ACADEMIC AND PLACEMENT TRACKER

A PROJECT REPORT

Submitted by

KEERTHIVASAN S J	**811722104074**
MUHUNDAN S	**811722104095**
RAJESH R	**811722104117**

in partial fulfillment of the requirements for the award degree of

Bachelor in Engineering

20CS7503 DESIGN PROJECT - 3

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM - 621112

NOVEMBER 2025

ACADEMIC AND PLACEMENT TRACKER

A PROJECT REPORT

Submitted by

KEERTHIVASAN S J	**811722104074**
MUHUNDAN S	**811722104095**
RAJESH R	**811722104117**

in partial fulfillment of the requirements for the award degree of

Bachelor in Engineering

20CS7503 DESIGN PROJECT - 3

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621112

NOVEMBER 2025

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)
SAMAYAPURAM - 621112

BONAFIDE CERTIFICATE

The work embodied in the present project report entitled "**ACADEMIC AND PLACEMENT TRACKER**" has been carried out by the students **KEERTHIVASAN S J, MUHUNDAN S, RAJESH R,** The work reported herein is original and we declare that the project is their own work, except where specifically acknowledged, and has not been copied from other sources or been previously submitted for assessment.

Date of Viva Voce: ……

Ms. R. Madhumathi ,M.E
SUPERVISOR
Assistant Professor
Department of CSE
K. Ramakrishnan College of Technology(Autonomous)
Samayapuram-621 112

Mr. R. Rajavarman , M.E.,(Ph.D.,)
HEAD OF THE DEPARTMENT
Assistant Professor(Sr. Grade)
Department of CSE
K. Ramakrishnan College of Technology(Autonomous)
Samayapuram-621 112

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The Academic and Placement Tracker is a web-based system designed to efficiently manage and monitor student's academic performance and placement activities. The main objective of this project is to provide an integrated platform for students, mentors, and placement officers to access and update relevant academic and placement information in a structured manner. In this system, students can log in to update and view their academic details such as semester wise CGPA, overall CGPA, certificates, self-study assessments, and company placement details including package (LPA). Mentors can access the details of their assigned mentees only, enabling them to monitor individual progress and provide guidance effectively. Placement officers (POs) have administrative access to view all student's academic and placement data across departments through a centralized dashboard, allowing them to analyse department-wise and overall placement performance. The system enhances transparency, reduces manual record maintenance, and simplifies data management for academic and placement tracking.

Keywords: Academic tracker, Placement tracker, Student portal, Mentor dashboard, Placement officer, Web-based system, CGPA tracking, Certificate management, Assessments, Communication bridge.

ACKNOWLEDGEMENT

We thank our **Dr. N.Vasudevan**, Principal, for his valuable suggestions and support during the course of my research work.

We thank our **Mr. R. Rajavarman**, Head of the Department, Assistant Professor (Sr. Grade), Computer Science and Engineering, for his valuable suggestions and support during the course of my research work.

We wish to record my deep sense of gratitude and profound thanks to my Guide **Ms. R. Madhumathi**, Assistant Professor, Computer Science and Engineering for her keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring this thesis into fruition.

We are extremely indebted to our project coordinator **Mr. M. Saravanan** , Assistant Professor, Computer Science and Engineering, for his valuable suggestions and support during the course of my research work.

We also thank the faculty and non-teaching staff members of the Computer Science and Engineering, K. Ramakrishnan College of Technology (Autonomous), Samayapuram, for their valuable support throughout the course of my research work.

Finally, we thank our parents, friends and our well wishes for their kind support.

SIGNATURE

_____

_____

_____

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CGPA	-	Cumulative Grade Point Average
LPA	-	Lakhs Per Annum
PO	-	Placement Officer
API	-	Application Programming Interface
ERP	-	Enterprise Resource Planning
KPI	-	Key Performance Indicator
REST	-	Representational State Transfer
AWS	-	Amazon Web Services
React JS	-	React JavaScript Library

CHAPTER 1

INTRODUCTION

1.1 DESCRIPTION

The Academic and Placement Tracker is a comprehensive digital system designed to monitor, manage, and analyze the academic growth and placement progress of students in educational institutions. The project addresses the challenges faced by mentors, students, and placement officers in maintaining accurate records, identifying performance gaps, and making informed decisions based on real-time academic and placement data. Traditional methods of tracking student progress—such as manual spreadsheets, paper-based records, or unstructured data collection—often lead to errors, delays, and difficulties in retrieving information. This system aims to streamline these processes by providing an integrated and user-friendly platform. The project supports three key users: Students/Mentees, Mentors, and the Placement Officer (PO). Students can update and view their academic details, such as semester-wise CGPA, assignments, arrears, and placement status. This helps them track their overall progress and stay motivated to improve.

Mentors use the system to analyze the performance of their assigned mentees, evaluate academic growth over time, and offer personalized guidance. The Placement Officer has access to complete student data, enabling them to analyze trends, shortlist eligible students, prepare placement reports, and track company wise selection details. The system offers features like CGPA tracking, assignment status monitoring, placement updates, performance analytics, and automated report generation. By analyzing academic patterns, mentors and placement teams can identify strengths, weaknesses, and the readiness levels of students for campus recruitment. The availability of structured data helps the placement cell make better decisions during drives, such as filtering eligible candidates for specific companies based on criteria like CGPA, skills, or academic consistency.

1.2 EDUCATIONAL TECHNOLOGY

Educational Technology focuses on using digital tools and software solutions to enhance learning, teaching, and academic management. It integrates technology into educational environments to improve efficiency, engagement, and accessibility. Systems such as learning management platforms, academic trackers, smart classrooms, and online assessments fall under this domain.

1.3 WEB TECHNOLOGY & FULL-STACK DEVELOPMENT

This domain involves designing and developing complete web applications, covering both frontend and backend components. Frontend development focuses on user interfaces using HTML, CSS, JavaScript, and frameworks, while backend development handles business logic, databases, authentication, and server management.

1.4 GENERATIVE AI & NATURAL LANGUAGE PROCESSING (NLP)

Generative AI and NLP focus on creating intelligent systems capable of understanding, generating, and processing human language. NLP techniques help machines analyze text, extract meaning, and interact conversationally, while generative AI produces new content such as text, summaries, answers, or recommendations.

1.5 DATA ANALYTICS & VISUALIZATION

Data Analytics and Visualization involve collecting, analyzing, and presenting data in meaningful formats. Using statistical methods, dashboards, and visual charts, this domain transforms raw academic and placement data into actionable insights. It helps institutions identify performance trends, track student progress, evaluate placement outcomes, and make informed decisions. Visualization tools like bar charts, graphs, and reports make complex information easier to understand.

CHAPTER 2

LITERATURE SURVEY

2.1 A MOBILE ATTENDANCE SYSTEM FOR UNIVERSITY STUDENTS USING ANDROID AND FIREBASE, INTERNATIONAL JOURNAL OF MOBILE COMPUTING

Kumar present an android technology with Firebase to manage student attendance seamlessly. The authors argue that universities often face challenges with manual attendance procedures, including time wastage, inaccuracies, and administrative inefficiencies. Their proposed system automates attendance tracking while ensuring data consistency and real-time availability. The application developed in the study includes a student interface for marking attendance and a faculty interface for session creation and monitoring. Firebase is used as the backend due to its cloud-hosted real-time database, authentication modules, and ease of integration with Android applications. Students authenticate using their assigned credentials, and once logged in, the app records attendance with a timestamp.

Teachers can view attendance in real time and generate reports for academic evaluation. One of the key contributions of the paper is its discussion on faculty-controlled validation, which reduces the likelihood of unauthorized attendance marking. Instead of allowing attendance any time, teachers activate time-bound attendance windows. This ensures students can only mark attendance during the class period, preventing misuse. The use of Firebase's offline persistence ensures the app functions well in areas with poor connectivity by temporarily storing data locally. These features help students track their progress and encourage better attendance behavior. The study concludes that the Androidhis–Firebase architecture is effective for developing scalable attendance systems.

2.2 ANDROID-BASED STUDENT ATTENDANCE MANAGEMENT SYSTEM USING FIREBASE, J. SOFTW. ENG.,

Patel were describe Firebase Authentication, Cloud fire store, and Firebase Storage as core components that ensure data accessibility, security, and scalability. A student attendance management system built on Android and powered by Google Firebase. Their work focuses on resolving issues associated with outdated attendance mechanisms, such as paper records and desktop-based systems, by exploiting the advantages of cloud computing and mobile technology. Firebase is selected due to its real-time database, easy integration, fast synchronization capabilities, and secure authentication features. The system allows students to mark attendance through an Android app, while teachers can monitor, verify, and manage student records. Unlike many location-based attendance systems, this design emphasizes real-time synchronization and multi-user data access, making it suitable for large academic institutions with multiple departments and sections.

The app workflow is simple: the teacher initiates a session, the student logs in using their unique credentials, and attendance is stored instantly in the cloud. The system also supports features like automatic time-stamping, student profile management, push notifications, and data backup. Teachers can retrieve detailed reports, export attendance summaries, and track irregularities in student participation. The authors highlight the strengths of Firebase, such as its ability to function even with limited internet connectivity due to local caching. This ensures smooth operation in classrooms with poor network conditions. Security measures, including role-based access control and encrypted data transmission, safeguard sensitive academic records. The paper includes test results demonstrating the system's high performance and response time. It also notes how cloud-based storage simplifies administrative tasks and reduces the risk of data loss. Although the system does not include geo-location or biometric verification, the authors argue that the ease of implementation, cost-effectiveness, and platform independence of Firebase make it an ideal solution for educational institutions.

2.3 ATTENDANCE MANAGER USING FIREBASE AND JAVA: EMPLOYEE ATTENDANCE SYSTEM, GITHUB.

Joseph emphasizes modularity: developers can integrate additional features such as location tracking, leave management, work-hour calculation, and payroll integration. One strength of this system is its simplicity and adaptability. While most academic papers focus on student attendance, this system is developed for managing employee attendance in workplaces, making it relevant for institutions seeking cross-domain solutions. The project uses Firebase for cloud storage, authentication, and real-time updates, while Java provides the application logic. The system allows employees to log in and mark attendance using their registered credentials. Firebase Authentication ensures secure user identity verification. Attendance records are stored in Cloud fire store, which makes them instantly accessible across devices. Administrative users can view employee attendance history, daily logs, and monthly summaries. The system aims to replace traditional biometric or manual punch-in methods with a simpler mobile or desktop interface.

The GitHub repository includes detailed documentation, code samples, and implementation guidelines, allowing developers to modify and extend the system. It requires minimal infrastructure investment because Firebase handles backend operations like server maintenance, scalability, and database security. The use of Java makes the system applicable to both Android and desktop-based environments. The repository demonstrates real-time attendance synchronization, which ensures that administrators always have access to updated attendance information. Firebase's offline capabilities allow attendance marking even when network connectivity is weak, with synchronization happening once the connection is restored. Overall, Joseph's project serves as a practical reference for developers and researchers working on cloud-based attendance solutions. It highlights the advantages of using Firebase for real-time applications and offers a flexible foundation for further research or academic implementation.

2.4 ATTENDANCE MONITORING: AN ANDROID APPLICATION FOR SCHOOL ATTENDANCE MANAGEMENT.

Magaraa highlights Firebase Real-time Database as the core backend, offering live synchronization between user devices and the cloud. This ensures that attendance records are instantly updated and accessible. The system focuses on providing a simple and efficient method for teachers to manage classroom attendance using Android devices. The project uses Firebase as the backend, making it suitable for real-time data operations and cloud-based storage. The system features two main modules: teacher and student management. Teachers can create class groups, add or remove students, and mark daily attendance using the app. The interface is designed to be simple, ensuring ease of use for schools where teachers may not be familiar with complex software. The project also includes features such as attendance summaries, daily reports, and automatic date-wise sorting. The system also integrates Firebase Authentication for secure login and user management. In addition to basic attendance functionalities, the project provides an admin dashboard for monitoring school-wide attendance patterns.

School administrators can use it to analyze student absenteeism, identify trends, and implement corrective measures. Since the project is open-source, developers can extend the system by adding features like QR code scanning, location verification, and SMS notifications. A unique strength of the project is its adaptability to rural schools or low-resource environments. Since Firebase supports offline data caching, the application remains functional even with limited connectivity. Once the device reconnects, data is automatically synchronized. Magaraa's project serves as a strong practical reference for developers looking to implement mobile attendance systems in schools. It demonstrates how open-source solutions can be used in real-world educational settings to modernize administrative processes. The authors conclude that their system enhances transparency, minimizes manual effort, and provides a strong foundation for future enhancements such as location verification and analytics.

2.5 DESIGN AND IMPLEMENTATION OF A LOCATION-BASED ATTENDANCE TRACKING APP FOR STUDENTS USING ANDROID, INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING APPLICATIONS

Narayan highlights Firebase Real-time Database as the core backend, The system architecture comprises three major components: the Android mobile app, a backend server, and a centralized database. Students use the mobile app to mark attendance only when their location falls within a predefined geofenced region. The app continuously verifies the coordinates using Android's fused location provider, which combines GPS, Wi-Fi, and mobile network signals to increase accuracy. For teachers, the system provides an interface to create attendance sessions, view real-time data, and generate reports. Android-based attendance tracking system that uses location-based technology to ensure accurate and secure attendance recording in educational institutions. The paper begins by identifying the limitations of traditional attendance methods such as manual roll calls, biometric machines, and RFID cards, all of which are susceptible to manipulation, delays, and data management complexities. To address these issues, the authors propose a mobile app that leverages GPS and network-based location services to validate a student's physical presence in a classroom.

The authors emphasize the importance of geofencing, which ensures attendance can only be recorded inside the classroom boundary.They also highlight security measures such as token-based authentication, encrypted data transmission, and server-side validation to prevent misuse or spoofing. Performance evaluations conducted during classroom trials show that the system is able to detect location with high accuracy, even inside buildings where GPS signals are typically weak. The use of hybrid location estimation techniques improves reliability. The study also reports a substantial reduction in time spent on attendance taking, enabling teachers to focus more on instruction. It concludes that the proposed system is scalable, cost-effective, and suitable for adoption in universities and colleges seeking to automate attendance while ensuring authenticity and real-time monitoring.

2.6 GEO-FENCING AND SMS INTEGRATION FOR MOBILE ATTENDANCE SYSTEMS IN EDUCATIONAL INSTITUTIONS, INTERNATIONAL JOURNAL OF ANDROID DEVELOPMENT

To solve these problems, Nair integrates GPS-based geofencing with automated SMS alerts to create a hybrid attendance management model. In the proposed system, the educational institution defines a geofenced boundary around classrooms. Students can mark attendance using their Android devices only when they are physically present inside this boundary. The system uses Android's Fused Location Provider to enhance location accuracy by combining GPS, Wi-Fi, and cellular network data. One unique feature of this system is SMS integration. Once a student successfully marks attendance, an automated SMS is sent to the parent or guardian informing them about the student's presence in class. This promotes transparency and strengthens communication between institutions and parents, which is particularly useful in schools. This ensures students can only mark attendance during the class period, preventing misuse. The authors also highlight the system's robust security measures, such as restricting database access through Firebase security rules and enabling encrypted data transactions.

Performance tests conducted on university networks show that the app synchronizes data quickly across all connected devices, even in congested network conditions. The backend system stores attendance logs and generates analytical reports for teachers and administrators. The study also emphasizes the importance of energy-efficient location tracking to prevent smartphone battery drain. Nair evaluates different geo-fencing radii and determines that a small radius (10–20 meters) provides optimal performance without compromising accuracy. Security measures include server-side validation and access control to prevent unauthorized attendance. The paper also suggests using additional layers like MAC address verification to further reduce the possibility of location spoofing. Overall, Nair demonstrates that combining geofencing with SMS alerts creates a robust and parent-friendly attendance system. It reduces administrative workload, supports fast data retrieval, and provides an intuitive user experience.

2.7 GEO-LOCO: LOCATION-BASED ATTENDANCE SYSTEM USING GEOFENCING TECHNOLOGY IN ANDROID (JAVA)

Porwal also provides insights on how geofencing reduces server load compared to continuous GPS tracking, as location checks occur only at boundary transitions rather than at frequent intervals. The system marks attendance only when the user is within a pre-set geographical boundary, ensuring reliable presence verification. The app uses Android's geofencing APIs, which monitor when a student enters or exits the classroom area. This reduces battery consumption compared to continuous GPS tracking. Attendance events are triggered automatically when the device detects entry into the geofenced region. The project also includes Firebase integration for storing attendance data, ensuring security and real-time synchronization. This makes the system suitable for large-scale deployments where hundreds of students may be entering or leaving the classroom simultaneously. The project's open-source nature encourages collaboration, allowing other developers to contribute enhancements such as integrating QR verification or combining geofencing with Bluetooth beacons for improved indoor accuracy.

Porwal concludes that geolocation-based attendance will become increasingly important as educational institutions shift toward fully digital administrative processes. Teachers can view logs, generate reports, and monitor attendance trends. Porwal demonstrates the practical use of geofencing in educational settings and highlights the system's potential for preventing proxy attendance. The open-source nature of the project allows developers to expand it with additional features like notifications, analytics, or biometric verification. The model enhances accountability and encourages continuous improvement through timely mentor intervention. The study concludes that such data-driven supervision mechanisms significantly improve student engagement and performance. For your Academic and Placement Tracker, this research supports implementing features like automatic notifications, performance graphs, and mentor remarks, ensuring efficient tracking and personalized academic mentoring across departments.

2.8 LOCATION-BASED ATTENDANCE SYSTEM: AN ANDROID APPLICATION FOR AUTOMATING CLASS ATTENDANCE. INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS

An Android Application for Automating Class Attendance" by Doe and Smith presents a modern solution to one of the most repetitive and time-consuming tasks in educational institutions attendance tracking. Traditional attendance methods such as manual roll calling, paper-based registers, or even basic biometric devices suffer from limitations including time consumption, proxy attendance, data maintenance difficulty, and lack of real-time monitoring. The authors propose an Android-based mobile application that uses GPS-based location verification to automate and secure the attendance process. The core concept of the system revolves around leveraging a student's smartphone as an authentication device. When a student enters a classroom or a predefined geographical area, the app uses GPS coordinates to verify the location. The teacher or institution sets a "geofenced" boundary around classrooms, ensuring that attendance is recorded only when students are physically present within the designated radius.

This reduces the chances of proxy attendance, as the system relies not just on device presence but on validated geographic position. The proposed model includes three major components: the mobile application for students, the faculty interface, and the backend database. The student application allows learners to check in for attendance automatically or with a single tap once the system detects that they are inside the geofenced area. The use of hybrid location estimation techniques improves reliability. The study also reports a substantial reduction in time spent on attendance taking, enabling teachers to focus more on instruction. The faculty interface provides tools for instructors to create sessions, monitor real-time attendance, and view reports. Meanwhile, the cloud-based backend database stores attendance logs securely and allows administrators to analyze overall trends.

2.9 REAL-TIME MONITORING AND ATTENDANCE TRACKING WITH ANDROID AND FIREBASE, INTERNATIONAL JOURNAL OF DIGITAL SYSTEMS

Sharma further explains that Firebase's built-in security features such as Firestore rules and encrypted communication help protect sensitive academic data from unauthorized access. The importance of instant data access and cloud-based storage for modern academic institutions. Firebase Real-time Database forms the backbone of the system, enabling immediate synchronization of attendance records across devices. The system provides role-based access for students, teachers, and administrators. Students log in through Firebase Authentication and mark attendance, while teachers monitor real-time data and generate reports. The system automatically timestamps each entry, providing an accurate and transparent attendance history. Sharma emphasizes the reliability and speed of Firebase in handling multi-user data transactions. Offline functionality allows attendance marking even in areas with weak connectivity. Performance testing shows fast data retrieval, robust synchronization, and minimal server latency. The system architecture supports scalability, meaning it can be adopted by institutions of various sizes without requiring major modifications.

The author also notes that implementing such cloud-based systems reduces dependency on physical storage, which minimizes the risk of data loss. Teachers benefit from quick access to attendance histories, helping them identify irregular students early. The paper suggests integrating additional modules like leave management, push notifications, and performance analytics to build a more comprehensive academic tracking ecosystem. The study concludes that integrating Android with Firebase results in a scalable and efficient attendance monitoring system. Additionally, integrating analytics improves tracking of placement performance over academic years. This study provides foundational concepts for your Academic and Placement Tracker, particularly the need for structured data flow, secure role management, and interactive dashboards. It underscores how automation and analytics can replace outdated manual systems and improve overall placement management efficiency.

2.10 STUDENT ATTENDANCE SYSTEM WITH REAL-TIME LOCATION TRACKING USING ANDROID, JOURNAL OF EDUCATIONAL TECHNOLOGY

Gupta were identify major problems with conventional attendance methods, such as proxy marking and manual effort. Their system ensures that attendance is recorded only when students are physically present within a defined location radius. The core functionality is built using Android's location APIs. When students launch the app, it reads their coordinates and compares them with the classroom's predefined geolocation. If the coordinates fall inside the allowed radius, attendance is accepted and logged. The system includes faculty controls for enabling and closing attendance sessions, ensuring students cannot mark attendance before or after class hours. Real-time data syncing allows teachers to view attendance instantly. The authors also include a notification system to inform students when attendance windows open or close. Experimental results from field deployment show that real-time tracking increases attendance accuracy by eliminating proxy attendance. Although GPS accuracy may fluctuate indoors, hybrid positioning using Wi-Fi networks improves reliability.

Their system includes a clean interface that guides students through the attendance marking process, ensuring ease of use even for first-time users. The authors emphasize that accuracy and transparency are essential features for any attendance monitoring tool in modern institutions. They also highlight the system's ability to export attendance data for administrative review, making it useful for examination eligibility verification and academic audits. The study concludes that integrating real-time location tracking into attendance mechanisms improves institutional discipline and contributes to better classroom engagement. The authors conclude that their system is effective, secure, and beneficial for both teachers and students. They recommend adding biometric verification or QR code scanning in future iterations.

CHAPTER 3

EXISTING SYSTEM

In most educational institutions, academic and placement tracking is still managed using traditional or semi-digital methods such as manual record keeping, spreadsheets, or separate standalone systems. These disconnected tools make it difficult to maintain accurate, real-time data about students' academic performance, certificates, and placement activities. The lack of integration between academic departments, mentors, and placement officers often results in delays, redundancy, and inconsistency of information. In the existing system, students are required to update their academic and placement information manually or through offline submissions. Academic data such as CGPA, marks, or certificates are often stored in hard copies or individual files maintained by faculty. This approach not only consumes time and resources but also increases the chances of data misplacement or human error.

Students have limited access to their academic progress records and no personalized platform to track their performance or placement readiness. Mentors, on the other hand, rely on fragmented data sources to monitor their mentees. They must manually collect updates from multiple students, verify grades, and provide feedback through physical meetings or emails. This manual monitoring process makes it difficult to maintain consistency in mentorship, especially for institutions with a large number of students. The Placement Officer (PO) faces even greater challenges. The placement process typically involves handling bulk data including company details, eligible student lists, interview schedules, and placement statistics. In the existing system, these tasks are often managed through spreadsheets, emails, or physical records, which are not scalable or secure.

The absence of a centralized placement management system results in confusion, duplication of data, and difficulty in generating accurate reports for institutional analysis. Furthermore, there is no analytical or predictive mechanism to evaluate student performance trends or placement probabilities. This makes it hard for the institution to identify at-risk students, provide early intervention, or enhance employability training programs. Communication between departments is also inefficient, leading to poor coordination and missed opportunities for students. Overall, the existing system lacks automation, integration, and real-time accessibility. It is time-consuming, error-prone, and fails to provide a comprehensive view of both academic and placement data. There is no single platform where students, mentors, and placement officers can collaborate efficiently.

3.1 BLOCK DIAGRAM OF EXISTING SYSTEM

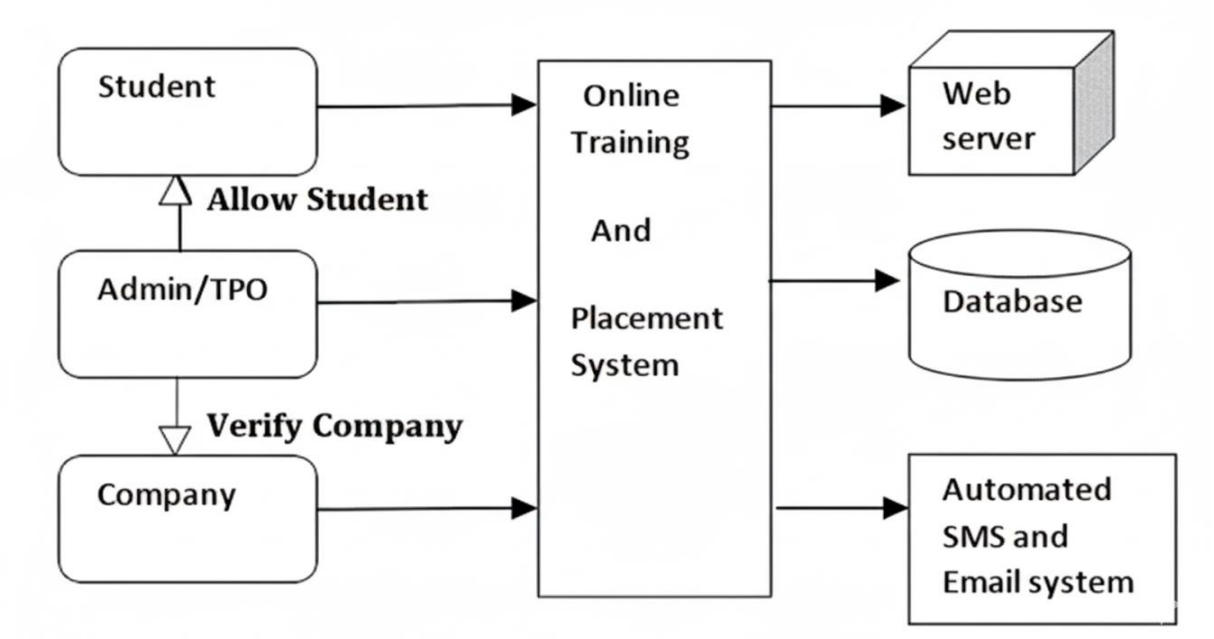

Figure. 3.1 Block diagram of existing system

CHAPTER 4

PROBLEMS IDENTIFIED

The existing Online Training and Placement System is designed to manage student registrations, verify companies, and communicate through automated SMS and emails. Although it digitizes some placement activities, the system suffers from several operational, technical, and functional limitations. These issues reduce the effectiveness of placement operations, limit decision-making capabilities, and create data management challenges for academic institutions.

1. Limited Automation

While the system uses automated notifications through email and SMS, most administrative processes such as student verification, company validation, and drive scheduling are performed manually. Placement officers often have to cross check documents, verify eligibility, and prepare reports by hand. This partial automation causes delays, increases human error, and reduces scalability when handling large student batches.

2. Lack of Integration between Academic and Placement Data

The current system focuses primarily on placement-related activities like company management and student registration but does not incorporate academic details such as CGPA, attendance, or course performance. Because of this separation, it becomes difficult to correlate academic progress with placement outcomes.

3. No Mentor Involvement

One of the major drawbacks of the existing system is the absence of mentor participation. Mentors play a crucial role in tracking students' academic and emotional development, but the current platform does not include mentor dashboards or access to mentee performance.

4. Inefficient Data Management

Although the system stores data in a centralized database, it lacks proper data organization, analytics tools, and visualization features. Information retrieval is often slow, and generating performance or placement reports requires manual compilation. Additionally, data redundancy and inconsistency occur because multiple entries are made for the same student or company.

5. Security and Privacy Concerns

Data security is one of the weakest areas of the existing system. The platform lacks role-based access control (RBAC), data encryption, and audit logs for monitoring user activities. Sensitive student details like CGPA, certificates, and contact information are at risk of unauthorized access or tampering.

6. No Predictive or Analytical Insights

The existing model does not include analytical tools or machine learning modules that can forecast placement success, skill gaps, or academic risks. Placement officers have to rely on manual analysis and intuition to assess student readiness. Without predictive analytics, it becomes challenging to identify students who require additional training or mentoring.

7. Communication Delays

Although an automated SMS and email system exists, it is not fully synchronized with real-time updates. Students may receive notifications about company drives or interviews late, leading to missed opportunities.

8. Lack of Scalability

The existing architecture is designed for small-scale usage and cannot handle heavy data loads when multiple departments or institutions are involved. As the number of students, companies, and drives increases, the system performance degrades. The lack of cloud-based storage or distributed database support further limits its scalability.

9. User Interface Limitations

The user interface (UI) of the current system is outdated and not intuitive for new users. Students often struggle to navigate the platform or upload documents, and placement officers face difficulties while generating reports.

10. Absence of Performance Tracking and Certification Modules

The existing platform does not support uploading or verifying student certifications, project work, or self-assessment assignments. This makes it hard for mentors and companies to evaluate students' skills beyond academic marks. Furthermore, there is no performance history tracking across semesters, preventing students from monitoring their progress over time.

CHAPTER 5

PROPOSED SYSTEM

The Academic and Placement Tracker System is designed to provide an integrated digital platform that automates and simplifies the monitoring of students' academic performance and placement activities. The proposed system addresses the challenges faced in traditional methods such as manual record maintenance, communication delays between students and mentors, and lack of centralized placement data. By combining academic progress tracking, mentoring feedback, and placement analytics into one unified web application, this system ensures transparency, accessibility, and efficiency for all stakeholders. The system enables students to log in securely, update their academic details, view CGPA history, upload certificates, and apply for placement drives. A personalized dashboard allows them to visualize their academic growth and placement readiness. Mentors can access their respective mentees' details, review progress, and provide academic or skill-based guidance. This ensures continuous mentorship and early identification of struggling students. The Placement Officer (PO) has access to all student data across departments, allowing efficient management of company drives, shortlisting of eligible candidates, and generation of placement reports.

The officer can also monitor departmental performance through analytical dashboards. Additionally, the Admin module manages user authentication, role allocation, and overall system maintenance while providing predictive analytics on placement probabilities. The system leverages modern technologies such as ReactJS for the front end, Node.js and Express for the backend, and MySQL for secure data storage.

RESTful APIs facilitate communication between modules, ensuring real-time data synchronization and scalability across departments. The proposed system not only reduces manual workload but also enhances decision-making through data visualization and analytics. It fosters better collaboration among students, mentors, and placement officers while providing a reliable digital record for institutional use. In the long term, this solution promotes academic accountability, improves placement efficiency, and supports data-driven career development initiatives within educational institutions.

5.1 BLOCK DIAGRAM OF PROPOSED SYSTEM

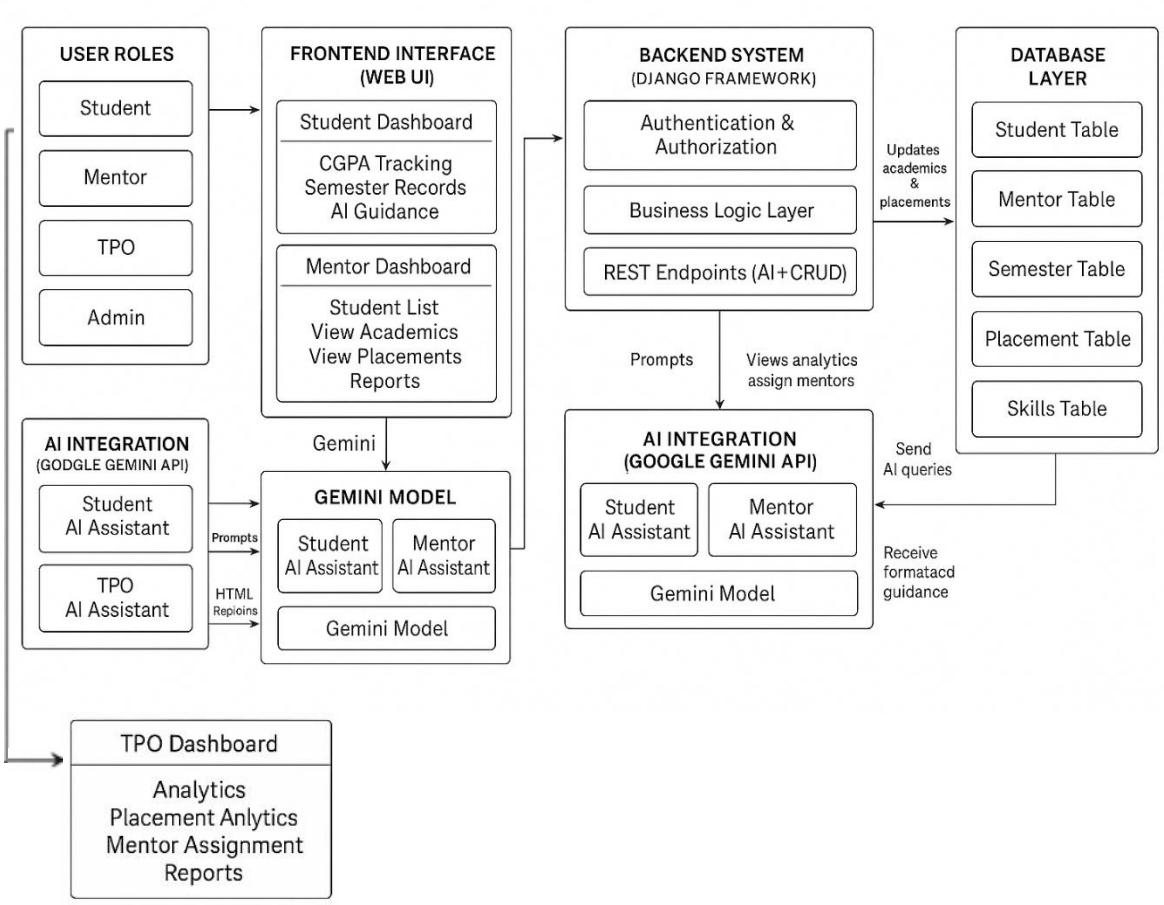

Figure. 5.1 Block Diagram

5.2 USECASE DIAGRAM

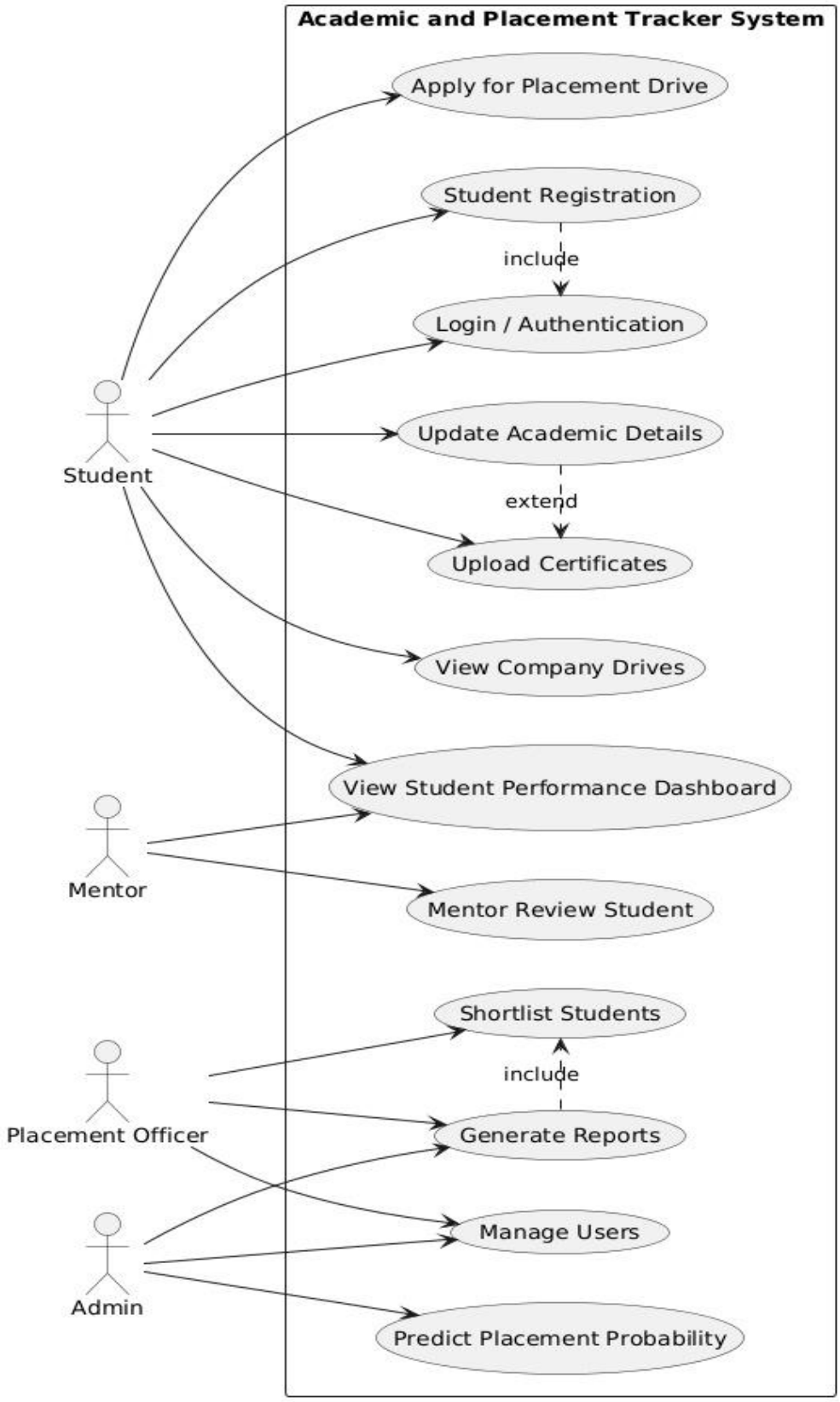

Figure. 5.2 Use case Diagram

5.3 CLASS DIAGRAM

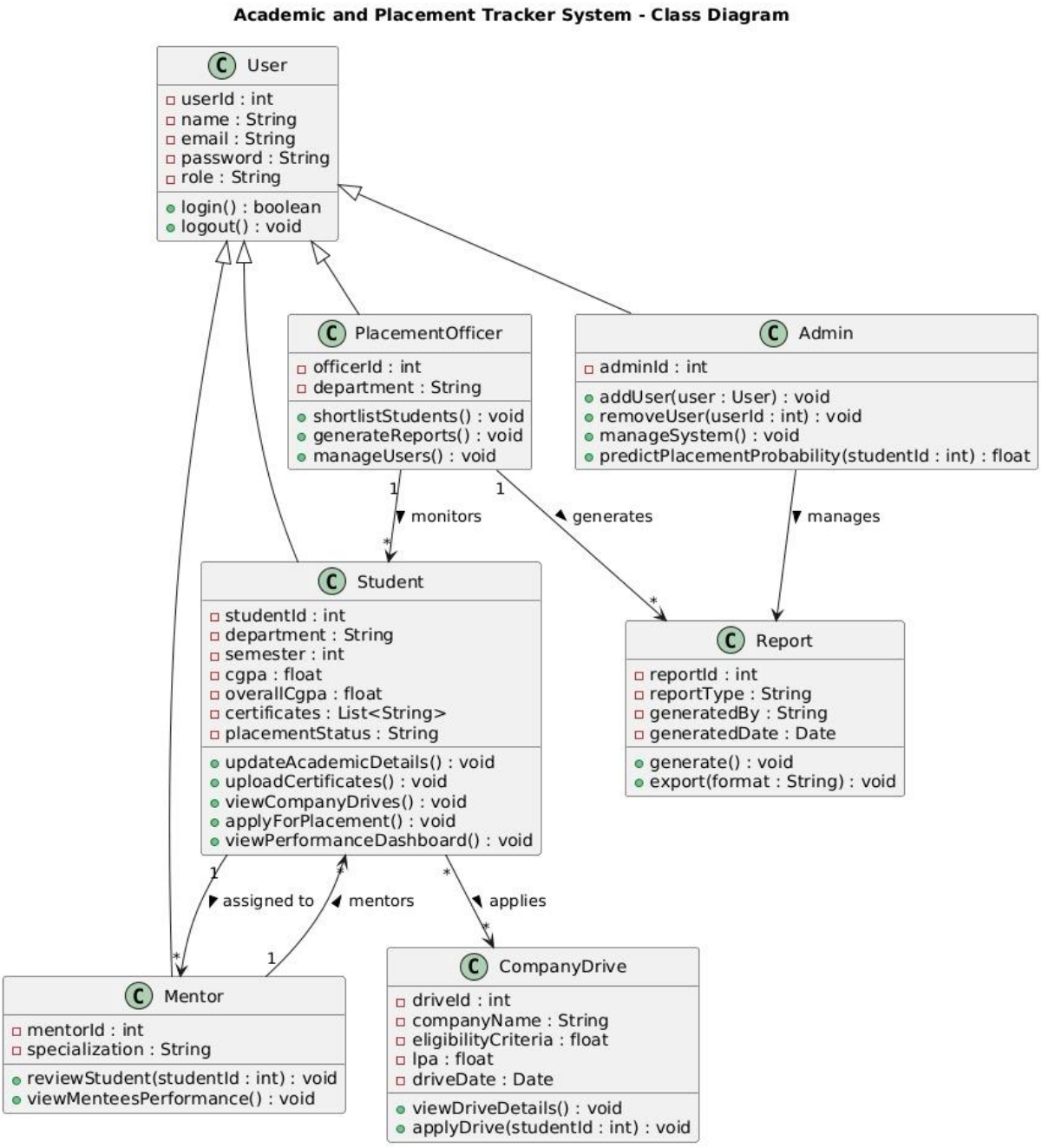

Figure. 5.3 Class Diagram

5.4 STATE DIAGRAM

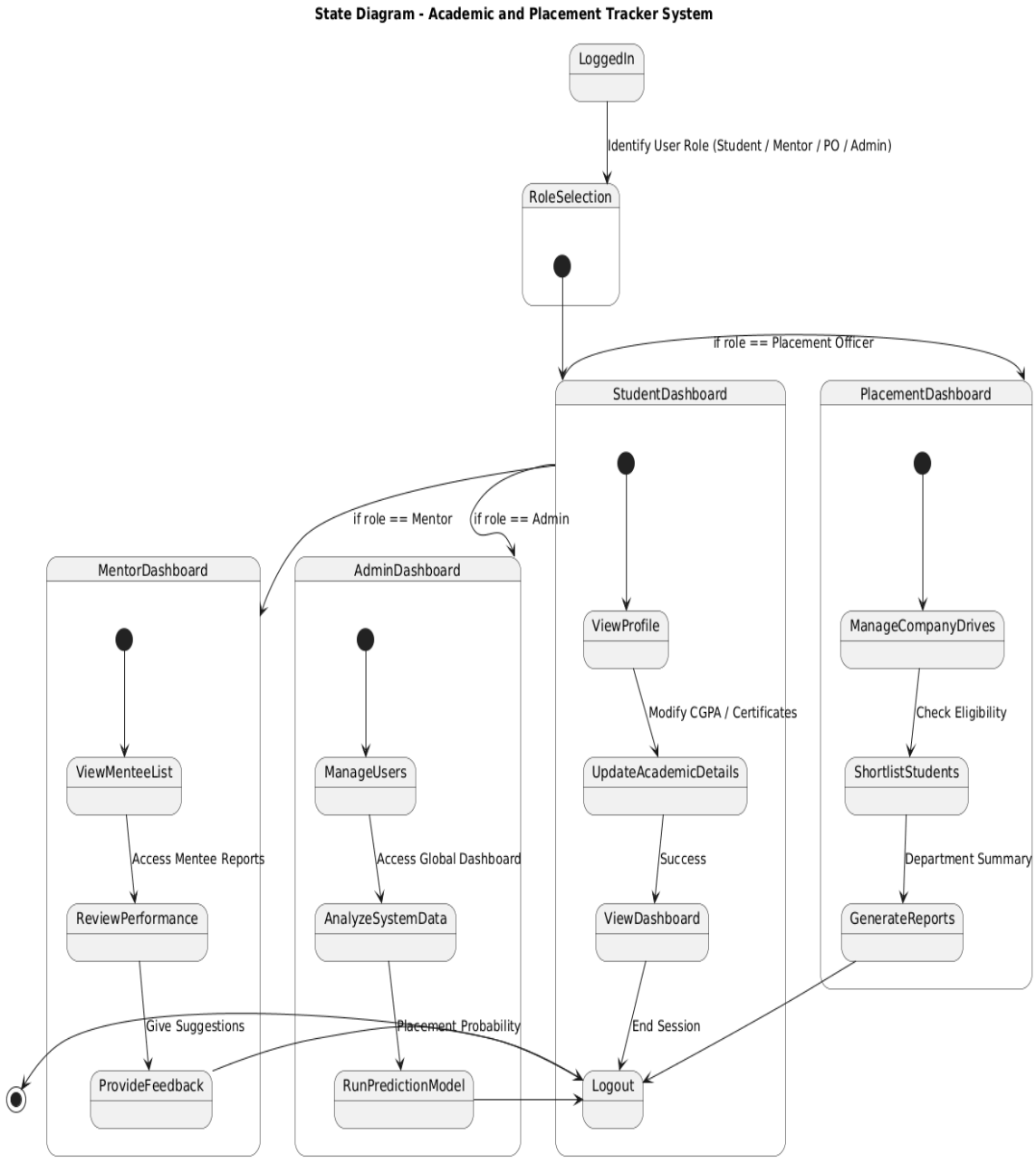

Figure. 5.4 State Diagram

5.5 ACTIVITY DIAGRAM

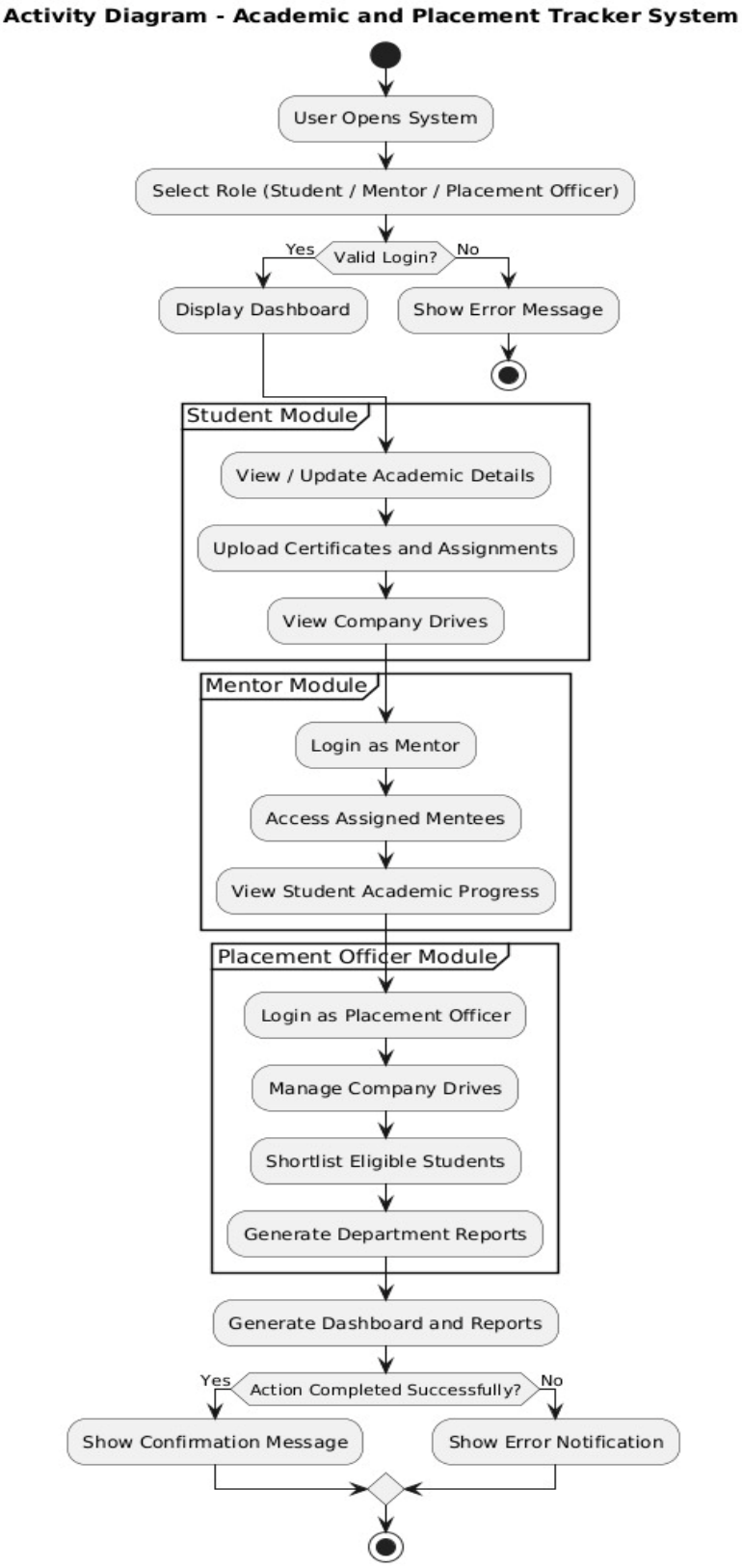

Figure. 5.5 Activity Diagram

5.6 SEQUENCE DIAGRAM

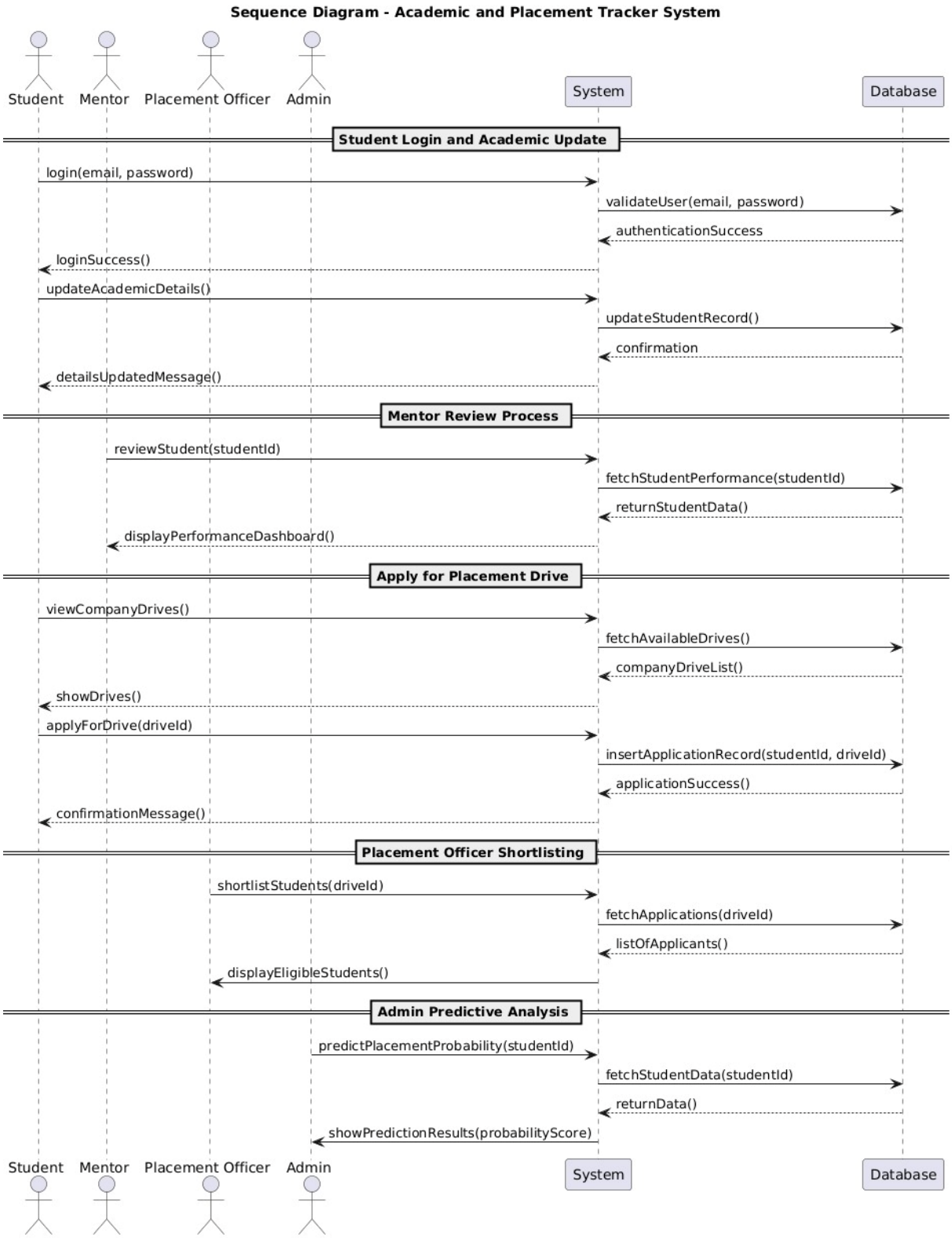

Figure. 5.6 Sequence Diagram

5.7 DEPLOYMENT DIAGRAM

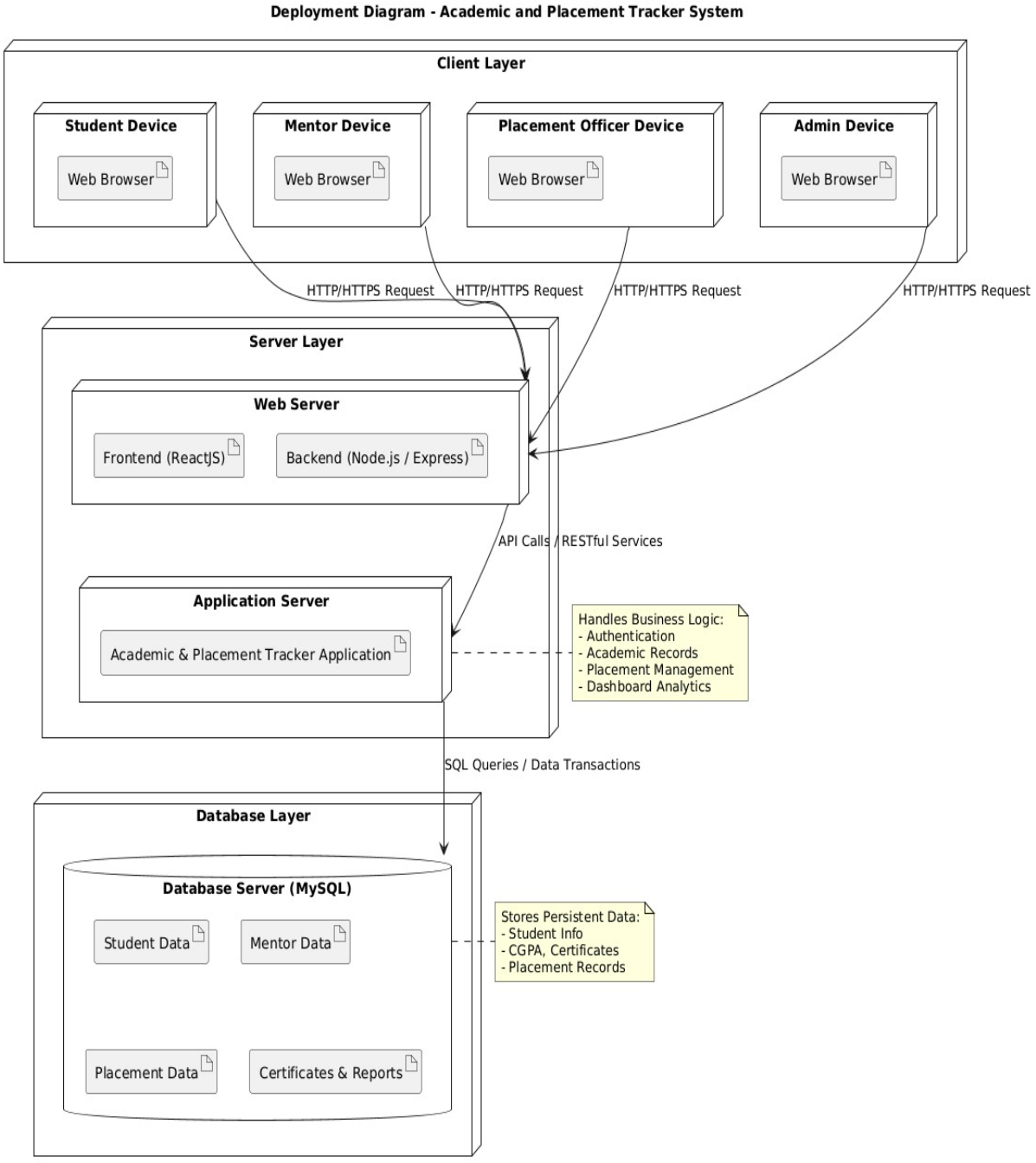

Figure. 5.7 Deployment Diagram

CHAPTER 6

SYSTEM REQUIREMENTS

6.1 HARDWARE REQUIREMENTS

Component	Specification
Processor	-Dual-Core CPU (Intel i3 / AMD equivalent, 2.0+ GHz)
RAM	-Minimum 4 GB
Storage	-50–100 GB HDD/SSD
Display	-1366 × 768 resolution or higher
Network	-Stable internet for package installation & API calls (Gemini)

6.2 SOFTWARE REQUIREMENTS

Component	Specification
Operating System	-Windows 10/11, Linux (Ubuntu 20.04+), or mac OS
Backend Framework	-Django 5.x
Database	-SQLite (development),MySQL / PostgreSQL
AI Integration	-Google Gemini API (google-generative package / google. genai)
Libraries	-Django, google- genai , tqdm, protobuf, pydantic, etc.

CHAPTER 7

SYSTEM IMPLEMENTATIONS

7.1 MODULES DESCRIPTION

Each module in the system is designed to perform specific tasks and interact seamlessly. The Student Module allows students to manage academic and placement details. The Mentor Module enables mentors to monitor and guide mentees. The Placement Officer Module provides access to all student records and placement drives. The Analytics Module generates dashboards and performance insights. The Authentication Module ensures secure, role-based access for all users, maintaining data accuracy and confidentiality.

7.2 LIST OF MODULES

7.2.1 Student Module
7.2.2 Mentor Module
7.2.3 Placement Officer Module
7.2.4 Analytics and Report Module
7.2.5 Authentication and Administration Module

7.2.1 Student Module

It enables students to log in and manage their academic and placement information. Students can enter and edit semester-wise CGPA, overall CGPA, uploaded certificates, self-study assessments, and placement details such as company name and LPA (Lakhs Per Annum). The dashboard provides visual progress tracking and downloadable reports. Allows students to manage personal

academic details such as semester CGPA, overall CGPA, uploaded certificates, and placement information including company name and LPA. It provides an interactive dashboard for easy access and updates.

7.2.2 Mentor Module

The mentor module is designed for faculty mentors to monitor their assigned mentees. Mentors can view academic performance, certificate submissions, and self-study assessments of their students. The system allows mentors to provide personalized feedback and suggest improvement plans. Access is restricted to only the mentor's mentees, ensuring data privacy. This module strengthens mentor–mentee communication and supports continuous academic supervision. Mentors can view and monitor the performance of their assigned mentees, track assessments, and provide academic guidance. Each mentor's access is limited to their respective students for security.

7.2.2 Placement Officer Module

Placement officers with administrative control to view and analyze the data of all students across departments. The PO can upload company details, placement notifications, and job requirements. They can generate department-wise and overall placement reports.

Placement officers can access data for all students, manage company drives, publish placement notifications, and generate department-wise reports. The placement officer manages all departmental data, including student profiles, placement records, and company partnerships. They can schedule placement drives, shortlist eligible candidates, and generate automated reports.

7.2.3 Analytics and Report Module

It generates insights such as department-wise CGPA trends, placement success rates, and mentor performance tracking. Predictive analytics can be incorporated to forecast student placement probabilities. These analytical tools assist mentors and placement officers in making data-driven decisions and identifying students who require academic or placement support.

It provides visual insights and statistical reports on academic progress and placement outcomes, assisting in decision-making and performance analysis. This module visualizes academic and placement data using graphs and charts. The dashboard provides analytical insights into student readiness and placement rates. Notifications and announcements can also be broadcast to all students directly from this module.

7.2.5 Authentication and Administration Module

It ensures secure access to the system through role-based authentication. Separate login credentials are provided for students, mentors, and placement officers. The administrator oversees user registration, role assignment, and access control. Security features such as password encryption, session management, and data validation are implemented to prevent unauthorized access.

It maintains the integrity and confidentiality of institutional data. Implements secure login and role-based access control for students, mentors, and officers, ensuring system integrity and data protection. Ensures secure access control through role-based logins (Student, Mentor, Placement Officer). It handles password encryption, session management, and data protection policies. Admins can add, modify, or delete user accounts and oversee system logs for transparency.

CHAPTER 8

SYSTEM TESTING

8. SYSTEM TESTING

System testing is a crucial phase in software development that ensures the Academic and Placement Tracker operates correctly, efficiently, and securely. It validates the system against functional and non-functional requirements and identifies any defects before deployment. Testing is performed in multiple stages to ensure comprehensive verification of all modules. System testing is a critical phase in the software development life cycle, ensuring that the Academic and Placement Tracker meets both functional and non-functional requirements.

It validates that all modules work correctly individually and collectively, identifies defects, and ensures the system performs efficiently under different conditions. Testing is performed in multiple stages, including unit testing, integration testing, system testing, performance testing, security testing, and usability testing, to provide a comprehensive verification of the system.

8.1 Unit Testing

Unit testing focuses on individual components or modules of the system to verify that each part functions as expected. In this project, modules such as the Student Module, Mentor Module, and Placement Officer Module are tested separately. Each function, like CGPA entry, certificate upload, or dashboard visualization, is validated with different inputs to ensure accurate outputs.

Unit testing helps detect early errors, simplifies debugging, and ensures that each module behaves correctly in isolation before integration. Unit testing

examines individual modules in isolation to verify that each component functions as intended. In this project, modules like the Student Module, Mentor Module, and Placement Officer Module are tested separately.

8.2 Integration Testing

Integration testing ensures that combined modules interact correctly. After unit testing, the Student, Mentor, Placement Officer, Analytics, and Authentication modules are integrated. Scenarios like mentor accessing mentee data, placement officer generating department-wise reports, or a student viewing placement notifications are tested. This stage checks data flow, interface consistency, and communication between modules. Integration testing ensures that modules collectively produce the expected results and handle errors gracefully. Integration testing focuses on the interaction between different modules after unit testing.

It ensures that modules work together correctly and data flows seamlessly across the system. Scenarios such as a mentor accessing mentee performance data, a placement officer generating department-wise placement reports, and a student receiving placement notifications are thoroughly tested. Integration testing verifies that inputs from one module are correctly processed by others and that combined functionalities produce accurate outputs, reducing the risk of interface mismatches or data inconsistencies.

8.3 System Testing

System testing evaluates the complete system as a whole. The Academic and Placement Tracker is tested for functionality, reliability, and compliance with project requirements. Test cases include registration, role-based access control, CGPA updates, certificate uploads, analytics dashboard functionality, and placement notifications.

The objective is to validate end-to-end workflows and confirm that the system meets the specified objectives. System testing evaluates the entire application as a single entity. The objective is to validate end-to-end workflows and ensure compliance with project requirements.

8.4 Performance Testing

Performance testing assesses the system's responsiveness and stability under varying workloads. The project tests database queries, dashboard rendering, report generation, and simultaneous logins by multiple users. Response time, data retrieval speed, and system throughput are measured to ensure optimal performance under normal and peak usage conditions. Performance testing assesses system responsiveness, stability, and speed under normal and peak loads.

Database queries for CGPA retrieval, certificate uploads, and analytics dashboards are tested for response time. The system is also evaluated under multiple concurrent logins by students, mentors, and placement officers to ensure smooth operation. Metrics such as page load time, transaction processing speed, and server throughput are monitored to confirm that the system can handle high user activity without degradation.

8.5 Security Testing

Security testing ensures that user data is protected and unauthorized access is prevented. Role-based authentication, password encryption, session management, and secure access to placement and academic records are verified. For example, the CGPA input function is validated by providing correct and incorrect numeric values to ensure proper validation and storage. Similarly, certificate uploads are tested with various file formats and sizes to ensure reliability. Unit testing helps in detecting early errors, simplifies debugging, and ensures that each module performs its designated tasks independently before integration with other modules.

8.6 Usability Testing

Usability testing evaluates the user interface and experience. Students, mentors, and placement officers interact with the system to assess navigation, ease of data entry, dashboard clarity, and accessibility. For example, the CGPA input function is validated by providing correct and incorrect numeric values to ensure proper validation and storage. Similarly, certificate uploads are tested with various file formats and sizes to ensure reliability.

Usability testing evaluates how intuitive and user-friendly the system is for all user roles. Students, mentors, and placement officers interact with the system to assess navigation, ease of data entry, dashboard clarity, and accessibility. Feedback from test users is incorporated to improve interface design, reduce errors, and enhance efficiency. This ensures that the system is not only functionally robust but also easy to use, increasing user satisfaction and adoption.

CHAPTER 9

RESULTS AND DISCUSSION

The Academic and Placement Tracker was developed to streamline the process of monitoring student performance and placement progress within an educational institution. After implementing and testing the system with sample student data, several significant results were observed. These results demonstrate improvements in academic monitoring efficiency, student progress visibility, and placement management accuracy. One of the key outcomes of the system is the centralization of student academic data. Previously, semester-wise CGPA and assignment details were scattered across spreadsheets, faculty records, or handwritten notes. With the new system, all information is stored in a structured database, enabling easy retrieval and analysis. Mentors reported that they could now quickly access historical performance trends for each student. This allowed them to identify academic weaknesses earlier and offer personalized guidance.

The availability of graphical analytics, such as CGPA trends and subject-wise performance charts, further helped in visualizing academic growth over multiple semesters. The system also generated positive results in mentee–mentor interaction. Before this system, many mentors struggled to track the performance of large groups of students due to time constraints and manual processes. With the tracker, mentors can view real-time performance summaries, assignment completion status, and attendance patterns. This allowed them to prioritize students who require more support. During the testing phase, mentors noted that the system reduced their administrative burden and increased the time available for actual mentoring. A major area of improvement observed was in placement readiness and monitoring. The placement officer (PO) often needed weeks to compile eligibility lists manually for different companies. With the Academic and Placement Tracker, eligibility checks became instantaneous. The system automatically filtered students based on selection criteria such as CGPA, backlogs, and skill data.

This reduced manual work and improved the accuracy of shortlisting. The PO also appreciated the ability to maintain detailed placement histories, including company details, interview stages, and final outcomes. The discussion of results highlights several broader implications. First, the system supports informed decision-making. Faculty and placement teams now rely on data-driven insights rather than manually calculated or estimated values. This enhances fairness and transparency in evaluating student performance and eligibility. Second, the system encourages students to take responsibility for their academic progress. When students can view their CGPA trends, assignments, and placement status, they become more aware of their preparedness and can work on improving weaker areas. Additionally, the system demonstrated strong potential for scalability. During testing, it handled multiple users accessing the database simultaneously without performance drops. This indicates that the system can be used across entire departments or institutions. The modular structure also allows future enhancements such as notifications, resume builders, skill-tracking modules, or integration with learning management systems (LMS).

However, some limitations were also observed. The system's effectiveness depends on consistent data entry by students and mentors. If academic or placement data is not updated regularly, the insights generated may become outdated. In some cases, students required training to understand how to navigate the interface properly. Furthermore, while the system supports CGPA and placement analysis, additional features like automated assignment submissions or attendance tracking could further enhance its usefulness. In conclusion, the results clearly show that the Academic and Placement Tracker improves academic monitoring, simplifies placement processes, and enhances communication among students, mentors, and placement officers. The system brings transparency, efficiency, and accuracy to academic and placement management, making it a valuable tool for educational institutions.

CHAPTER 10
CONCLUSION AND FUTURE WORK

10.1 CONCLUSION

The Academic and Placement Tracker is designed as a comprehensive digital solution to address the growing complexity of academic monitoring and placement management in educational institutions. Traditional methods of tracking student performance, mentoring, and placement activities often involve extensive paperwork, manual coordination, and time-consuming reporting. This system consolidates all these activities into a single, integrated platform, enhancing efficiency, transparency, and reliability. From a functional perspective, the system allows students to manage and update their academic records, including semester-wise CGPA, overall CGPA, uploaded certificates, self-assessment reports, and placement-related information such as company applications and offers with LPA details. Students can visualize their academic progress through interactive dashboards, which provide insights into trends, performance gaps, and overall growth.

By empowering students with timely information about their academic and placement status, the system encourages proactive self-improvement and better preparation for placement opportunities. For mentors, the system provides role-based access to their respective mentees' records, enabling focused supervision and personalized feedback. Mentors can monitor academic performance, review uploaded certificates and assignments, and suggest corrective or enrichment measures. This structured mentorship support reduces the chances of students falling behind academically and ensures targeted guidance for skill development and career readiness. Placement officers benefit from a holistic overview of all students, including department-wise summaries, eligibility statistics, and placement trends. They can efficiently manage recruitment drives, generate automated reports, and maintain accurate records of student placements. This centralization eliminates redundant administrative tasks and facilitates informed decision-making at the institutional level.

The integration of analytics dashboards supports data-driven strategies for improving placement outcomes, identifying departmental strengths and weaknesses, and optimizing resource allocation. Technically, the system employs a modular design, separating responsibilities across Student, Mentor, Placement Officer, Analytics, and Authentication modules. The architecture supports scalability, allowing multiple departments to operate concurrently while maintaining data security and consistency. Role-based authentication and secure access controls ensure that sensitive academic and placement data is protected against unauthorized access. Real-time data synchronization and validation enhance reliability, preventing discrepancies between modules and maintaining high data integrity. Overall, the Academic and Placement Tracker demonstrates how digital transformation can enhance academic monitoring and placement efficiency. By integrating student academic performance, mentorship supervision, and placement management into a unified platform, the system addresses the limitations of traditional approaches. It not only improves operational efficiency but also promotes data-driven decision-making, enhances communication among stakeholders, and provides a foundation for future technological enhancements such as predictive analytics and AI-based career guidance.

10.2 FUTURE ENHANCEMENTS

The Academic and Placement Tracker can be further enhanced to increase functionality, scalability, and user engagement. One significant improvement is the integration of predictive analytics and machine learning, which would allow the system to forecast student placement outcomes, identify skill gaps, and suggest targeted interventions for at-risk students. Developing a mobile application for Android and iOS would enable students, mentors, and placement officers to access dashboards, notifications, and updates on-the-go, improving convenience and engagement. Cloud-based deployment can enhance system scalability, ensure high availability, and facilitate multi-institution support while providing robust data storage and disaster recovery.

Enhancing the notification system with automated email and SMS alerts will ensure timely communication about placement drives, document submissions, and mentor feedback. The integration of third-party learning and certification platforms can enrich student profiles, providing mentors and placement officers with deeper insights for data-driven decision-making. Advanced analytics dashboards with trend analysis, departmental comparisons, and visual reports will help administrators monitor performance and optimize placement strategies.

Strengthening security measures, such as two-factor authentication, encryption, and audit logging, will protect sensitive student data. Incorporating gamification elements, such as achievement badges, progress scores, and leaderboards, can motivate students to actively participate in academic and placement activities. Finally, AI-powered career guidance, including Chatbot's and recommendation engines, can provide personalized advice, company suggestions, and skill improvement recommendations, further bridging the gap between academic performance and career readiness. These enhancements collectively aim to make the system more intelligent, accessible, secure, and user-centric, supporting long-term institutional efficiency and student success.

APPENDIX – A
SOURCE CODE

models.py

```
from django.db import models
from django.contrib.auth.models import User
from django.db.models.signals import post_save
from django.dispatch import receiver
from django.db.models import Avg
class Profile(models.Model):
    USER_TYPES = [
        ('student', 'Student'),
        ('mentor', 'Mentor'),
        ('tpo', 'TPO'),
        ('principal', 'Principal'),
    ]
    user = models.OneToOneField(User, on_delete=models.CASCADE,
related_name="profile")
    user_type = models.CharField(max_length=20, choices=USER_TYPES)
    phone = models.CharField(max_length=15, blank=True, null=True)
    def full_name(self):
        return self.user.first_name + " " + self.user.last_name
```

```
    def _str_(self):
        return f"{self.user.username} ({self.user_type})"
class Mentor(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE,
related_name='mentor_profile')
    name = models.CharField(max_length=100)
    email = models.EmailField()
    department = models.CharField(max_length=100, default="Computer
Science")
    phone = models.CharField(max_length=15, blank=True, null=True)
    def _str_(self):
        return self.name
class Student(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE,
related_name='student_profile')
    name = models.CharField(max_length=100)
    email = models.EmailField()
    branch = models.CharField(max_length=50)
    mentor = models.ForeignKey(Mentor, on_delete=models.SET_NULL,
null=True, blank=True, related_name='students')
    cgpa = models.FloatField(default=0.0)
    attendance = models.PositiveIntegerField(default=0)
```

```
    credits = models.PositiveIntegerField(default=0)
    # NEW FIELD
    current_semester = models.PositiveIntegerField(default=1)
    @property
    def package_lpa(self):
        return float(self.package) / 100000  # convert INR to LPA
    @property
    def top_offer(self):
        # Get only accepted offers
        offers = self.placements.filter(status="Accepted")
        if not offers.exists():
            return None
        # max based on numeric package
        return max(offers, key=lambda o: float(o.package))
    @property
    def full_name(self):
        return f"{self.user.first_name} {self.user.last_name}"
    def _str_(self):
        return self.name
@receiver(post_save, sender=Student)
def create_semesters(sender, instance, created, **kwargs):
    if created:
```

```
        for sem_num in range(1, 9):
            Semester.objects.get_or_create(student=instance,
semester_number=sem_num, defaults={'gpa': 0.0})
class Semester(models.Model):
    student = models.ForeignKey(Student, on_delete=models.CASCADE,
related_name='semesters')
    semester_number = models.PositiveIntegerField()
    gpa = models.FloatField(default=0.0)
    class Meta:
        unique_together = ('student', 'semester_number')
        ordering = ['semester_number']
    def _str_(self):
        return f"{self.student.name} - Sem {self.semester_number}: {self.gpa}"
class Placement(models.Model):
    STATUS_CHOICES = [
        ('Pending', 'Pending'),
        ('Accepted', 'Accepted'),
        ('Rejected', 'Rejected'),]
    UNIT_CHOICES = [
        ('LPA', 'LPA'),
        ('K', 'K'),
    ]
```

```
    student = models.ForeignKey(Student, on_delete=models.CASCADE,
related_name='placements')
    company = models.CharField(max_length=100)
    position = models.CharField(max_length=100)
    package = models.FloatField(default=0.0, help_text="Enter numeric value
only")
    package_unit = models.CharField(max_length=3, choices=UNIT_CHOICES,
default='LPA')
    status = models.CharField(max_length=10, choices=STATUS_CHOICES,
default='Pending')
    created_at = models.DateTimeField(auto_now_add=True)
    @property
    def package_in_lpa(self):
        """Return package in LPA for comparison."""
        if self.package_unit == 'LPA':
            return self.package
        elif self.package_unit == 'K':
            return self.package / 100  # convert thousands to LPA
        return self.package
    def _str_(self):
        return f"{self.student.name} - {self.company} ({self.status})"
```

APPENDIX – B

SCREENSHOTS

Sample Output

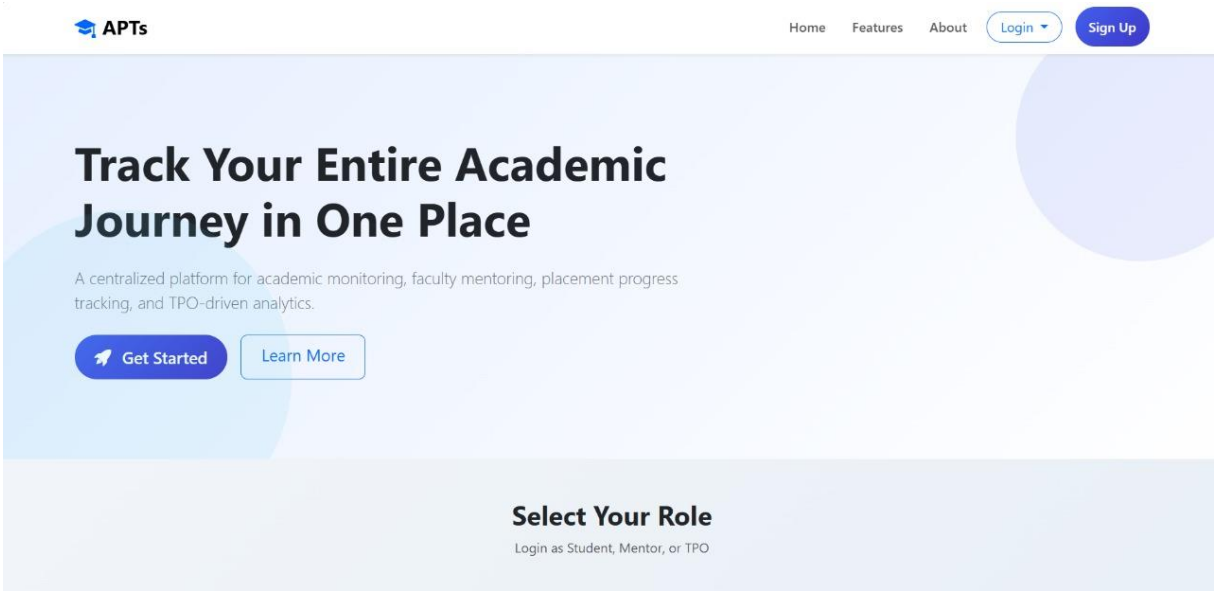

Figure. B.1.Landing Page

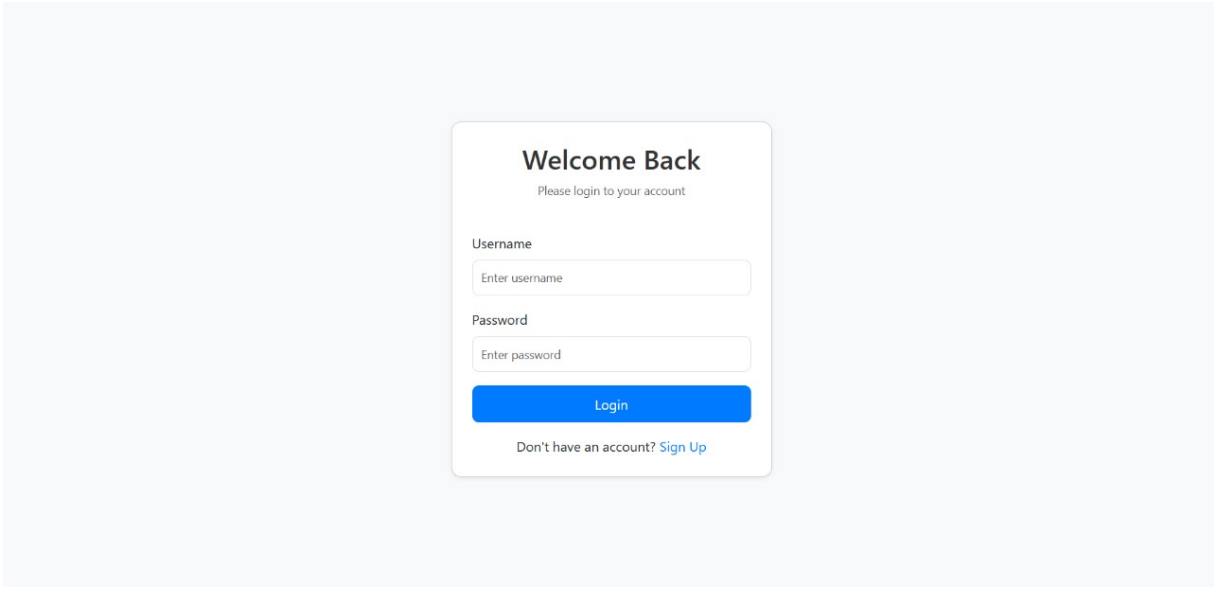

Figure. B.2. Login Page

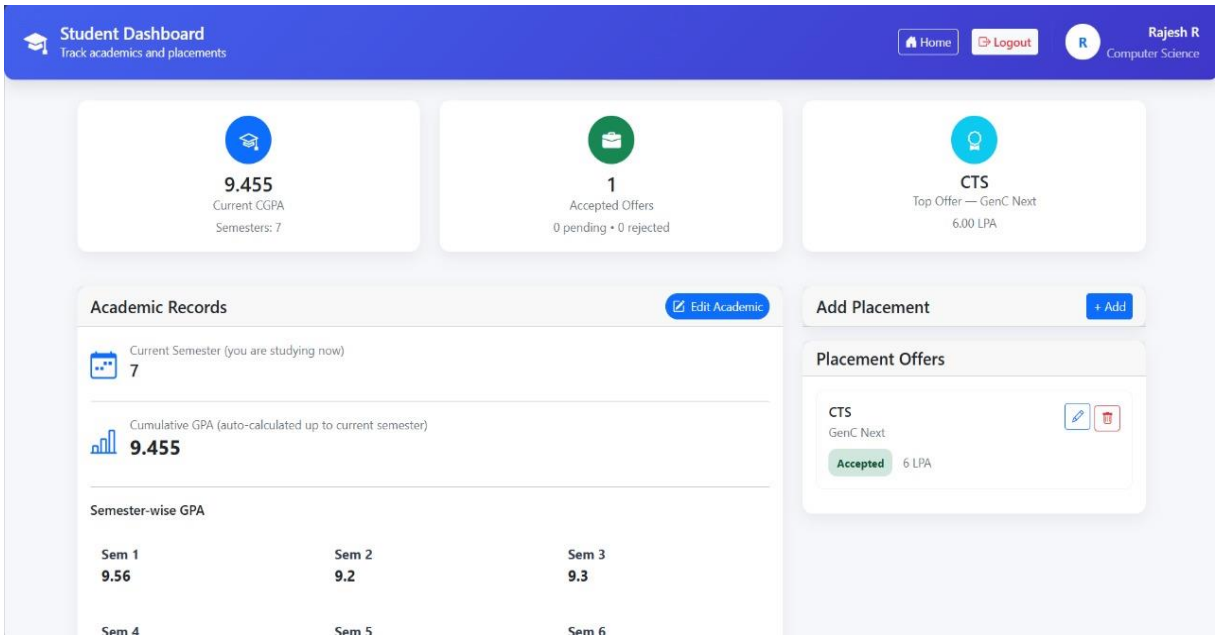

Figure. B.3 Student Dashboard

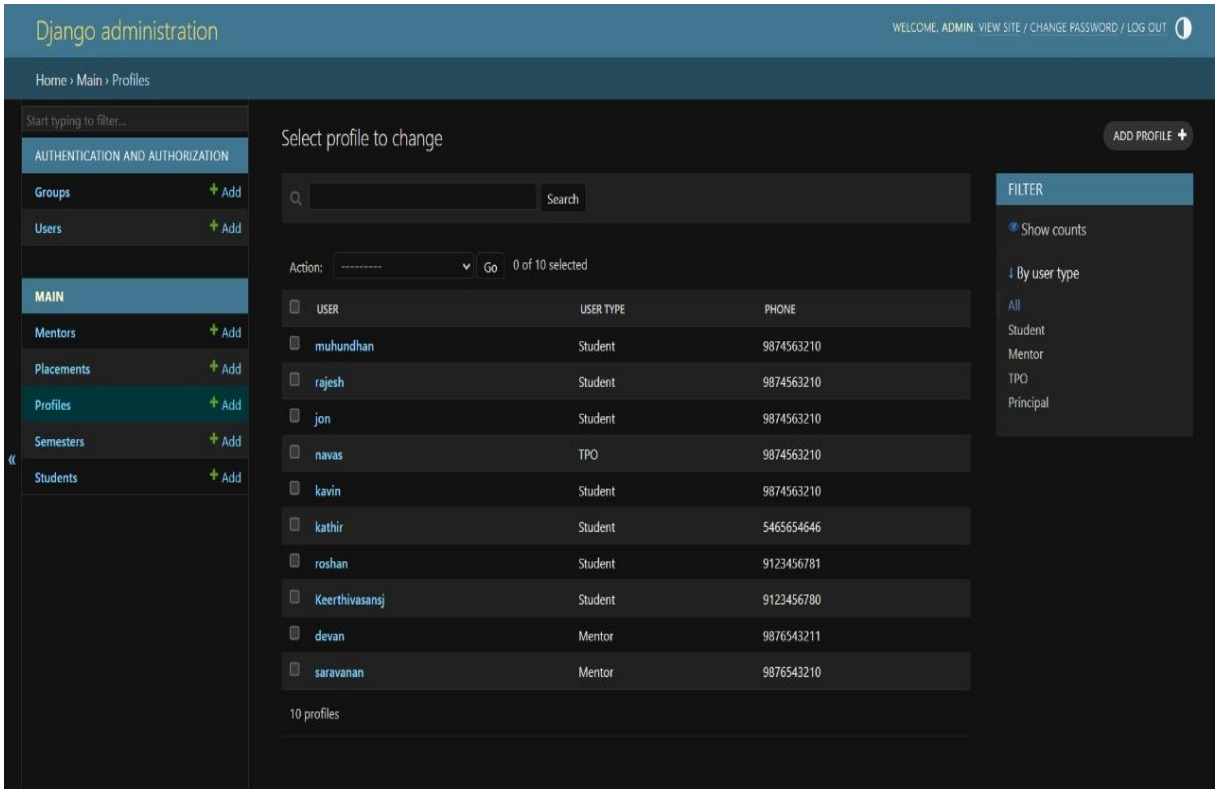

Figure. B.4 Admin Dashboard

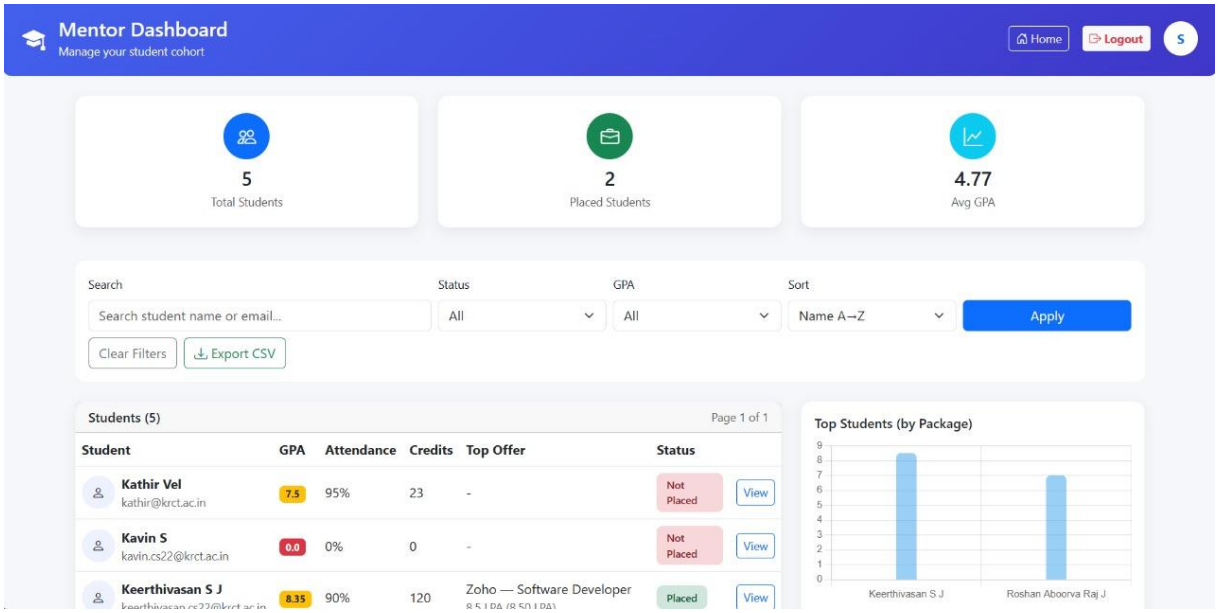

Figure. B.5. Mentor Dashboard

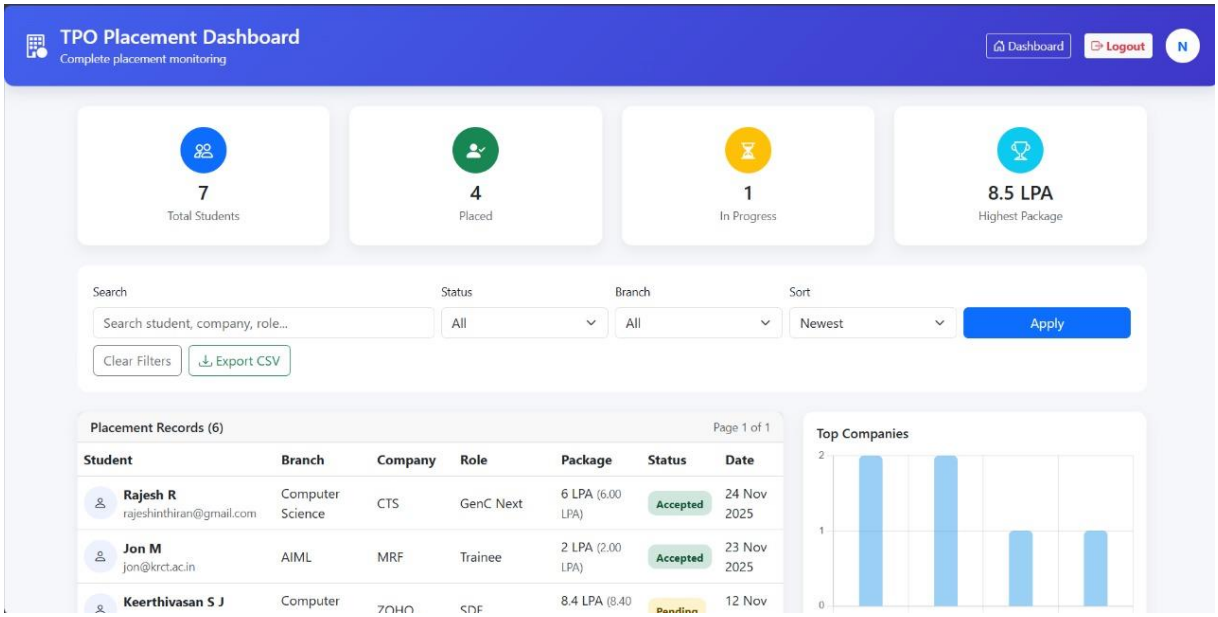

Figure. B.6. TPO Dashboard

REFERENCES

1. A mobile attendance system for university students using Android and Firebase, International Journal of Mobile Computing, Vol. 7, No. 5, pp. 210–218, R. Shankar and P. Kumar (2023).

2. Android-based student attendance management system using Firebase, J. Softw. Eng., Vol. 15, No. 2, pp. 112–119, Smithe and M. Patel (2023).

3. Attendance Manager Using Firebase and Java: Employee Attendance System, GitHub, G. Joseph (2023).

4. Attendance Monitoring: An Android Application for School Attendance Management, GitHub, Magaraa (2022).

5. Design and implementation of a location-based attendance tracking app for students using Android, International Journal of Software Engineering Applications, Vol. 13, No. 4, pp. 77–89, D. Narayan and A. Kumar (2023).

6. Geo-fencing and SMS integration for mobile attendance systems in educational institutions, International Journal of Android Development, Vol. 6, No. 4, pp. 52–65, V. Nair (2022).

7. Geo-loco: Location-Based Attendance System Using Geofencing Technology in Android (Java), GitHub, A. Porwal (2021).

8. Location-Based Attendance System: An Android Application for Automating Class Attendance. International Journal of Computer Applications, 182(3), 45–52, Doe, J., & Smith, A. (2023).

9. Real-time monitoring and attendance tracking with Android and Firebase, International Journal of Digital Systems, Vol. 5, No. 8, pp. 93–100, H. Sharma (2022).

10. Student attendance system with real-time location tracking using Android, Journal of Educational Technology, Vol. 8, No. 3, pp. 115–121, S. Patel and R. Gupta (2022).