A Project Report

on

"NeoLearn – AI-Powered Learning Platform"

By

Shadakwala Mahirabanu (2303396160011)
Sharama Khushi (2303396160012)
Bariya Parth (2303396160060)
(5th sem Computer Engineering Department)

INTERNAL GUIDE

Prof. Ashish Pandey
Prof. Ankit Ramani
(Computer Engineering Department)

Submitted To

PARUL INSTITUTE OF ENGINEERING & TECHNOLOGY (DIPLOMA STUDIES)

LIMDA, VAGHODIA, VADODARA

<u>CERTIFICATE</u>

This is to Certify that Major Project Report by **Shadakwala Mahirabanu(2303396160011), Sharama Khushi(2303396160012), Bariya Parth(2303396160060)** of Computer Engineering Department of **PARUL INSTITUTE OF ENGINEERING & TECHNOLOGY (DIPLOMA STUDIES), LIMDA** is the record of work carried out by them under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination.

GUIDE(1)	**HOD**	**PRINCIPAL**
Prof. Ashish Pandey	**Dr. Hetal Bhaidasna**	**Prof. (Dr.) Ruchi Shrivastava**
Department of Computer Engineering	Department of Computer Engineering	Principal
PIET(DS), LIMDA	PIET(DS), LIMDA	PIET(DS), LIMDA

GUIDE(2)

Prof. Ankit Ramani

Department of Computer Engineering

PIET(DS), LIMDA

ACKNOWLEDGEMENT

ABSTRACT

This report presents a comprehensive **NeoLearn – AI-Powered Learning Platform** developed using **Flask** and **MySQL**. The system aims to streamline various educational processes, including user management (admin, teacher, student roles), material sharing, quiz management, and interactive features like an AI chatbot and a voting system. It provides a centralized platform for administrators to oversee operations, teachers to manage content and assess students, and students to access learning resources, take quizzes, and engage in collaborative activities. The system focuses on enhancing communication, organization, and interaction within an educational environment.

TABLE OF CONTENTS

No.	Chapter Name	Page No.

LIST OF FIGURES

LIST OF IMPLEMENTATION FIGURE

Chapter-1 INTRODUCTION

1.1 Introduction

NeoLearn is a modern, AI-Powered Learning Platform designed to transform traditional educational environments into dynamic, interactive, and efficient digital spaces. Built with the powerful Flask framework and a robust MySQL database, the system offers a centralized solution for managing all aspects of classroom operations. The platform's core strength lies in its ability to cater to three distinct user roles—administrators, teachers, and students—each with a customized dashboard and a unique set of tools. Administrators can oversee user registration and system-wide communication, while teachers are empowered with features for uploading materials and creating quizzes. Students, in turn, gain access to essential learning resources, can take quizzes, and engage with an integrated AI chatbot for academic support. By streamlining administrative tasks and enhancing teacher-student interaction through features like a dynamic voting system, NeoLearn aims to foster a more organized, engaging, and effective learning experience for all users.

1.2 Existing System

Currently, many educational institutions rely on fragmented systems or manual processes for managing classes, sharing materials, conducting assessments, and facilitating communication. These existing methods often lead to inefficiencies, inconsistencies in data management, and limited engagement opportunities for students.

1. Google Classroom

-Assignments and grading

-Material sharing

-Calendar integration

-Works with Google Meet

2. Byju's

- Animated video lessons

- Interactive tests

- Mobile app available

3. Microsoft Teams for Education

- Live classes

- Assignments

- File sharing

- Calendar/events

4. Moodle

- Quizzes & forums

- Material uploads

- Calendar/events

- Highly customizable

5. Toppr

- Doubt clearing

- Practice tests

- Video lessons

1.3 Proposed System

The proposed NeoLearn system aims to overcome the limitations of existing systems by offering a unified, intelligent platform. Our work focuses on developing a robust web application using Flask and MySQL, incorporating features such as secure user authentication, role-based access control, dynamic content management, an interactive AI chatbot, and a flexible voting system. The platform will streamline administrative tasks, empower teachers with efficient tools, and provide students with an engaging and accessible learning environment.

1. **Centralized Learning Platform**

   – Combines quizzes, materials, live classes, feedback, and events in one portal.

2. **Role-Based Access**

   – Separate dashboards for Admin, Teacher, and Student.

3. **AI Chatbot Assistance**

   – Built-in LLaMA 3 chatbot for instant doubt solving and academic queries.

4. **Face Detection**

   – Uses MediaPipe + OpenCV to verify student identity for attendance/exams.

5. **Smart Quiz Management**

   – Automatic scoring, result storage, and analytics.

6. **Study Material Management**

   – Upload, organize, and access materials by subject/class.

7. **Voting & Feedback System**

   – Students can vote on events, sessions, or give feedback anonymously.

8. **Live Classes Integration**

   – Built-in video conferencing using Jitsi Meet API.

9. **Event Calendar**

   – FullCalendar integration for scheduling classes, exams, and activities.

1.4 Advantages

NeoLearn offers several advantages:

1. **Enhanced Efficiency**: Automates administrative tasks, material distribution, and quiz management.

2. **Improved Communication**: Facilitates seamless messaging between admins, teachers, and students.

3. **Centralized Resources**: Provides a single platform for all learning materials and assessments.

4. **Interactive Learning**: Integrates an AI chatbot for instant query resolution and a voting system for engagement.

5. **Secure Environment**: Implements robust authentication and role-based access control.

6. **Data-Driven Insights**: Enables tracking of quiz scores and voting history for better decision-making.

1.5 Scope

The scope of NeoLearn encompasses:

- User management for Admin, Teacher, and Student roles.
- Secure login and logout functionalities.
- Dashboard interfaces tailored to each user role.
- Material upload, storage, and access.
- Quiz creation, administration, and score tracking.
- Messaging system for inter-role communication.
- An AI-powered chatbot for student support.
- A dynamic voting system for polls and feedback.
- Calendar event management.
- Responsive design for accessibility across devices.

Chapter: 2 SYSTEM REQUIREMENT STUDY

2.1 Feasibility Study

A feasibility study was conducted to assess the viability of developing NeoLearn.

1. Technical Feasibility

- **Technologies**: The project utilizes established technologies like **Flask (Python)** and **MySQL**, which are well-suited for building robust and scalable web applications.
- **AI Integration**: The integration of an AI model, specifically **Llama3**, is technically feasible via a local API, confirming that the core functionality is within reach of the development team.

2. Operational Feasibility

- **User-Friendly Design**: The system is designed with an intuitive interface to minimize the learning curve and training requirements for users, making it easy to integrate into existing educational practices.
- **Workflow Enhancement**: Instead of disrupting existing processes, NeoLearn aims to enhance them by providing a centralized and efficient platform for managing classes, sharing materials, and communication.

3. Economic Feasibility

- **Open-Source Technologies**: The project leverages open-source tools such as Flask and MySQL, which helps in minimizing licensing costs.
- **Cost-Effective Solution**: The efficient allocation of development resources, combined with the use of free, open-source software, makes the overall solution cost-effective.

2.2 Requirements of Project

The project requirements are categorized into functional and non-functional requirements:

Functional Requirements:

- **User Management**: Registration, login, role assignment (Admin, Teacher, Student).
- **Content Management**: Upload, edit, delete study materials; create, modify, view quizzes.
- **Communication**: Send messages to specific roles or all users.
- **Assessment**: Administer quizzes, calculate and store scores, view score reports.
- **Interactive Features**: Implement an AI chatbot for student queries; create and participate in voting polls.
- **Reporting**: View student voting history; generate quiz score reports.
- **Event Management**: Create and view calendar events.

Non-Functional Requirements:

- **Security**: User authentication, password hashing, protection against common web vulnerabilities.
- **Usability**: Intuitive user interface, easy navigation, clear feedback.
- **Performance**: Fast loading times, efficient database queries.
- **Scalability**: Ability to handle an increasing number of users and data.
- **Reliability**: System uptime, data integrity, error handling.
- **Maintainability**: Modular code, clear documentation.
- **Compatibility**: Support for modern web browsers.

2.3 Tools & Technology

1. Frontend Technologies

- **HTML5** – For structuring the web pages.
- **CSS3** – For styling and layout design.
- **JavaScript** – For interactive UI components.

2. Backend Technologies

- **Flask (Python Framework)** – For building the web application and handling backend logic.
- **Python 3.x** – Core programming language for application logic and AI integration.

3. Database

- **MySQL** – Relational database for storing user data, quiz results, materials, and event details.

4. AI

- **LLaMA 3** – AI chatbot model for answering student queries.
- **MediaPipe + OpenCV** – For face detection in attendance and verification processes.
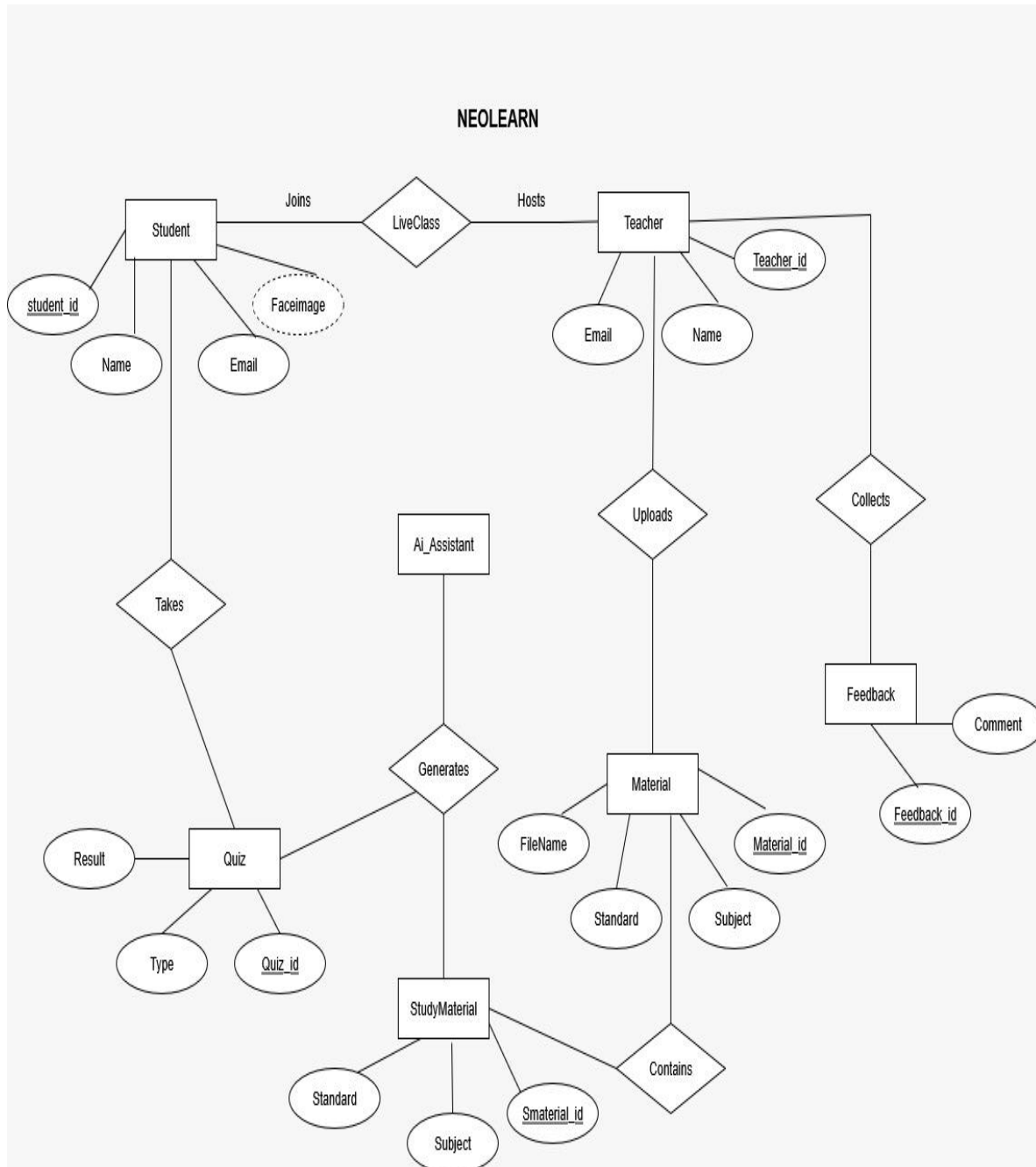
5. APIs & Libraries

- **Jitsi Meet API** – For live video conferencing and online classes.
- **FullCalendar** – For scheduling events and activities.
- **Flask-Login** – For secure authentication.
- **Flask-SQLAlchemy** – For database ORM handling.

6. Development Tools

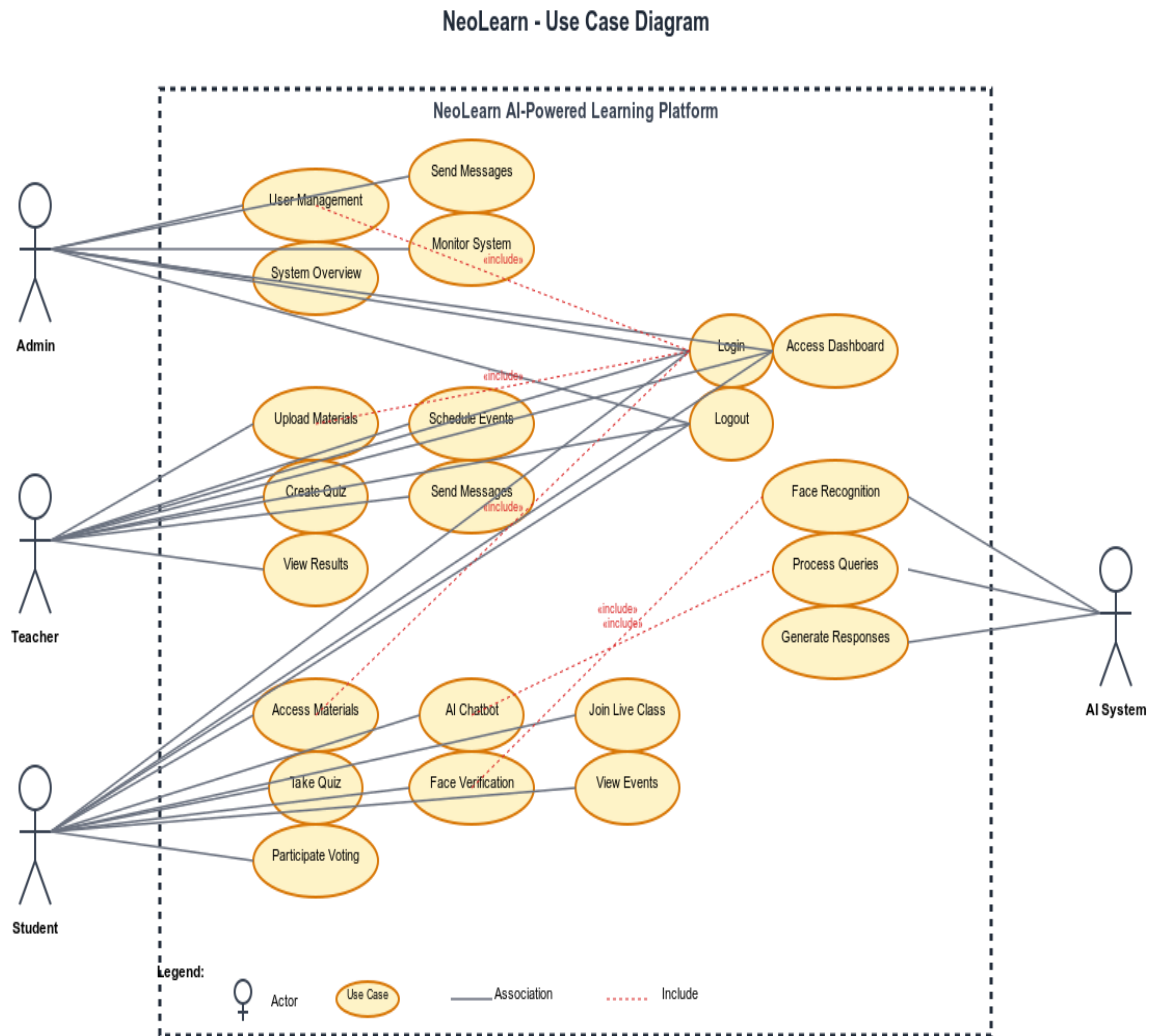- **VS Code** – Code editors for development.
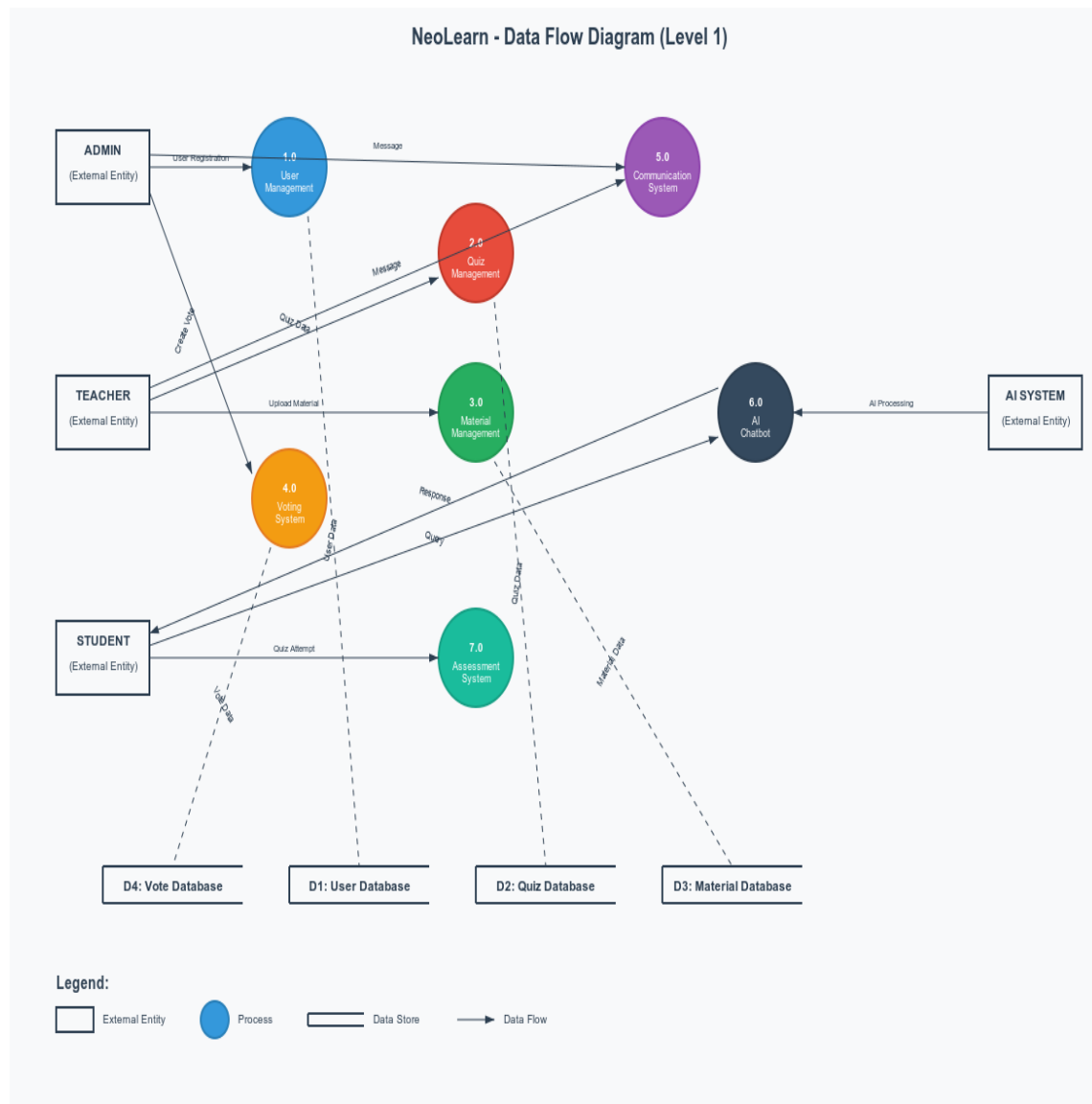
Chapter: 3 DIAGRAM'S

3.1 ER Diagram

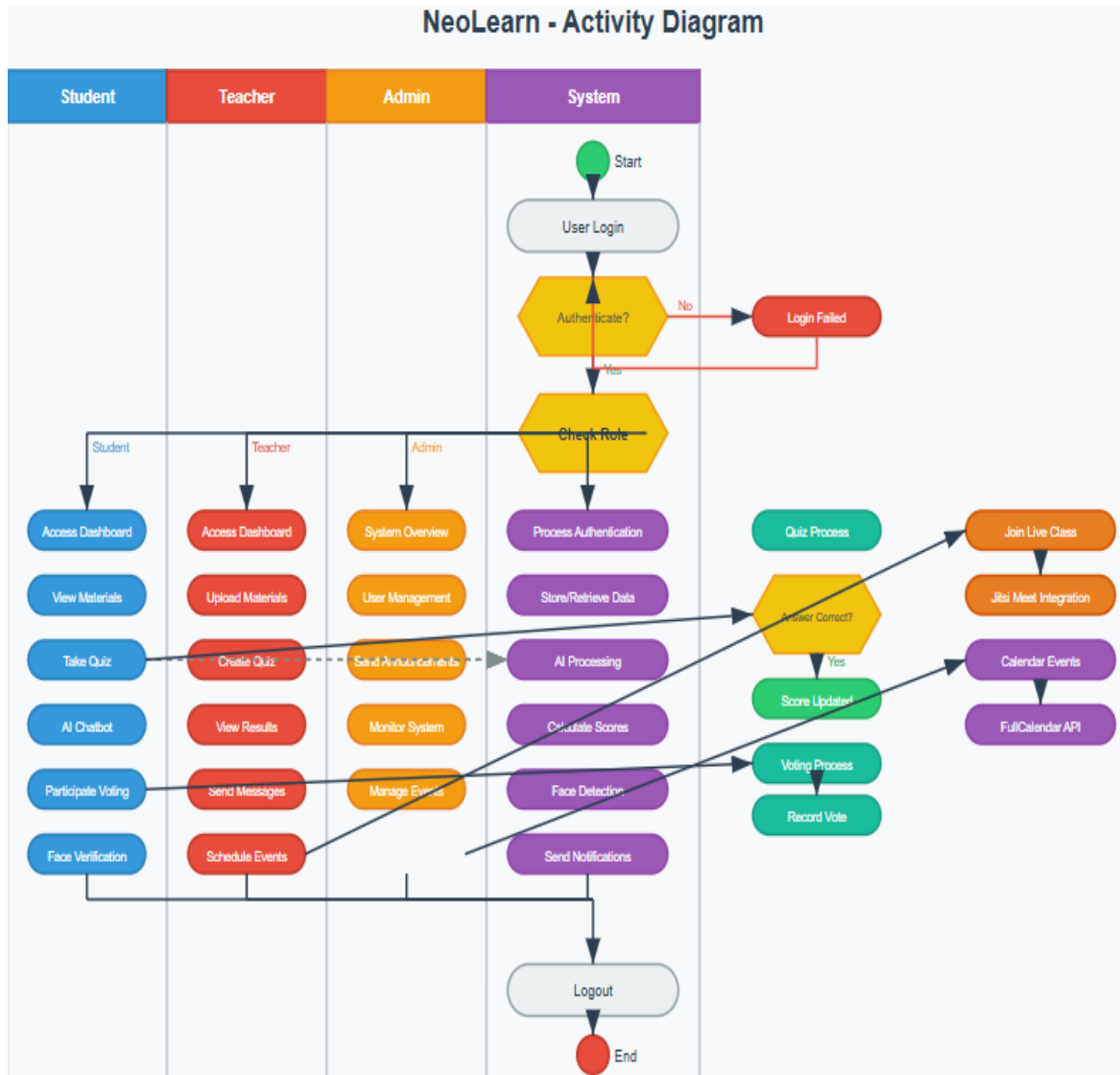
NEOLEARN

1. ER Diagram

3.2 Use case Diagram

NeoLearn - Use Case Diagram

2. Use Case Diagram

3.3 DFD Diagram

NeoLearn - Data Flow Diagram (Level 1)

3. DFD Diagram

3.4 Activity Diagram

4. Activity Diagram

3.5 Class Diagram

NeoLearn - Class Diagram

User

- user_id: int
- username: string
- email: string
- password: string
- role: UserRole
- created_at: datetime

+ login(): boolean
+ logout(): void

Material

- material_id: int
- title: string
- file_path: string
- upload_date: datetime
- teacher_id: int

+ uploadFile(): void

Quiz

- quiz_id: int
- title: string
- description: string
- created_date: datetime
- teacher_id: int

+ createQuiz(): void

QuizResult

- result_id: int
- quiz_id: int
- student_id: int
- score: int
- completed_at: datetime

+ calculateScore(): int

Admin

+ manageUsers(): void
+ systemOverview(): void
+ sendMessages(): void
+ monitorSystem(): void

Teacher

+ uploadMaterials(): void
+ createQuiz(): void
+ viewResults(): void
+ scheduleEvents(): void
+ sendMessages(): void

Student

+ accessMaterials(): void
+ takeQuiz(): void
+ participateVoting(): void
+ useChatbot(): void
+ faceVerification(): void
+ joinLiveClass(): void
+ viewEvents(): void

Message

- message_id: int
- sender_id: int
- content: string
- recipient_role: string
- sent_at: datetime

+ sendMessage(): void

Vote

- vote_id: int
- student_id: int
- question: string
- option: string
- voted_at: datetime

+ castVote(): void

Event

- event_id: int
- title: string
- description: string
- event_date: datetime
- created_by: int

+ scheduleEvent(): void

ChatHistory

- chat_id: int
- student_id: int
- query: string
- response: string
- timestamp: datetime

+ saveChatLog(): void

AISystem

- model_name: string
- api_endpoint: string

+ processQuery(): string
+ generateResponse(): string
+ faceRecognition(): boolean

FaceVerification

- verification_id: int
- student_id: int
- image_path: string
- verified: boolean
- verified_at: datetime

+ verifyFace(): boolean

Legend:

ClassName
- attribute: type
+ method(): type

→ Inheritance

— Association

1..* = One to Many

5. Class Diagram

3.6 Data Dictionary
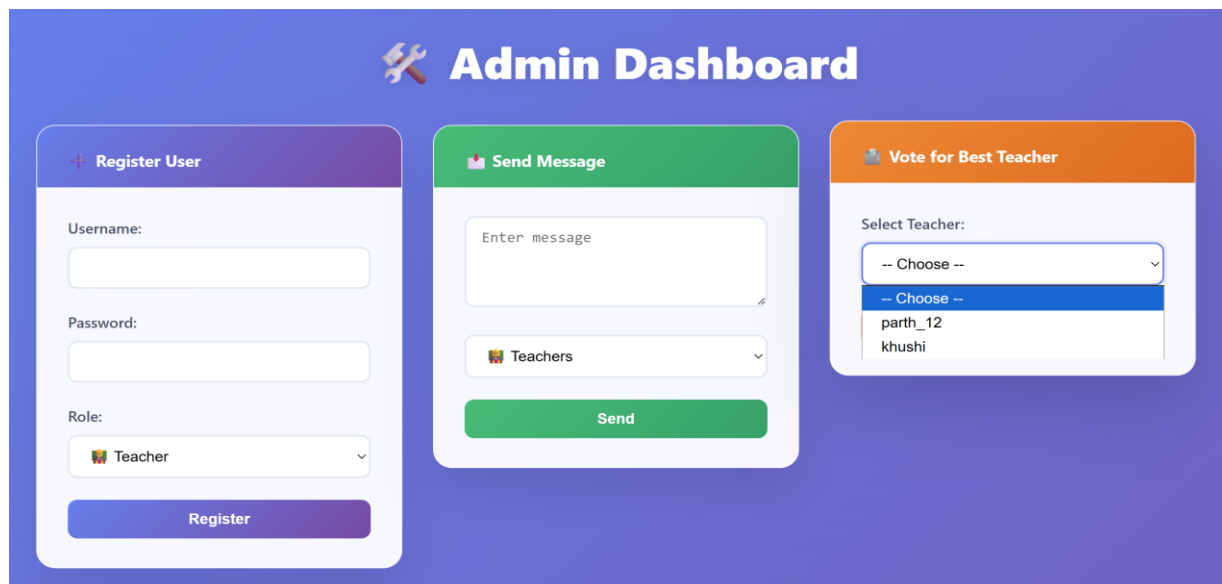
QUIZ
- quiz_id (PK)
- title
- description
- created_by (FK)

MATERIAL
- material_id (PK)
- title
- file_path
- uploaded_by (FK)

VOTE
- vote_id (PK)
- question
- options
- created_by (FK)

USER
- user_id (PK)
- username
- email
- password_hash
- role
- full_name

MESSAGE
- message_id (PK)
- sender_id (FK)
- subject
- content

QUIZ_RESULT
- result_id (PK)
- user_id (FK)
- quiz_id (FK)
- score

EVENT
- event_id (PK)
- title
- event_date
- created_by (FK)

VOTE_RESPONSE
- response_id (PK)
- user_id (FK)
- vote_id (FK)
- response

Legend:
- Primary Key (PK)
- Foreign Key (FK)
- Relationship

6. Data Dictionary

Chapter: 4 IMPLEMENTATION

4.1 Login Dashboard

2. Login Dashboard

4.2 Admin dashboard

Register,send message ,vote section, Custom voting,vote history, Vote result,delet vote

2. Register User, Send message, Vote section

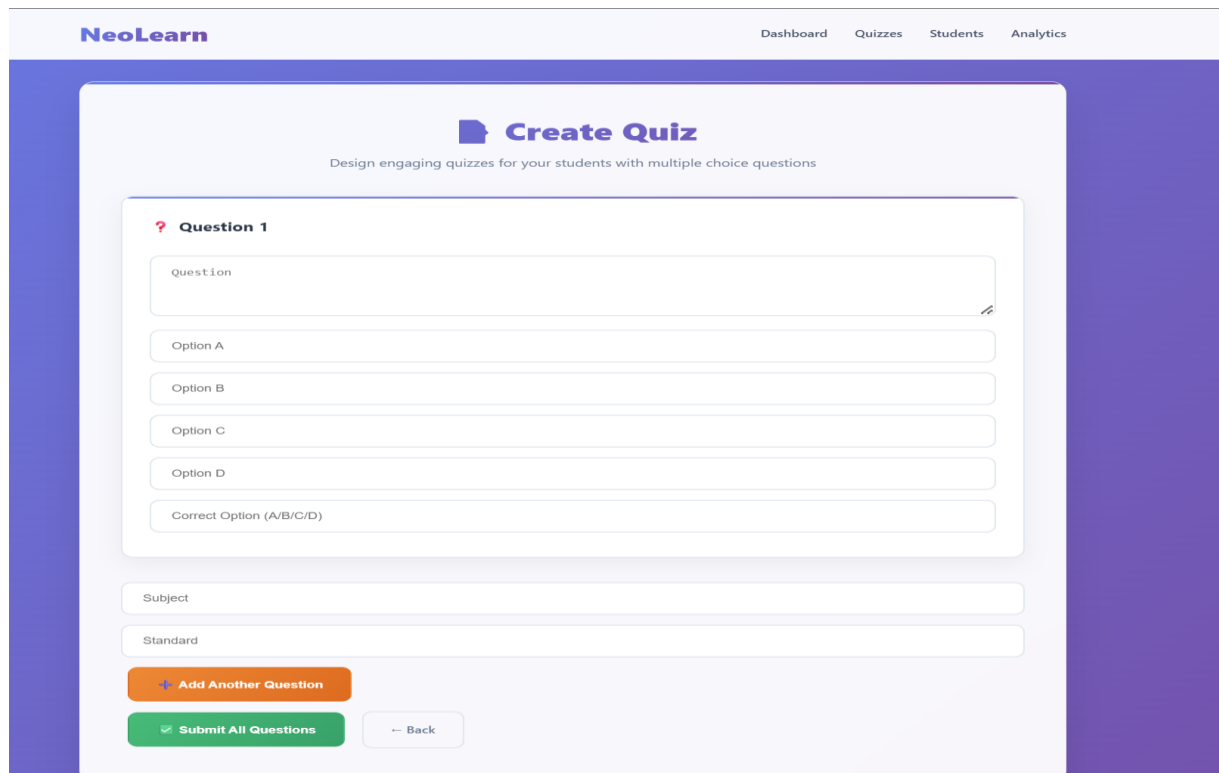**2.Custom voting, Voting History**

2. Voting result, Delete vote

4.3 Teacher Dashboard
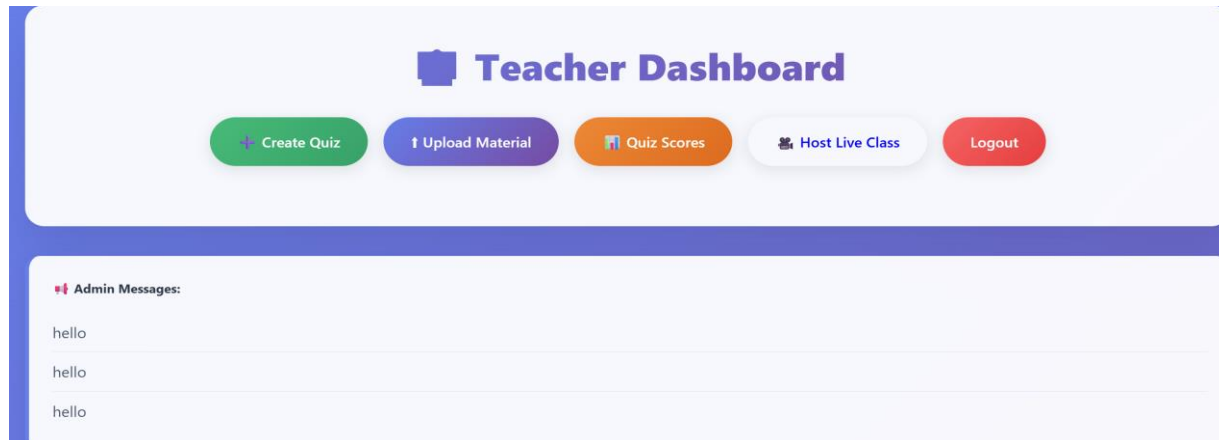
3. Teahcer Dashboard

4.4 Creating Quiz and Quiz Scores

4. Creating Quiz and Quiz Scores

📊 Quiz Scores

Name	Standard	Subject	Score	Total	Taken At
parthbariya	2	maths	1	1	2025-06-21 14:12:01
parthbariya	2	maths	1	1	2025-06-21 14:12:27
parthbariya	2	maths	3	3	2025-06-21 14:16:51
parth	2	maths	2	3	2025-06-29 03:29:51
mahira	3	coding	0	1	2025-06-29 03:30:48
mahira	3	coding	1	1	2025-06-29 03:31:30
parth	2	maths	3	3	2025-07-10 09:05:41

Back

4.Quiz Score

4.5 Upload Material

5. Upload Material

Uploaded Materials

Standard	Subject	Title	Preview	Actions
1	maths	study	PDF File	👁 View ↓ Download ✏ Edit 🗑 Delete
2	maths	study	PDF File	👁 View ↓ Download ✏ Edit 🗑 Delete

5. Uploading Materials

4.6 Live Meeting

Live Class: classroom_khushi

jitsi

Join meeting
Classroom Khushi

khushi

Join meeting

You need to enable microphone and camera access

K
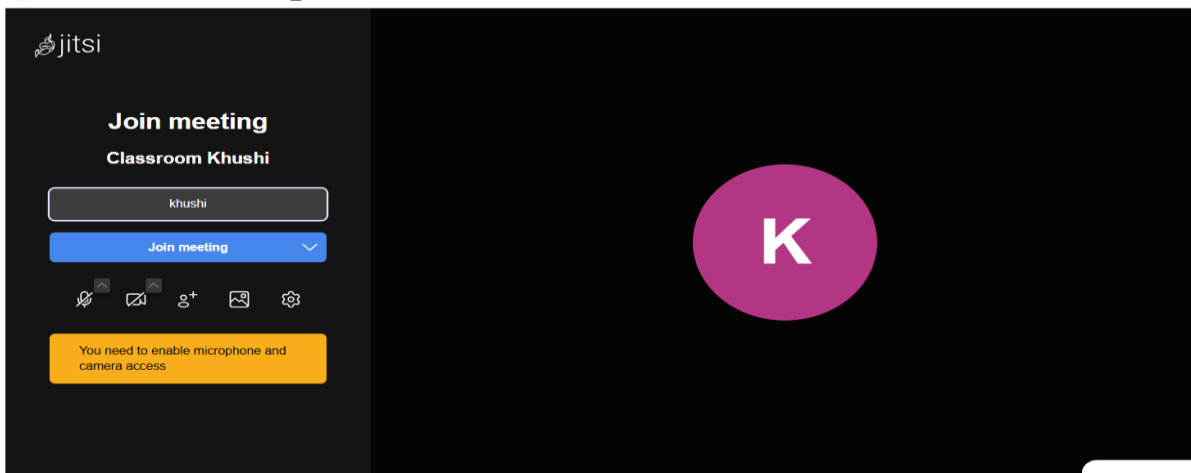
Participants: 1
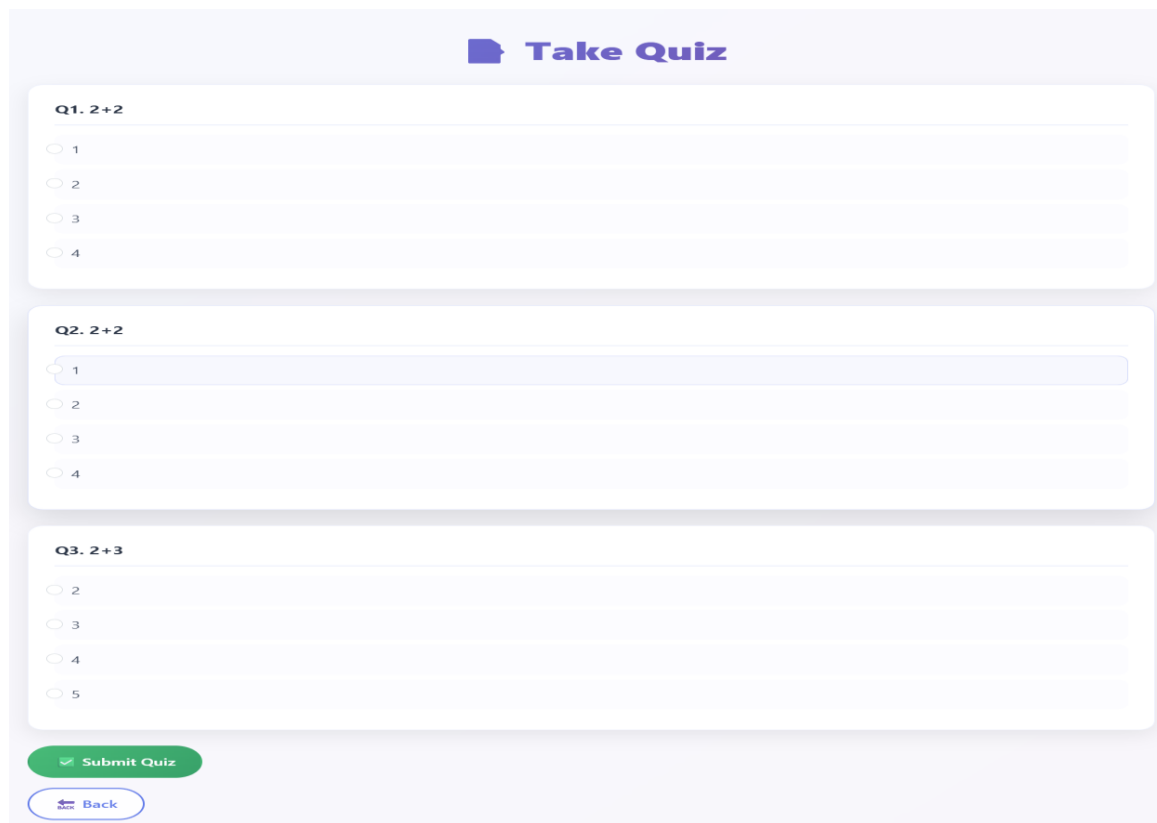
🔊 Download Attendance

6. Live Meeting
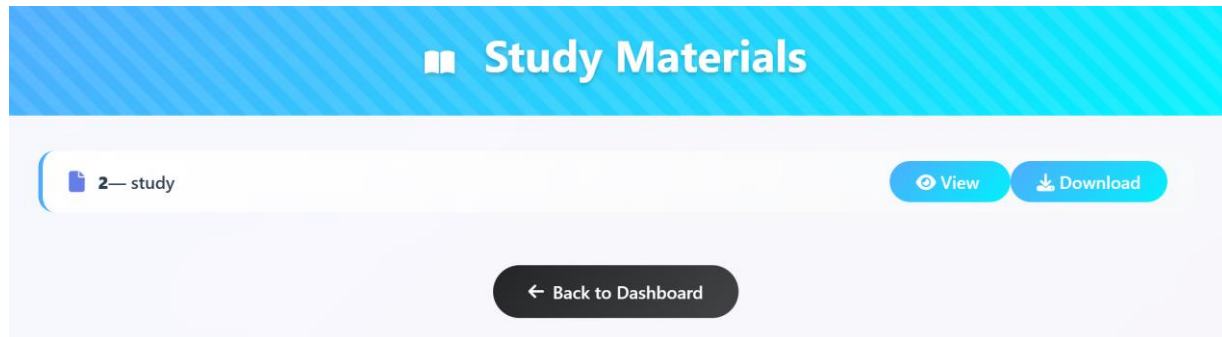
4.7 Student Dashboard
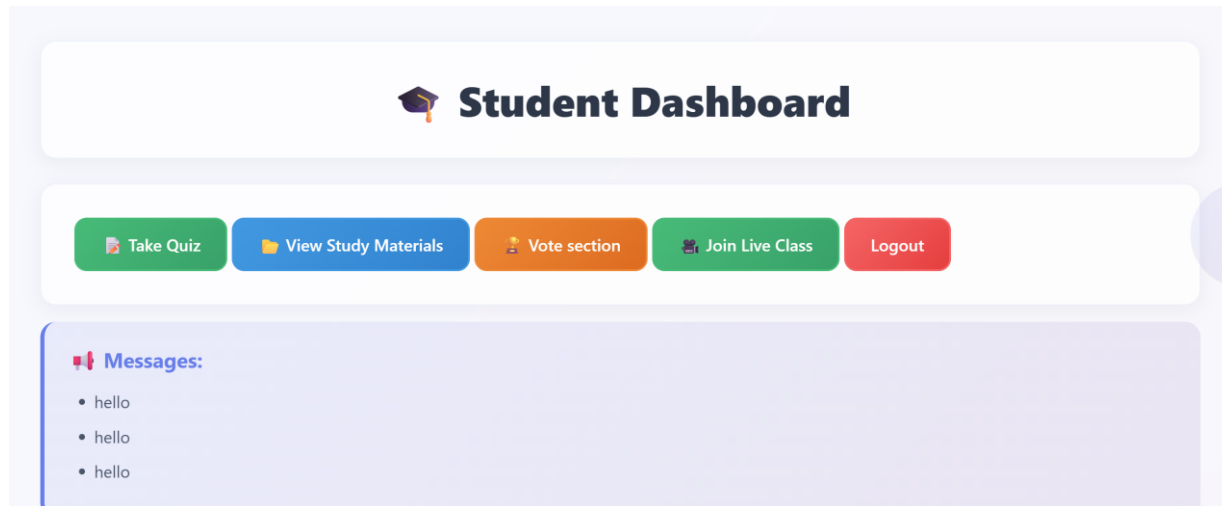
7. Student Dashboard

4.8 Take Quiz

8. Take Quiz

4.9 View Study Material
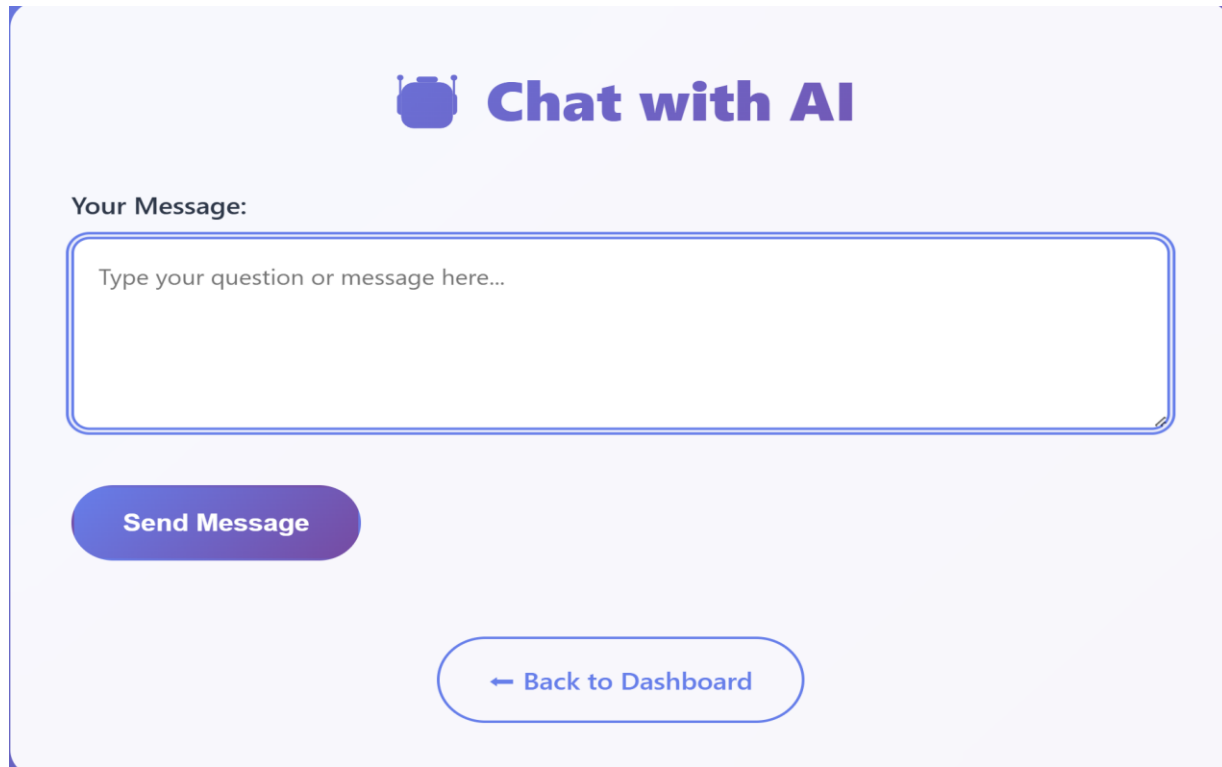
9. View Study Material

4.10 Vote Section
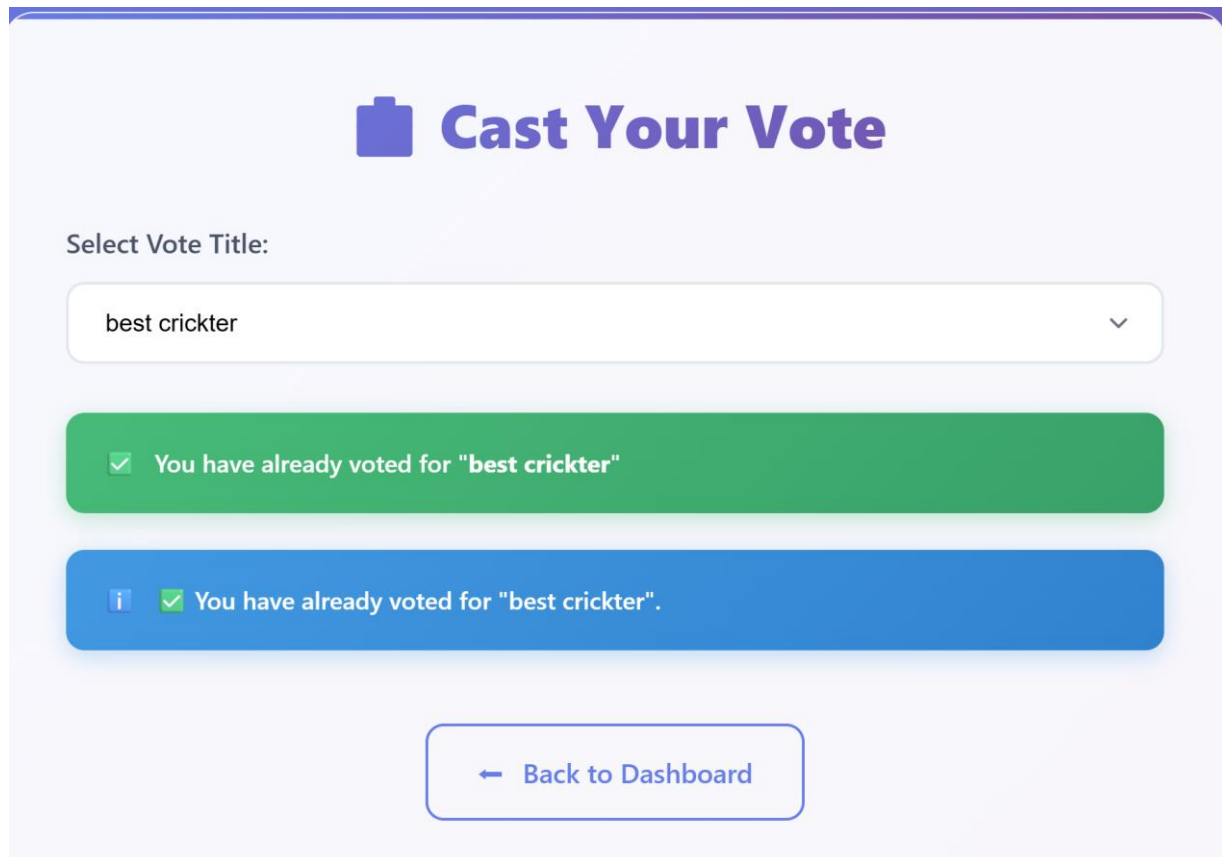
10. Vote Section

4.11 AI Integration

11. AI Integration

Chapter: 5 CONCLUSION

NeoLearn is designed to overcome the limitations of existing educational platforms by bringing together all essential academic tools into one centralized system. With features such as AI chatbot assistance**,** face detection for verification**,** role-based dashboards**,** smart quiz management**,** study material sharing**,** voting and feedback collection, and live class integration, it offers a comprehensive solution for modern learning needs.Built using Flask, MySQL, and AI technologies, NeoLearn ensures scalability, security, and ease of use.By integrating advanced features with a simple, user-friendly interface, NeoLearn enhances both teaching efficiency and student engagement, making it a valuable step toward the future of digital education.

Chapter: 6 FUTURE SCOPE

The future enhancements for NeoLearn include:

- **Personalized Learning Paths**: Implementing AI algorithms to suggest tailored learning materials and quizzes based on student performance.
- **Gamification**: Integrating game-like elements such as badges, leaderboards, and rewards to increase student engagement.
- **Real-time Collaboration Tools**: Adding features like live chat for group projects or virtual classrooms.
- **Advanced Analytics**: Providing more in-depth reports and insights for teachers and admins on student performance and engagement trends.
- **Mobile Application**: Developing native mobile applications for iOS and Android to enhance accessibility.
- **Integration with External APIs**: Connecting with other educational tools or platforms (e.g., calendaring services, video conferencing).
- **Automated Content Curation**: Using AI to help teachers find and suggest relevant external learning resources.

Chapter: 7 BIBLIOGRAPHY

- **Flask Documentation** – https://flask.palletsprojects.com/

- **MySQL Documentation** – https://dev.mysql.com/doc/

- **LLaMA 3 Model Overview** – Meta AI Research

- **Jitsi Meet API Documentation** – https://jitsi.github.io/handbook/docs/dev-guide/dev-guide-iframe

- **FullCalendar Documentation** – https://fullcalendar.io/docs

- **OpenCV Documentation** – https://docs.opencv.org/

- **MediaPipe Documentation** – https://developers.google.com/mediapipe

- **Python 3 Official Documentation** – https://docs.python.org/3/

- **Bootstrap Documentation** – https://getbootstrap.com/