Capstone Project-II

Report

On

EduCrypt (Learning Management System)

Submitted By

Kunj Mehta (202202626010070) Ayushi Soni M (202202626010092)

Pooja Bagul (202302626020160) Het Hirani (202202626010059)

Under the Guidance of

Dr. Tejas Bhatt
Assistant Professor

Faculty of Engineering and Technology

GLS University

Academic Year

(2024-2025)

CERTIFICATE

This is to certify that the project report entitled EduCrypt (Learning Management System) has been satisfactorily carried out by the following students

Kunj Mehta	202202626010070
Pooja Bagul	202302626020160
Ayushi Soni M	202202626010092
Het Hirani	202202626010059

under my guidance in the fulfillment of the course Capstone Project-II (2601606) work during the academic year 2024-2025.

Dr. Tejas Bhatt
Internal Guide

(Dr. Madhuri Chopade)
Capstone Project-II Coordinator

Acknowledgment

We would like to express our deepest gratitude to everyone who contributed to the successful completion of our Capstone Project-II titled EduCrypt (Learning Management System).”

First and foremost, we are sincerely thankful to Dr. Tejas Bhatt, our internal guide, for his continuous support, expert guidance, and valuable feedback throughout the duration of this project. His insightful suggestions and encouragement helped us navigate challenges and enhance the quality of our work.

We also extend our heartfelt thanks to Dr. Madhuri Chopade, Dr. Harshal Arolkar sir and Dr. Savita Mam Capstone Project-II Coordinator, for providing us with the opportunity to work on this project and for her constant motivation and guidance.

We are grateful to the Faculty of Engineering and Technology, GLS University, for providing the platform and resources necessary to explore, innovate, and implement our ideas into a meaningful project.

A special thanks to our peers and teammates for their collaboration, brainstorming, and dedicated efforts that turned this concept into a functional system. Every team member played a vital role in the successful execution of the project.

Sincerely,

Kunj Mehta (202202626010070)

Pooja Bagul (202302626010160)

Ayushi Soni M (202202626010092)

Het Hirani (202202626010059)

Abstract

The shift toward digital education has created a strong demand for powerful and secure Learning Management Systems (LMS) that offer a seamless learning experience. EduCrypt is a web-based LMS built to meet this demand with a feature-rich platform designed for both students and teachers. It simplifies online learning by offering key academic tools, secure communication, and efficient content delivery—all within a visually appealing, user-friendly interface.

EduCrypt is developed using HTML, CSS, and JavaScript for the frontend, Node.js and Express.js for the backend, and MongoDB as the database. The system also leverages REST APIs for efficient data flow. Designed with modular architecture and responsive design principles, EduCrypt delivers an interactive and intuitive UI that enhances user engagement.

Key features include AI ChatBot integration for real-time academic assistance, a note-making system, an online compiler for programming practice, and encrypted file submission to ensure data privacy. It also includes essential LMS components like course materials, quizzes, announcements, and performance tracking.

The project was developed following agile methodology with iterative feedback and testing. EduCrypt aims to redefine the digital classroom experience by offering a smart, secure, and visually engaging learning platform that adapts to modern educational needs.

Content

List Of Tables

List of Figures

Chapter 1: Introduction

1.1 Project Detail

EduCrypt is a modern web-based Learning Management System (LMS) developed to redefine the way educational content is delivered and managed in academic institutions. In the wake of global shifts towards digital learning, especially following the COVID-19 pandemic, the need for secure, interactive, and centralized platforms has become more vital than ever.

EduCrypt addresses these needs by providing an intuitive, role-based web platform that connects students, faculty, and administrators through a seamless interface. It facilitates efficient management of educational content, quizzes, assignments, note-taking, and real-time feedback. The system's responsive UI, secure backend, and modular architecture ensure scalability and performance across devices and user bases. The primary vision is to offer an innovative digital environment that supports both teaching and learning while focusing on data integrity, accessibility, and user engagement.

1.2 Purpose

The purpose of EduCrypt is to provide a secure, feature-rich online platform that simplifies digital learning and academic communication. It bridges the gap between educators and students by integrating necessary learning tools, offering a streamlined experience for content access, interaction, and evaluation.

With the inclusion of AI chatbot support and real-time compiler integration, it ensures learners are supported beyond just content delivery. It also empowers educators to manage assessments, assignments, and learning materials more efficiently while ensuring user data is encrypted and protected.

1.3 Scope

EduCrypt is built for deployment across colleges, universities, and online institutions. It supports multiple user roles including students, teachers, and administrators. Students can access lecture materials, practice code in an embedded compiler, take notes, and submit assignments in encrypted format.

Teachers can manage multiple courses, post assignments and quizzes, track student progress, and provide feedback. Admins can monitor usage, manage roles, and maintain system integrity. The system is scalable and can be extended with features such as video conferencing, performance analytics, and multilingual support.

Its backend is designed to be REST API-based, which means EduCrypt can be integrated with third-party education tools in the future. The system supports role-based access and ensures that academic data remains secure and confidential.

1.4 Objective

- To build a fully functional, web-based LMS tailored for digital education
- To integrate secure and encrypted file submission mechanisms
- To enhance user engagement with AI chatbot and built-in compiler support
- To provide an intuitive and clean UI for seamless user experience
- To create scalable and customizable architecture using modern web technologies
- To support educators in tracking academic performance and managing course content effectively.

1.5 Literature Review

Learning Management Systems have evolved significantly in the last decade. Platforms like Moodle, Google Classroom, and Canvas have been widely adopted due to their collaborative features. However, these systems often require external tools or plugins for advanced functionality.

Research and user feedback suggest the growing demand for LMS platforms that offer native support for AI assistance, real-time coding environments, and secure submissions. EduCrypt is designed with these expectations in mind. It integrates tools directly into the ecosystem—minimizing reliance on third-party plugins and improving system efficiency.

The use of MongoDB as a flexible backend, combined with Node.js and Express.js for scalable logic processing, ensures high performance. Studies also emphasize the importance of responsive UI and accessibility, both of which are core to EduCrypt's design philosophy. With more institutions transitioning to hybrid or fully online education, the LMS landscape is shifting toward platforms that are adaptable, secure, and rich in user-centric features.

Chapter 2: About The System

2.1 System Requirement Specification (Functional and Non-Functional)

The success of any software system largely depends on its ability to meet user needs and expectations. For EduCrypt, a robust web-based Learning Management System, the functional and non-functional requirements were carefully identified to ensure it meets performance, usability, and security standards effectively.

Functional Requirements

● **User Authentication:** The system must support login and registration functionalities for students, teachers, and admin roles using secure credentials.

● **Role-based Dashboards:** Different user roles (Student, Teacher, Admin) must have access to unique dashboards offering functionalities specific to their responsibilities.

● **Course Management:** Teachers should be able to create, edit, and delete courses, upload learning materials, and manage assessments.

● **Assignment and Quiz Management:** Teachers can assign quizzes and tasks, students can submit their responses or files (with encryption support), and teachers can view, evaluate, and provide feedback.

● **Note-Making System:** Students should be able to write, save, and organize notes under subject-specific sections.

● **AI ChatBot Integration:** Students should have access to an AI chatbot for instant navigation help, and learning support.

● **Compiler Integration:** A web-based code compiler must allow students to write and execute code snippets for supported languages directly within the platform.

● **Encrypted File Submission:** Students must be able to submit assignments securely with encryption applied to uploaded files.

● **Admin Panel:** Admin users can monitor system-wide activities, manage user accounts, and view analytical reports regarding platform usage.

Non-Functional Requirements

- **Security:** The system must ensure data privacy and integrity by using encryption for file submission and implementing two-factor authentication for login.
- **Scalability:** The system should support expansion in terms of users, modules, and storage without performance degradation.
- **Performance:** The platform should load within 2–3 seconds and must support concurrent users efficiently.
- **Usability:** The system should have a responsive and intuitive UI that is accessible across all modern browsers and devices.
- **Availability:** The LMS must be available 24/7 with minimum downtime, ensuring continuous access to learning materials and services.
- **Maintainability:** The system should be built using modular and reusable code practices to ensure easier debugging and future enhancement.

2.2 Project Planning

To ensure efficient execution, the EduCrypt project followed the **Agile Development Methodology**, focusing on iterative development and frequent feedback incorporation. The project was planned and executed in the following phases:

Requirement Analysis Phase

- Identification of user personas (student, teacher, admin) and their use cases
- Brainstorming sessions for feature planning based on academic needs.

Design Phase

- Designing low-fidelity wireframes and high-fidelity UI mockups using Figma
- Structuring the system architecture and database design based on MERN stack (MongoDB, Express.js, Node.js)

Development Phase

- Implementation of frontend using HTML, CSS, and JavaScript for responsiveness and UI interactivity
- Backend logic and API development using Node.js and Express.js
- Database schema creation and integration with MongoDB

Testing Phase

● Unit testing of individual modules

● Integration testing for functionality validation

● UI/UX testing for responsiveness and user flow

Deployment and Evaluation Phase

● Hosting of the application on a server with necessary environment configuration

● Final evaluation by team members and guide, ensuring all planned modules work seamlessly

● Feedback gathering for improvements and future scalability

The entire team worked collaboratively with clear task distribution, frequent team meetings, and sprint-based tracking to meet project goals within the allocated academic timeline.

Chapter 3: Analysis The System

1.1) Use-Case Diagram

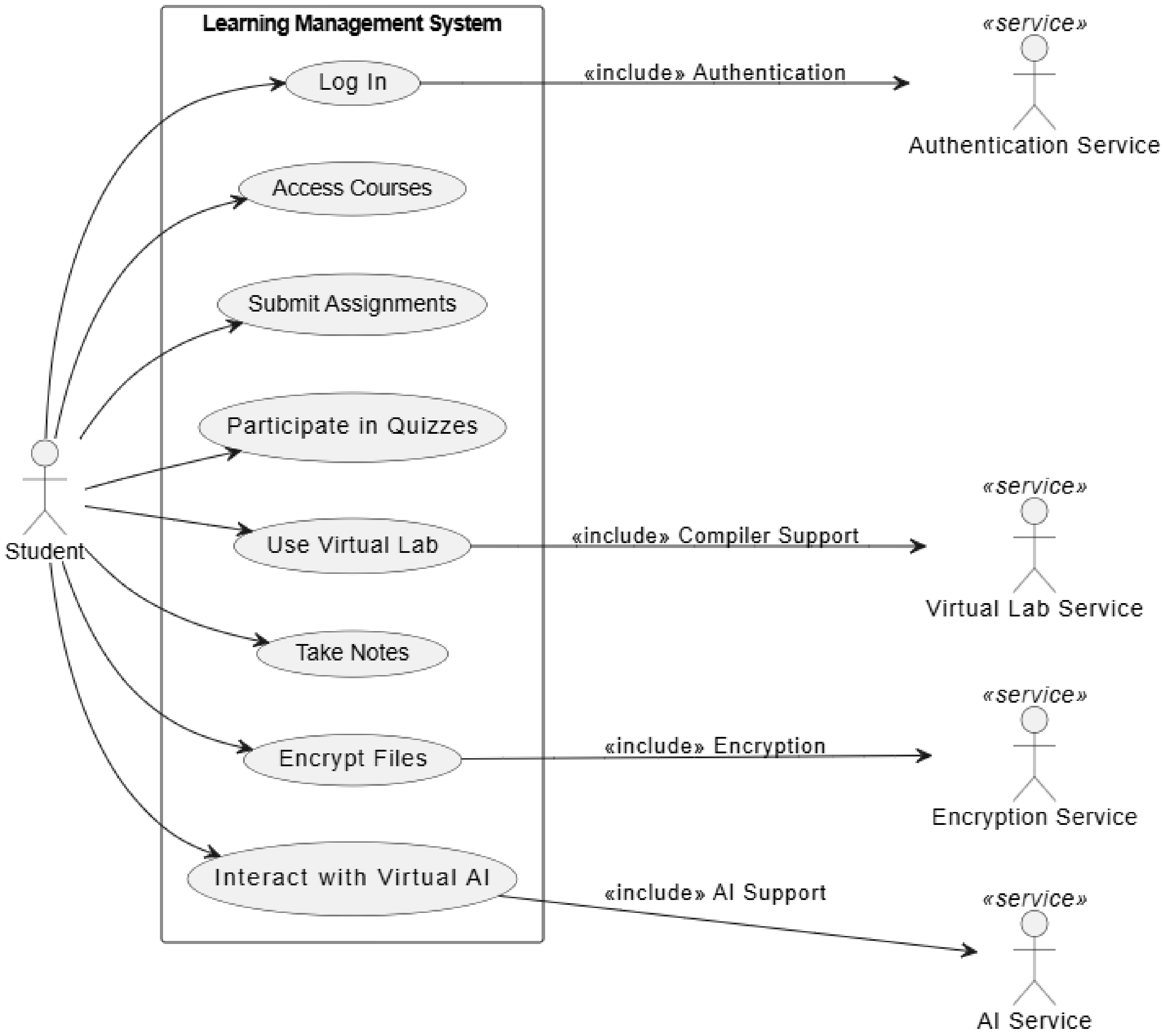


Fig 3.1 Use-Case Diagram for Teacher

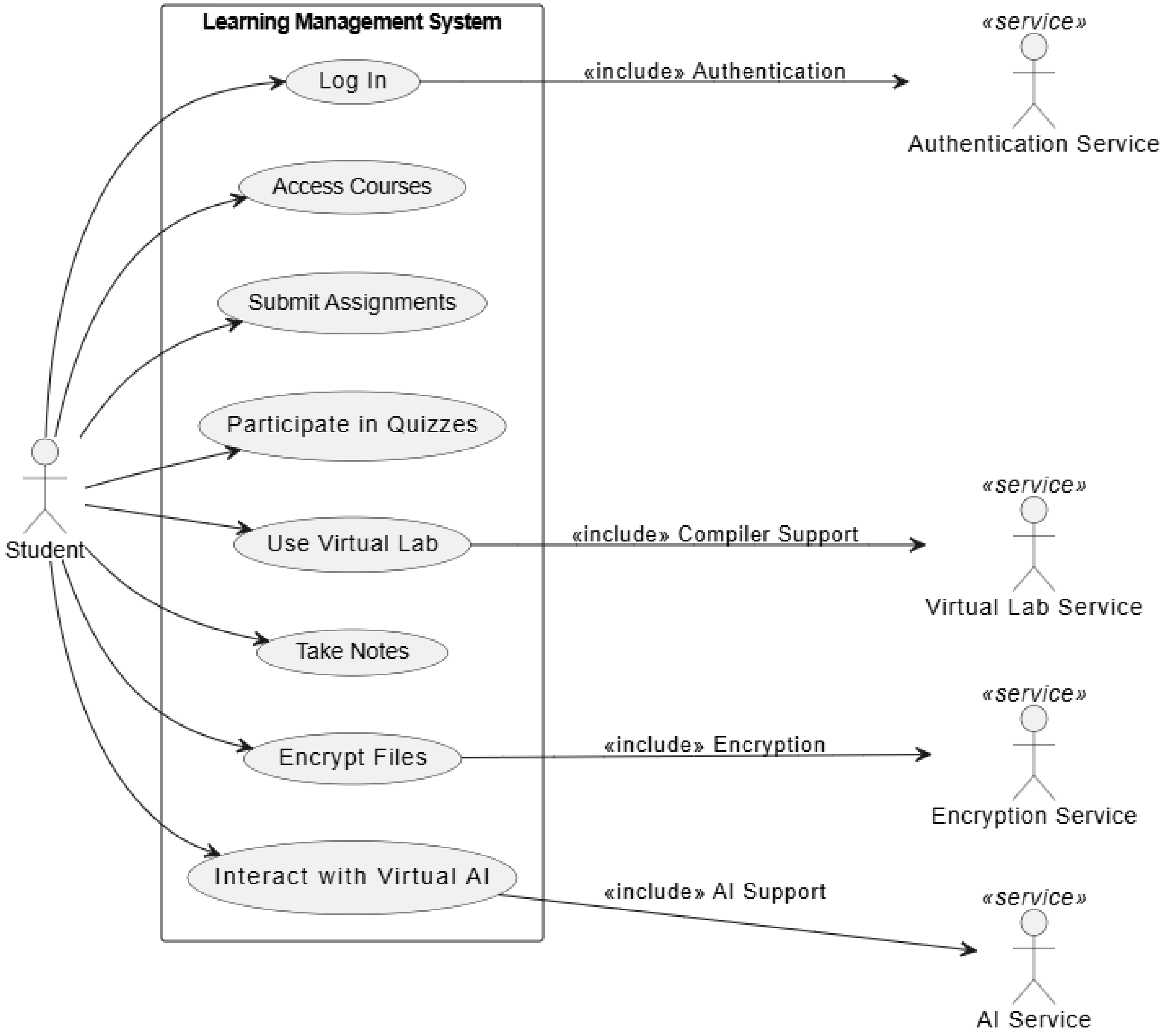


Fig 3.2 Use-Case Diagram for Student

1.2) Sequence Diagram

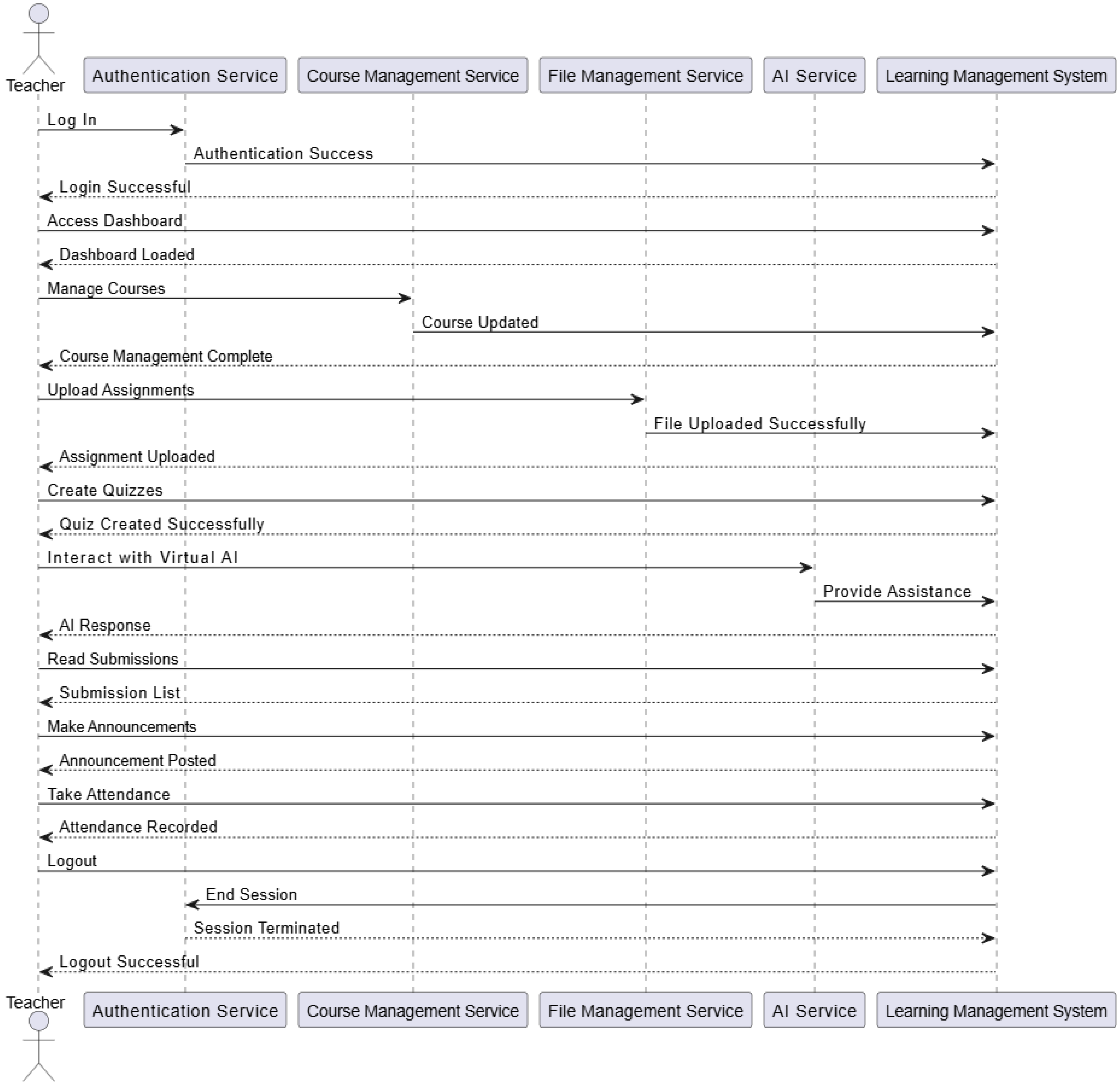


Fig-3 Sequence Diagram for Teacher

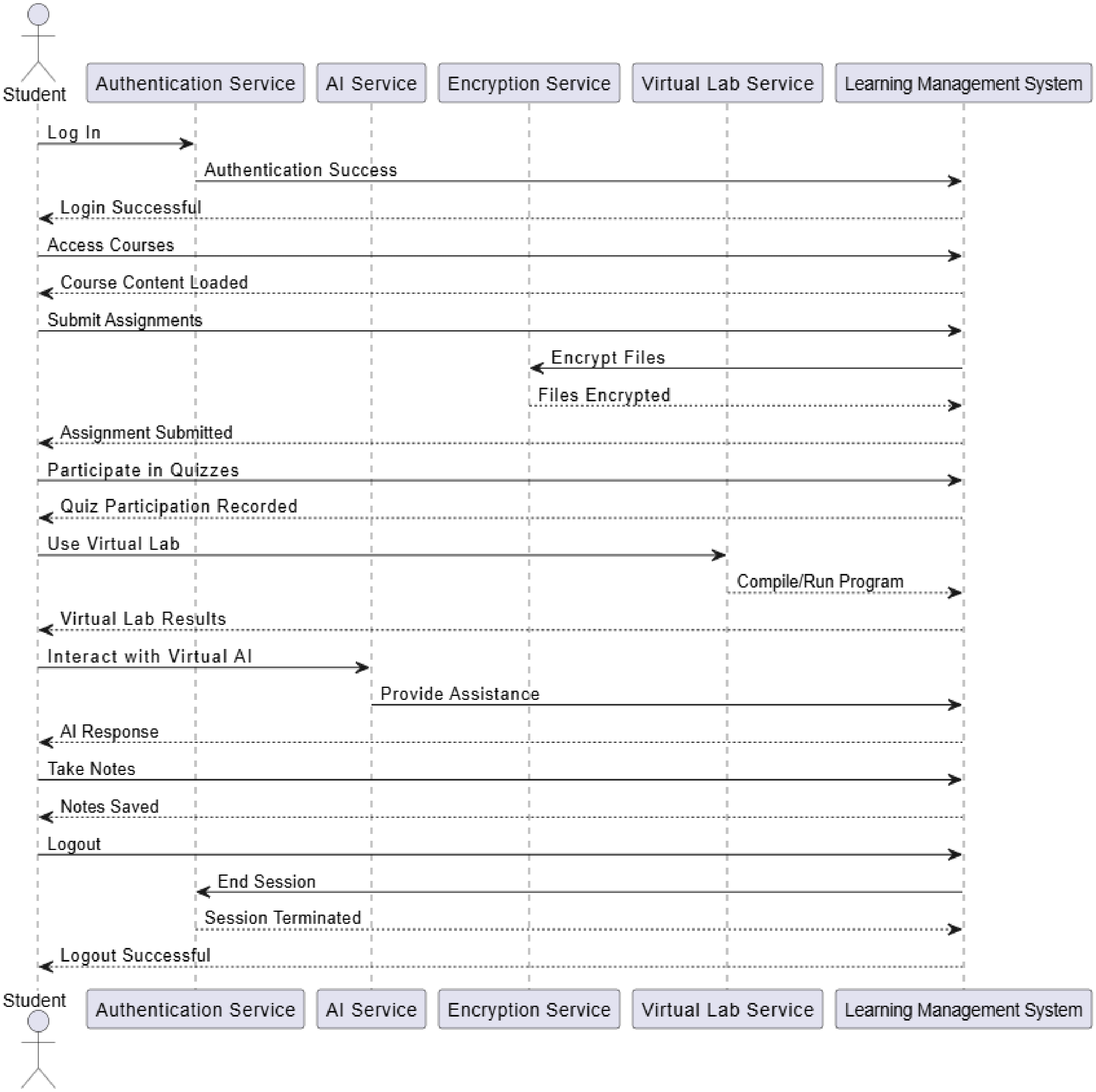


Fig 3.4 Sequence Diagram for Student

1.3) Activity Diagram

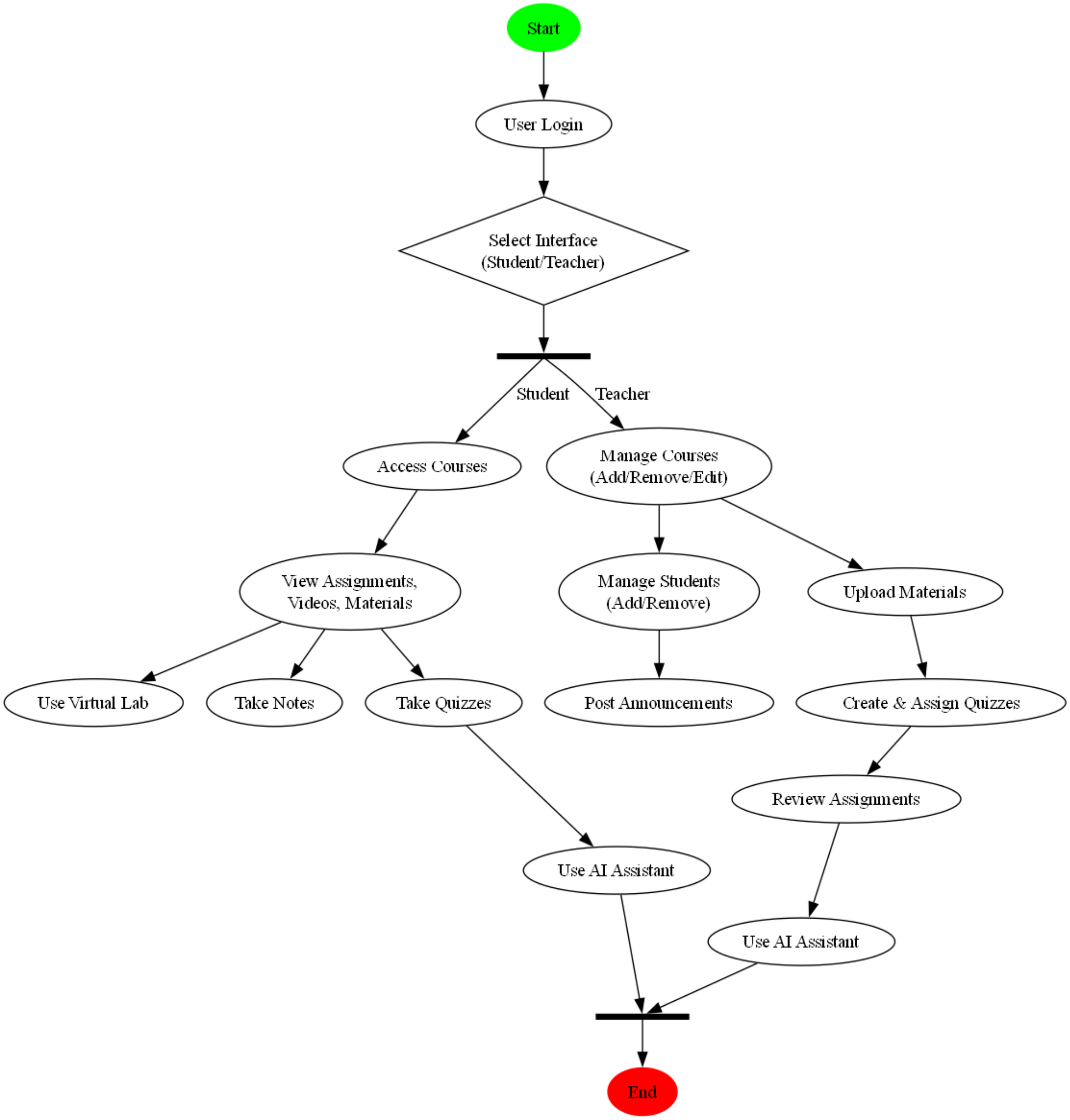


Fig 3.5 Activity Diagram

1.4) Data Flow Diagram

1.4.1) Data Flow Diagram (Level-0)

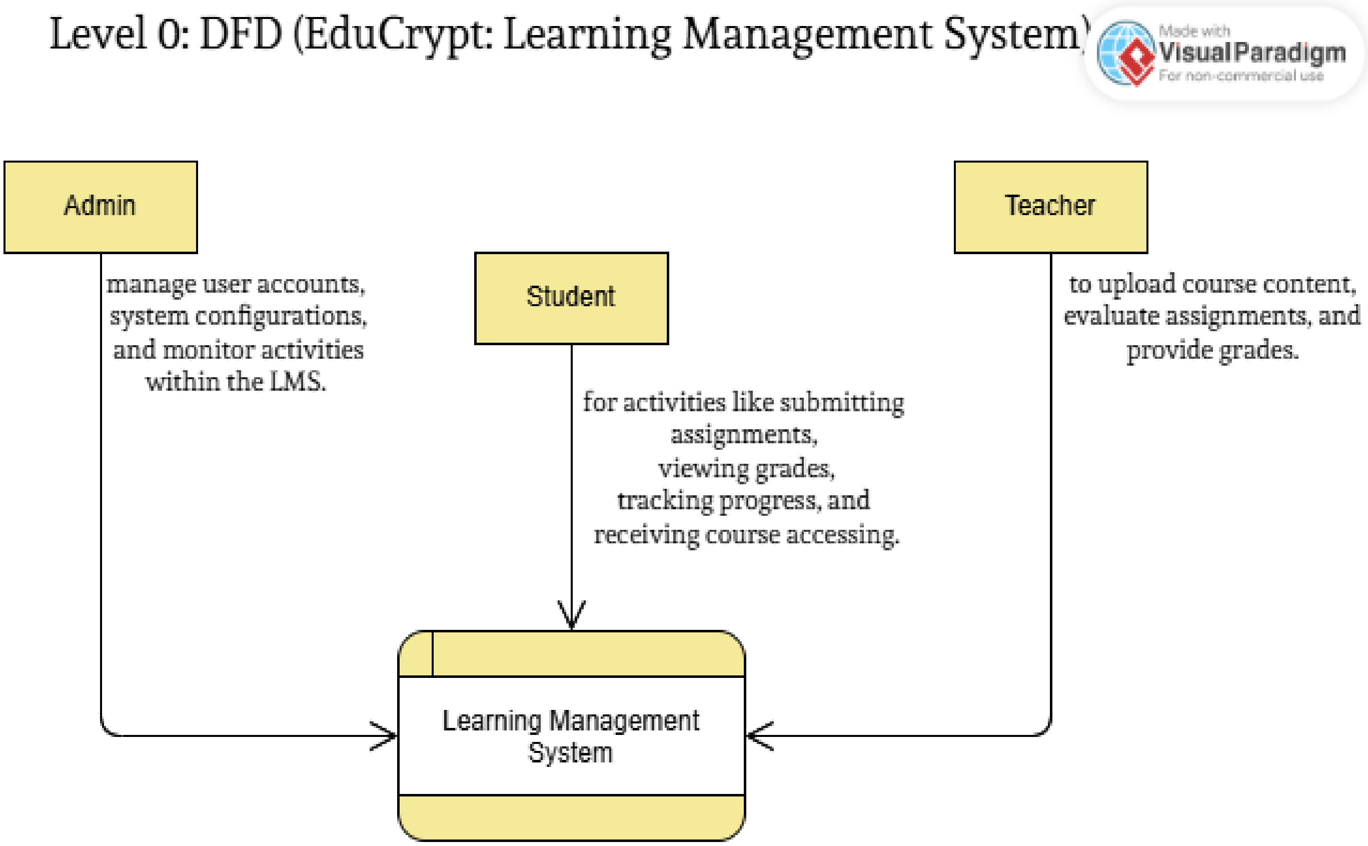

Level 0: DFD (EduCrypt: Learning Management System)
Made with Visual Paradigm For non-commercial use
Admin
manage user accounts, system configurations, and monitor activities within the LMS.
Student
for activities like submitting assignments, viewing grades, tracking progress, and receiving course accessing.
Teacher
to upload course content, evaluate assignments, and provide grades.
Learning Management System

Fig 3.6 Data Flow Diagram (Level-0)

1.4.2) Data Flow Diagram (Level-1)

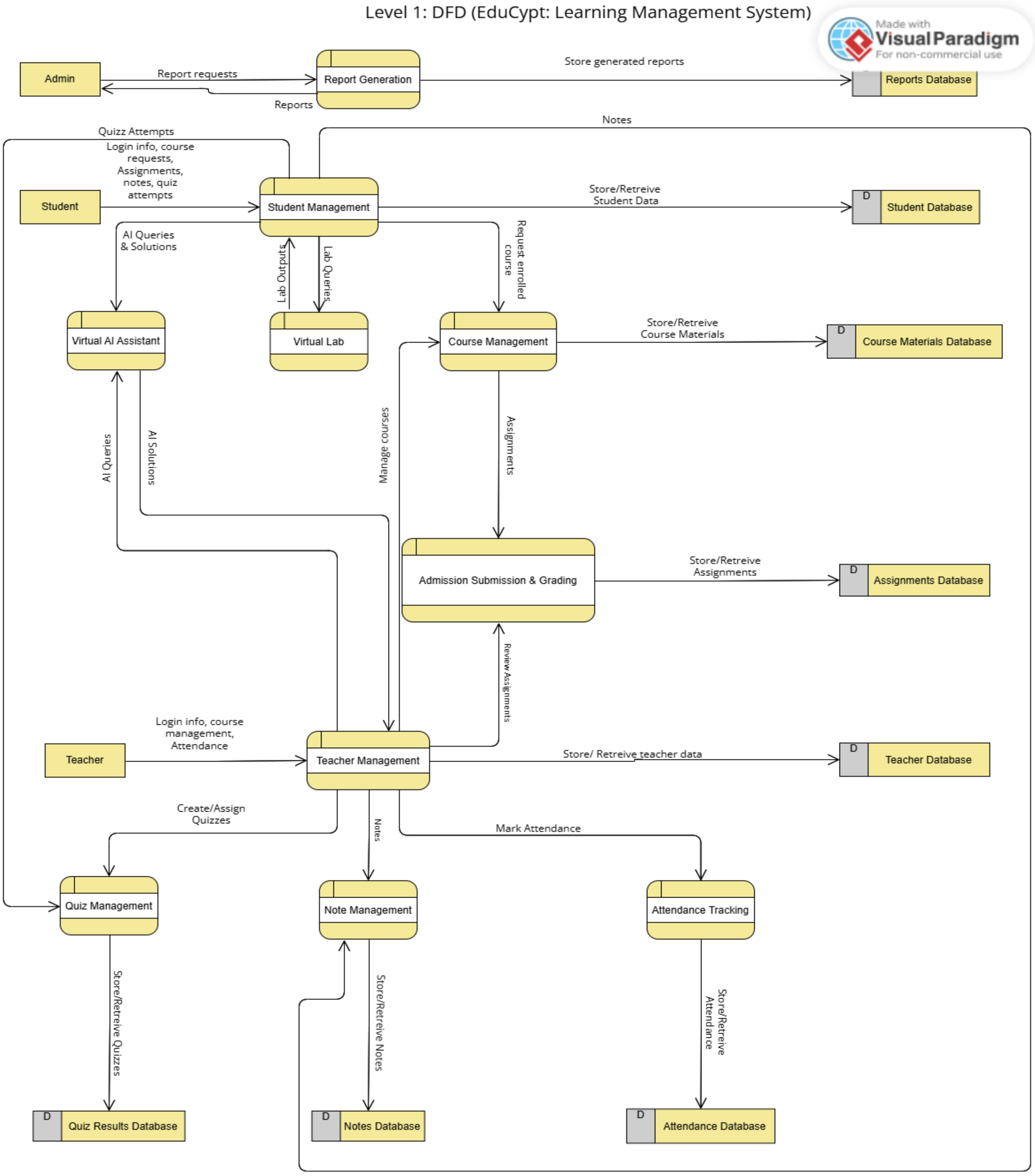


Fig 3.7 Data Flow Diagram (Level-0)

1.5) E.R Diagram

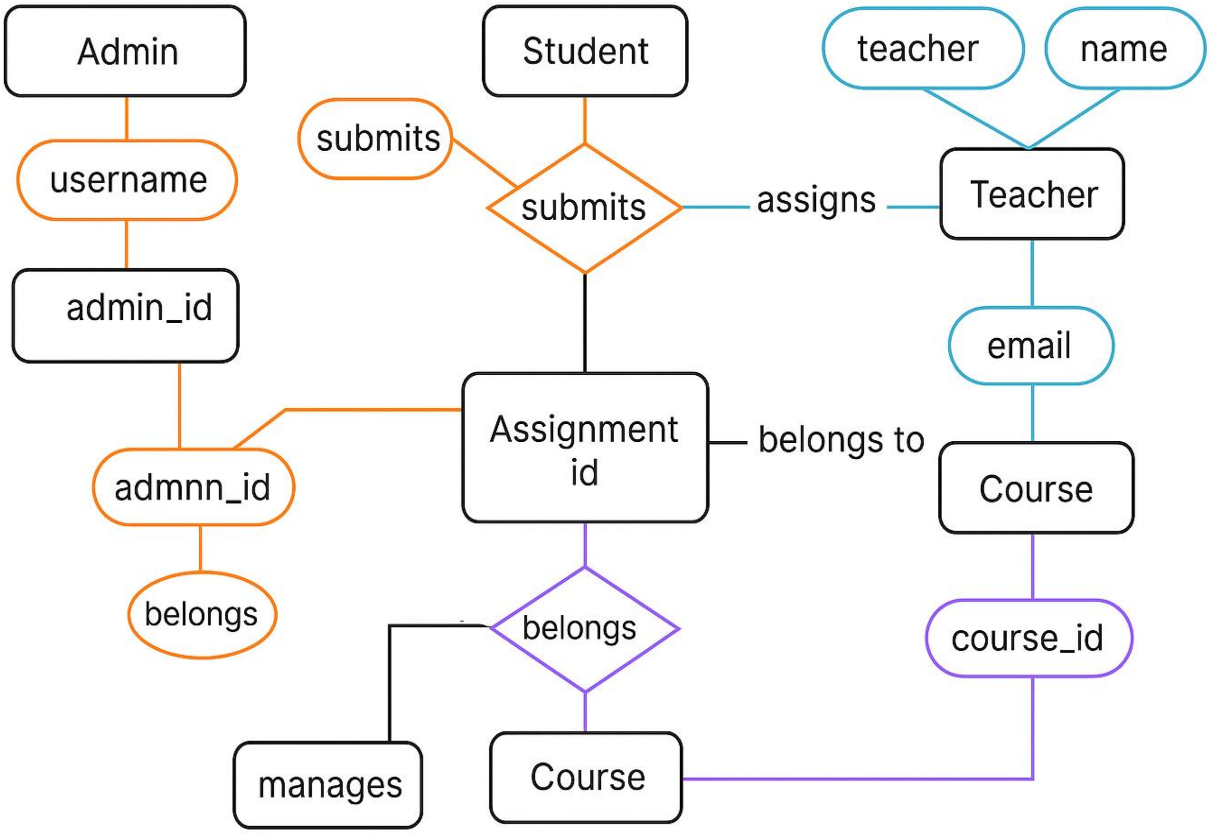


Fig 3.8 E. R Diagram

1.6) Class Diagram

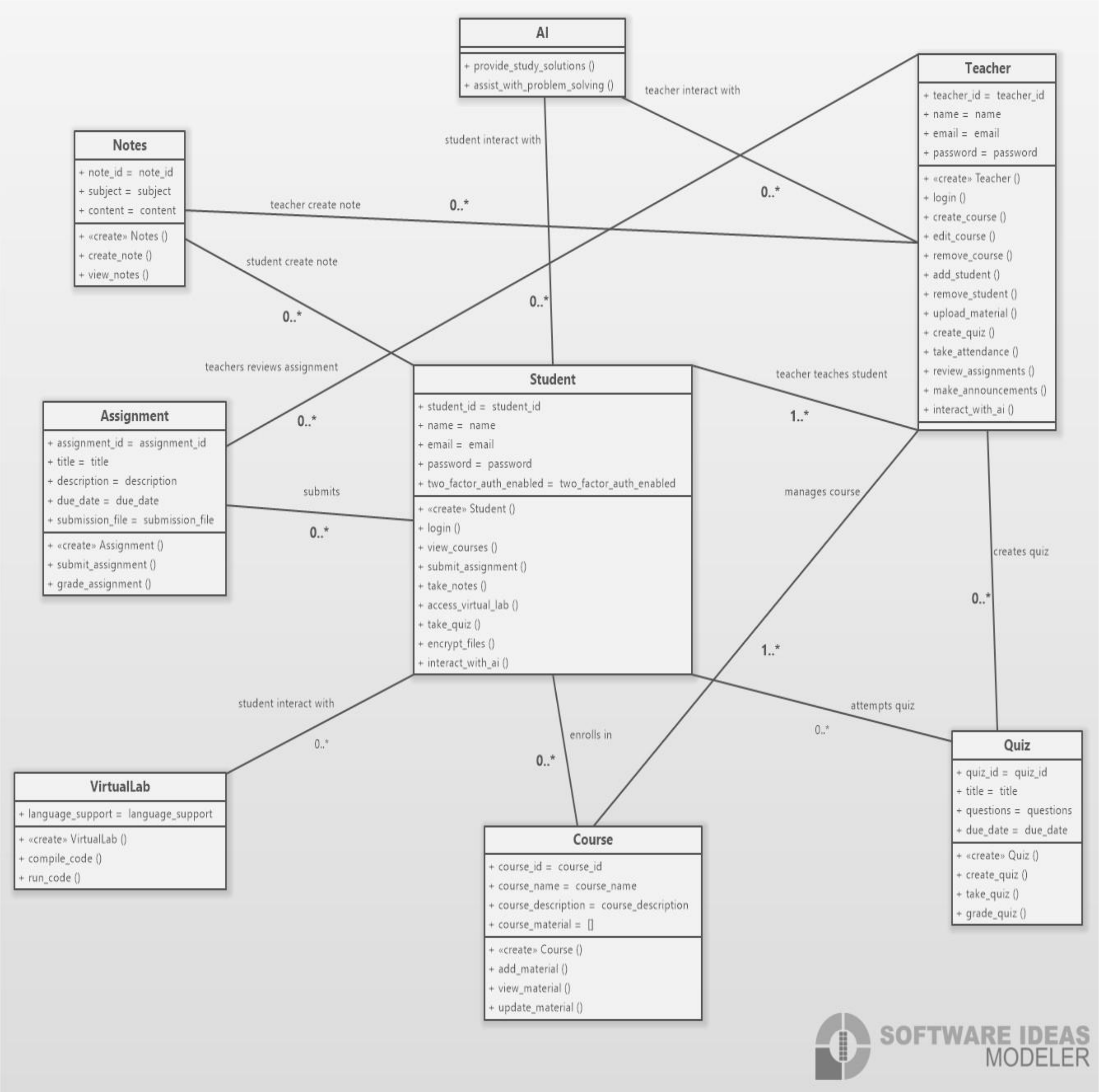
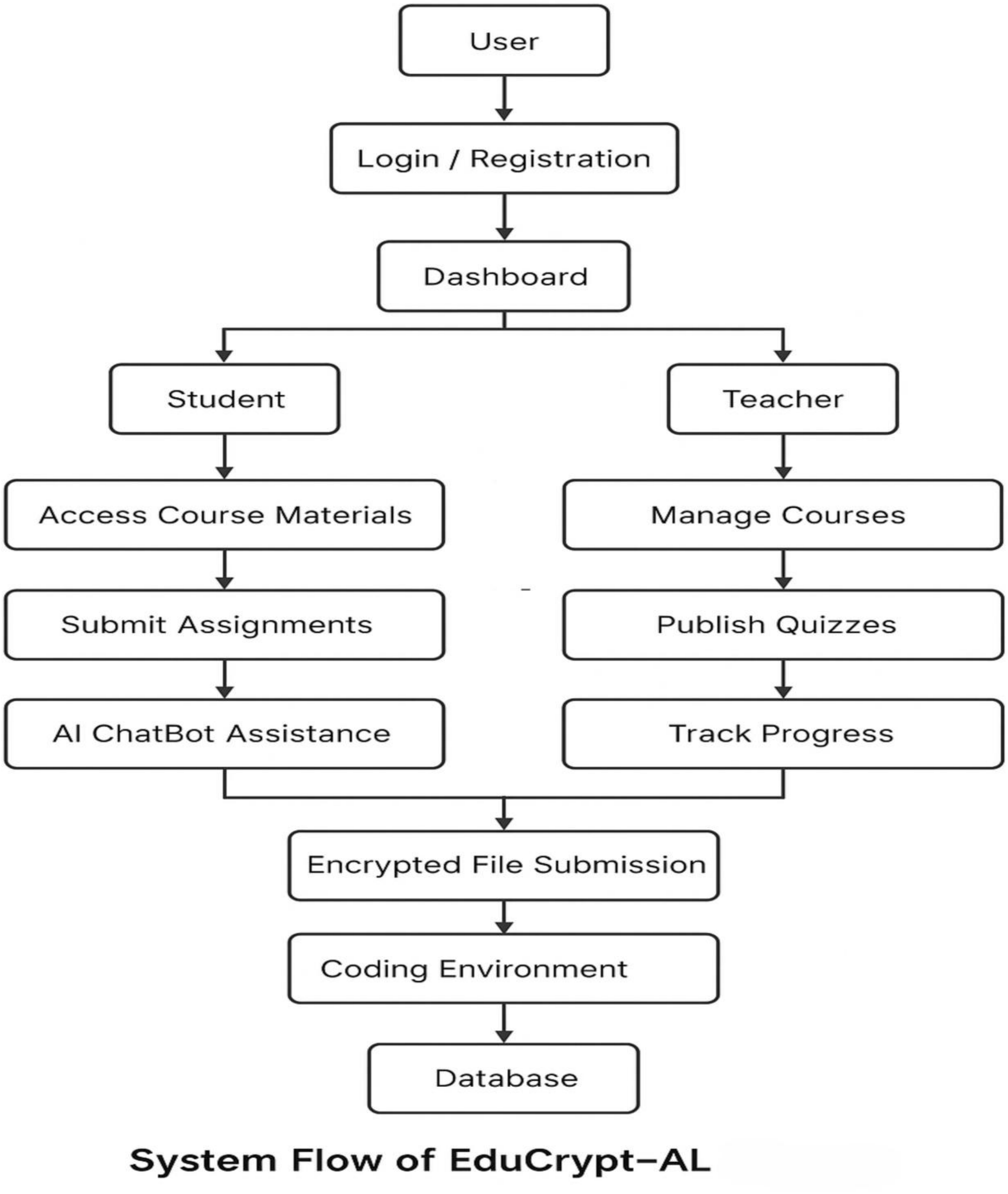


Fig 3.9 Class Diagram

Chapter-4: Design

4.1) System Flow Diagram

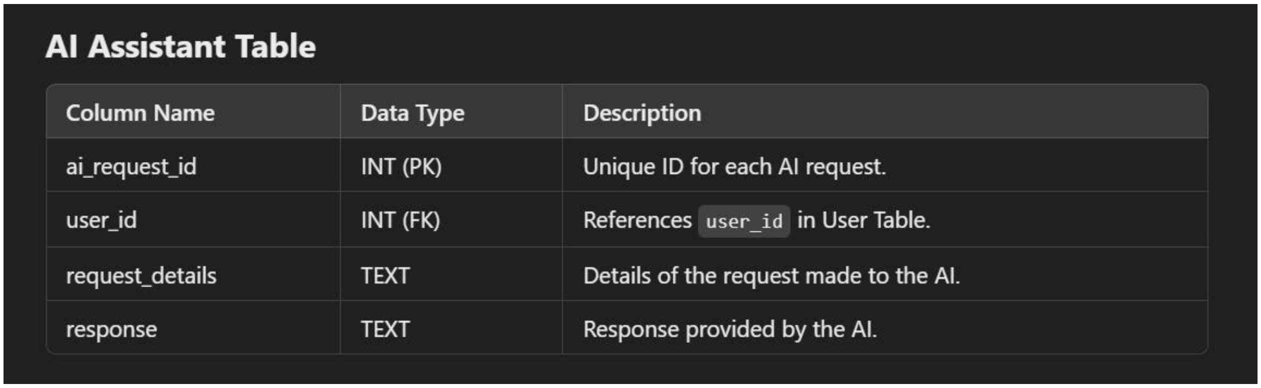


System Flow of EduCrypt–AL

Fig 4.1 System Flow Diagram

4.2) Data Dictionary

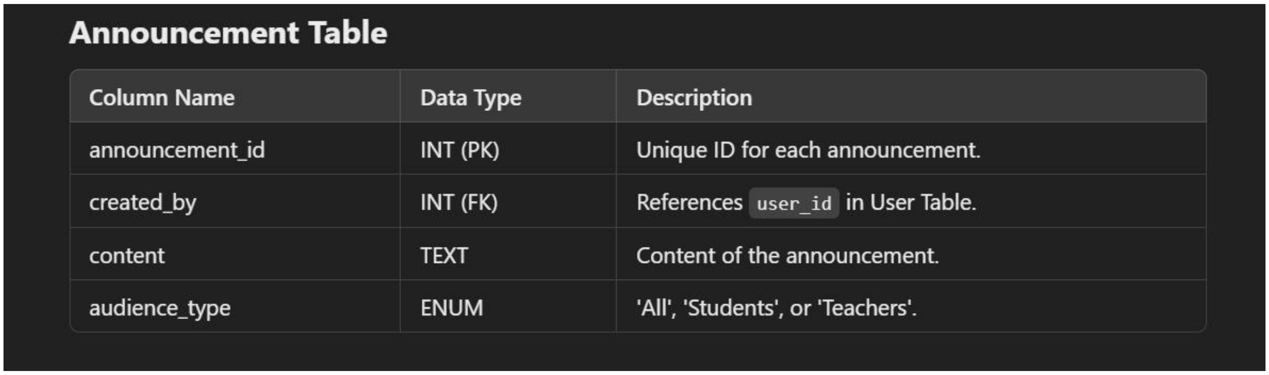
Table 4.1 A.I assistant table



AI Assistant Table

Column Name	Data Type	Description
ai_request_id	INT (PK)	Unique ID for each AI request.
user_id	INT (FK)	References `user_id` in User Table.
request_details	TEXT	Details of the request made to the AI.
response	TEXT	Response provided by the AI.

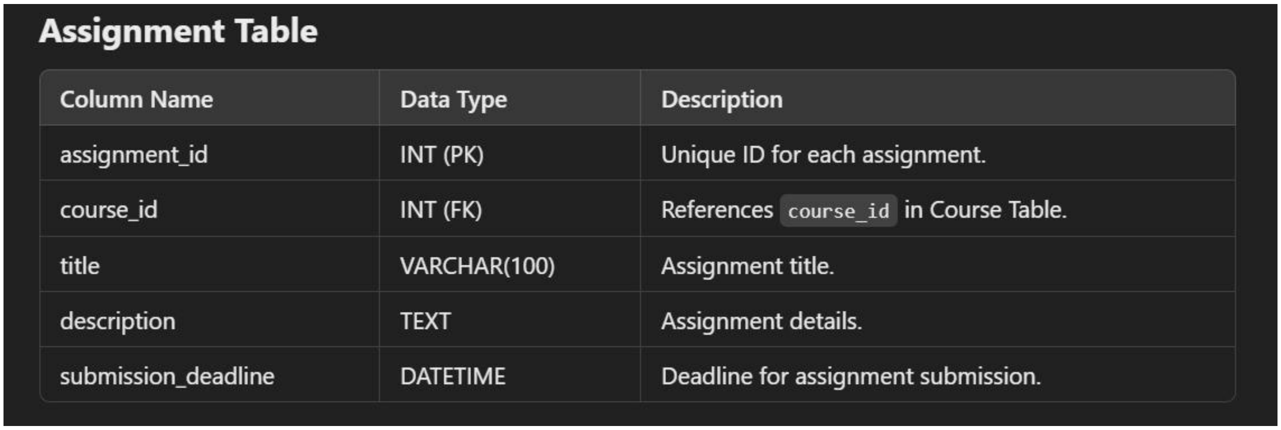
Table 4.2 Announcement table



Announcement Table

Column Name	Data Type	Description
announcement_id	INT (PK)	Unique ID for each announcement.
created_by	INT (FK)	References `user_id` in User Table.
content	TEXT	Content of the announcement.
audience_type	ENUM	'All', 'Students', or 'Teachers'.

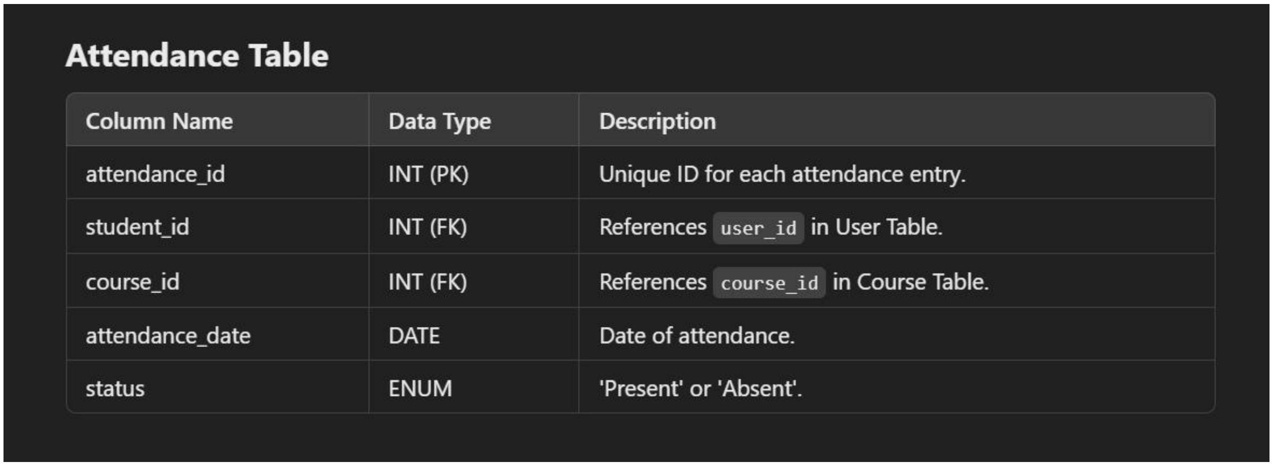
Table 4.3) Assignment table



Assignment Table

Column Name	Data Type	Description
assignment_id	INT (PK)	Unique ID for each assignment.
course_id	INT (FK)	References `course_id` in Course Table.
title	VARCHAR(100)	Assignment title.
description	TEXT	Assignment details.
submission_deadline	DATETIME	Deadline for assignment submission.

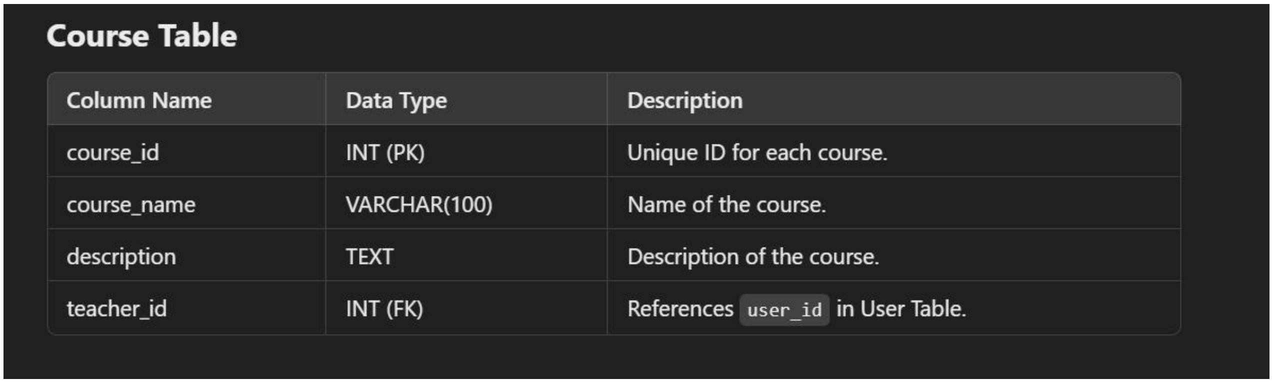
Table 4.4 Attendance table



Attendance Table

Column Name	Data Type	Description
attendance_id	INT (PK)	Unique ID for each attendance entry.
student_id	INT (FK)	References `user_id` in User Table.
course_id	INT (FK)	References `course_id` in Course Table.
attendance_date	DATE	Date of attendance.
status	ENUM	'Present' or 'Absent'.

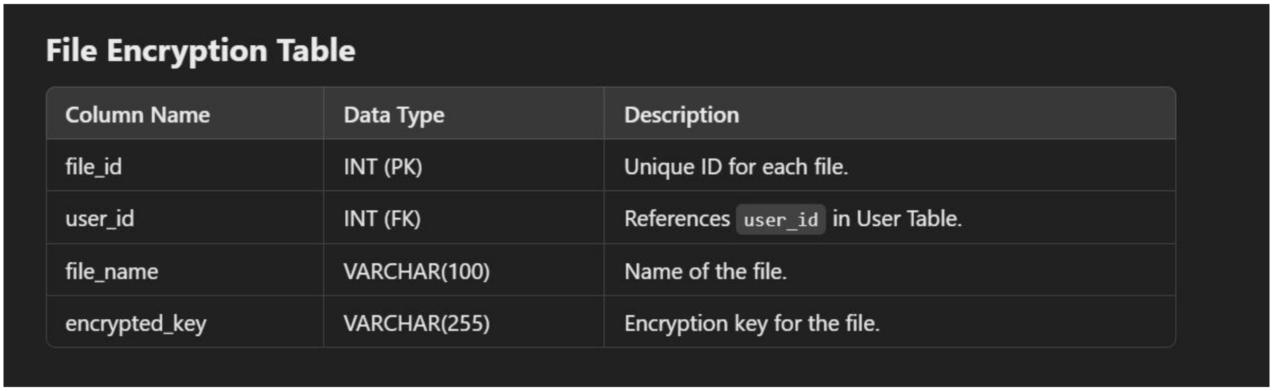
Table 4.5 Course table



Course Table

Column Name	Data Type	Description
course_id	INT (PK)	Unique ID for each course.
course_name	VARCHAR(100)	Name of the course.
description	TEXT	Description of the course.
teacher_id	INT (FK)	References `user_id` in User Table.

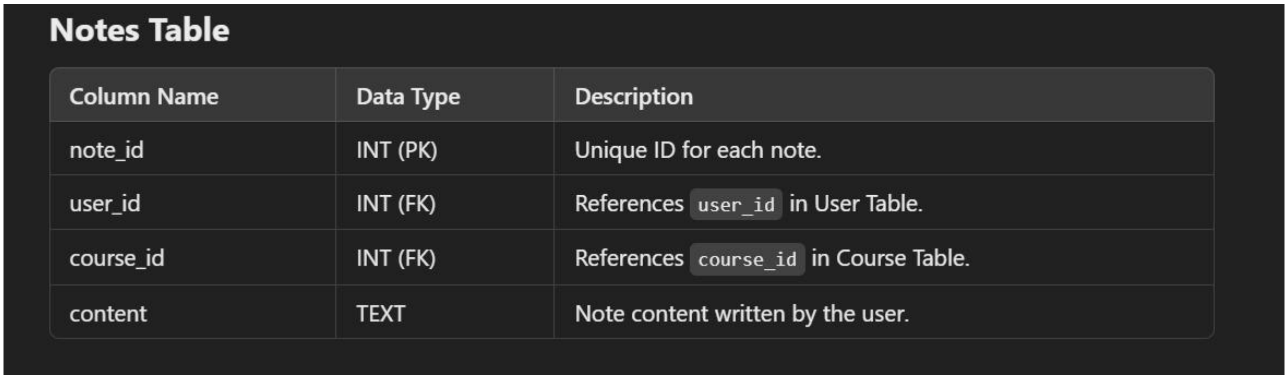
Table 4.6 File Encryption table



File Encryption Table

Column Name	Data Type	Description
file_id	INT (PK)	Unique ID for each file.
user_id	INT (FK)	References `user_id` in User Table.
file_name	VARCHAR(100)	Name of the file.
encrypted_key	VARCHAR(255)	Encryption key for the file.

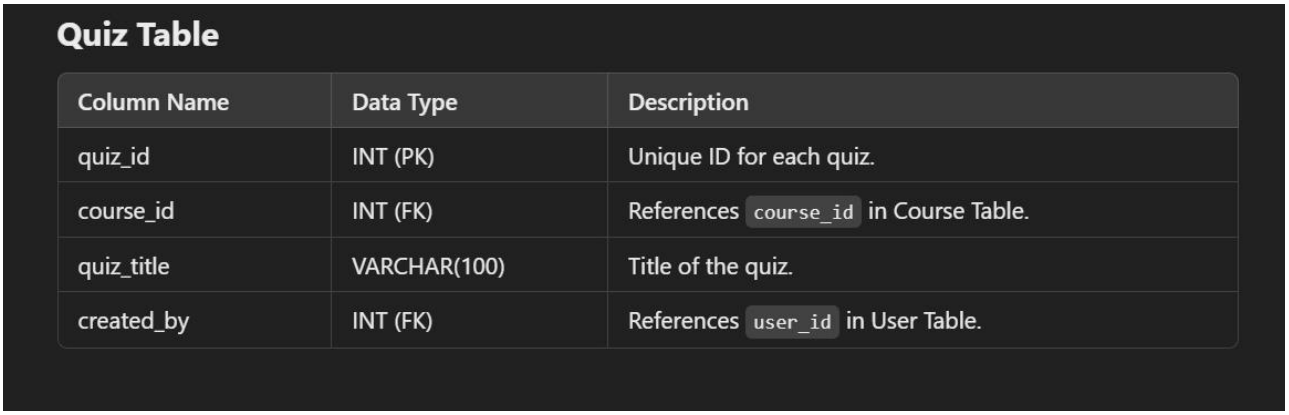
Table 4.7 Notes table



Notes Table

Column Name	Data Type	Description
note_id	INT (PK)	Unique ID for each note.
user_id	INT (FK)	References `user_id` in User Table.
course_id	INT (FK)	References `course_id` in Course Table.
content	TEXT	Note content written by the user.

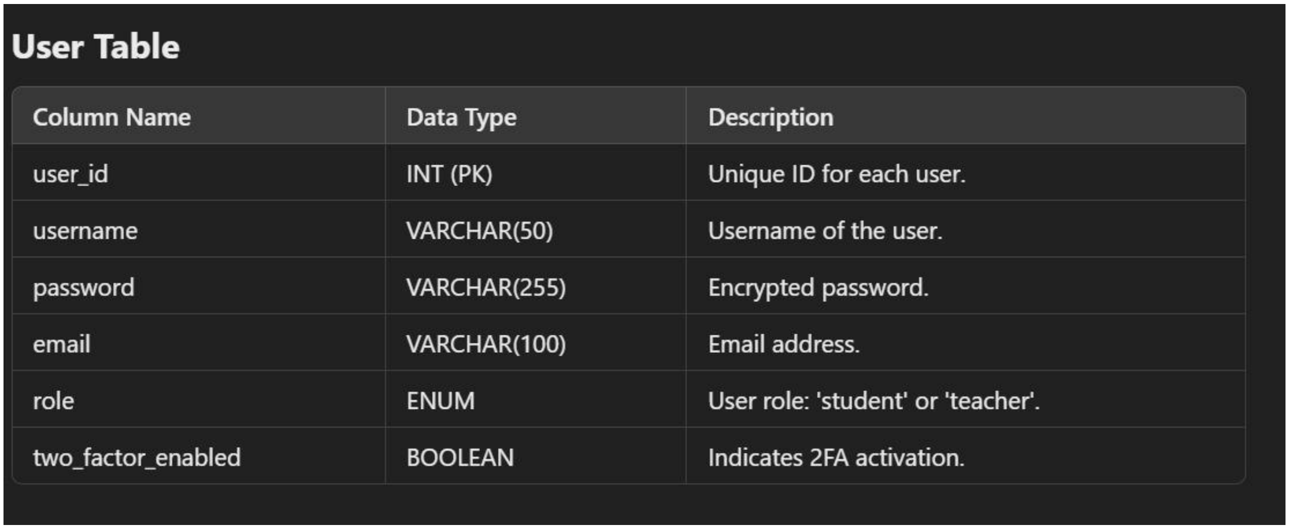
Table 4.8 Quiz table



Quiz Table

Column Name	Data Type	Description
quiz_id	INT (PK)	Unique ID for each quiz.
course_id	INT (FK)	References `course_id` in Course Table.
quiz_title	VARCHAR(100)	Title of the quiz.
created_by	INT (FK)	References `user_id` in User Table.

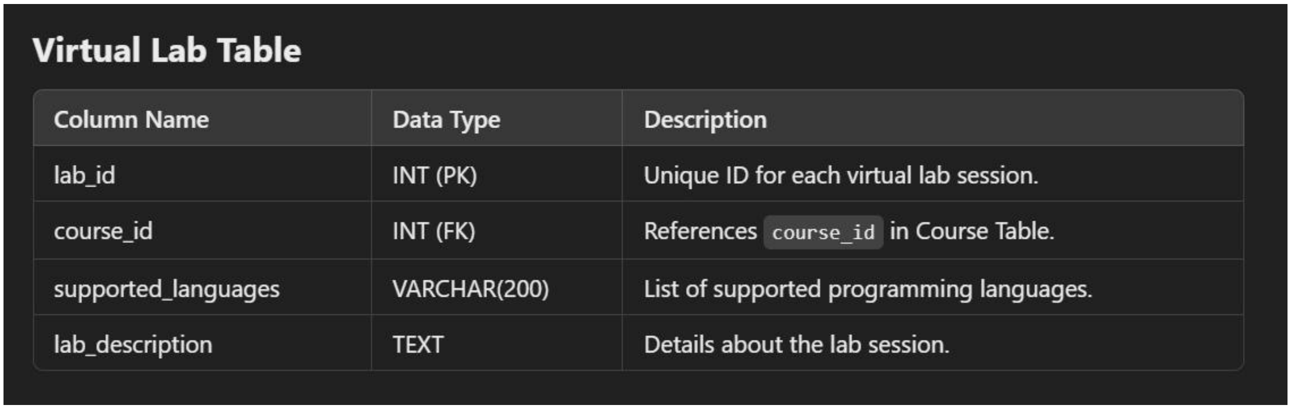
Table 4.9 User table



User Table

Column Name	Data Type	Description
user_id	INT (PK)	Unique ID for each user.
username	VARCHAR(50)	Username of the user.
password	VARCHAR(255)	Encrypted password.
email	VARCHAR(100)	Email address.
role	ENUM	User role: 'student' or 'teacher'.
two_factor_enabled	BOOLEAN	Indicates 2FA activation.

Table 4.10 Virtual lab table



Virtual Lab Table

Column Name	Data Type	Description
lab_id	INT (PK)	Unique ID for each virtual lab session.
course_id	INT (FK)	References `course_id` in Course Table.
supported_languages	VARCHAR(200)	List of supported programming languages.
lab_description	TEXT	Details about the lab session.

4.3) Relationship of Table (From Data Base System)

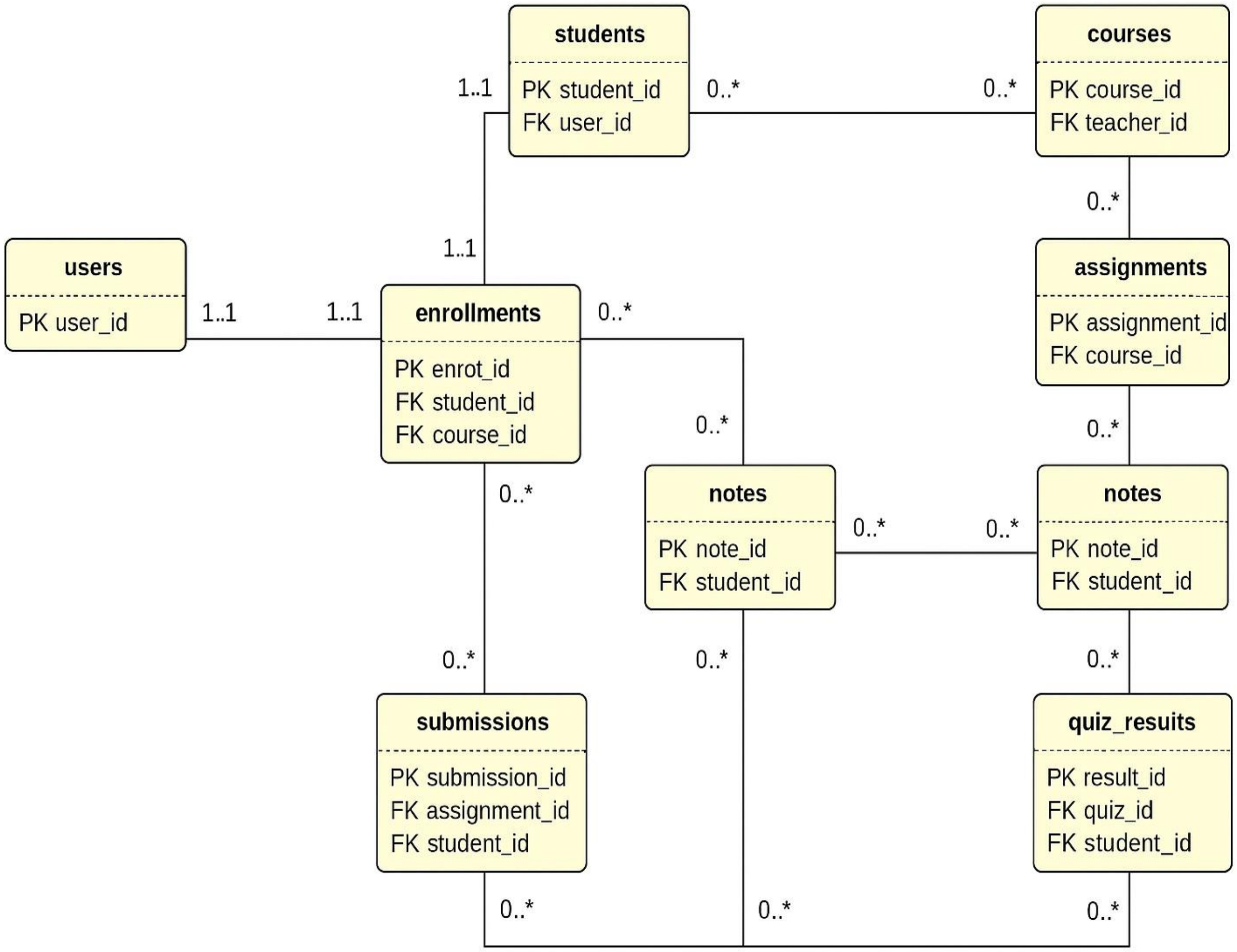
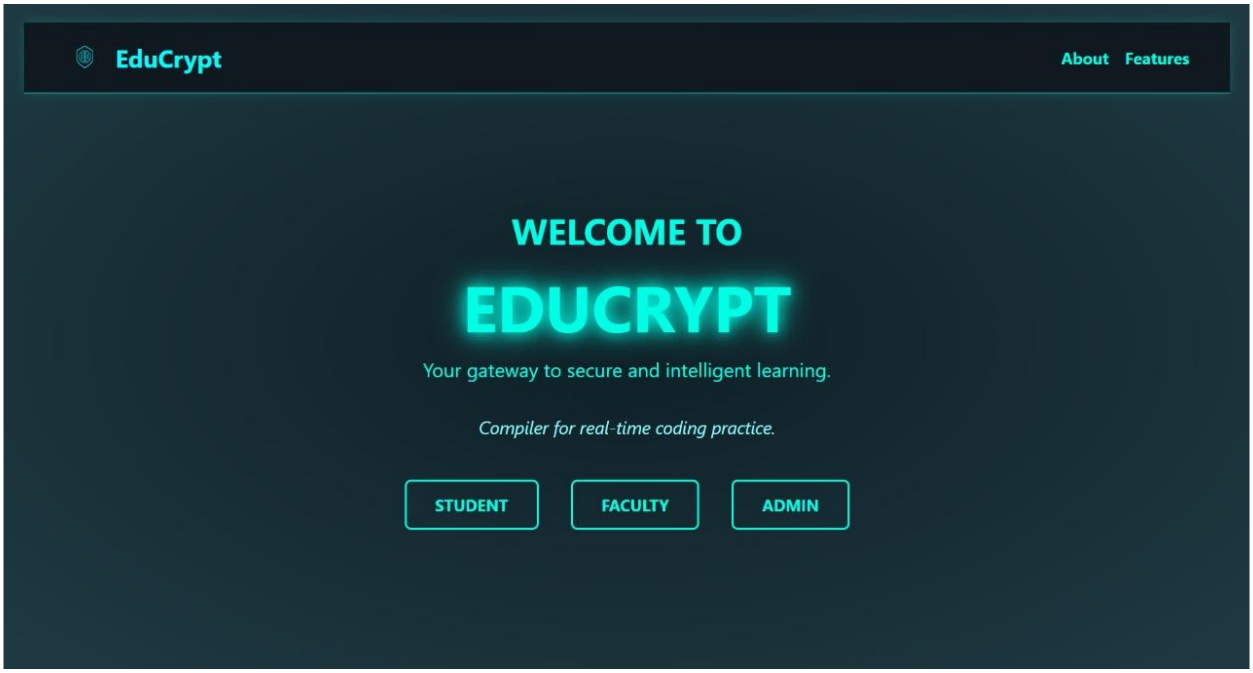


Fig 4.2 Relationship of Table

4.4) User-Interface



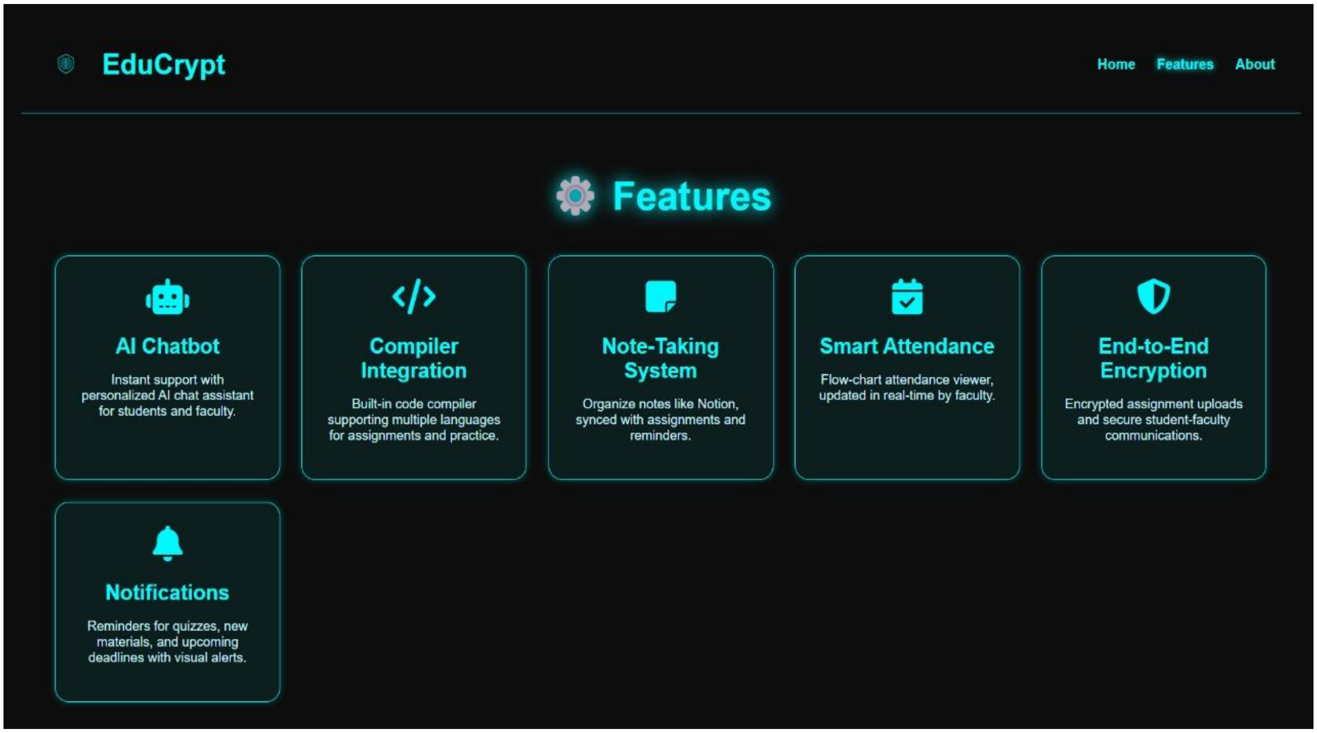
WELCOME TO

EDUCRYPT

Your gateway to secure and intelligent learning.

Compiler for real-time coding practice.

Fig 4.3 Home Page



Features

AI Chatbot

Instant support with personalized AI chat assistant for students and faculty.

Compiler Integration

Built-in code compiler supporting multiple languages for assignments and practice.

Note-Taking System

Organize notes like Notion, synced with assignments and reminders.

Smart Attendance

Flow-chart attendance viewer, updated in real-time by faculty.

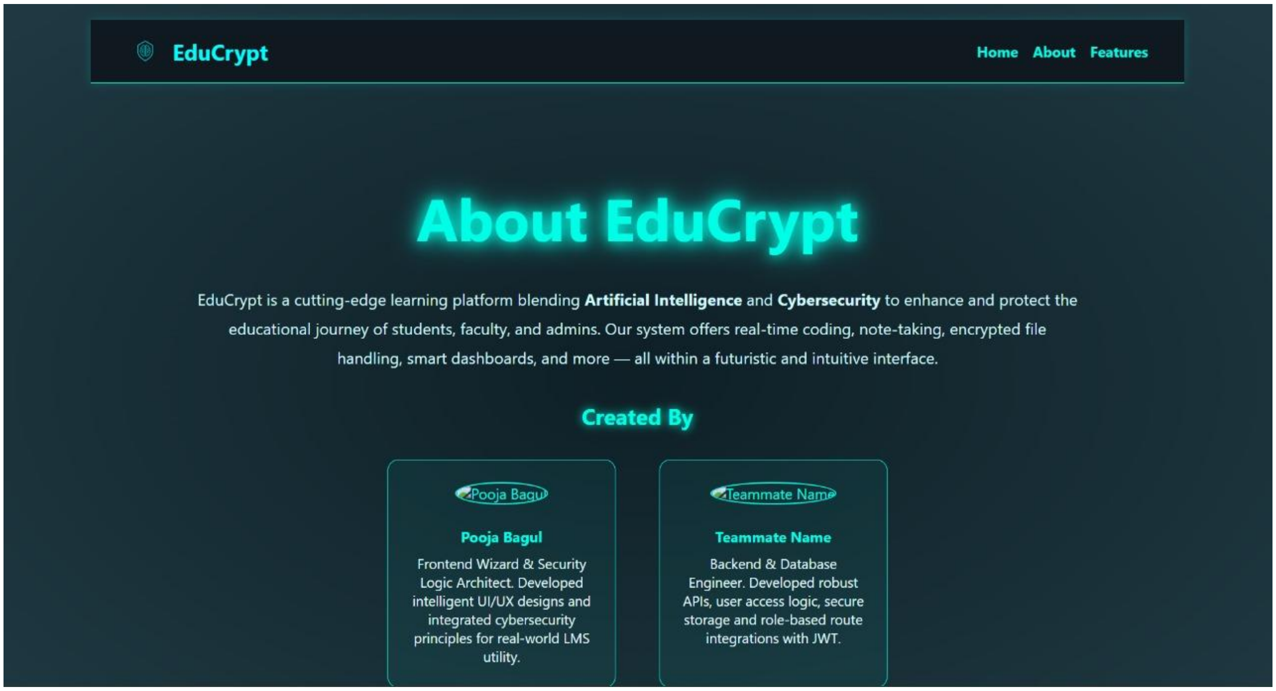
End-to-End Encryption

Encrypted assignment uploads and secure student-faculty communications.

Notifications

Reminders for quizzes, new materials, and upcoming deadlines with visual alerts.

Fig 4.4 Features



About EduCrypt

EduCrypt is a cutting-edge learning platform blending **Artificial Intelligence** and **Cybersecurity** to enhance and protect the educational journey of students, faculty, and admins. Our system offers real-time coding, note-taking, encrypted file handling, smart dashboards, and more — all within a futuristic and intuitive interface.

Created By

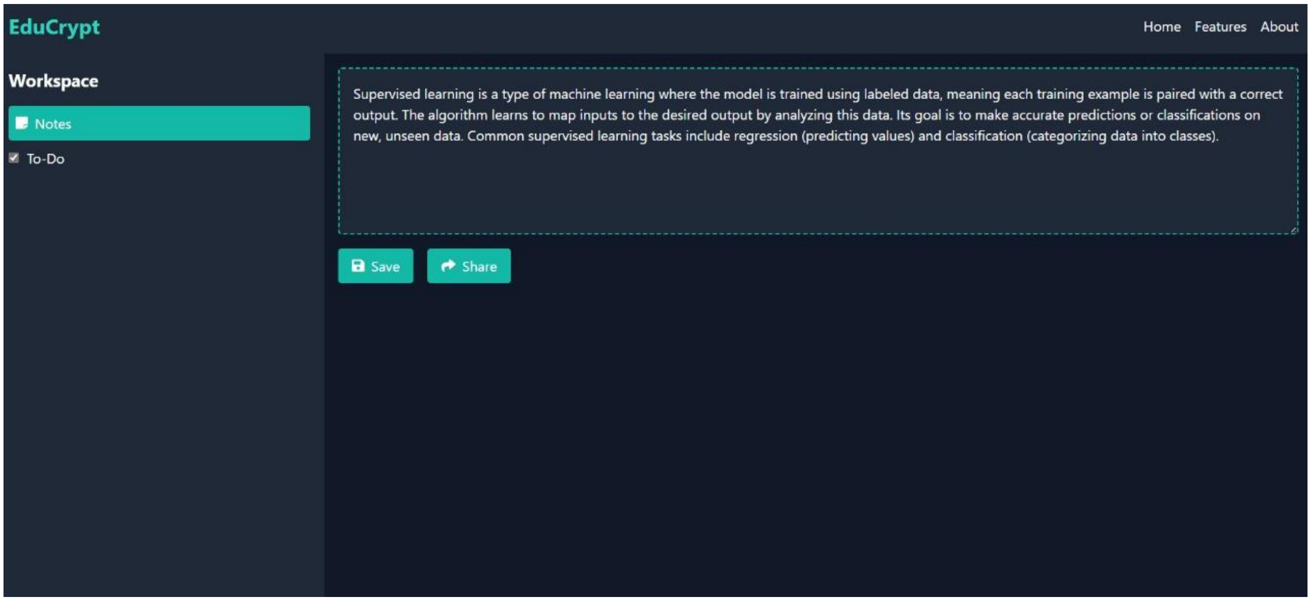
Pooja Bagul

Frontend Wizard & Security Logic Architect. Developed intelligent UI/UX designs and integrated cybersecurity principles for real-world LMS utility.

Teammate Name

Backend & Database Engineer. Developed robust APIs, user access logic, secure storage and role-based route integrations with JWT.

Fig 4.5 About Educrypt



Supervised learning is a type of machine learning where the model is trained using labeled data, meaning each training example is paired with a correct output. The algorithm learns to map inputs to the desired output by analyzing this data. Its goal is to make accurate predictions or classifications on new, unseen data. Common supervised learning tasks include regression (predicting values) and classification (categorizing data into classes).

Fig 4.6 Notes Taking System

Chapter-5: Implementation

5.1 Implementation Environment (Tools & Technology)

The implementation of EduCrypt was carried out using a modern technology stack that ensures speed, flexibility, scalability, and maintainability. The development tools and frameworks were selected based on their performance, developer support, and compatibility with LMS requirements.

Frontend (Client Side)

- **HTML5** – Used to structure the content of the web application.
- **CSS3** – Employed for layout design and responsiveness across devices.
- **JavaScript** – Enabled dynamic user interaction and data manipulation on the client side.

Backend (Server Side)

- **Node.js** – Used as the JavaScript runtime for handling backend logic.
- **Express.js** – A minimalist web framework used for creating RESTful APIs and routing.
- **REST API Client** – Facilitates communication between client and server by processing HTTP requests and responses efficiently.

Database

- **MongoDB** – A NoSQL document-based database used for storing user data, courses, assignments, submissions, and results. It offers scalability and flexibility for managing structured and semi-structured data.

Development & Design Tools

- **VS Code** – Source-code editor with support for debugging, extensions, and Git integration.
- **Postman** – Used to test REST APIs during development.
- **Figma** – Used for UI/UX wireframing and interface design.
- **Git & GitHub** – Version control and collaborative development platform.

5.2 Security Feature

Security was an integral part of the EduCrypt design, considering that it deals with user credentials, student submissions, and teacher resources.

Key Security Implementations

● **Authentication & Authorization:**

- Role-based login system for students, teachers, and admin.
- JWT (JSON Web Tokens) used for session management and secure API access.

● **Data Encryption:**

- Files uploaded by students (assignments) are encrypted before being stored.
- Sensitive user information is hashed and securely stored in the database.

● **Two-Factor Authentication (2FA):**

- Optional OTP-based verification for added account security.

● **HTTPS Deployment:**

- Ensures all data transmission between client and server is encrypted.

● **Rate Limiting and Input Validation:**

- Prevents brute-force attacks and SQL/NoSQL injection by sanitizing inputs.

5.3 Coding Standard

To ensure maintainability, consistency, and readability, the EduCrypt development team followed standard coding practices throughout the lifecycle of the project.

Coding Practices Followed

● **Modular Programming:**

- The backend code was broken into controllers, routes, models, and middleware to separate logic.

● **Naming Conventions:**

- CamelCase used for JavaScript variables and functions.
- Consistent naming conventions across files and APIs.

● **Code Commenting:**

- Functions and complex logic blocks are properly commented for readability.

● **Linting Tools:**

- ESLint used to ensure syntax rules and to avoid errors or warnings.

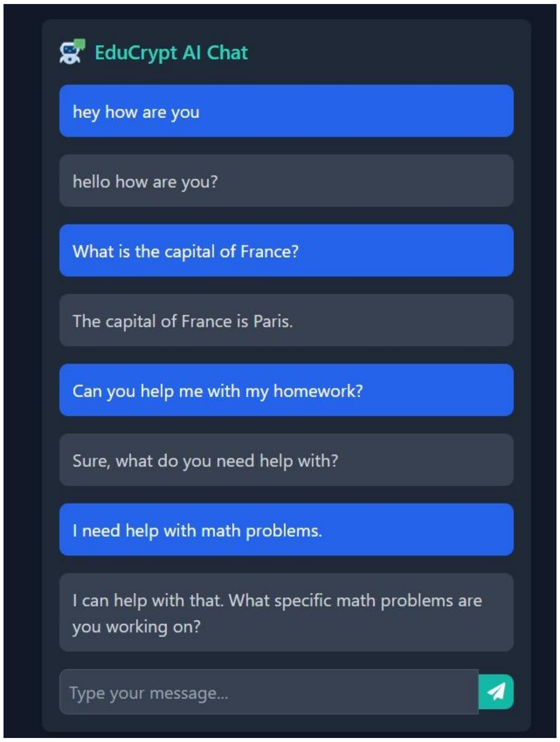
● **Version Control:**

    - Every major change is committed with meaningful messages to GitHub, enabling traceability and collaboration.

- **API Design Standard:**
    - REST principles were followed: GET, POST, PUT, DELETE mapped correctly with endpoints.
    - Responses follow consistent JSON structure with success/error messaging.

Chapter-6: Project Screenshots

6.1) Prototype with Screenshots/Results



EduCrypt AI Chat

hey how are you

hello how are you?

What is the capital of France?

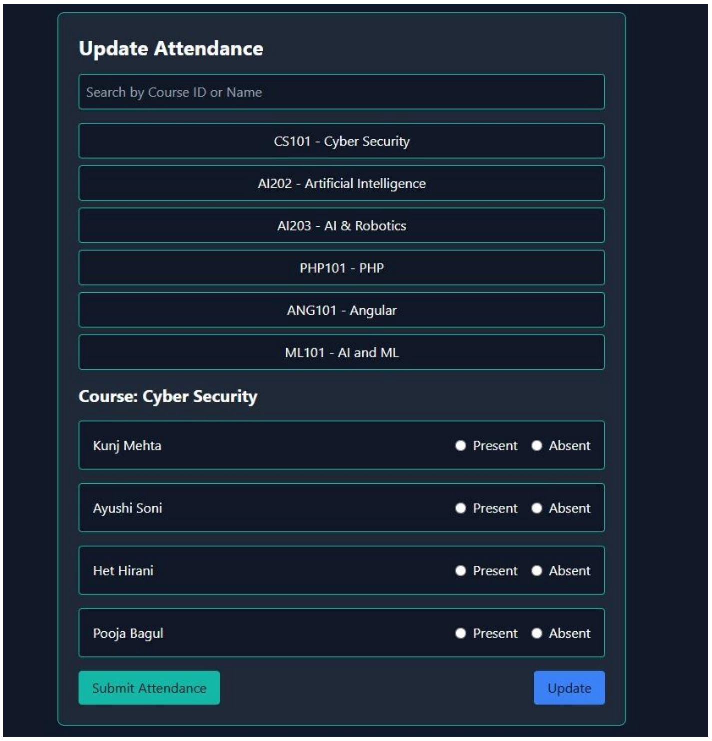
The capital of France is Paris.

Can you help me with my homework?

Sure, what do you need help with?

I need help with math problems.

I can help with that. What specific math problems are you working on?



Update Attendance

CS101 - Cyber Security

AI202 - Artificial Intelligence

AI203 - AI & Robotics

PHP101 - PHP

ANG101 - Angular

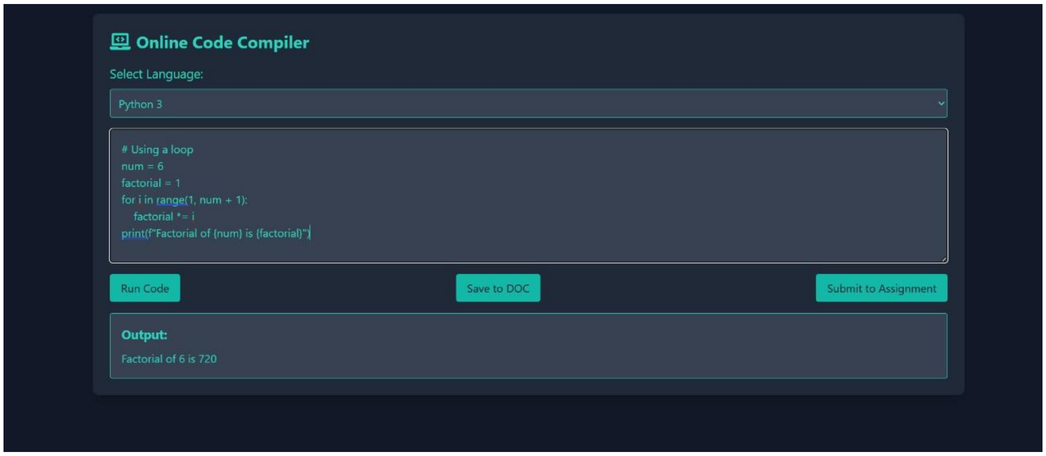
ML101 - AI and ML

Course: Cyber Security

Kunj Mehta	● Present	● Absent
Ayushi Soni	● Present	● Absent
Het Hirani	● Present	● Absent
Pooja Bagul	● Present	● Absent

Fig 6.1 AI ChatBot

Fig 6.2 Student Attendance Update



Online Code Compiler

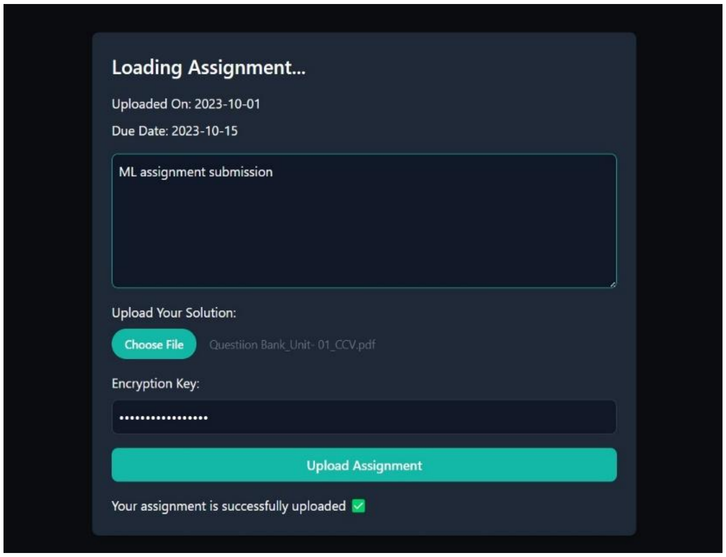
Select Language: Python 3

```
# Using a loop
num = 6
factorial = 1
for i in range(1, num + 1):
   factorial *= i
print(f"Factorial of {num} is {factorial}")
```

Output:

```
Factorial of 6 is 720
```

Fig 6.3 In Built Compiler



Loading Assignment...

Uploaded On: 2023-10-01

Due Date: 2023-10-15

ML assignment submission

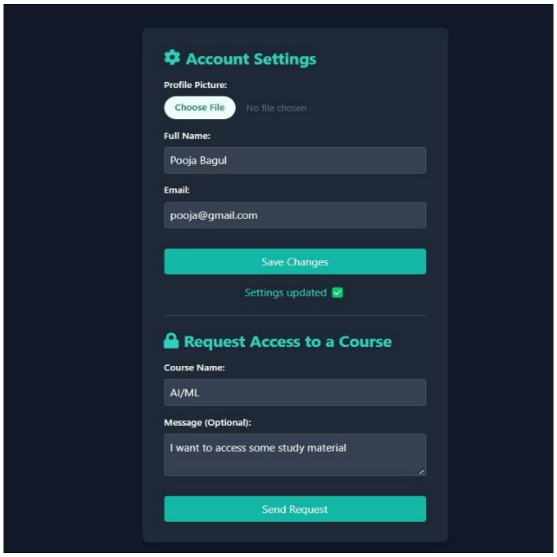
Upload Your Solution:

Questiion Bank_Unit- 01_CCV.pdf

Encryption Key:

•••••••••••••••••

Your assignment is successfully uploaded ✓



Account Settings

Profile Picture:

No file chosen

Full Name:

Pooja Bagul

Email:

pooja@gmail.com

Settings updated ✓

Request Access to a Course

Course Name:

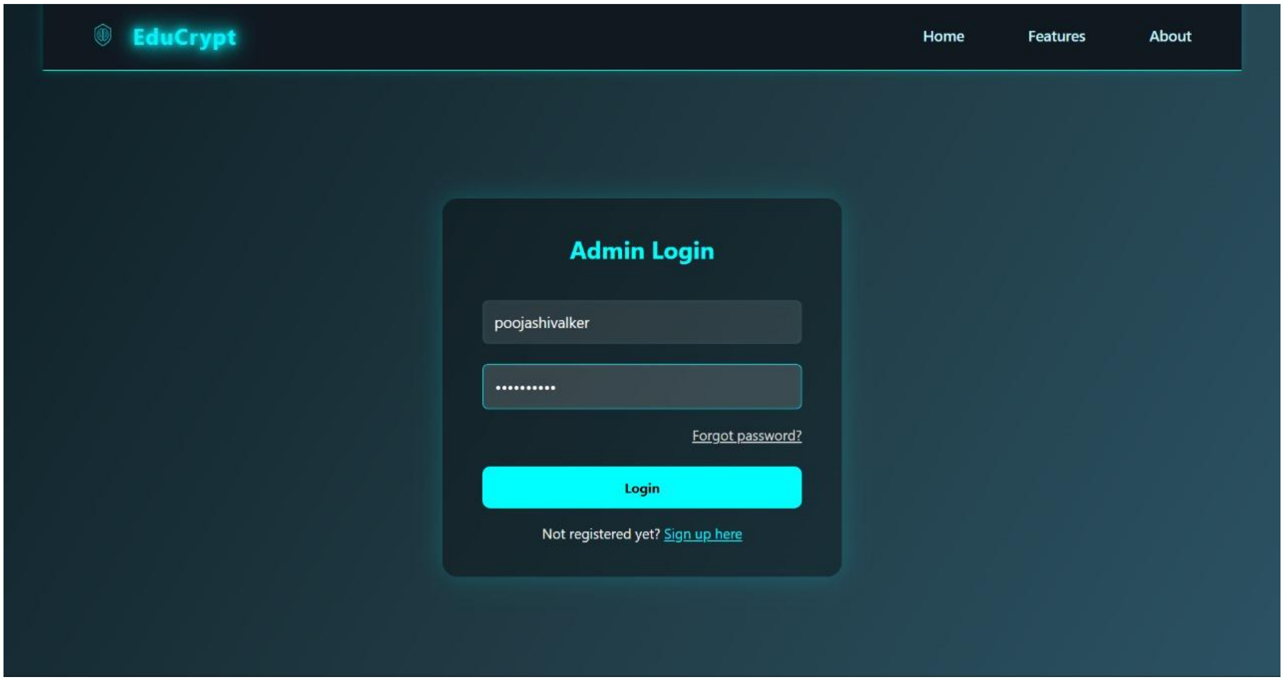
AI/ML

Message (Optional):

I want to access some study material

Fig 6.4 Assignment Submission with Encryption

Fig 6.5 Account Settings and Update



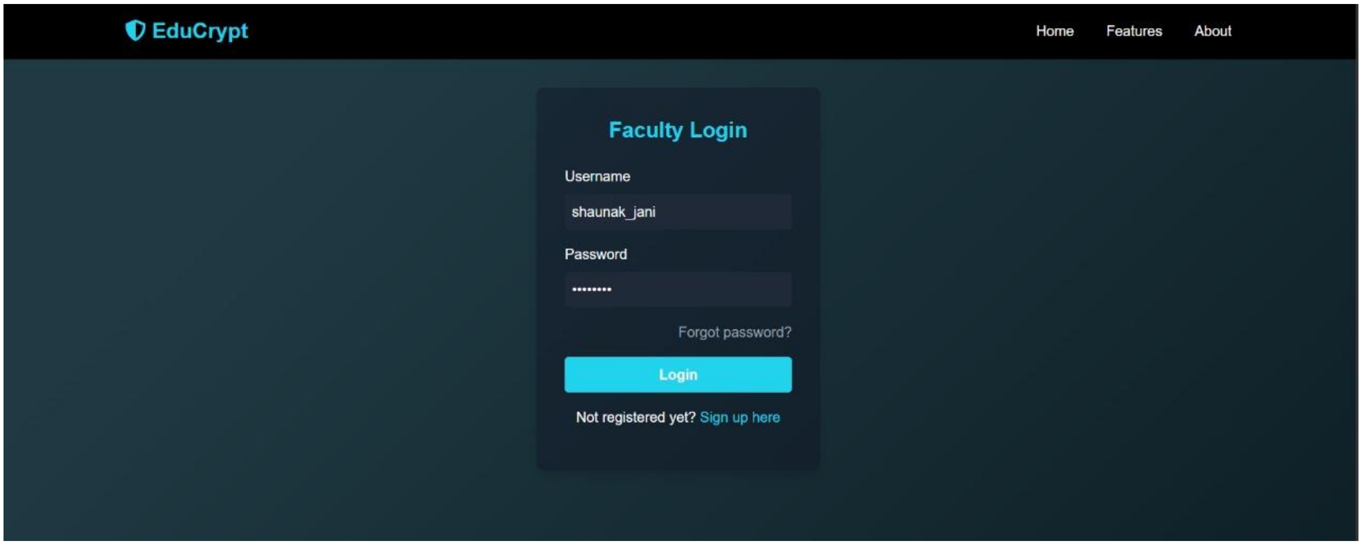
Admin Login

poojashivalker

••••••••••

Not registered yet? Sign up here

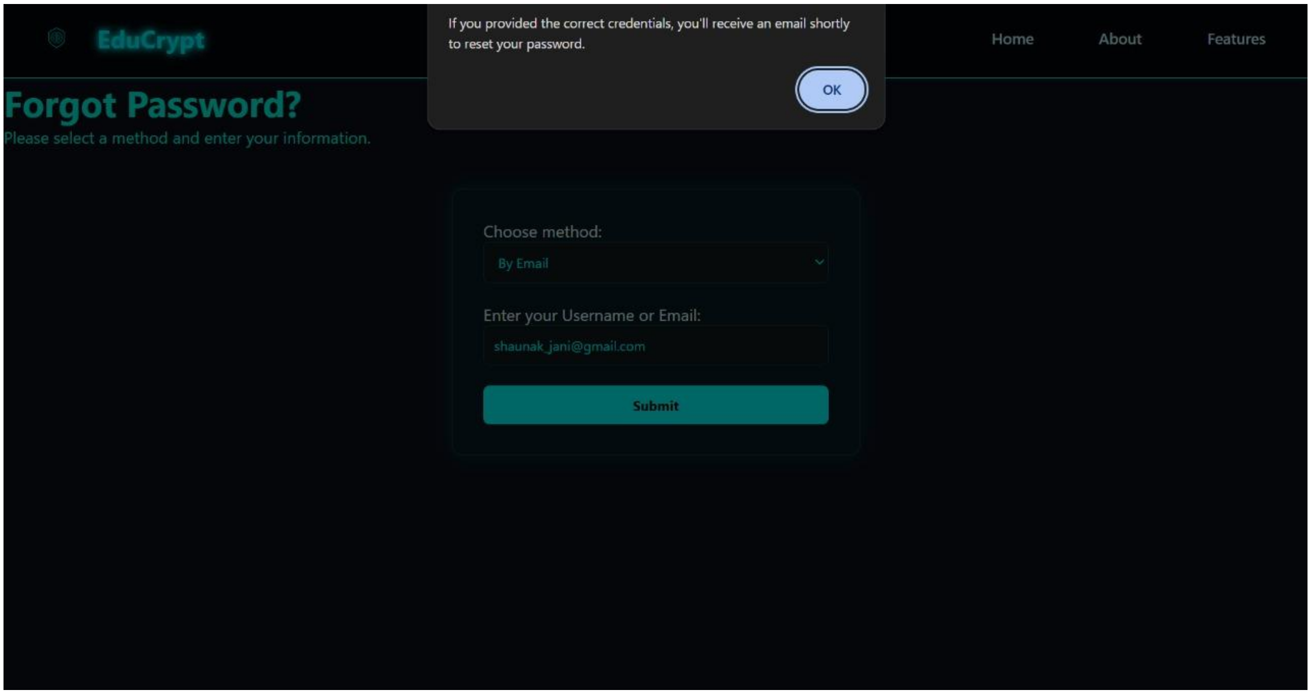
Fig 6.6 Admin Login



Username: shaunak_jani

Password: ••••••••

Fig 6.7 Faculty Login



If you provided the correct credentials, you'll receive an email shortly to reset your password.

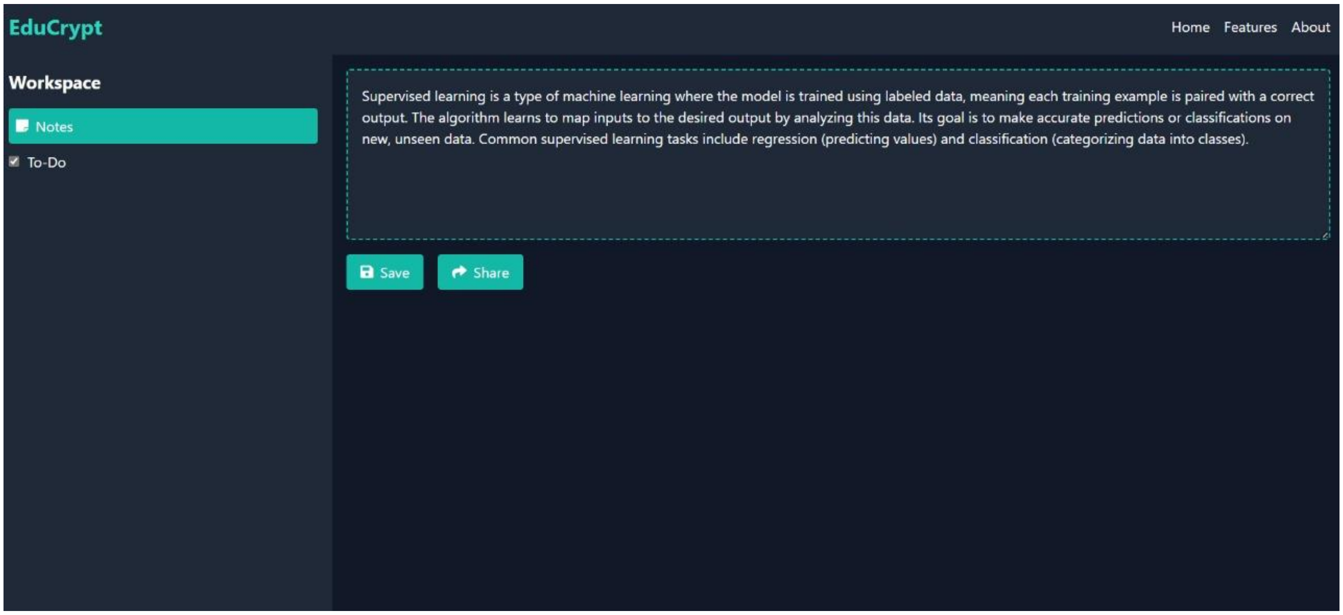
Forgot Password?

Please select a method and enter your information.

Choose method: By Email

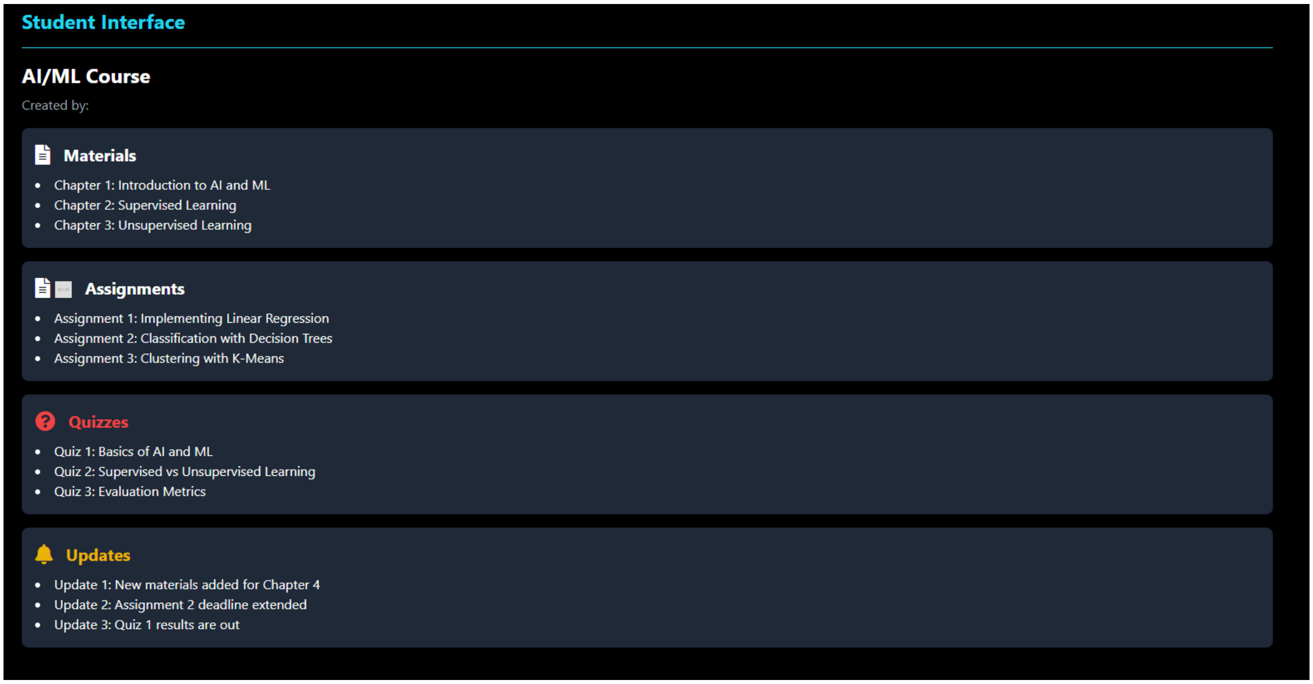
Enter your Username or Email: shaunak_jani@gmail.com

Fig 6.8 Forgot Password



Supervised learning is a type of machine learning where the model is trained using labeled data, meaning each training example is paired with a correct output. The algorithm learns to map inputs to the desired output by analyzing this data. Its goal is to make accurate predictions or classifications on new, unseen data. Common supervised learning tasks include regression (predicting values) and classification (categorizing data into classes).

Fig 6.9 In built Notes making



Student Interface

AI/ML Course

Created by:

Materials

- Chapter 1: Introduction to AI and ML
- Chapter 2: Supervised Learning
- Chapter 3: Unsupervised Learning

Assignments

- Assignment 1: Implementing Linear Regression
- Assignment 2: Classification with Decision Trees
- Assignment 3: Clustering with K-Means

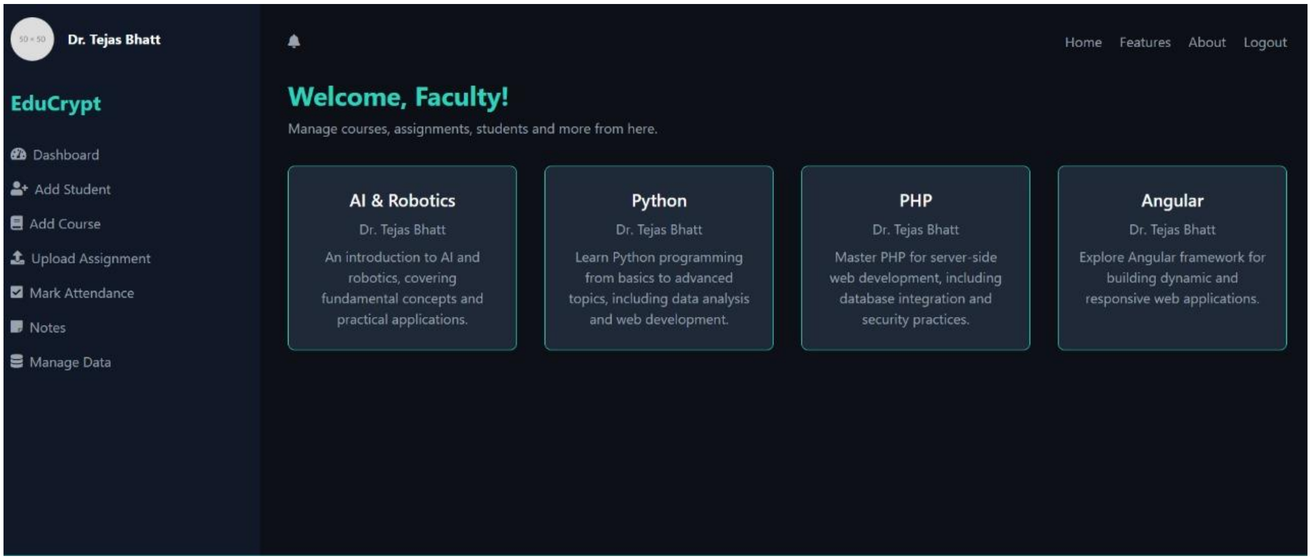
Quizzes

- Quiz 1: Basics of AI and ML
- Quiz 2: Supervised vs Unsupervised Learning
- Quiz 3: Evaluation Metrics

Updates

- Update 1: New materials added for Chapter 4
- Update 2: Assignment 2 deadline extended
- Update 3: Quiz 1 results are out

Fig 6.10 Student Interface



Welcome, Faculty!

Manage courses, assignments, students and more from here.

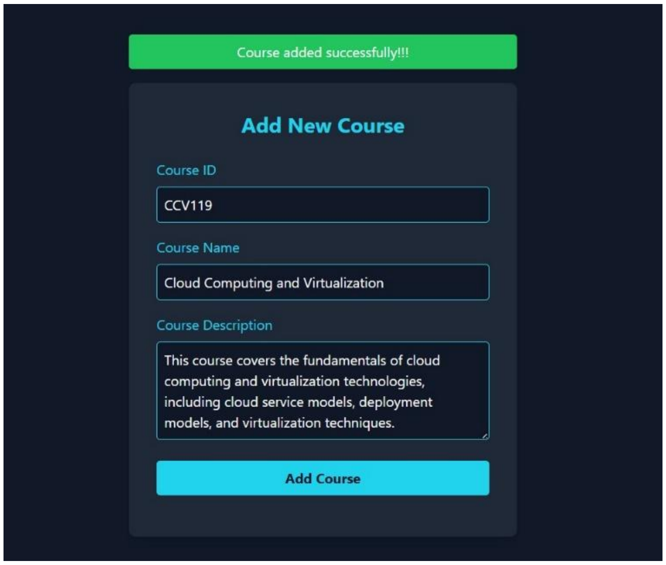
AI & Robotics
Dr. Tejas Bhatt
An introduction to AI and robotics, covering fundamental concepts and practical applications.

Python
Dr. Tejas Bhatt
Learn Python programming from basics to advanced topics, including data analysis and web development.

PHP
Dr. Tejas Bhatt
Master PHP for server-side web development, including database integration and security practices.

Angular
Dr. Tejas Bhatt
Explore Angular framework for building dynamic and responsive web applications.

Fig 6.11 Faculty Dashboard



Course added successfully!!!

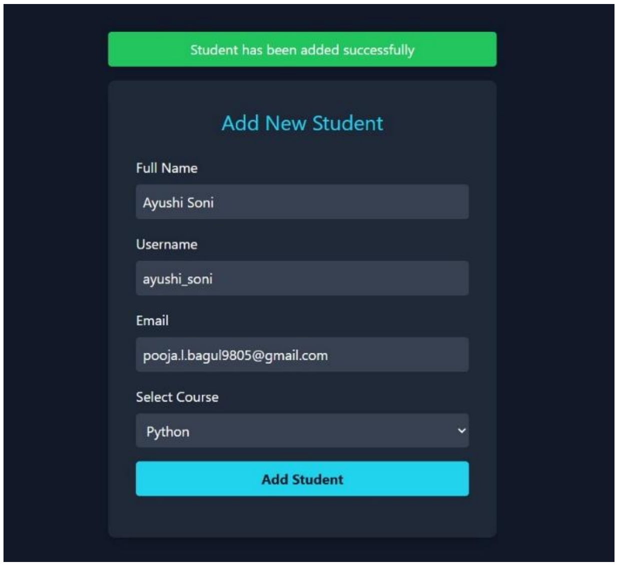
Add New Course

Course ID: CCV119

Course Name: Cloud Computing and Virtualization

Course Description: This course covers the fundamentals of cloud computing and virtualization technologies, including cloud service models, deployment models, and virtualization techniques.

Fig 6.12 Add New Course



Student has been added successfully

Add New Student

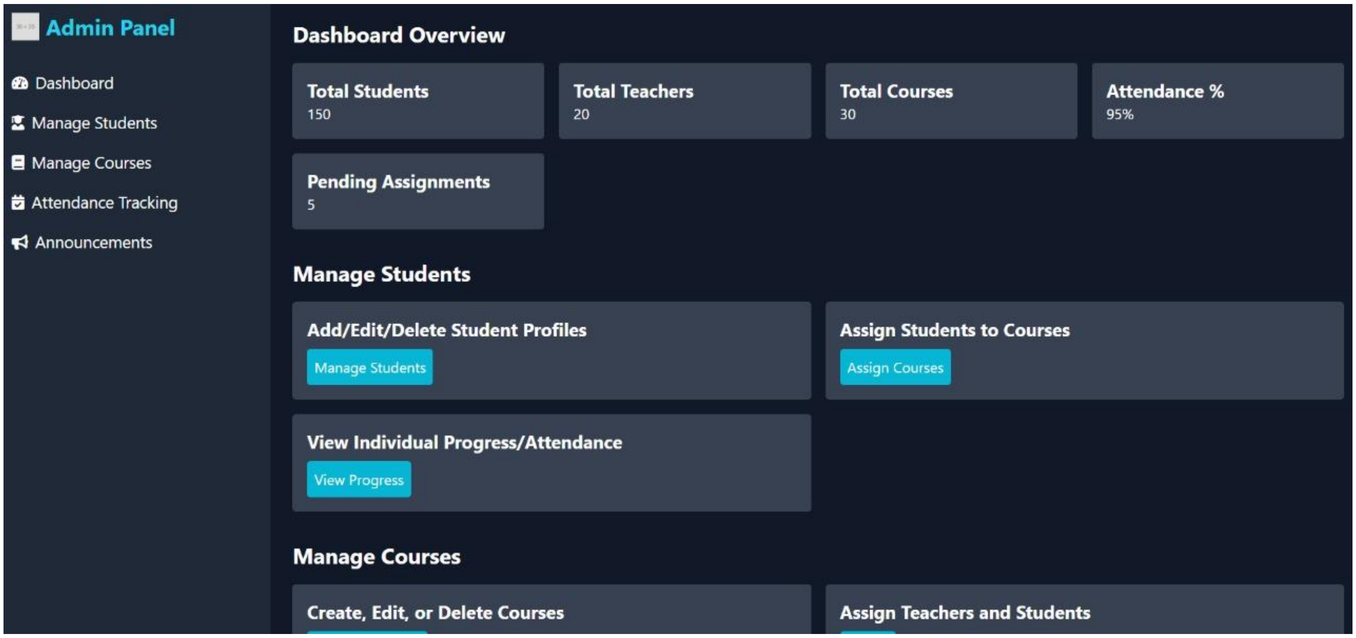
Full Name: Ayushi Soni

Username: ayushi_soni

Email: pooja.l.bagul9805@gmail.com

Select Course: Python

Fig 6.13 Add New Student



Dashboard Overview

Total Students
150

Total Teachers
20

Total Courses
30

Attendance %
95%

Pending Assignments
5

Manage Students

Add/Edit/Delete Student Profiles

Assign Students to Courses

View Individual Progress/Attendance

Manage Courses

Create, Edit, or Delete Courses

Assign Teachers and Students

Fig 6.14 Admin Panel

Chapter-7: Conclusion and Future Work

7.1 Conclusion

EduCrypt has been successfully developed as a web-based **Learning Management System (LMS)** that addresses the modern educational needs of institutions, students, and faculty. The platform offers secure and seamless interaction through user-specific dashboards, intuitive navigation, and feature-rich modules. With integrated functionalities such as course management, encrypted file submissions, AI chatbot assistance, quizzes, notes, and a built-in coding compiler, EduCrypt provides a centralized system for collaborative learning.

The implementation of modern technologies such as Node.js, Express.js, MongoDB, and responsive front-end design ensures high performance, scalability, and user engagement. Role-based access control, encrypted communications, and adherence to coding standards have contributed to a reliable and secure product. The use of agile methodology facilitated continuous feedback integration, resulting in a robust and dynamic learning platform. EduCrypt serves as a complete academic solution for managing day-to-day academic workflows and has the potential to be scaled and customized for any educational institution. The project also reflects the collaborative skills, technical expertise, and problem-solving capabilities of the development team.

7.2 Future Work

Although EduCrypt includes many core and advanced features, there is still significant scope for expansion and improvement in future iterations. The proposed enhancements include:

Planned Enhancements

- **Video Lecture Integration:**

  - Integration of video conferencing APIs (e.g., Zoom, Jitsi Meet) for live sessions.
- **AI-Powered Recommendation Engine:**
  - Personalized course and resource recommendations based on user behaviour and performance.
- **Blockchain Certification:**
  - Issue tamper-proof digital certificates using blockchain technology for course completion.
- **Multilingual Support:**
  - Interface translation and accessibility for users in multiple regional languages.
- **Dark Mode & Accessibility Improvements:**
  - UI enhancements for low-light usage and accessibility-friendly interfaces (screen reader support).
- **Admin Analytics Dashboard:**
  - Advanced dashboards with charts and usage analytics for performance monitoring.
- **Mobile App Extension:**
  - Development of a cross-platform mobile app for Android and iOS using the same backend.

References

1. Firebase Documentation – https://firebase.google.com/docs
2. MongoDB Documentation – https://www.mongodb.com/docs/
3. Node.js Official Docs – https://nodejs.org/en/docs/
4. Express.js Guide – https://expressjs.com/
5. W3Schools HTML, CSS, JS Reference – https://www.w3schools.com/
6. MDN Web Docs (Mozilla Developer Network) – https://developer.mozilla.org/
7. Postman API Platform – https://www.postman.com/
8. Draw.io (Diagrams.net) – https://draw.io/
9. Figma – UI/UX Design Tool – https://www.figma.com/
10. GeeksforGeeks Articles on LMS Systems – https://www.geeksforgeeks.org/
11. JWT.IO – JSON Web Token Documentation – https://jwt.io/introduction
12. OWASP Security Guidelines – https://owasp.org/
13. Research Paper: "A Survey on Learning Management Systems" – International Journal of Computer Applications (IJCA), 2019
14. Moodle: btech.glsmoodle.in