A Theme Based Project Report

on

EDUTRACK

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

in

Computer Science and Engineering

by

Gunturi.Gayathri (2451-23-733-203)

Gannoju. Anjali (2451-23-733-215)

Akula. Kalyanram (2451-23-733-223)

Under the Guidance of

Dr.Rajesh Kulkarni

Associate Prof., Dept. of CSE, MVSREC

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MATURI VENKATA SUBBA RAO ENGINEERING COLLEGE

(An Autonomous Institution, Sponsored by Matrusri Education Society - Estd. 1980) Affiliated to Osmania University & Recognized by AICTE

Nadergul (PO), Balapur (M), Hyderabad, Telangana, India – 501510

Academic Year: 2024 – 2025

CERTIFICATE

*This is to certify that the Theme Based project work entitled "**EduTrack**" is a bonafide work carried out by **Gunturi.Gayathri** (2451-23-733-203),**Gannoju.Anjali** (2451-23-733-215), **Akula.Kalyanram**(2451-23-733-223) in partial fulfillment of the requirements for the award of degree of Bachelor of Engineering in Computer Science and Engineering from Maturi Venkata Subba Rao(MVSR) Engineering College, affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2024-2025 under our guidance and supervision.*

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide **Project Coordinator** **Head of the Department**

Dr.Rajesh Kulkarni **Prof.J. Prasanna Kumar**

Associate Prof.,

Dept. of CSE, MVSREC Professor &
HoD,CSE,
MVSREC

External Examiner

DECLARATION

This is to certify that the work reported in the present Theme Based project entitled "**EduTrack**" is a record of bonafide work done by us in the Department of Computer Science and Engineering, MVSR Engineering College, Osmania University. The reports are based on the work done entirely by us and not copied from any other source. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

Gunturi.Gayathri(2451-23-733-203)

Gannoju.Anjali(2451-23-733-215)

Akula.Kalyanram(2451-23-733-223)

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and indebtedness to our project guide **Dr.Rajesh Kulkarni**, Associate. Prof., Dept. of CSE, MVSREC for his valuable suggestions and interest throughout the course of this project.

We are also thankful to our Project coordinators **Dr. Vikram Narayandas, P. Subhashini Dr. Namitha Parati and Dr. Rajesh Kulkarni**. We also thank **Prof. J. Prasanna Kumar**, HoD, Dept. of CSE, MVSREC and **Prof. Vijaya Gunturu**, Principal, MVSREC for providing excellent infrastructure and a nice atmosphere for completing this partial fulfillment of the requirements for the award of Bachelor of Engineering in Computer Science and Engineering.

Finally, we would like to take this opportunity to thank our family and friends for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

Gunturi.Gayathri (2451-23-733-203)

Gannoju.Anjali (2451-23-733-215)

Akula.Kalyanram(2451-23-733-223)

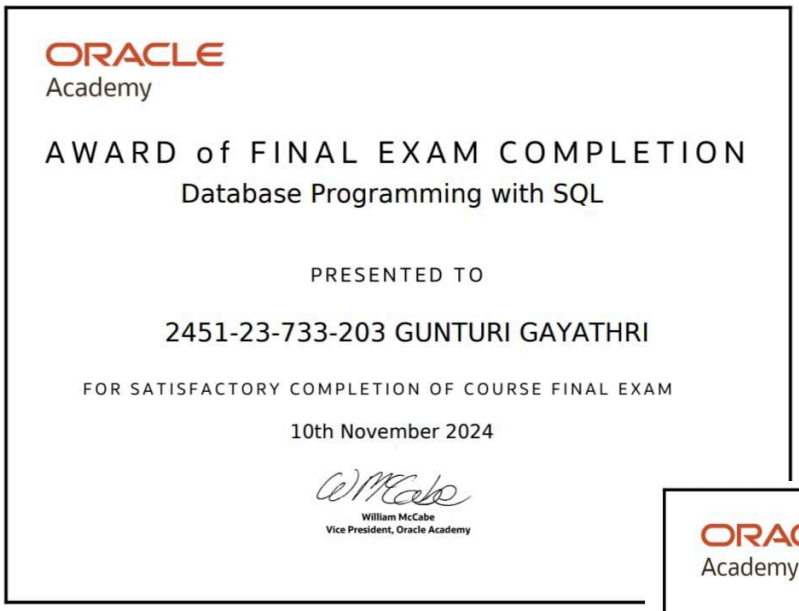
ABSTRACT

The modern education system demands seamless access to academic resources and effective management tools to enhance student learning outcomes. However, students often face challenges in tracking their attendance, accessing academic results, managing certificates, and engaging with quizzes and assignments. These inefficiencies can lead to disengagement and lower academic performance.

EduTrack is an innovative, interactive platform designed to address these challenges by providing a unified digital ecosystem for students and faculty. It allows students to track attendance records in real-time, view academic results and certificates, and participate in faculty-assigned quizzes and assignments through a centralized portal. Additionally, the platform includes gamification elements, automatic grading, and progress tracking to create a personalized and engaging learning experience.

For faculty, *EduTrack* simplifies the process of creating and managing quizzes, assignments, and feedback, enabling real-time updates for students. The platform's seamless integration with existing Learning Management Systems (LMS) ensures smooth adoption and scalability. By streamlining academic and administrative tasks, *EduTrack* enhances engagement, fosters a collaborative learning environment, and empowers students and faculty alike to achieve their educational goals efficiently.

CERTIFICATIONS


ORACLE
Academy

AWARD of FINAL EXAM COMPLETION

Database Programming with SQL

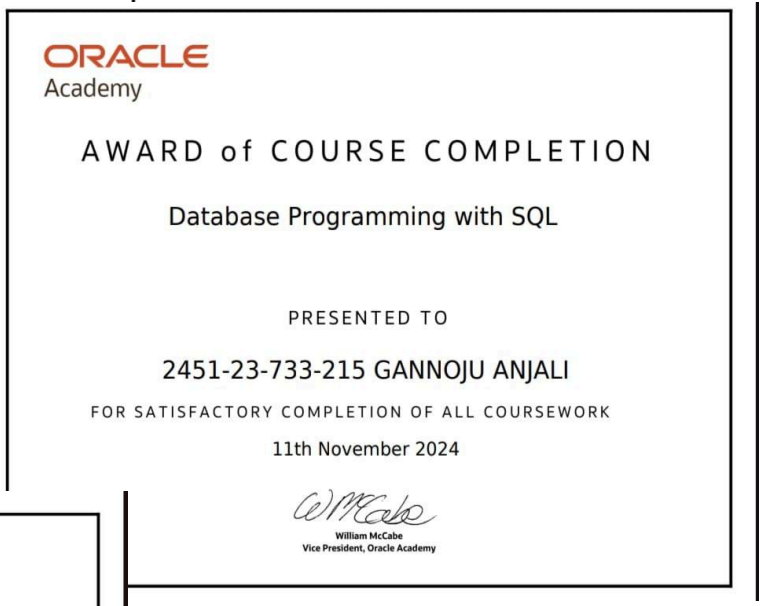
PRESENTED TO

2451-23-733-203 GUNTURI GAYATHRI

FOR SATISFACTORY COMPLETION OF COURSE FINAL EXAM

10th November 2024

William McCabe
Vice President, Oracle Academy


ORACLE
Academy

AWARD of COURSE COMPLETION

Database Programming with SQL

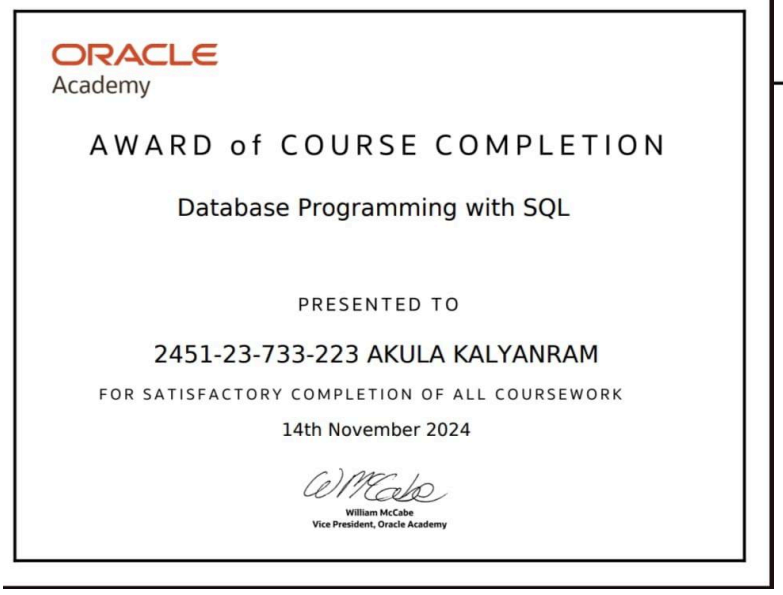
PRESENTED TO

2451-23-733-215 GANNOJU ANJALI

FOR SATISFACTORY COMPLETION OF ALL COURSEWORK

11th November 2024

William McCabe
Vice President, Oracle Academy


ORACLE
Academy

AWARD of COURSE COMPLETION

Database Programming with SQL

PRESENTED TO

2451-23-733-223 AKULA KALYANRAM

FOR SATISFACTORY COMPLETION OF ALL COURSEWORK

14th November 2024

William McCabe
Vice President, Oracle Academy

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision

To revolutionize education by providing an interactive platform that enhances engagement, streamlines academic management, and empowers students and educators to achieve excellence.

Mission

- ➢ To create an engaging platform that simplifies access to assignments, attendance, and results, enhancing student productivity.
- ➢ To offer a centralized hub for managing academic certificates and course references efficiently.
- ➢ To integrate advanced tools for personalized learning, empowering students to excel academically and professionally.
- ➢ To promote seamless collaboration between students and faculty through intuitive and innovative solutions.

Program Educational Objectives (PEOs)

The Bachelor's program in Computer Science and Engineering is aimed at preparing graduates who will: -

PEO-1: Achieve recognition through demonstration of technical competence for successful execution of software projects to meet customer business objectives.

PEO-2: Practice life-long learning by pursuing professional certifications, higher education, or research in the emerging areas of information processing and intelligent systems at a global level.

PEO-3: Contribute to society by understanding the impact of computing using a multidisciplinary and ethical approach.

Program Outcomes (POs)

PO 1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems.

PO 2: Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO 3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.

PO 4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO 6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.

PO 8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions

PO 11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOS)

PSO-1: Demonstrate competence to build effective solutions for computational real-world problems using software and hardware across multi-disciplinary domains.

PSO-2: Adapt to current computing trends for meeting the industrial and societal needs through a holistic professional development leading to pioneering careers or entrepreneurship.

COURSE OBJECTIVES AND OUTCOMES

Course Title: Theme Based Project

Course Code: U22PW481AL

Course Objectives

- Enhanced Academic Support
- Streamline Assignment and Quiz Management
- Efficient attendance & grade tracking.
- Real-time notification and reminders

Course Outcomes

- ➢ Apply academic knowledge to develop a unified platform for student engagement and academic tracking.
- ➢ Assess and implement technical solutions for efficient integration of attendance, assignments, and certificates.
- ➢ Manage the development process, ensuring effective project planning, execution, and delivery.
- ➢ Use modern technologies to create a scalable and user-friendly system that supports both students and faculty.

List Of Figures

List Of Tables

TABLE OF CONTENTS

CHAPTER - 1

INTRODUCTION

EduTrack is a comprehensive web-based platform designed to streamline and enhance the educational experience for both students and educators.[1]It addresses common challenges faced by students, such as tracking academic records, managing assignments and quizzes, and monitoring attendance, while also providing instructors with powerful tools to manage and assess student performance.

The platform allows students to access and track their attendance, results, and academic progress in real time. It offers a centralized space where students can view their grades, download certificates, and engage with assignments and quizzes posted by mentors.

[2]EduTrack aims to make learning more interactive by facilitating direct submission of assignments, participation in quizzes, and receiving timely feedback.For educators, EduTrack provides an intuitive interface to manage quizzes, assignments, and attendance. Mentors can post assignments, create quizzes, and track student progress across subjects with ease. The system supports personalized assessments, allowing mentors to evaluate students based on their performance and provide constructive feedback to enhance learning outcomes. Additionally, the platform includes options for mentors to upload course references, additional learning materials, and create detailed quizzes with varied question types to assess student knowledge effectively.

With features like real-time progress tracking, automatic grading for quizzes, and detailed student performance reports,[3]EduTrack is designed to simplify administrative tasks, enabling instructors to focus more on teaching and less on managing paperwork. The platform also integrates gamification elements, where students can see their progress and be motivated by achievements, making the learning process more engaging.

By combining simplicity with powerful features, EduTrack provides an efficient solution for managing assignments, tracking attendance, and fostering better communication between students and instructors. The ultimate goal of the platform is to enhance the educational experience by making learning more accessible, transparent, and interactive for everyone involved.

1.1 Problem statement

Students face challenges in accessing academic records, tracking attendance, and managing quizzes and exams.[4] EduTrack offers a unified portal where students can view their attendance, results, certificates, and participate in faculty-posted quizzes and assignments, enhancing engagement.

1.2 Existing systems

Traditional education systems rely on multiple disconnected platforms for managing academic activities like attendance, assignments, and results. Students often struggle to track their progress across these various systems, and instructors face challenges in managing and grading assignments.[5]Additionally, accessing academic records, certificates, and tracking progress can be time-consuming and inefficient. This fragmentation leads to confusion and inefficiencies for both students and faculty. EduTrack addresses these challenges by offering a unified platform where students can view attendance, grades, assignments, and certificates in one place, streamlining the academic experience for both learners and instructors.

1.3Proposed System

The proposed EduTrack web application aims to transform the academic experience by addressing the challenges students and instructors face with managing academic records, assignments, and assessments.[6]This innovative system provides a unified platform where students can access their attendance, results, certificates, and assignments in one place. EduTrack enhances engagement by allowing instructors to post quizzes and assignments while providing real-time feedback to students. The platform is designed for ease of use, offering personalized learning experiences, tracking student progress, and promoting interactive learning. By integrating all essential academic features into a single system, EduTrack streamlines the learning process and improves accessibility for both students and faculty.

1.4 Scope : EDUTRACK is an academic management platform designed to provide faculty, students, and administrators with secure, role-based access to tools for managing attendance, grades, assignments, and schedules.[7]It features an intuitive interface accessible on both desktop and mobile. Future enhancements will include exam scheduling and feedback modules to further support academic management.

CHAPTER - 2

2. SYSTEM ARCHITECTURE SPECIFICATION

2.1 Software Requirements

- IDE Used : Visual Studio
- Frontend : HTML, CSS, JavaScript
- Backend : PHP
- Database: Mysql

2.1.1 Functional Requirements

1. **User Authentication**

- Users can create an account and log in using secure authentication.
- The system will store user credentials securely and allow users to access their personalized dashboards.

2. **Assignment and Quiz Management**

- Instructors can post assignments and quizzes.
- Students can view and attempt quizzes and assignments posted by their instructors.
- The system will allow instructors to provide real-time feedback on assignments and quizzes.

3. **Result and Attendance Tracking**

- Students can view their attendance records and academic results on the platform.
- The system will automatically update and store attendance and results in a secure database.
- Both instructors and students can track academic performance and progress.

4. **Certificate Management**

- The application will allow students to store and access certificates earned from completed courses or assessments.
- Certificates will be available for download or printing as required by the student.

5. **Progress and Feedback**

- The platform will provide personalized feedback based on student performance.
- Instructors can offer suggestions for improvement and track student progress over time.
- Students will receive detailed feedback for their submissions, including quizzes, assignments, and overall performance.

6. **Data Storage and Reporting**

- All data, including attendance, results, assignments, and feedback, will be securely stored in a database.
- The system will generate weekly progress reports summarizing assignments, quizzes, attendance, and feedback.
- Reports will be accessible to both instructors and students to monitor progress and improvements

2.1.2 Non Functional requirements

1. **Performance**

- The application will provide real-time responses to user interactions, ensuring minimal latency during usage.
- The system will generate and display weekly progress reports within a reasonable timeframe (no longer than 2-3 seconds).

2. **Security**

- User data, including personal information, academic records, and feedback, will be securely stored and transmitted using industry-standard encryption methods.
- The system will implement secure user authentication (e.g., password encryption, two-factor authentication) to protect sensitive data.

3. **Usability**

- The application will feature an intuitive and user-friendly interface that is easy to navigate, ensuring a seamless experience for both students and instructors.
- Clear instructions and real-time feedback will be provided to guide users through the platform, especially when completing assignments, quizzes, or reviewing performance data.

2.2 Hardware Requirements

1. **For Hosting/Server**:

- RAM: 8GB (or higher)
- Processor: Dual-core (Intel i5 or equivalent)
- Storage: 500GB SSD

- Internet Speed: 10 Mbps (or higher)

2. **For Mentor Side**:

- RAM: 4GB (minimum)
- Processor: Dual-core (Intel i3 or equivalent)
- Browser: Latest versions of Chrome, Firefox, Safari, or Edge
- Internet Speed: 5 Mbps (minimum)

3. **For Student Side**:

- RAM: 4GB (minimum)
- Processor: Dual-core (Intel i3 or equivalent)
- Browser: Latest versions of Chrome, Firefox, Safari, or Edge
- Internet Speed: 5 Mbps (minimum)

2.3 SYSTEM ARCHITECTURE

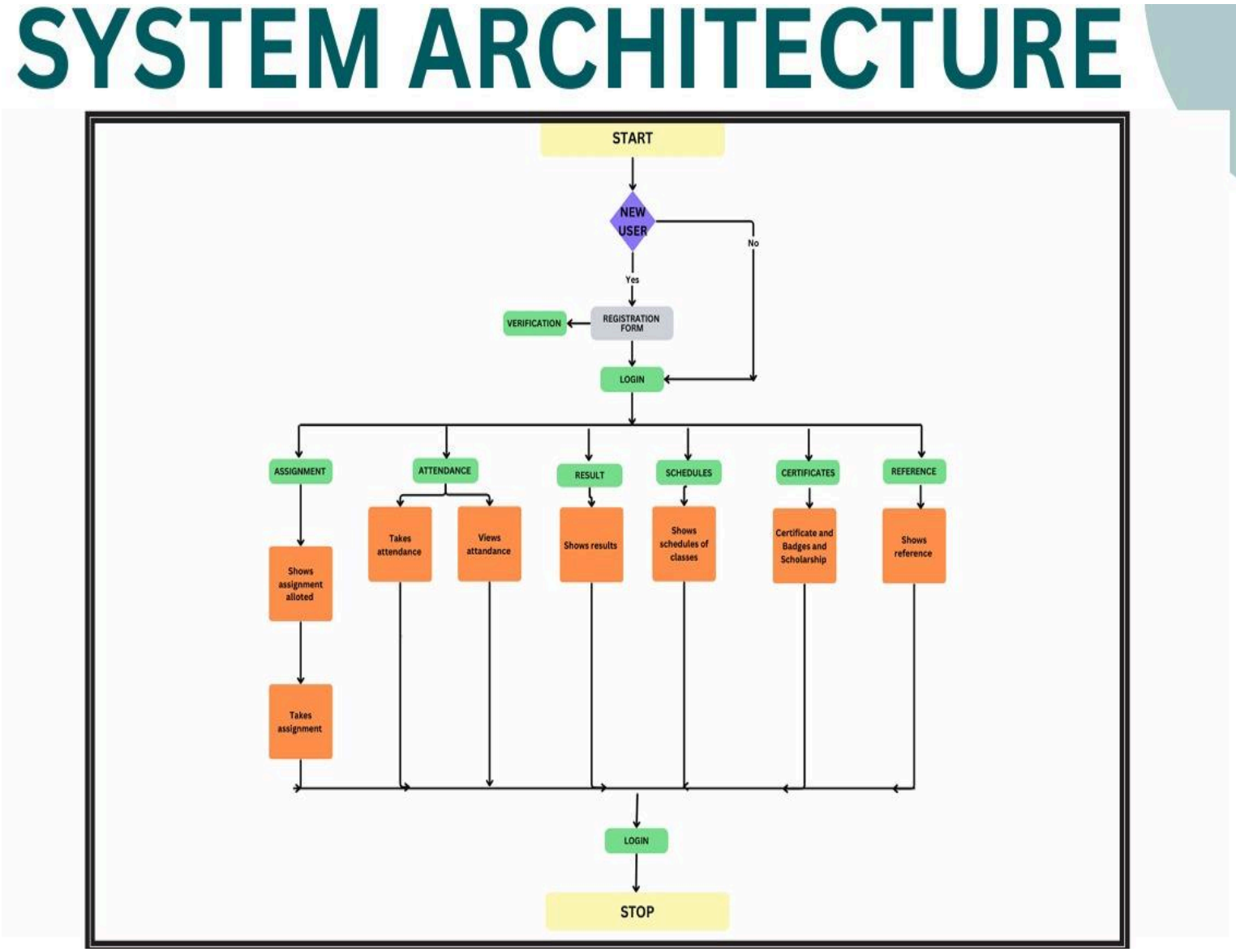


Fig 2.1 System Architecture

This Fig. outlines the structure and components that make up the EduTrack platform and to achieves their requirement in a single platform.

CHAPTER - 3

3. System Design

The system design phase for *EduTrack* aims to establish a solid architectural foundation and define detailed components for the platform. This phase focuses on creating a scalable and efficient system architecture, which incorporates both the client-server model and the database design. The design also includes UI/UX wireframes to ensure the frontend is user-friendly and easy to navigate for both students and mentors. Additionally, API specifications will be created to facilitate smooth communication between the frontend and backend. All design components will be compiled into a comprehensive document, outlining the blueprint for the development and deployment of *EduTrack*.

3.1 Methodology

The methodology for *EduTrack* is structured to ensure the development of a seamless, efficient, and user-centric web platform for both students and mentors. The process can be divided into the following key stages:

- **Planning and Requirement Analysis**: This phase involves gathering the specific needs and requirements of the users (students and mentors), ensuring that both sides of the platform are adequately catered to.
- **System Design**: The system design phase focuses on establishing the platform's architecture, including client-server models, database structure, and frontend wireframes, ensuring ease of use and clarity for users.
- **Implementation**: During this phase, the system components will be developed according to the design specifications. Frontend development will begin with setting up the React environment, followed by building the UI components based on the wireframes. For the backend, the Node.js environment will be configured to build server-side logic and RESTful APIs. Firebase will be used for user authentication and to provide real-time data updates, ensuring seamless user experiences. The database schema will be implemented to efficiently handle student and mentor data, assignments, quizzes, and results.
- **Testing and Deployment**: Rigorous testing will be carried out to identify and fix any bugs or issues, ensuring a stable and high-performance platform. Once the system is fully tested, it will be deployed for use by students and mentors.

3.1.1 Algorithms Description

EduTrack will leverage modern algorithms and technologies to enhance the learning experience for both students and mentors. Some of the key features include:

- **User Authentication**: The platform will utilize secure authentication methods, allowing users to create accounts and log in using Firebase. This ensures secure and seamless access for both students and mentors.
- **Content Personalization**: To deliver a tailored experience, *EduTrack* will utilize algorithms that analyze user data, such as previous quiz scores, assignments, and attendance patterns. This allows the platform to offer personalized learning resources, recommendations, and progress tracking to students.
- **Real-Time Feedback**: Using real-time data updates and user input, the platform will provide students with instant feedback on assignments, quizzes, and other activities. This feature helps foster continuous engagement and motivates students to improve.
- **Data Visualization**: *EduTrack* will visualize student performance data, such as quiz scores and attendance trends, using graphical representations like line charts and histograms. These visualizations will help students track their progress over time and provide mentors with valuable insights for assessment and guidance.
- **Analysis of Student Data**: The system will also implement basic data analysis techniques, such as identifying trends in student performance and engagement, to provide actionable insights to both mentors and students. This helps mentors offer targeted support and ensures students stay on track with their learning goals.

By combining a user-friendly interface with sophisticated data handling and analysis techniques, *EduTrack* will deliver a comprehensive platform for education management, tracking, and personalized learning.

3.2 **Behavioral Diagrams**

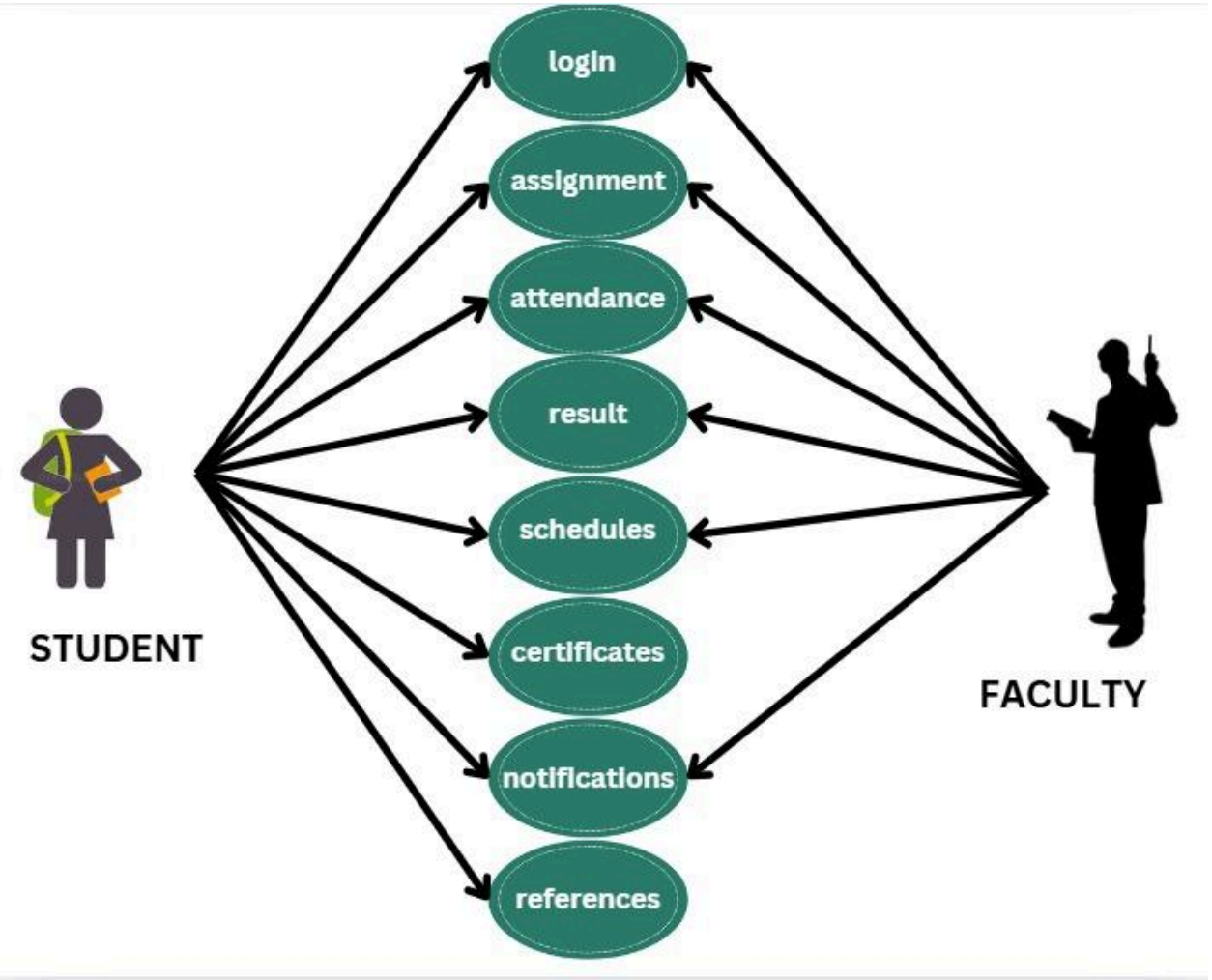


Fig 3.1 Use case Diagram

This fig. Shows the streamlines educational workflows, offering real-time insights for faculty and students.

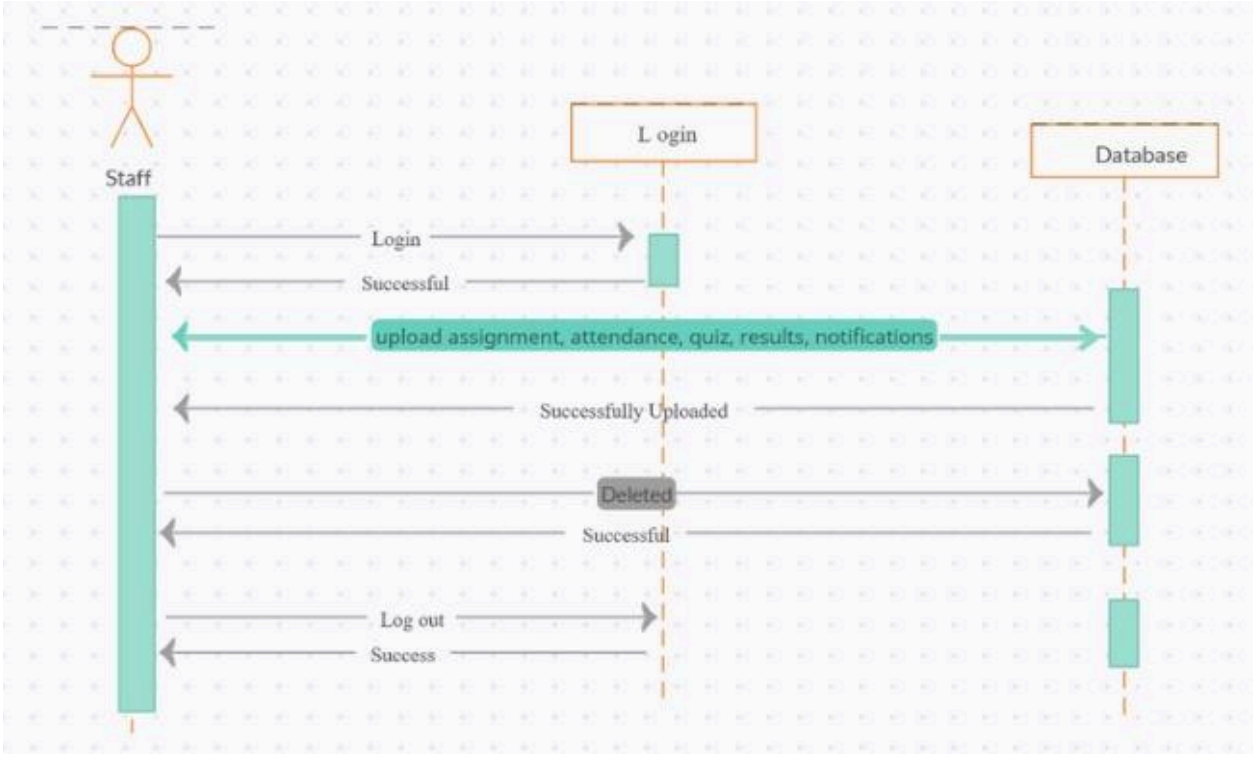


Fig 3.2 Mentor Sequence Diagram

The figure showcasing the interaction flow between mentors and the system.

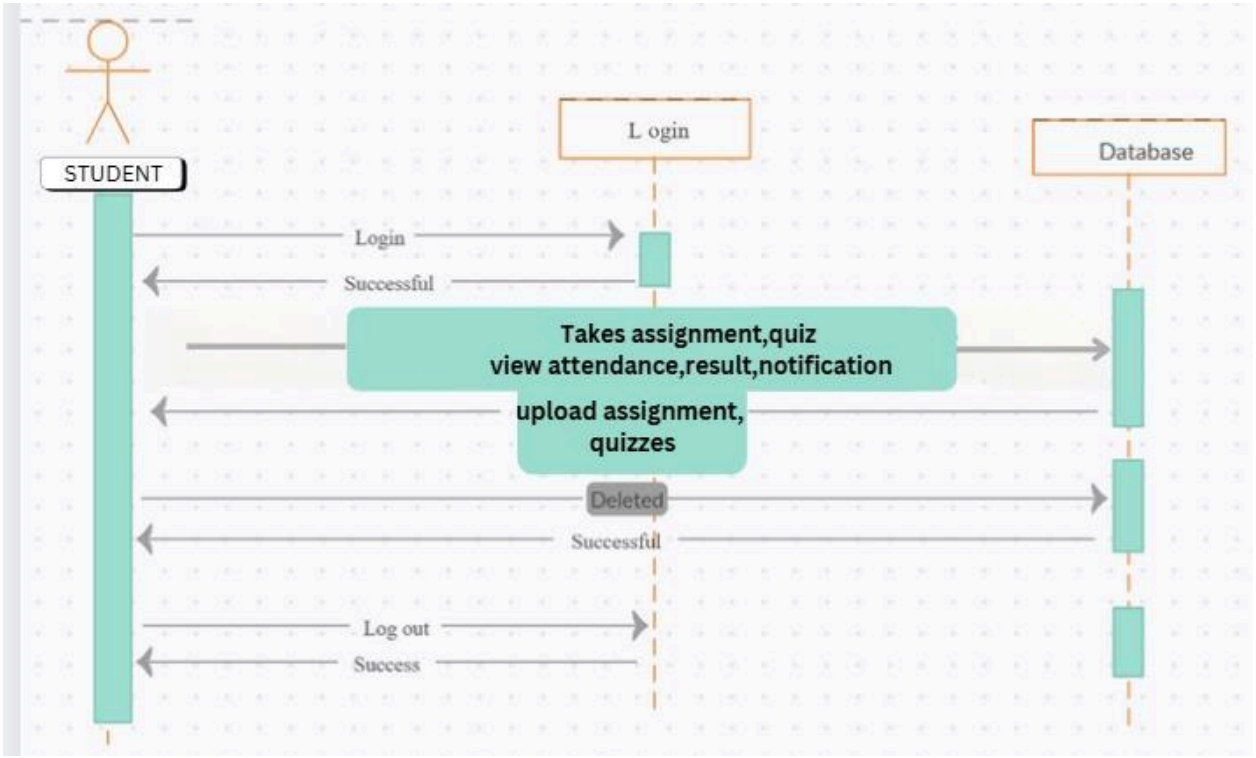


Fig 3.3 Student Sequence diagram

The figure showcasing the interaction flow between students and the system.

CHAPTER - 4

4. Implementation

4.1 Environmental Setup

Installing an Editor: Visual Studio Code

1. Download Visual Studio Code for your operating system from the official website: Visual Studio Code Download

Setting Up PHP 2. Install PHP on your machine to run server-side scripts. Download PHP from the official website: PHP Download Ensure that PHP is added to your system's PATH for easy execution.

Setting Up MySQL Database 3. Install MySQL to manage and store application data such as user information, assignments, quizzes, results, and attendance records. Download MySQL from: MySQL Download Once installed, use MySQL Workbench or phpMyAdmin to create the necessary databases and tables.

Installing XAMPP for Local Development 4. For local development, install XAMPP, which bundles Apache, MySQL, and PHP in a single package. This will allow you to run PHP and MySQL locally: XAMPP Download

Setting Up Frontend with HTML, CSS, and JavaScript 5. Create the HTML structure for different pages of the platform (Student Dashboard, Mentor Dashboard, Assignments, Quizzes, etc.). Style the pages using CSS for a user-friendly and professional interface. 6. Use JavaScript (JS) for client-side interactivity and form validation (e.g., assignment submissions, quiz attempts). JS will also be used to dynamically fetch and display data from the server using AJAX.

Backend Development with PHP 7. Develop the backend of the system using PHP to handle user authentication, session management, and CRUD operations for assignments, quizzes, attendance, and results.

- **User Authentication**: Implement a login system for students and mentors with secure password storage (e.g., bcrypt hashing).
- **CRUD Operations**: Use PHP to create, read, update, and delete data related to assignments, quizzes, and results from the MySQL database.

Database Design with MySQL 8. Design the database with tables such as:

- **Users** (for student and mentor information)
- **Assignments** (to store assignment details)
- **Quizzes** (for storing quiz data)
- **Attendance** (for tracking student attendance)
- **Results** (for storing quiz and assignment results)

AJAX Integration for Real-Time Updates 9. Use JavaScript and AJAX to interact with the PHP backend without reloading the page:

- Fetch and submit assignment, quiz, and attendance data asynchronously.

- Update the frontend in real-time as students submit assignments or attempt quizzes.

Implementing Mentors and Student Dashboards 10. **Mentor Dashboard**:

- Provide mentors with the ability to create and manage assignments and quizzes.
- Display student results and feedback options.

11. **Student Dashboard**:

- Allow students to access and submit assignments and quizzes.
- Display attendance records and results dynamically using data fetched from the backend.

API Development with PHP 12. Develop API endpoints in PHP to handle client-side requests:

- **GET** requests to fetch assignments, quizzes, and results.
- **POST** requests to submit completed assignments or quiz answers.
- **PUT** requests to update assignment feedback or quiz results.

Testing and Debugging 13. Test the application for:

- **User authentication**: Ensure secure login and session management.
- **CRUD functionality**: Ensure assignments, quizzes, and results are managed properly.
- **AJAX updates**: Verify that real-time data updates work without page reloads.
- **Cross-browser compatibility**: Test the app on different browsers for consistent performance.

Deployment 14. Deploy the application to a live web server using PHP and MySQL hosting. You can use a hosting provider such as Bluehost, HostGator, or any other that supports PHP and MySQL.

- Upload your project files via FTP.
- Configure the MySQL database on the server and ensure all PHP scripts are correctly configured for production

4.2 Implementation Details

The **EduTrack** project involves building a comprehensive web application for managing assignments, quizzes, attendance, results, and certifications for both students and mentors. The steps include frontend and backend development, database management, and integration of core features such as user authentication, dynamic content display, and real-time updates.

Frontend Development (HTML, CSS, JavaScript)

Initialize Project:

- Set up the project structure for the frontend using basic HTML, CSS, and JavaScript.
- Use a text editor (like Visual Studio Code) for writing and organizing the code.

Create Components:

- **Dashboard**: Develop the student and mentor dashboards for managing assignments, quizzes, results, and attendance.

- **Assignment Section**: Create components for assignment submission and feedback display.
- **Quiz Section**: Develop components for quiz-taking and displaying results.
- **Attendance and Results**: Display sections for tracking and displaying attendance and results.

Set Up Routing:

- Use JavaScript to manage page navigation and routing. For example, redirect users between the dashboard, assignments, quizzes, and results.

Styling:

- Style components using CSS to ensure a user-friendly and visually appealing interface.
- Use responsive design to ensure compatibility with both mobile and desktop views.

Backend Development (PHP and MySQL)

Initialize Project:

- Set up a PHP project structure with proper directories and files to handle client-server communication.
- Install necessary dependencies for MySQL integration and session management.

Define Routes:

- Create PHP routes for handling different pages and functionalities like user authentication, assignment submissions, and quiz attempts.

Create Services:

- Implement PHP functions to handle backend processes, such as assignment creation, quiz management, and attendance tracking.
- Implement logic for CRUD operations related to assignments, quizzes, and results.

Define Database Models:

- Use MySQL to design tables and schemas for users, assignments, quizzes, results, and attendance.
- Set up relationships between tables (e.g., linking students to assignments, quizzes, and results).

Set Up Middleware:

- Implement middleware for authentication, session management, and error handling.
- Ensure that the mentor and student dashboards are only accessible after proper login/authentication.

User Authentication:

- Set up PHP scripts to handle user login, registration, and session management.
- Ensure passwords are securely hashed using PHP’s password hashing functions.

Integrate APIs:

- Use PHP to write service functions to interact with APIs for real-time data fetching, updating assignments and quizzes, and displaying results.
- Integrate third-party services as needed for additional functionality.

Real-time Updates (AJAX):

- Use JavaScript and AJAX to fetch and submit data asynchronously between the frontend and backend without reloading the page.
- Enable real-time updates for assignments, quizzes, and results to provide a smooth user experience.

Testing and Debugging:

- Test the application for user authentication, assignment and quiz submissions, attendance tracking, and results display.
- Ensure the frontend and backend communicate seamlessly through APIs and AJAX calls.

Deployment:

- Deploy the EduTrack application on a live server.
- Use a hosting service that supports PHP and MySQL (e.g., Bluehost, HostGator).
- Ensure all configurations are set up for production, including database connections and API integrations.

4.3 Results

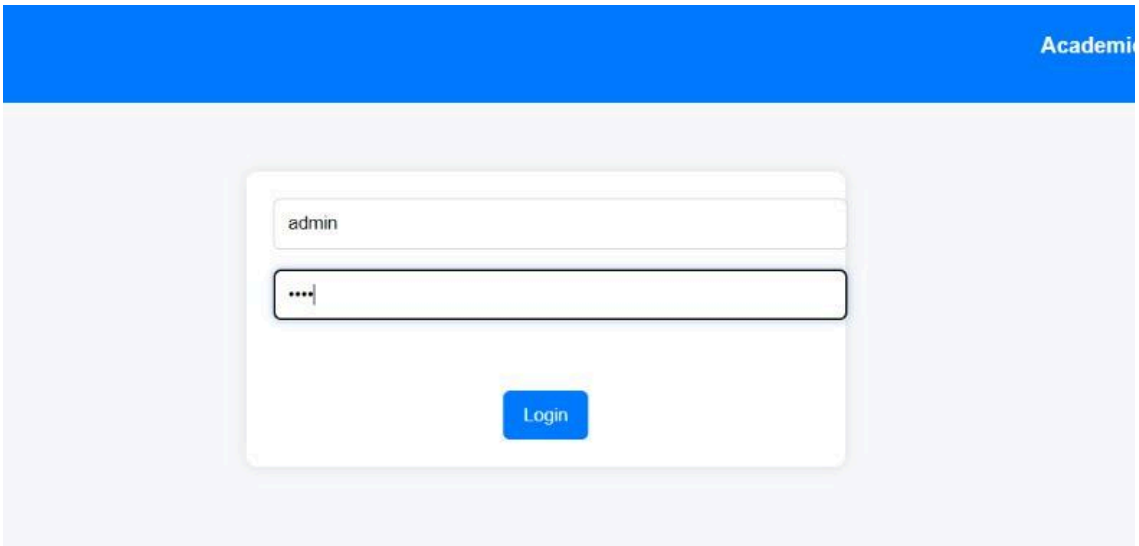


Fig 4.1 Login Page for mentor

This fig. provides access to personalized portals for mentors. In which they can access with their credentials to upload their requirements

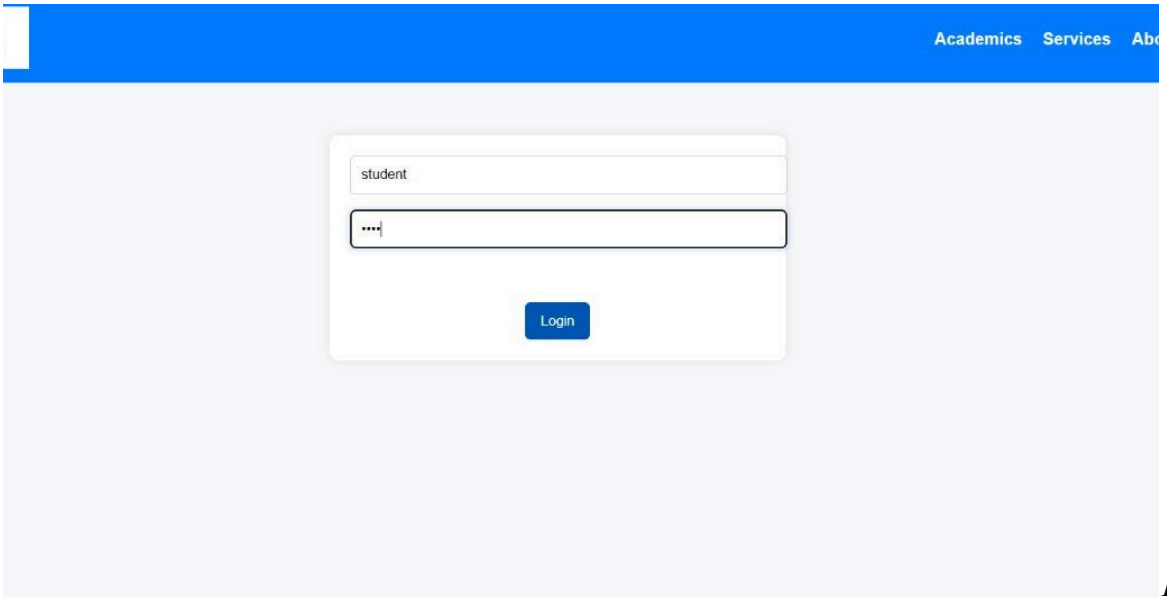


Fig 4.2 Student Login Page

This fig. provides access to personalized portals for students. In which student can check the uploaded information and can submit their works

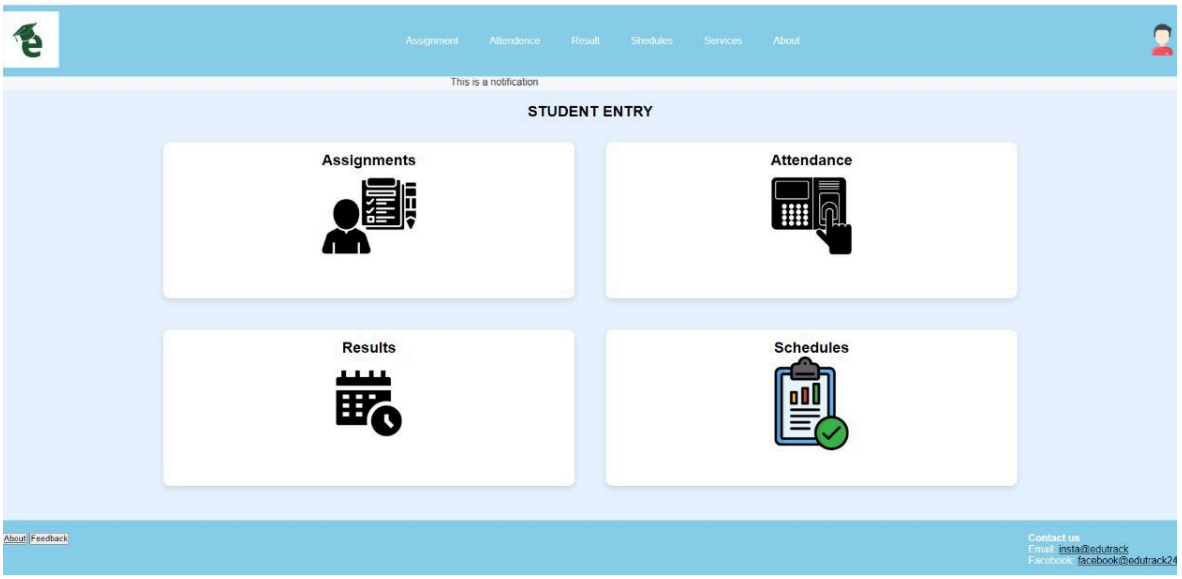


Fig 4.3 Mentor Home Page

This Fig. serves as the central portal for faculty-related information.In which the mentor can upload/modify/update assignment,attendance,results and schedules

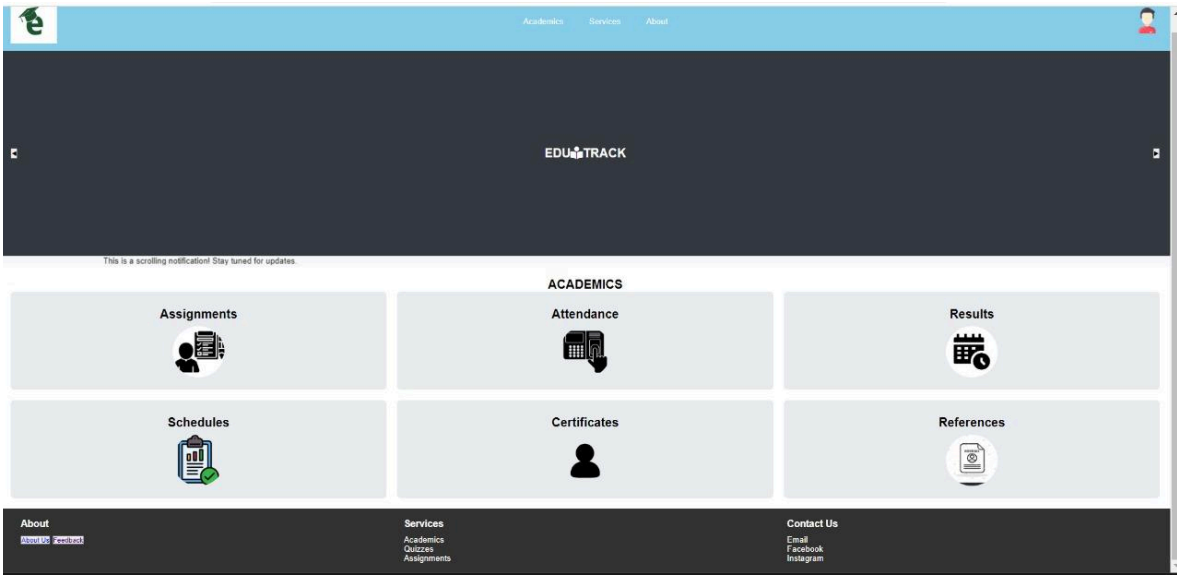


Fig 4.4 Student Home Page

This Fig. serves as the central portal for student-related information.It shows the modules that are going to be accessed by the students to fulfil their needs

Cases

1. Test Case Table for Mentor:

Test Case	Description	Input	Output	Status
Login	Mentor Login	Enter valid username and password	Mentor is logged In and redirected to the dashboard	Pass
Post Assignment	Mentor posts an assignment	Click on”Add Assignment” fill in details,upload image and submit	Assignment is successfully posted and appears in the student view	Pass
Post Quiz	Mentor posts a new quiz	Click on”Add Quiz”,enter questions,choices and correct answer then submit	Quiz is successfully posted visible to students	pass
Post Schedule	Mentor posts a new schedule for classes or assignments	Enter schedule details such as date,time and description,then submit	Schedule is posted and visible to students under the “Schedules” section	Pass
Post Attendance	Mentor records or updates student attendence	Enter quiz results for a specific student or group of students	Attendance is updated successfully and reflects in the student's attendance record	Pass
Post Results	Mentor posts results for quizzes or assignments	Enter quiz results for a specific student or group of students	Results are posted successfully and visible in the "Results" section for students	Pass
View All Students Data	Mentor views all student data (attendance, results, assignments)	Go to the respective section (attendance, results, etc.	Data for all students is displayed for review, with the option to edit or delete	Pass
Edit/Delete Posted Content	Mentor edits or deletes previously posted content (assignments, quizzes, etc.)	Select content (assignment, quiz, schedule) and click on edit/delete option	Content is successfully edited or deleted, and changes are reflected in the student view	Pass

2. **Test Case Table for Student:**

Test Case	Description	Input	Output	Status
Login	Student Login	Enter valid username and password	Student is logged in and redirected to the dashboard	Pass
Access Assignment	Student accesses an assignment posted by mentor	Click on the assignment under the "Assignments" section	Assignment details are displayed, student can view or download assignment	Pass
Submit Assignment	Student submits an assignment	Upload completed assignment file or enter work in text field and click Submit	Assignment is successfully submitted, confirmation message displayed	Pass
Attempt Quiz	Student attempts a quiz posted by mentor	Click on the quiz, select answers for each question, then click Submit	Quiz is submitted successfully, results displayed in a scoreboard	Pass
View Results	Student views their quiz or assignment results	Navigate to the "Results" section and click on the desired result to view	Quiz or assignment results are displayed with the correct/incorrect answers highlighted	Pass
View Schedule	Student views the class or assignment schedule	Navigate to the "Schedules" section	Schedule is displayed, showing upcoming classes or assignment deadlines	Pass
View Attendance	Student views their attendance record	Go to the "Attendance" section, select the term/date range	Student's attendance record is displayed for the selected period	Pass
Add Certificate	Student adds a certificate to their profile	Go to the "Certificates" section, click on "Add Certificate", upload certificate details and file	Certificate is successfully added to the student's profile	Pass
View Course References	Student views course references	Navigate to the "References"	Course references and	Pass

	posted by mentor	section, select a course and view/download materials	materials are displayed and available for download	

CHAPTER-5

CONCLUSION & FUTURE ENHANCEMENTS

Conclusion

In a world where educational management systems often lack integration and efficiency, **EduTrack** offers a transformative solution that streamlines academic functions. By consolidating attendance tracking, assignment management, grading, schedules, and certificate handling into a single platform, EduTrack bridges the gap between faculty and students. Through its intuitive, user-friendly interface, EduTrack ensures that both students and mentors can navigate academic tasks with ease, fostering a more organized and efficient learning environment.

With its role-based access, EduTrack ensures that the platform remains secure while delivering personalized experiences tailored to the needs of students, mentors, and administrators. This centralized approach makes EduTrack an invaluable tool for educational institutions, enabling seamless communication, real-time updates, and enhanced student engagement. As a result, EduTrack is not only a tool for academic management but also a platform that enhances the overall educational experience for all users.

Future Enhancements

In the future, EduTrack could integrate more advanced features such as AI-driven learning suggestions, personalized learning paths, and automated progress tracking, allowing for even more tailored educational experiences. Additionally, incorporating integrations with external tools such as cloud storage and third-party learning resources could further enrich the platform. Enhanced analytics for students and faculty, along with predictive insights, could provide proactive support and improve academic outcomes.

REFERENCES:

[1] P. Goyal, V. Nigam, S. A. Khan and S. K. Singh, "Automated Attendance Management System," *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2023, pp. 452-455, doi: 10.1109/ICAC3N60023.2023.10541627.keywords: {Face recognition;Manuals;Organizations;Fingerprint recognition;Recording;Mobile applications;Active appearance model;biometric recognition;RFID;computer vision;attendance tracking;student information system;efficiency}

URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10541627&isnumber=10541309

[2] N. I. Khan *et al*., "Towards Developing an Automated Attendance Management System using Fingerprint Sensor," *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, Bangladesh, 2020, pp. 1-6, doi: 10.1109/ETCCE51779.2020.9350882.keywords: {Cloud computing;Laboratories;Manuals;Sensor systems;Real-time systems;sensors;Security;Automation;biometric identification;cloud computing;edge computing;fingerprint sensors}

URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9350882&isnumber=9350859

[3] **MVSREC** official website

URL : MVSR Official Website

[4] J. Ye, X. Bin, D. Luo, Z. Wang and C. Shu, "Design and Implementation of an Online Learning System," 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA), Changsha, China, 2018, pp. 259-262, doi: 10.1109/ICDSBA.2018.00054. keywords: {Videos;Courseware;Histograms;IEEE Sections;Learning systems;Clustering algorithms;online learning;learning platform}

URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8588923&isnumber=8588869

[5] F. Domenach and G. Portides, "PASS — A programming assignment submission system," 2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL), Thessaloniki, Greece, 2015, pp. 195-199, doi: 10.1109/IMCTL.2015.7359585. keywords: {Programming profession;Simple object access protocol;Syntactics;Computer languages;Databases;Online submission;programming languages;automated assessment;elearning},

URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7359585&isnumber=7359535

[6] M. Novák, M. Biňas, M. Michalko and F. Jakab, "Student's progress tracking on programming assignments," 2012 IEEE 10th International Conference on Emerging eLearning Technologies and Applications (ICETA), Star Lesn, Slovakia, 2012, pp. 279-282, doi: 10.1109/ICETA.2012.6418321. keywords: {Control systems;Programming profession;Electronic learning;Java;Libraries},

URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6418321&isnumber=6418280

[7] B. Zorić, M. Dudjak, D. Bajer and G. Martinović, "Design and development of a smart attendance management system with Bluetooth low energy beacons," 2019 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2019, pp. 86-91, doi: 10.1109/ZINC.2019.8769433. keywords: {Bluetooth;Tracking;Web services;Mobile applications;Radiofrequency identification;Task analysis;Wireless communication;attendance data management;attendance tracking;beacons;bluetooth low energy},

URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8769433&isnumber=8769334

Appendix

Pseudo code

```
// Initialization Phase

initializeConfig()

loadLibraries()

// User Authentication - Login

function handleUserLogin(username, password):

   // Query database to check if user exists and credentials match

   user = queryDatabase("SELECT * FROM users WHERE username = username AND
password = password")

   if user exists:

      session = startSession(user)

      return session  // User logged in successfully

   else:

      display("Invalid username or password.")

      return null



// User Registration (for new users)

function handleUserRegistration(username, password, role):

   // Save new user details into the database

   queryDatabase("INSERT INTO users (username, password, role) VALUES (username,
password, role)")

   display("Registration successful.")

   return login(username, password)  // Automatically log in after registration
```

```
// Attendance Management - Mentor

function recordAttendance(mentor, student, date, status):

    // Mentor records attendance for a student

    queryDatabase("INSERT INTO attendance (mentor_id, student_id, date, status) VALUES (mentor.id, student.id, date, status)")

    display("Attendance recorded successfully.")



// Attendance Management - Student

function viewAttendance(student):

    // Student views their attendance record

    attendance = queryDatabase("SELECT * FROM attendance WHERE student_id = student.id")

    display(attendance)

    return attendance



// Assignment Management - Mentor

function createAssignment(mentor, title, description, dueDate):

    // Mentor creates a new assignment

    queryDatabase("INSERT INTO assignments (mentor_id, title, description, due_date) VALUES (mentor.id, title, description, dueDate)")

    display("Assignment created successfully.")



// Assignment Management - Student
```

```
function viewAssignments(student):

    // Student views the list of assignments

    assignments = queryDatabase("SELECT * FROM assignments WHERE student_id =
student.id")

    display(assignments)

    return assignments



// Submit Assignment - Student

function submitAssignment(student, assignmentId, submissionData):

    // Student submits an assignment

    queryDatabase("INSERT INTO submissions (student_id, assignment_id,
submission_data) VALUES (student.id, assignmentId, submissionData)")

    display("Assignment submitted successfully.")



// Quiz Management - Mentor

function createQuiz(mentor, title, questions):

    // Mentor creates a new quiz

    queryDatabase("INSERT INTO quizzes (mentor_id, title, questions) VALUES (mentor.id,
title, questions)")

    display("Quiz created successfully.")



// Quiz Management - Student

function viewQuiz(student):

    // Student views available quizzes
```

```
    quizzes = queryDatabase("SELECT * FROM quizzes WHERE student_id = student.id")

    display(quizzes)

    return quizzes



// Submit Quiz - Student

function submitQuiz(student, quizId, answers):

    // Student submits their quiz answers

    queryDatabase("INSERT INTO quiz_submissions (student_id, quiz_id, answers) VALUES
(student.id, quizId, answers)")

    calculateQuizResults(student, quizId, answers)

    display("Quiz submitted successfully.")



// Calculate Quiz Results

function calculateQuizResults(student, quizId, answers):

    // Compare student's answers with correct answers and calculate score

    correctAnswers = queryDatabase("SELECT * FROM quiz_answers WHERE quiz_id =
quizId")

    score = calculateScore(answers, correctAnswers)

    saveQuizResults(student, quizId, score)

    display("Quiz results calculated.")



// Calculate Score Function

function calculateScore(submittedAnswers, correctAnswers):

    score = 0
```

```
    for each answer in submittedAnswers:
        if answer matches correct answer:
            score += 1
    return score

// Save Quiz Results to Database
function saveQuizResults(student, quizId, score):
    queryDatabase("INSERT INTO quiz_results (student_id, quiz_id, score) VALUES
(student.id, quizId, score)")

// Certificates Management - Mentor
function issueCertificate(mentor, student, certificateDetails):
    // Mentor issues a certificate to a student
    queryDatabase("INSERT INTO certificates (mentor_id, student_id, certificate_details)
VALUES (mentor.id, student.id, certificateDetails)")
    display("Certificate issued successfully.")

// Certificates Management - Student
function viewCertificates(student):
    // Student views their certificates
    certificates = queryDatabase("SELECT * FROM certificates WHERE student_id =
student.id")
    display(certificates)
    return certificates
```

```
// Main Application Loop
while userIsActive:
    // Get user input for different operations (login, assignments, quizzes, etc.)
    userInput = getUserInput()

    // If user input is login
    if userInput is "login":
        username = getInput("Enter username:")
        password = getInput("Enter password:")
        session = handleUserLogin(username, password)
        if session is null:
            display("Login failed.")
        else:
            display("Welcome, " + username)

    // If user input is view attendance
    else if userInput is "view attendance":
        student = getUserSession()
        attendance = viewAttendance(student)
        display(attendance)

    // If user input is create assignment
```

```
else if userInput is "create assignment":
    mentor = getUserSession()
    title = getInput("Enter assignment title:")
    description = getInput("Enter assignment description:")
    dueDate = getInput("Enter assignment due date:")
    createAssignment(mentor, title, description, dueDate)

// If user input is submit assignment
else if userInput is "submit assignment":
    student = getUserSession()
    assignmentId = getInput("Enter assignment ID:")
    submissionData = getInput("Enter submission data:")
    submitAssignment(student, assignmentId, submissionData)

// If user input is view quizzes
else if userInput is "view quizzes":
    student = getUserSession()
    quizzes = viewQuiz(student)
    display(quizzes)

// If user input is submit quiz
else if userInput is "submit quiz":
    student = getUserSession()
```

```
    quizId = getInput("Enter quiz ID:")
    answers = getInput("Enter answers:")
    submitQuiz(student, quizId, answers)

// If user input is issue certificate
else if userInput is "issue certificate":
    mentor = getUserSession()
    student = getInput("Enter student ID:")
    certificateDetails = getInput("Enter certificate details:")
    issueCertificate(mentor, student, certificateDetails)

// If user input is view certificates
else if userInput is "view certificates":
    student = getUserSession()
    certificates = viewCertificates(student)
    display(certificates)
```