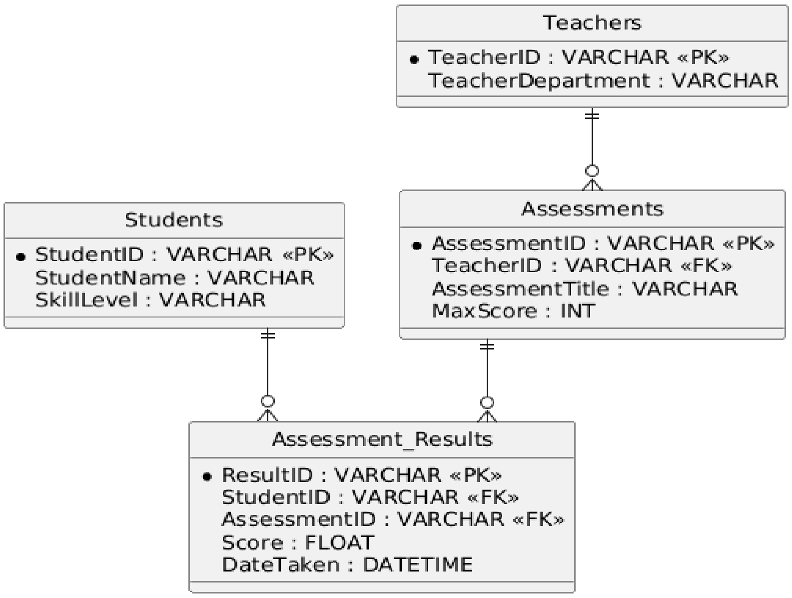

Fig 5.3 Third Normal Form

Data Storage Structure and Access Method

This data will be held in a Relational Database Management System where PostgreSQL will be used in this particular case. In this system, data will be arranged into rigid tables using primary and foreign keys. This ensures that there is no duplication of the data and that the data stays consistent.

The app will access this data through an Object-Relational Mapper such as SQLAlchemy on a Flask backend. This enables the app to retrieve and modify records in the database using Python code instead of string SQL. This makes data extraction fast, keeps the database safe from code injections and makes it very easy to pass large sets of data to the predictive machine learning model.

3.5.4 System Architecture Design

Three Tiered Client/Server Information System Architecture has been suggested for the Intelligent Academic Support and Progress Tracking Information System. It entails breaking down the system into three distinct tiers that will coordinate to provide responses to the user queries, process and store information.

High level architecture is as follows:

1. Presentation Layer (Front End) This is the layer where the interaction between the students and teachers takes place through the use of a web browser or mobile application. It displays the dashboards, collects information from the users and displays the results. It communicates with the backend through Application Programming Interfaces (APIs).

2. Application Layer (Back End) This is the "brain" of the system. It holds the logic of the whole system and does all the hard work. This layer receives requests from the user, accesses the database and implements the machine learning models.

i. **Web Server:** Written in Python programming language using the Flask framework, this server manages the user login, file upload and lecture scheduling.
ii. **Predictive Model Engine:** This module implements the machine learning algorithm which is used to find out which students are at risk of failing using their activity logs.
iii. **External Service Integration:** This layer securely connects to the external service, more specifically to the pre-trained Gemini AI model for implementing the digital tutor.

3. Data Layer (Database) This layer ensures the storage and retrieval of all system data securely. The relational database keeps track of users, course material, assessment score and the logs of the system activity. The back end application accesses the data through the Object-Relational Mapper (ORM).

System Architecture Diagram

The following figure represents a diagram of the connection and communication between the three layers and the external services.

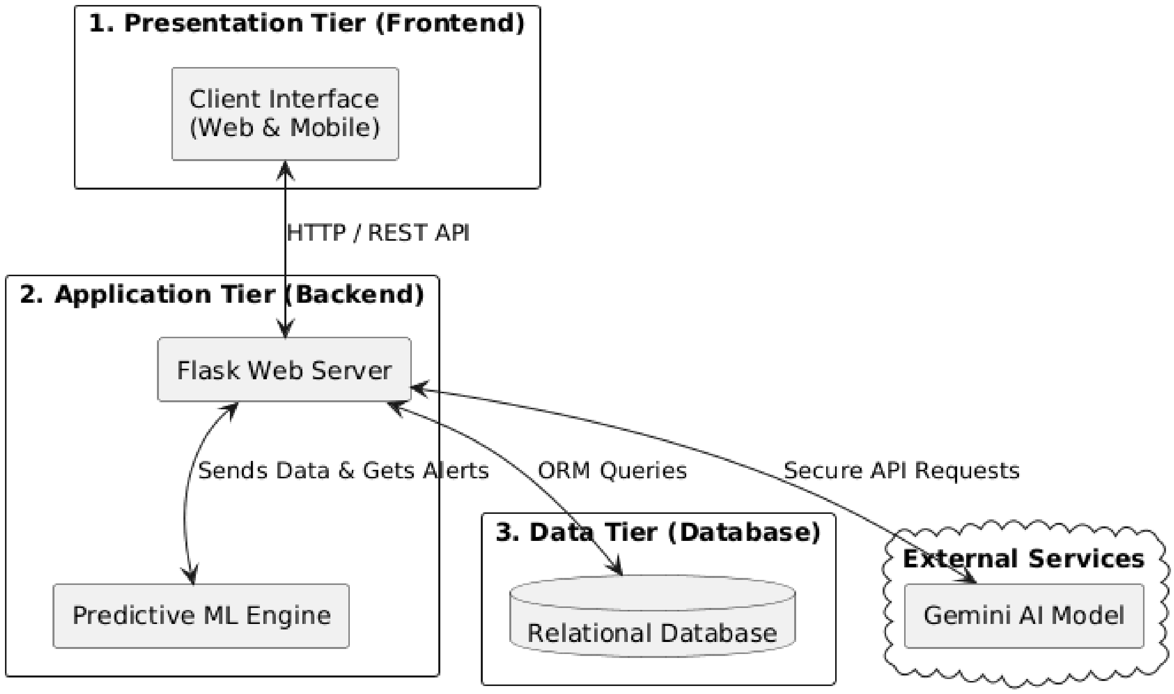

Fig. 6.0 System Architecture Diagram

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.0 Introduction

This chapter deals with the construction of the intelligent academic assistance and monitoring system. This chapter will provide information about the software and hardware utilized, the programming languages that were used, and the data processing and training of the predictive model. In addition to this, the testing phase and the results achieved from the system performance will also be discussed in this chapter.

4.1 Development Environment and Tools

Creating such a system involved the use of frontend and backend frameworks, database solutions, and AI libraries that could help make the design work.

4.1.1 Development Platform and Hardware Requirements

i. **IDE:** The project was done using Visual Studio Code (VS Code) throughout as it was an IDE with several good attributes. It helped us develop, debug, and organize the backend code written in Python and the frontend code written in Dart/Flutter.

ii. **Hardware:** The platform was designed using a machine having at least 8GB of RAM and a powerful multi-core processor. Such a hardware configuration was required to allow comfortable development, running of the local server, doing the database work, and testing the mobile application simultaneously.

4.1.2 Programming Languages and Frameworks

i. **Frontend:** Our frontend design uses the Dart language and the Flutter platform to create an effective and responsive user interface for our application.

ii. **Backend:** We have chosen the Python language as our primary backend technology since it supports integration of artificial intelligence and machine learning.

4.1.3 Backend Libraries and Dependencies

In order to achieve efficient backend operation, the project uses a selected set of Python libraries, classified based on their functionality as follows:

i. **Core Server and Security:** Flask was used for creating the web server, while the task of enabling effective interaction between the frontend and backend was carried out using Flask-CORS. For the sake of security, Flask-Bcrypt was used to protect and encrypt user passwords; Flask-JWT-Extended for creating tokens during the login process; python-dotenv for securing our hidden environment variables.

ii. **Database Management:** Flask-SQLAlchemy was used to facilitate our database management process, together with Flask-Migrate for managing database structure migrations. psycopg2-binary adapter was used to ensure a solid connection with our relational database.

iii. **Machine Learning and Data Processing:** For the purposes of training the early warning system and for manipulating the data, we have utilized libraries such as scikit-learn, pandas, and numpy. When the training phase of the model had ended, joblib was used for saving the model and data scalers so they can be easily reused later on.

iv. **Artificial Intelligence Integration:** The digital AI tutor uses google-generativeai library in order to be able to interpret and answer students' questions. In order to facilitate its ability to understand and process the information contained in the uploaded course materials, we have used libraries such as langchain for document handling, sentence-transformers for interpreting the meaning of texts, chromadb for storing and searching information. Finally, requests were used to handle all internet-based communications.

v.

4.2 Dataset Description and Preprocessing

To train our early warning system on identifying students who are under performing, we made use of a real-life dataset available from Kaggle called Student_Engagement_Dataset.csv.

i. **Overview of the Dataset:** Our dataset includes 200 records of the engagement level of the students including data such as the number of hours they studied, number of assignments submitted, and average quiz score.
ii. **Preprocessing of the Data:** In the given dataset, all students were identified as "Inactive" in the labels which is not useful for training the model. Therefore, we have assigned a label risk_level. Students whose average quiz score is less than 60 are identified as high risk and the others are low risk. Then we have standardized our measurements by using a standard scaler.
iii. **Splitting of the Data**: We split the data set in two parts; 80% of the data is used for training the model and the rest 20% is kept aside for future validation.

4.3 Model Architecture and Training

A machine learning classifier known as the Random Forest Classifier was applied as the brain of our Early Warning System.

i. **Architecture:** The choice to apply the Random Forest algorithm was made due to its outstanding capacity to process structured data and innate resistance to wrong conclusions drawn by the model.
ii. **Training:** Training of the classifier was carried out with the help of 100 separate decision trees. The process involved an analysis of structured data about student performance and missing homework. Once the training was complete, the final version of the model and data scaler were saved in .pkl format for quick loading by the server.

4.4 System Implementation and Modules

The system was divided into three major parts for efficiency:

i. **Frontend (Presentation):** This is an interactive mobile app that was developed using Flutter technology. It consists of the Student Dashboard where one can view their classes and communicate with the AI tutor, and the Teacher Dashboard where they can view class lists and get notifications on those students who struggle.

ii. **Backend (Application):** The Flask server works as a coordinator of all operations within our system. It takes care of user logins, the Random Forest model and the interaction between students' chat messages and the Gemini AI.

iii. **Database (Data Module):** iii. Database (Data Module): Our relational database is secure and stores all users' accounts, course documents, and students' test results.

4.5 Testing and Evaluation

The whole system was extensively tested to ensure that all components functioned properly and users had an excellent experience.

4.5.1 Evaluation Metrics and Model Performance

To measure the performance of our model, we conducted testing to assess the efficiency of the identification of students at risk of failure. The outcome was quite impressive:

i. **Accuracy (90.0%):** The system could distinguish between high-risk and low-risk students 90% of the time, indicating high reliability of the system.

ii. **Precision (89.3%):** When the system warned teachers about "high risk" for the certain student, it was correct 89.3% of the time. Thus, teachers would have to face few alerts which would not be real.

iii. **Recall (78.1%):** The model managed to detect 78.1% of those who needed help. Though the result is rather high, there is still a possibility for about 22% of students at risk to remain unnoticed and teachers should be attentive.

iv. **F1 Score (83.3%):** This result is a perfect compromise between precision and recall and thus demonstrates that the model is useful and not sensitive.

Breakdown of Confusion Matrix: We have made use of matplotlib, which is a visualization tool, to show the performance of the model on the 200 test records in a diagrammatic format as shown in Figure 7.0 below:

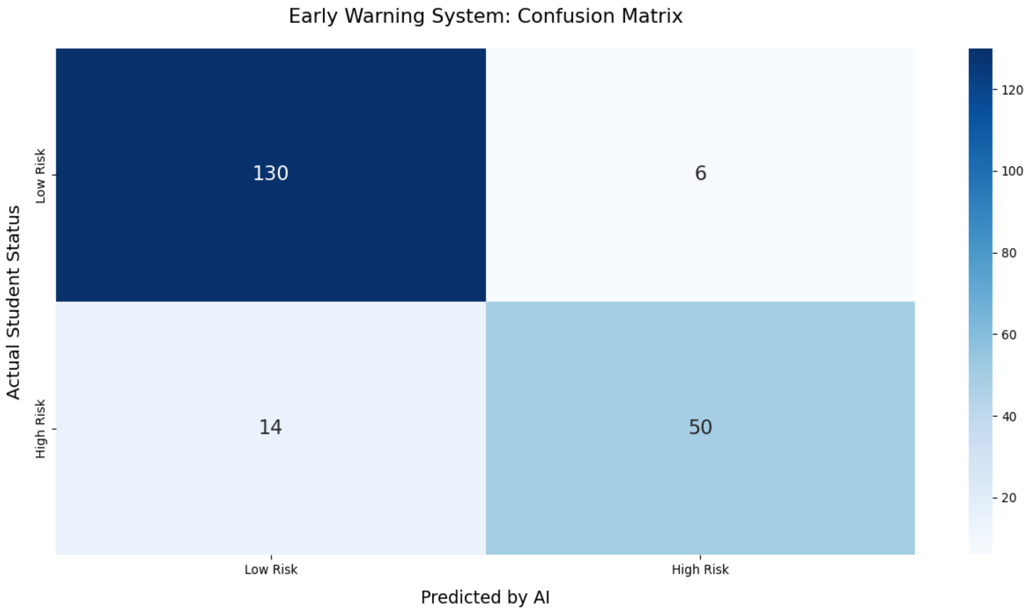

Fig. 7.0 Confusion Matrix of the Early Warning System

i. **130 True Negatives:** These are good-performing students who have been categorized as such.

ii. **50 True Positives:** These are poor-performing students who have been flagged as risky.

iii. **6 False Positives:** Good-performing students who have been flagged as risky, thus making them false positives.

iv. **14 False Negatives:** Poor-performing students who have not been detected by the model.

The above statistics clearly indicate that our early warning system is excellent in warning teachers about at-risk students.

4.5.2 System testing

Integration testing was performed to check the seamless communication between the Flutter mobile app and the Python backend. Moreover, we tested the early warning system by feeding it the mock study periods to verify its output for risk labels. Lastly, we did AI Tutor testing to confirm that it uses the uploaded course material prior to answering the questions.

4.5.3 User interface testing and walkthrough

We checked whether the user interface elements work flawlessly in our app. To be more precise, we performed the test on the teacher's risk alert cards in order to check if they change the color to red once there is a student at high risk. Also, we tested the chat screen to check that student messages and AI responses are located on the right sides.

4.6 Results Presentation and Analysis

4.6.1 Sample Output and Interface Screens

i. Digital Classroom and AI Tutor: Here, students enjoy a sleek, user-friendly chat box where they can pose any question and receive prompt and meaningful responses from the AI. This figure is illustrated below, depicting a student being assisted with her physics assignment.

ii. Teacher Alerts and Overviews: There is an independent dashboard for the teacher where all his/her students appear altogether. Figure 9.0 depicts the same page, marking each student as Low, Medium, and High Risk depending on the scores.

iii. Practice Assessments: It provides facility for making practice tests. In Figure 10.0, you will find a student examining herself through AI generated multiple choice practice test.

iv. Personalized Study Plans: Students can benefit from personalized study plans in order to get back on track. Figure 11.0 illustrates an AI made schedule telling the student which readings to make and which assignments to submit.

v. Live Class Management: In Figure 12.0, we have demonstrated how teachers control live classes, showing who is participating in the class and providing a link to the ongoing class.

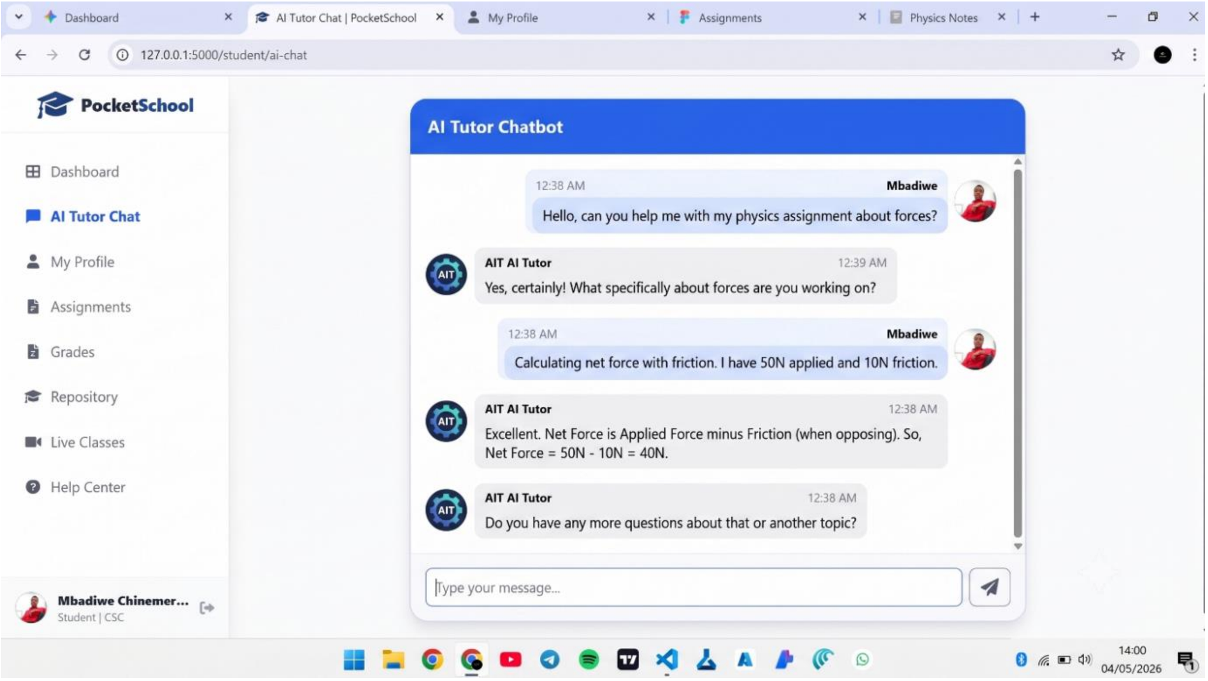

Fig. 8.0 Chat Screen Showing Conversation Between the AI Tutor and a Student

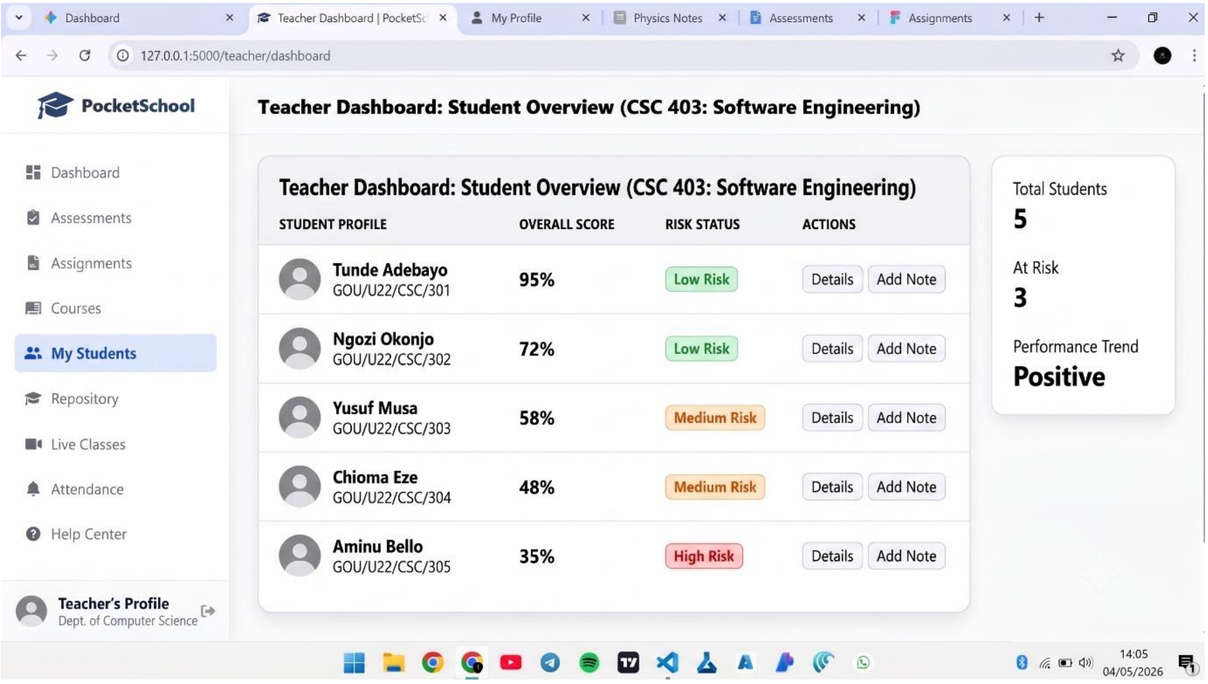

Fig. 9.0 Teacher Dashboard Showing Student Risk Status

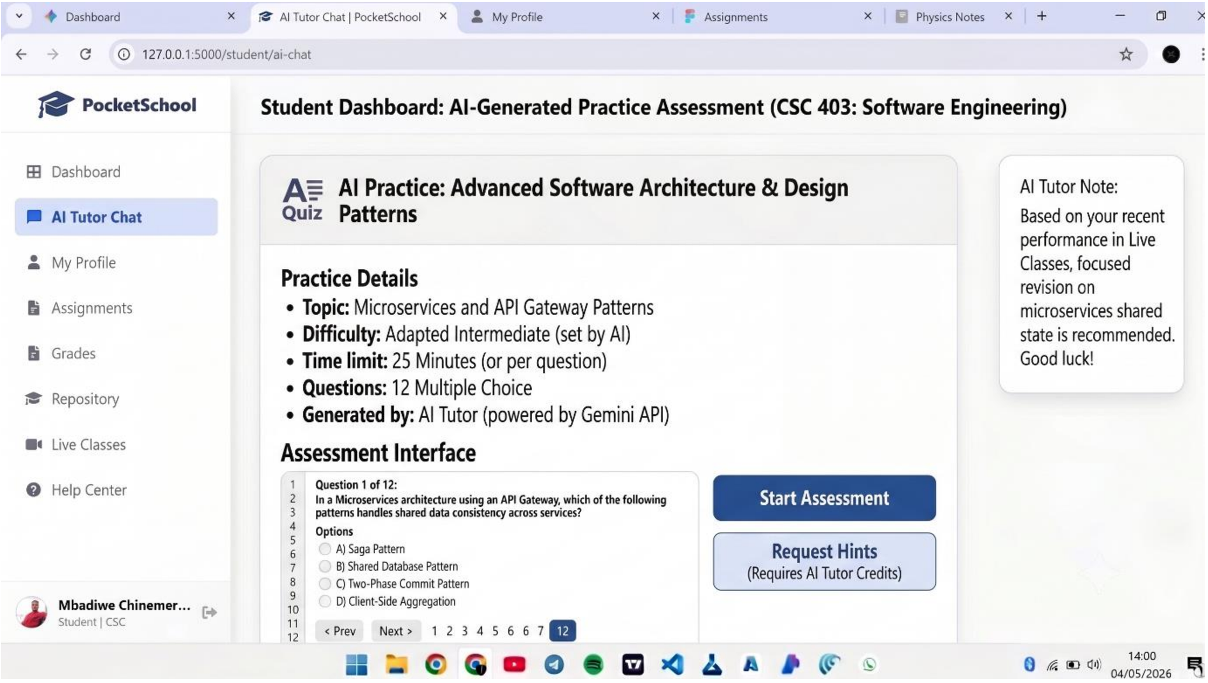

Fig. 10.0 Student Dashboard Showing an AI generated Practice Assessment

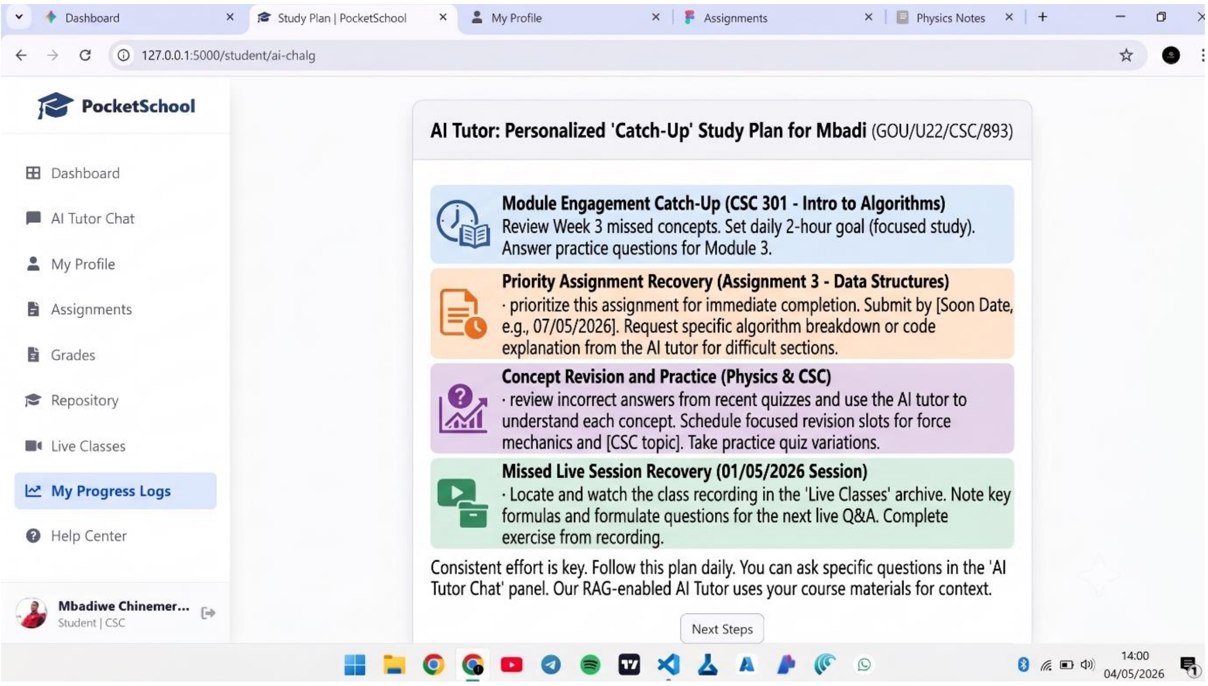

Fig. 11.0 Student Dashboard Showing AI generated Study Plan

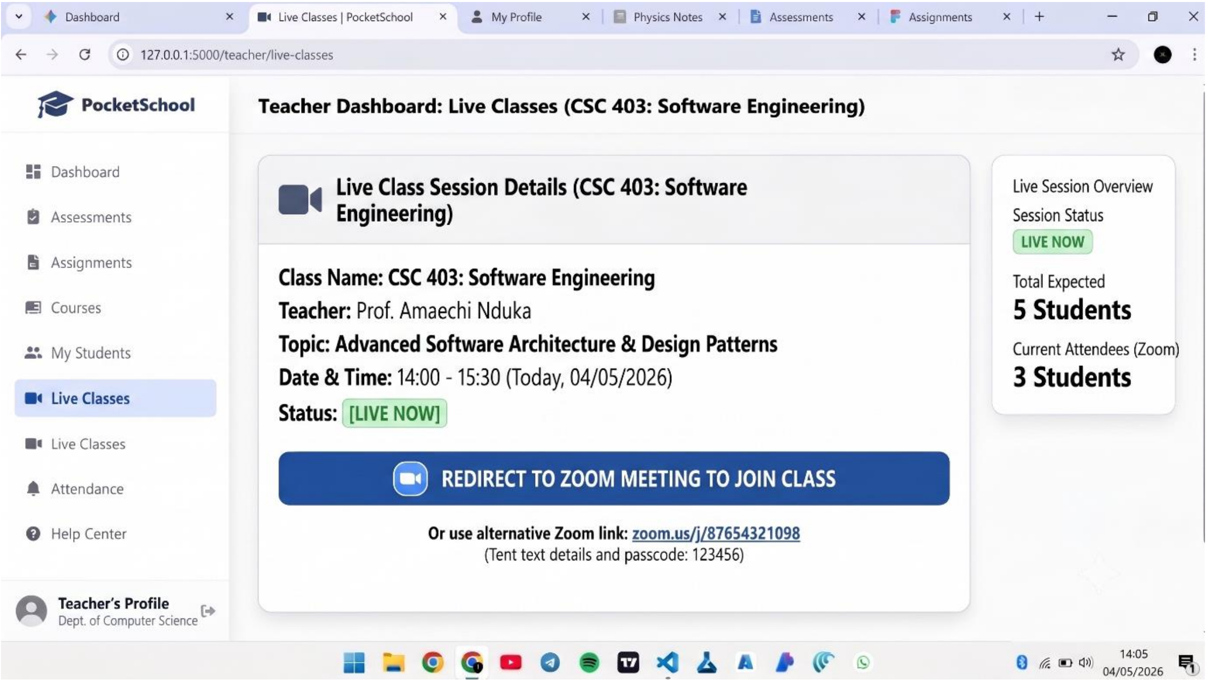

Fig. 12.0 Teacher's Dashboard Showing a Live Class Link

Images presented in the report below demonstrate how the process of interaction of a student with the AI tutor, personalized learning resources, as well as alerts from the teacher dashboard looks like when a particular student has been identified. The user interface was created in such a way as to provide information via visual means and to be simple enough for usage by both students and teachers.

4.6.2 Discussion of Results and System Performance

Integration of the Random Forest model into the system resulted in an extremely efficient tool for identifying struggling students. Along with this, integration of the Gemini 1.5 Flash model made our digital tutor not only incredibly fast but also very accurate. Thus, this project managed to merge data tracking and artificial intelligence into one proactive and round-the-clock system.

CHAPTER FIVE
SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter will involve revisiting the activities carried out in this project and the analysis of how our objective was achieved. It will involve a review of the entire research, the conclusion made on the findings and recommendations for further research works and practical implementation of the system in the real world.

5.2 Summary of the Study

It was established that institutions with poor resource bases face challenges with late feedback, and thus we developed an innovative student progress monitoring tool. The key objectives of the development included providing personal assistance all throughout the day, providing rapid responses to inquiries, progress tracking, and detecting struggling students at very early stages. A literature review helped to establish that there is no existing solution which provides early warning signs along with an extremely accurate artificial intelligence digital student tutor. For software development purposes, Agile approach was adopted while for developing the predictive machine learning risk model, CRISP-DM framework was employed. A design that comprised of three components depicted interaction between students, educators, and intelligent system with the help of UML. The frontend of the system was developed using Flutter while the backend with Python Flask and Random Forest model was used for predicting risks in academia. Google Gemini AI was integrated for tutoring students and through testing process, we were able to establish that the proposed solution is effective for detecting very accurate responses of struggling learners.

5.3 Conclusion

i. The primary objective of this project was the development of an intelligent Learning Management System to monitor progress and provide individual tutoring to students. The system designed met this objective in full measure by providing an AI tutor that gave accurate responses to difficult course content questions.

ii. The predictive model successfully checked student actions to find struggling learners early, which allows human teachers to offer extra needed help.
iii. This platform changes the school experience into a fast action, helping students get early guidance long before they fail their classes.
iv. The model learned from only two hundred student records, so researchers must test it with larger datasets to prove value.
v. The system needs fast internet to work perfectly, which is a big problem where computer networks are totally weak or missing.

5.4 Recommendations

i. Such a system will be used in schools to reduce the heavy burden on human teachers and discourage students from leaving school.
ii. In future, the team will have to apply the early warning model on larger data sets in order to ensure that the students remain engaged with their education.
iii. Another aspect that has to be considered while developing future versions of the system is its ability to function without fast Internet access.
iv. It should also contain additional information about real-life settings and social skills development for better support of the students' learning styles.

References

Aldino, A.A., et al. (2025). Analytics of Learner-Centered Feedback: A Large-Scale Case Study in Higher Education. Journal of Computers and Education 237 (2025) 105360.

Assiri, A., AL-Ghamdi, A. A., & Brdesee, H. (2020). From Traditional to Intelligent Academic Advising: A Systematic Literature Review of e-Academic Advising. International Journal of Advanced Computer Science and Applications (IJACSA).

Balogun, M., et al. (2024). Deep Learning-Based Early Warning Systems for Identifying At-Risk Students in Higher Education: A Socio-Technical Framework for Precision and Intervention. Research Gate, presented at University of Ibadan.

Duan, C., Shi, Q., Chen, X., & Lin, H. (2024). Research on the Personalized Learning Support Function of Intelligent Tutoring System in Higher Education Teaching. ICIEAI 2024, Kaifeng, China.

Henriquez-Nunez, Y., et al. (2023). Adaptive Model for the Selection of Resources and Activities in a Virtual Learning Environment. Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 1, pages 256-263.

IEEE. (n.d.). Definition of Intelligent System. IEEE Standard 1484.11.1-2022. (2022). Definition of Learning Management System (LMS).

Kurniawan, et al. (2021). Analyzing the Impact of AI Text Generators on Learning Styles, Technological Dependency, and Critical Thinking among Accounting Students IJIET 2025 Vol.15(11): 2465-2475.

Lampropoulos, G. (2025). Integration of Intelligent Tutoring Systems into Augmented Reality and Virtual Reality Environments. Applied Sciences, MDPI.

Li, H., & Liu, Z. (2025). An Intelligent Educational System: Analyzing Student Behavior and Academic Performance Using Multi-Source Data. Electronics, 14(16), 3328.

SAP. (n.d.). Definition of bias in AI. Retrieved from www.SAP.com.

Saufi, S. (2025). From Traditional Classrooms to Digital Learning: Exploring Research Trends in Contemporary E-Learning Challenges Education. SunText Review of Arts and Social Sciences 6(1) January 2025

Shpolianskaya, I., & Seredkina, T. (2020). Intelligent Support System for Personalized Online Learning. BRAIN. Broad Research in Artificial Intelligence and Neuroscience, 11(3), 29-35.

Sirohi, T., et al. (2025). From Static Platforms to Intelligent Assistants: Reviewing LMS Challenges and the Role of Retrieval Augmented Generation. 2025 International Conference on Future Technologies (ICFT).

Soomro, A. A., et al. (2025). AI-driven academic advising systems and student retention. The Critical Review of Social Sciences Studies.

Study.com. (n.d.). Definition of Academic Support Systems.

Sule, J. (2025). Design and Implementation of a Learning Management System. Bachelor of Science (BSc.), Nasarawa State University, Keffi.

tchop.io. (n.d.). Definition of Adaptive Feedback.

Wang, Y. (2025). Hybrid AI-based model for monitoring academic performance. Scientific Reports.

Zhan, Y., et al. (2025). Generative artificial intelligence as an enabler of student feedback engagement: a framework. Higher Education Research & Development 2025, VOL. 44, NO. 5, 1289–1304.

Appendix

1. Backend Prediction Route (Python and Flask)

This code shows how the backend server uses the trained Random Forest model to predict if a student is at risk of failing.

Python

```python
from flask import Flask, request, jsonify
import joblib
import numpy as np

app = Flask(__name__)
model = joblib.load('random_forest_model.pkl')
scaler = joblib.load('standard_scaler.pkl')

@app.route('/predict', methods=['POST'])
def predict_risk():
    data = request.get_json()

    features = [
        data['study_time_hours'],
        data['missed_assessments'],
        data['quiz_average']
    ]

    features_array = np.array(features).reshape(1, -1)
    scaled_features = scaler.transform(features_array)
    prediction = model.predict(scaled_features)

    risk_label = "High Risk" if prediction[0] == 1 else "Low Risk"

    return jsonify({'risk_level': risk_label})

if __name__ == '__main__':
    app.run(port=5000)
```

2. Frontend Teacher Alert Card (Dart and Flutter)

This code builds the visual card that shows the early warning alerts on the teacher's dashboard.

Dart

```dart
import 'package:flutter/material.dart';

class RiskAlertCard extends StatelessWidget {
  final String studentName;
  final String riskLevel;
  final String reason;

  const RiskAlertCard({
    Key? key,
    required this.studentName,
    required this.riskLevel,
    required this.reason,
  }) : super(key: key);
```

```
  @override
  Widget build(BuildContext context) {
    bool isHighRisk = riskLevel == 'High Risk';

    return Card(
      color: isHighRisk ? Colors.red.shade100 : Colors.green.shade100,
      margin: const EdgeInsets.all(8.0),
      child: ListTile(
        title: Text(
          studentName,
          style: const TextStyle(fontWeight: FontWeight.bold),
        ),
        subtitle: Text(reason),
        trailing: Text(
          riskLevel,
          style: TextStyle(
            color: isHighRisk ? Colors.red.shade900 :
Colors.green.shade900,
            fontWeight: FontWeight.bold,
          ),
        ),
      ),
    );
  }
}
```

3.Sample Dataset

This table shows a small part of the `Student_Engagement_Dataset.csv` downloaded from Kaggle. The machine learning model uses this data to learn student patterns.

student_id	study_time_hours	missed_assessments	quiz_average	risk_level
STU-001	45.5	0	85.0	0
STU-002	12.0	3	42.5	1
STU-003	30.2	1	65.0	0
STU-004	8.5	4	38.0	1

student_id	study_time_hours	missed_assessments	quiz_average	risk_level
STU-005	50.0	0	92.0	0

Table 3.0 Sample Dataset

4.System Requirements

To run this system, you need to install the following software and tools on your computer:

- **Code Editor:** Visual Studio Code.
- **Database:** PostgreSQL.
- **Frontend Environment:** Flutter SDK and Dart SDK.
- **Backend Environment:** Python.
- **Python Libraries:** Flask, Flask-SQLAlchemy, Flask-Bcrypt, Scikit-learn, Pandas, Numpy, Google-generativeai.
- **External APIs:** Google Gemini API Key for the AI tutor.