Opportunifi-where talent meets opportunity

A

Project Report

Submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

In

Information Technology

Submitted to

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA,
BHOPAL (M.P.)

Guided By
Prof. Kapil Sahu

Submitted By
Sneha Shrivastava (0827IT221137)

Palak Sanjay Gahile(0827IT221103)

Shruti Sharma(0827IT221136)

Mauli Patel(0827IT221091)

DEPARTMENT OF INFORMATION TECHNOLOGY
ACROPOLIS INSTITUTE OF TECHNOLOGY & RESEARCH,
INDORE (M.P.) 452020
2024-2025

Declaration

I hereby declared that the work, which is being presented in the project entitled **Opportunifi-where talent meets opportunity** partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology**, submitted in the department of Information Technology at **Acropolis Institute of Technology & Research, Indore** is an authentic record of my own work carried under the supervision of "**Prof. Kapil Sahu**". I have not submitted the matter embodied in this report for the award of any other degree.

Sneha Shrivastava (0827IT221137)

Palak Sanjay Gahile(0827IT221103)

Shruti Sharma (0827IT221136)

Mauli Patel(0827IT221091)

Prof. Kapil Sahu

Supervisor

+

Project Approval Form

I hereby recommend that the project **Opportunifi-where talent meets opportunity** prepared under my supervision by **Sneha Shrivastava (0827IT221137), Palak Sanjay Gahile (0827IT221103), Shruti Sharma(0827IT221136), Mauli Patel(0827IT221091)** be accepted in partial fulfillment of the requirement for the degree of Bachelor of Technology in Information Technology.

Prof. Kapil Sahu

Supervisor

Recommendation concurred in 2024-2025

Prof. Monika Choudhary

Project Incharge

Prof. Deepak Singh Chouhan

Project Coordinator

Acropolis Institute of Technology & Research

Department of Information Technology

Certificate

The project work entitled **Opportunifi-where talent meets opportunity** submitted by **Sneha Shrivastava (0827IT221137), Palak Sanjay Gahile(0827IT221103), Shruti Sharma(0827IT221136), Mauli Patel(0827IT221091)**is approved as partial fulfillment for the award of the degree of Bachelor of Technology in Information Technology by Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.).

Internal Examiner

Name:……………….

Date: …./…/………..

External Examiner

Name: ……………..

Date: …./…/………..

Acknowledgement

With boundless love and appreciation, we/I would like to extend our/my heartfelt gratitude and appreciation to the people who helped us/me to bring this work to reality. We/I would like to have some space of acknowledgement for them.

Foremost, our/I would like to express our/ my sincere gratitude to our/my supervisor, **Prof. Kapil Sahu** whose expertise, consistent guidance, ample time spent and consistent advice that helped us/me to bring this study into success.

To the project in-charge **Prof. Monika Chaudhary** and project coordinator **Prof. Deepak Singh Chouhan** for their constructive comments, suggestions, and critiquing even in hardship.

To the honorable **Prof. (Dr.) Prashant Lakkadwala**, Head, Department of Information Technology for his favorable responses regarding the study and providing necessary facilities.

To the honorable **Dr. S.C. Sharma**, Director, AITR, Indore for his unending support, advice and effort to make it possible.

Finally, I/we would like to pay my/our thanks to faculty members and staff of the Department of Computer Science & Engineering for their timely help and support.

We/I also like to pay thanks to our/my **parents** for their eternal love, support and prayers without them it is not possible.

Sneha Shrivastava (0827IT221137)

Palak Sanjay Gahile(0827IT221103)

Shruti Sharma(0827IT221136)

Mauli Patel(0827IT221091)

Abstract

This project developed an automated placement portal aimed at simplifying and enhancing the placement process for students and coordinators. The platform serves as a centralized system for managing job postings, tracking applications, and streamlining communication, ultimately reducing manual work and improving efficiency for everyone involved.

The need for such a platform arose from the challenges faced with traditional placement methods, which often relied on manual processes, disorganized tracking of applications, and delays in communication between students and coordinators. These issues created confusion and slowed down the recruitment process. The portal was created to address these challenges by automating key tasks, ensuring timely notifications, and providing a single, easy-to-use platform for all placement activities.

The system was built using modern web technologies, including frontend tools like HTML, CSS, and JavaScript, and backend frameworks such as Node.js and Django. Cloud-based infrastructure was used to ensure the platform could scale and perform reliably. The platform’s key features included automated email notifications, job posting management, and application tracking, all designed with both students and coordinators in mind. The platform was rigorously tested to ensure it was functional, secure, and easy to use.

Once implemented, the portal proved to be a success. It saved valuable time by automating repetitive tasks like sending notifications and managing applications. Students benefited from receiving timely updates about job opportunities and interviews, while coordinators found the process more organized and transparent. Positive feedback from users highlighted how much easier and more efficient the placement process had become.

The significance of these findings lies in how automation can transform traditional processes, making them more efficient and user-friendly. The platform’s success also provides a strong foundation for future improvements, such as adding analytics, personalized job recommendations, and integrating with other job portals, ensuring it remains a valuable tool for the long term.

Table of Content

List of Figures

List of Tables

Abbreviations

1. RDBMS - Relational Database Management System
2. SMTP - Simple Mail Transfer Protocol
3. AWS SES - Amazon Web Services Simple Email Service
4. API - Application Programming Interface
5. UI - User Interface
6. IDE - Integrated Development Environment
7. HTML - Hypertext Markup Language
8. CSS - Cascading Style Sheets
9. AI - Artificial Intelligence
10. ML - Machine Learning
11. GDPR - General Data Protection Regulation
12. CCPA - California Consumer Privacy Act

CHAPTER 1: INTRODUCTION

The project aims to build an automated placement portal that streamlines communication and collaboration between students and placement coordinators. This platform will serve as a centralized hub for managing placement activities, including job postings, application tracking, and updates on recruitment processes. One of the key features of the system is the automatic email notification system, which eliminates the need for manual intervention by sending timely updates to students regarding job openings, interview schedules, results, and other important information.

1.1 Rationale

The placement process in educational institutions often involves multiple stakeholders, including students, placement coordinators, and potential employers. A key challenge faced in this process is ensuring effective communication and timely updates between these parties. Traditionally, emails or notifications related to placement activities require manual effort, leading to delays and potential oversights.

This project aims to streamline and automate the interaction between students and placement coordinators by implementing an automated email system. When a relevant action occurs, such as a student submitting their resume or a new job opportunity being posted, an email will automatically be sent to the necessary parties, ensuring immediate and consistent communication. This automation not only reduces the manual workload for coordinators but also guarantees that students receive real-time updates, fostering an efficient and organized placement process.

By introducing this automated communication feature, the project seeks to enhance the user experience for both students and placement coordinators, ultimately improving the efficiency and effectiveness of the placement system as a whole.

1.2 Existing System

1.2.1 Creatrix Campus’ Placement Management Software

1.2.1.1 Features

- Job Posting and Management

- Resume Building and Parsing

1.2.1.2 Limitations / Drawbacks

- Integration Challenges
- Customization
- Complexity

1.2.2 Superset

1.2.2.1 Features

- Student Profile Management
- Placement Coordinator Dashboard

1.2.2.2 Limitations / Drawbacks

- Limited Customization
- Complexity For New Users

1.2.3 Vercel Placement Portal

1.2.3.1 Features

- Job Posting Management
- Application Tracking System

1.2.3.2 Limitations / Drawbacks

- Automation
- Performance

1.3 Problem Formulation

There are many existing systems for this topic but have certain limitations and problems related to it. Such portals and problems related to it are mentioned below.

1.3.1 Superset

1. Limited Customization: Superset offers a standardized set of features, and institutions may find it challenging to customize the platform based on their needs.
2. Complexity for New Users: For students or placement coordinators who are not tech-

savvy, navigating the platform’s features can be overwhelming. There is often a learning curve, especially for users unfamiliar with automated systems.

1.3.2 Vercel placement portal

1.3.2.1. User Experience: Enhancing navigation and user interface for better flow.

1.3.2.2. Automation: Adding features for automatic notifications and data updates.

1.3.2.3. Performance: Optimizing database queries and page load times for smoother experience.

1.3.2.4. Security: Strengthening authentication for safer user data handling.

1.3.3 Creatrix Campus' Placement Management Software

1.3.3.1. Integration Challenges: May not easily integrate with existing institutional systems.

1.3.3.2. Customization: Requires significant customization to suit specific needs.

1.3.3.3. Complexity: Some advanced features may have a learning curve for users.

The existing placement process between students and coordinators is manual, inefficient, and prone to delays. Students face difficulties in finding job opportunities, tracking application status, and receiving timely updates. Coordinators struggle with managing job postings, student applications

1.4 Proposed System

The proposed solution is a web Based Application:

1.4.1 User Research and Testing

- Gather feedback from students and coordinators.
- Test usability to ensure a smooth experience.
- Create user personas to identify key needs.

1.4.2 Data Analytics and Reporting

- Show student progress and job application success.

- Predict job trends for better planning.
- Create custom reports for coordinators.

1.4.3 Mobile Optimization

- Make the platform mobile-friendly.

1.4.4 Data Security

- Use encryption for secure data transfer.
- Ensure compliance with data protection laws.

1.4.5 Personalized Support and Training.

- Offer personalized training modules for students.
- Give coordinators tools to manage job postings and reports.

1.4.6 Automated Email Notifications

- The system automatically sends emails to students for various events, such as: Campus drives, interview scheduling etc.

1.5 Objectives

1.5.1 Centralized Placement Platform: Create a single platform where students and placement coordinators can interact, access relevant information, and manage placement-related activities efficiently.

1.5.2 Automated Email Notifications: Implement a system that sends automatic email alerts to students and coordinators regarding important updates like job openings, application deadlines, interview schedules, etc., without the need for manual intervention.

1.5.3 User Management and Role-Based Access: Provide different access levels for students, coordinators, and administrators to control the flow of information and communication in a structured manner.

1.5.4 Real-Time Tracking of Applications: Allow students and coordinators to track application status in real time, and trigger emails when changes occur (e.g., shortlisting, interview scheduling).

1.6 Contribution of the Project

The project aims to connect students and placement coordinators through a seamless platform, automating email communications to ensure timely and efficient updates without manual intervention. Below is an analysis of its market potential, innovativeness, and usefulness.

1.6.1 Market Potential

Target Audience: Colleges, universities, and institutions offering placement services.

Demand: With the increasing emphasis on employment opportunities for graduates, a system to streamline placement activities is essential.

Scalability: The project can be scaled to multiple institutions, industries, or even regions.

Opportunity for integration with company recruitment platforms for better collaboration.

Competitive Edge: Time-saving automation. Real-time communication reduces delays and errors, giving it an edge over traditional manual systems.

1.6.2 Innovativeness

Automation of Communication: Automatic email triggers for announcements, application deadlines, interview schedules, and results. Reduces dependency on manual intervention, ensuring prompt updates.

Centralized System: A unified platform for students and coordinators to interact, track progress, and manage placement activities.

Integration of Modern Technologies: Could include AI to match student profiles with job opportunities. Real-time notifications and dashboards for improved visibility.

User Experience: Intuitive and user-friendly design. Improves engagement and satisfaction for both coordinators and students.

1.6.3 Usefulness

For Students: Timely updates ensure they never miss critical opportunities. Provides a structured and organized view of their placement journey.

For Placement Coordinators: Reduces workload and manual effort, enabling them to focus on strategic tasks. Ensures consistent and reliable communication with students.

For Institutions: Enhances the credibility of the placement cell. Better placement statistics due to increased efficiency and reduced errors.

Future Applications: Potential to integrate analytics for tracking placement performance. Support for alumni connections and job updates post-placement.

This project not only addresses current pain points but also introduces innovative solutions that could become a standard for placement activities in educational institutions.

1.7 Report Organization

Provide a roadmap of the report structure, summarizing what each section contains. For example:

Section 1: Introduction – Describes the background, objectives, scope, and significance of the project.

Section 2: Literature Review – Discusses existing systems and highlights the need for the proposed solution.

Section 3: System Analysis – Analyzes the requirements and describes the proposed sysem.

Section 4: System Design – Details the architecture and module design of the system.

Section 5: Implementation – Explains the tools, technologies, and development process.

Section 6: Contribution of the Project – Highlights market potential, innovativeness, and usefulness.

Section 7: Testing and Evaluation – Discusses testing strategies and results.

Section 8: Conclusion – Summarizes findings and suggests future enhancements.

CHAPTER 2: REQUIREMENT ENGINEERING

Requirement Engineering involves systematically gathering, analyzing, and documenting the needs for a system. For a project connecting students and placement coordinators with automated email notifications, the process includes functional and non-functional requirements

2.1 Feasibility Study (Technical, Economical, Operational)

A feasibility study for a project that connects students and placement coordinators with automated email functionality involves evaluating its technical, economic, and operational feasibility.

2.1.1 Technical Feasibility

This aspect assesses whether the technical resources and capabilities exist to execute the project.

Key Considerations:

Platform: Web-based application or mobile app using frameworks like Django, ReactJS, or Node.js for backend and frontend development.

Database: Use databases like MySQL, PostgreSQL, or MongoDB to store user profiles, job postings, and email logs.

Automation Tools: Integrate tools like SMTP, SendGrid, or AWS SES for automated email dispatching.

Integration: APIs for third-party integration to connect calendars or job portals (e.g., Google Calendar API, LinkedIn API).

Infrastructure: Host the application on cloud platforms (e.g., AWS, Azure, or Google Cloud) to ensure scalability and uptime.

Challenges: Ensuring reliable email delivery and handling failures (e.g., spam issues). Setting up secure authentication and role-based access for students and coordinators.

Conclusion: Technically feasible with existing tools and technologies.

2.1.2 Economic Feasibility

This aspect evaluates the cost-effectiveness of the project.

Key Considerations:

Development Costs: Developer salaries or outsourcing costs. Initial design, coding, and testing.

Operational Costs: Cloud hosting and database storage. Subscription to email automation services (e.g., SendGrid starts at ~$15/month).

Maintenance Costs: Regular updates, server upkeep, and bug fixes.

Potential Revenue Models: Subscription fees for companies posting jobs. Freemium model for students, with premium features for coordinators.

Cost-Benefit Analysis: Savings in manual effort and time justify the automation cost. Investment required is moderate and can be recovered through subscription or sponsorship.

Conclusion: Economically feasible if managed within budget constraints and revenue plans.

2.1.3 Operational Feasibility

This aspect assesses whether the project can be implemented and used effectively in the target environment.

Key Considerations:

User Adoption: Easy-to-use interface for students and coordinators to upload resumes, post jobs, and track applications.

Training: Minimal training required due to user-friendly design.

Scalability: Ability to handle a growing number of users and emails without performance issues.

Compliance: Ensure compliance with GDPR or CCPA for data privacy and protection.

Risks: System downtime could disrupt communication. Spam filters could block emails.

Conclusion: Operationally feasible with the right planning, user feedback, and regular system monitoring.Overall Feasibility

Technically: Achievable with modern frameworks and tools.

Economically: Cost-effective with scalable revenue models.

Operationally: Practical and easy to implement with low user friction.

The project is feasible and offers significant value to placement activities.

2.2 Requirement Collection

To collect and analyze the requirements for a project connecting students and placement coordinators with automated email notifications, the following steps can be undertaken:

2.2.1 Discussion Phase

Stakeholders: Identify and involve key stakeholders like placement coordinators, students, college administration, and technical team members.

Purpose: Establish the core objectives of the system.

Example: Automating placement communication, maintaining a student database, tracking placement statuses.

Preferred Features: Automated email generation and scheduling. User-friendly dashboards for students and coordinators. Role-based access controls.

Pain Points: Understand the manual challenges currently faced, like repetitive communication or data tracking inefficiencies.

2.2.2 Requirement Analysis

Functional Requirements:

- User Registration: Students and coordinators should have separate login portals.
- Profile Management:

Students: Add academic and personal details.

Coordinators: Add company details, job roles, eligibility criteria.

- Job Posting: Coordinators should be able to post job openings with deadlines.
- Automated Emails:

Email notifications for: Job postings (sent to eligible students).Interview schedules and updates. Result announcements.

- Application Tracking: Students can view the status of their applications. Coordinators can track applications and shortlisted candidates.

Non-Functional Requirements:

Scalability: Support a growing number of users.

Performance: Ensure emails are sent promptly without delays.

Security: Protect sensitive student data and email credentials.

Availability: System should be accessible 24/7.

Technology Stack:

Frontend: React, Angular, or similar.

Backend: Node.js, Python (Flask/Django).

Database: MySQL, PostgreSQL, or MongoDB.

Email Service: Integration with SMTP, AWS SES, or SendGrid.

Third-Party Integrations: Email APIs for automated emails.

Cloud hosting for deployment (AWS, Azure, or GCP).

2.3 Requirements

2.3.1 Functional Requirements

2.3.1.1 Statement of Functionality

The system should enable seamless communication between students and placement coordinators for placement activities. It should support automatic email notifications to eliminate manual intervention.

- Student Registration and Profile Management
- Students should be able to register using their college ID and create a profile with necessary details like name, contact info, qualifications, and skills.
- Placement Coordinator Portal

- Placement coordinators should have access to manage student data, view applications, and post job/internship opportunities.
- Job/Internship Posting
- Coordinators should be able to create job postings, specifying requirements, deadlines, and selection criteria.
- Automatic Email Notifications
- The system should send automatic emails to students and coordinators.
- To Students: Notify about new job postings, deadlines, or application status.
- To Coordinators: Notify about student applications or updates.
- Application Submission
- Students should be able to apply for jobs or internships online through their portal.
- Dashboard and Analytics
- Both students and coordinators should have dashboards to view:
- For students: Available opportunities and application statuses.
- For coordinators: Student applications and placement statistics.
- Search and Filtering
- Students can search and filter job opportunities.Coordinators can filter students based on eligibility criteria.

2.3.2 Nonfunctional Requirements

2.3.2.1 Statement of Functionality

The system must ensure reliability, performance, scalability, and security while maintaining ease of use.

- Performance
- The system should handle multiple users simultaneously with no noticeable lag (e.g., up to 500 concurrent users).
- Reliability
- The email notification feature must work 99.9% of the time, ensuring no emails are missed.
- Scalability

- The system should be scalable to accommodate a growing number of students and job postings.
- Security
- Data encryption should be used for sensitive information (e.g., email IDs, job applications).
- Role-based access control should be implemented for students and coordinators.
- Ease of Use
- The interface should be user-friendly, with minimal training required for students and coordinators.
- Accessibility
- The system should follow accessibility standards to ensure usability by individuals with disabilities.
- System Integration

The system should integrate seamlessly with email servers and support tools like Google Calendar for scheduling interviews.

- Response Time

Automatic email notifications should be triggered within 1 minute of a relevant event (e.g., job posting, application submission).

2.4 Hardware & Software Requirements

2.4.1 Hardware Requirement (Developer & End User)

2.4.1.1 Server and Client-side Requirements:

- Processor: 2.5 GHz Quad-Core (or higher)
- RAM: 16 GB (or higher)

2.4.1.2 Network Requirements:

- Internet Connectivity: High-speed internet connection

2.4.1.3 Mobile Device for Running the web Application

2.5 Use-case Diagrams

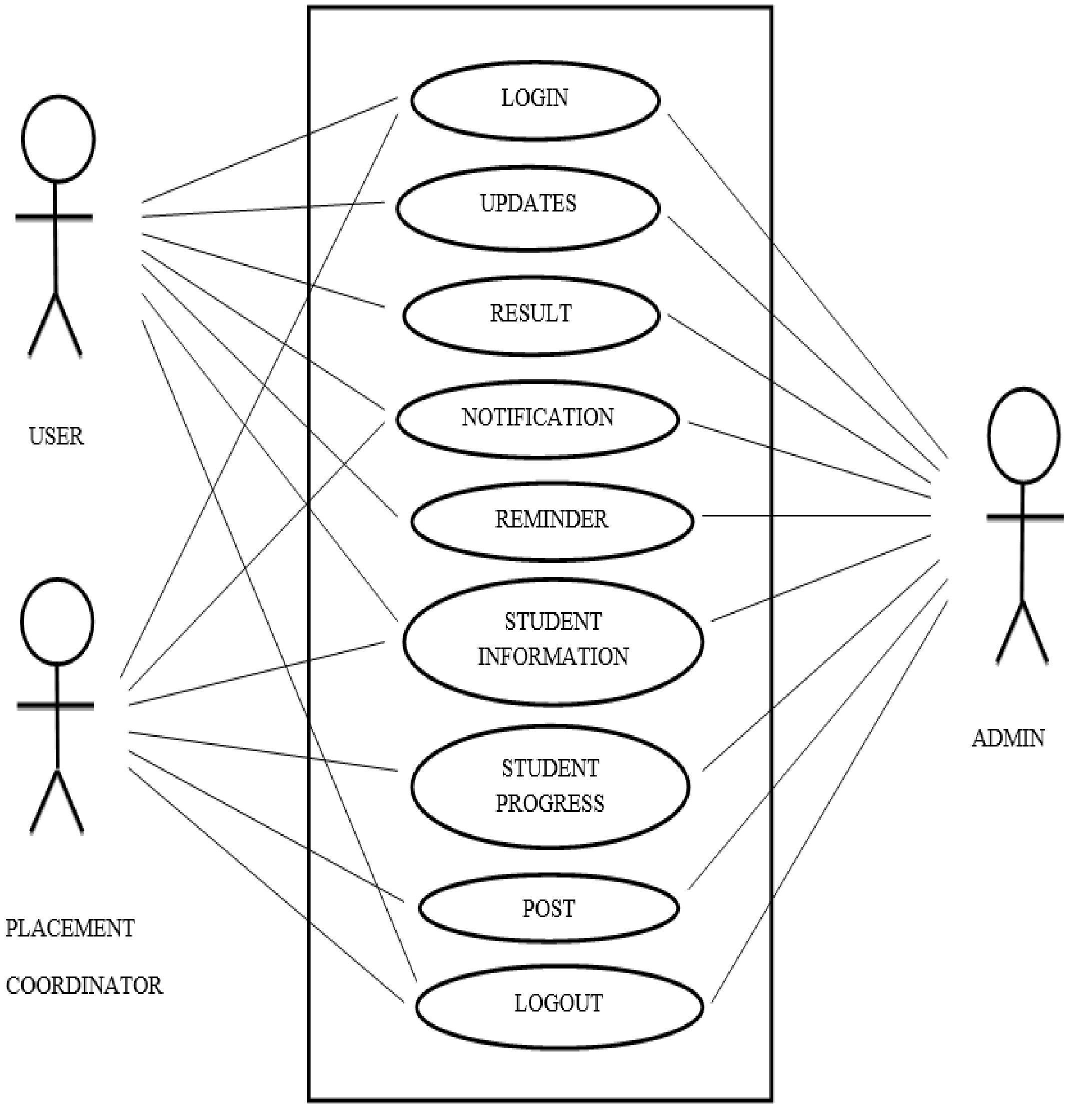


Fig.1 Use Case Diagram of User and Placement coordinator

2.5.1 Use-case Descriptions

Actors: Student, Placement Coordinator, System

Description: Enables students to apply for jobs and track status while coordinators manage postings and applications. The system automates updates and notifications.

Main Flow:

- Student: Logs in, applies for jobs, and tracks updates.
- Coordinator: Posts jobs, manages applications, and updates statuses.
- System: Sends real-time notifications and updates.

CHAPTER 3: ANALYSIS & CONCEPTUAL DESIGN & TECHNICAL ARCHITECTURE

3.1 Technical Architecture

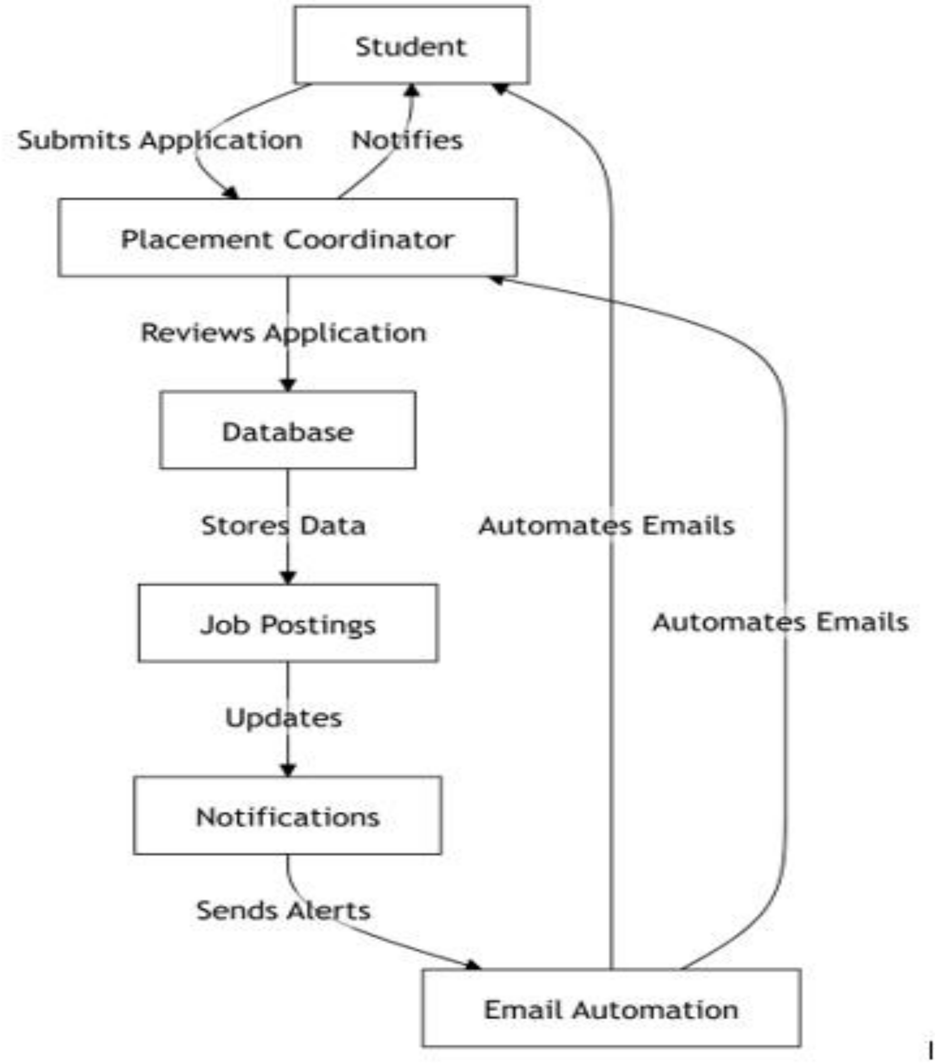


Fig.2 Technical Architecture

3.2 Sequence Diagrams

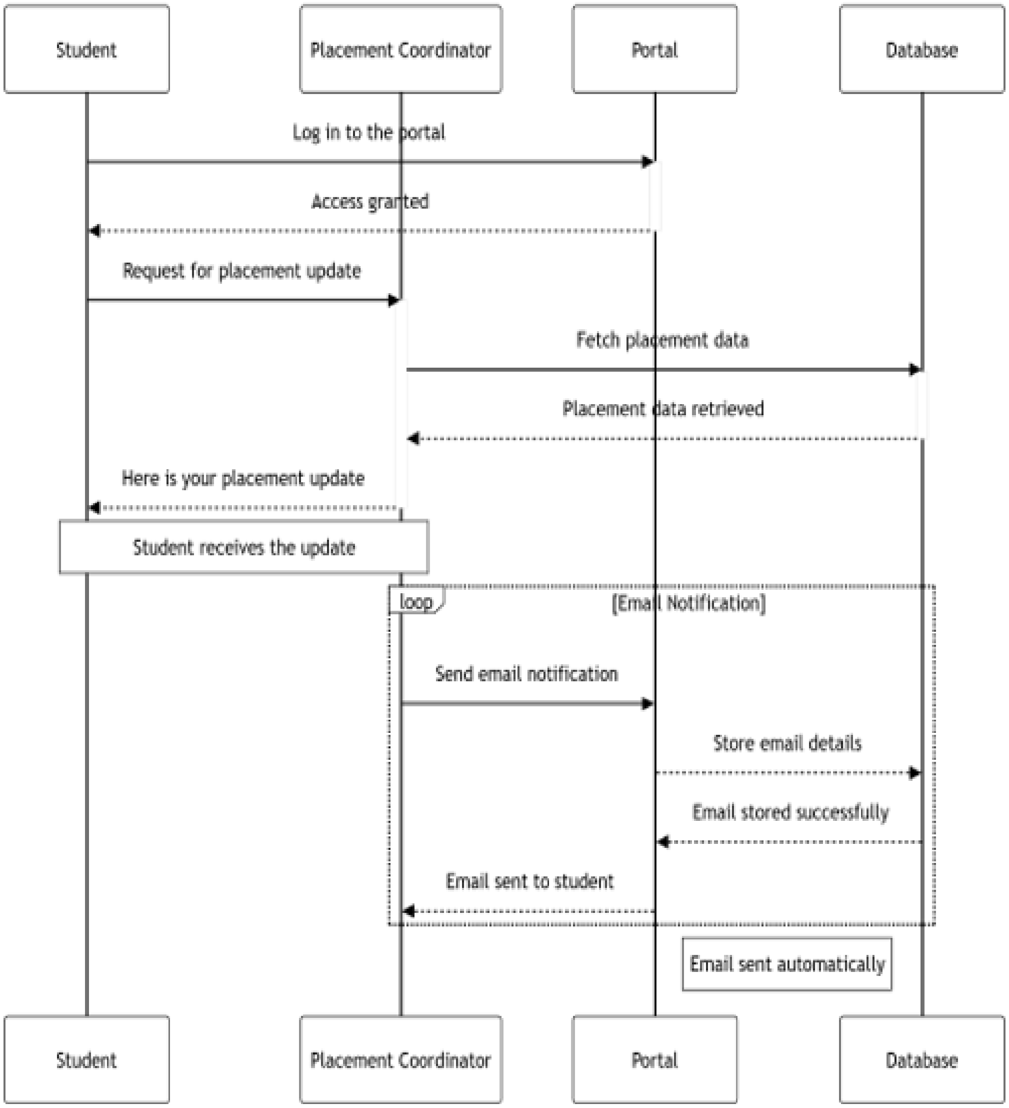


Fig.3 Sequence Diagram

Sequence of Events

3.2.1 The Student initiates an application for a placement.

3.2.2 The Placement Coordinator receives the application.

3.2.3 The Placement Coordinator calls the Email Service to send a notification email to the student.

3.2.4 The Email Service sends the email to the Student.

3.3 Class Diagrams

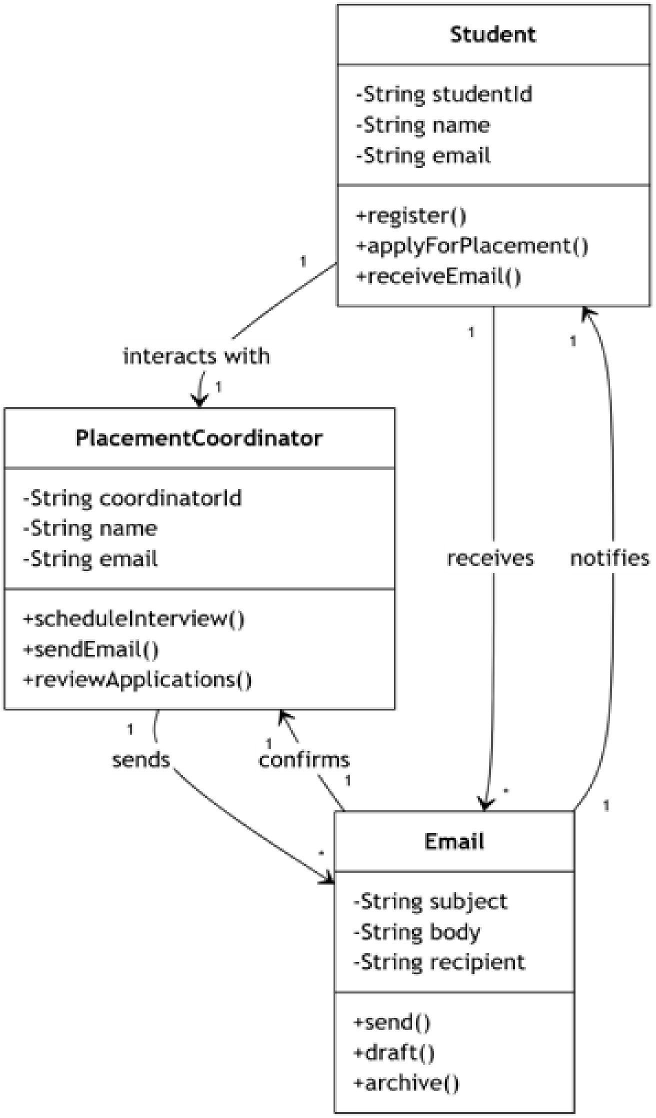


Fig.4 Class diagram

3.3 DFD

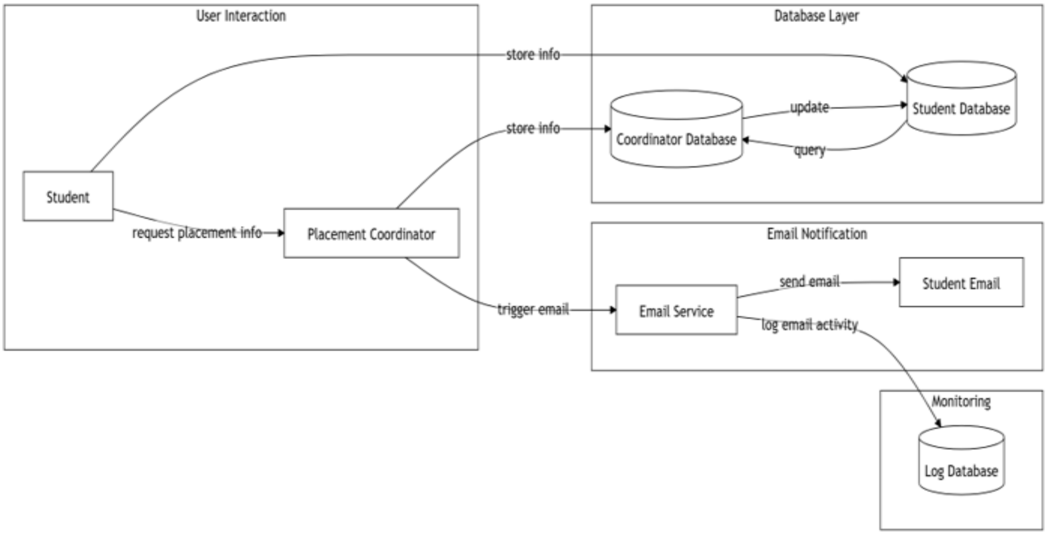


Fig.5 Data Flow Diagram

3.5 User Interface Design

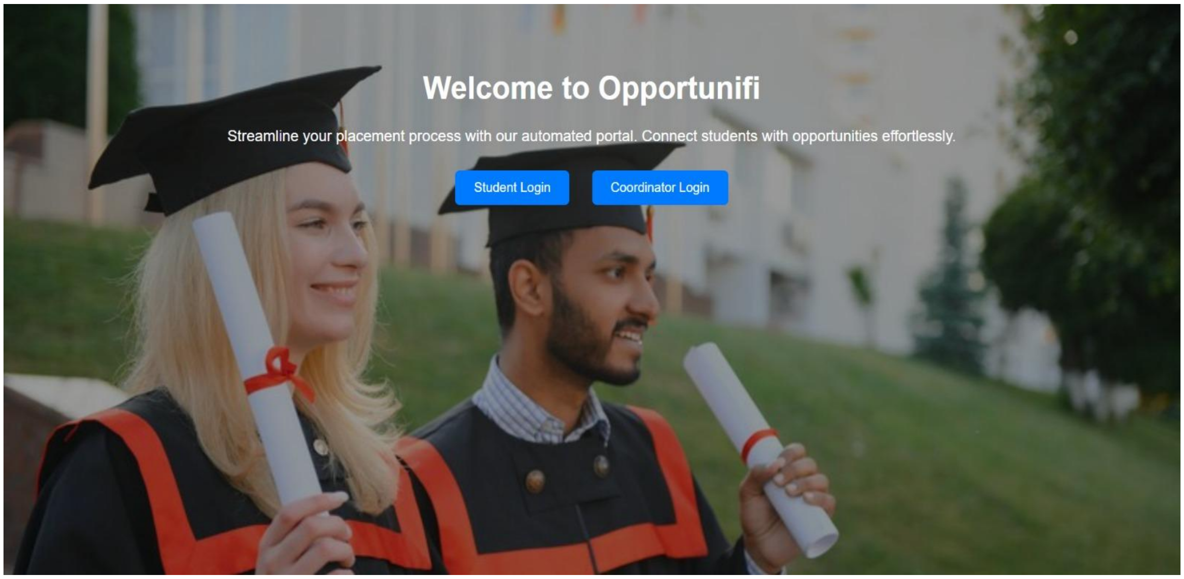
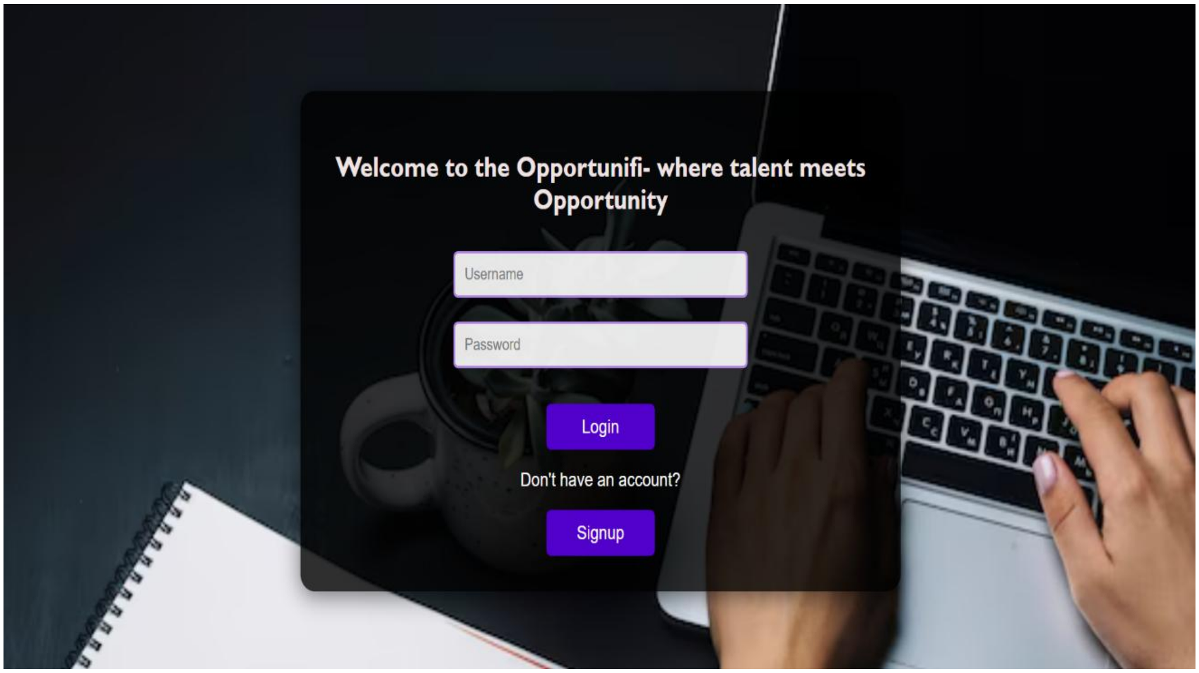
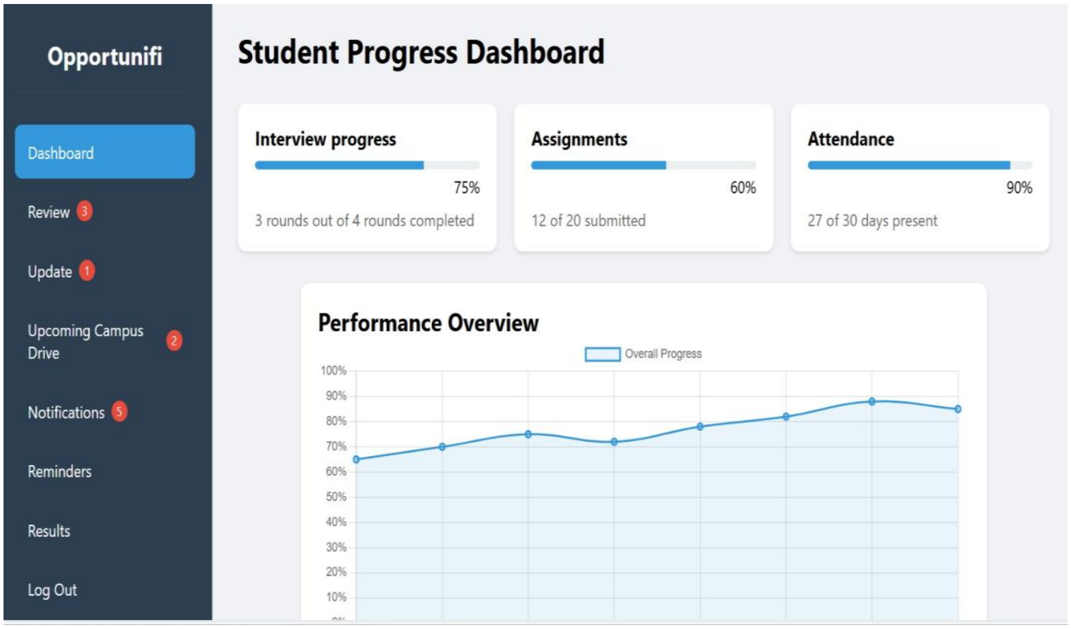

Welcome to Opportunifi
Streamline your placement process with our automated portal. Connect students with opportunities effortlessly.
Student Login
Coordinator Login

Fig. 6 User Interface Design



Welcome to the Opportunifi- where talent meets Opportunity

Fig..6.1 User Interface Diagram



Student Progress Dashboard

Interview progress

75%

3 rounds out of 4 rounds completed

Assignments

60%

12 of 20 submitted

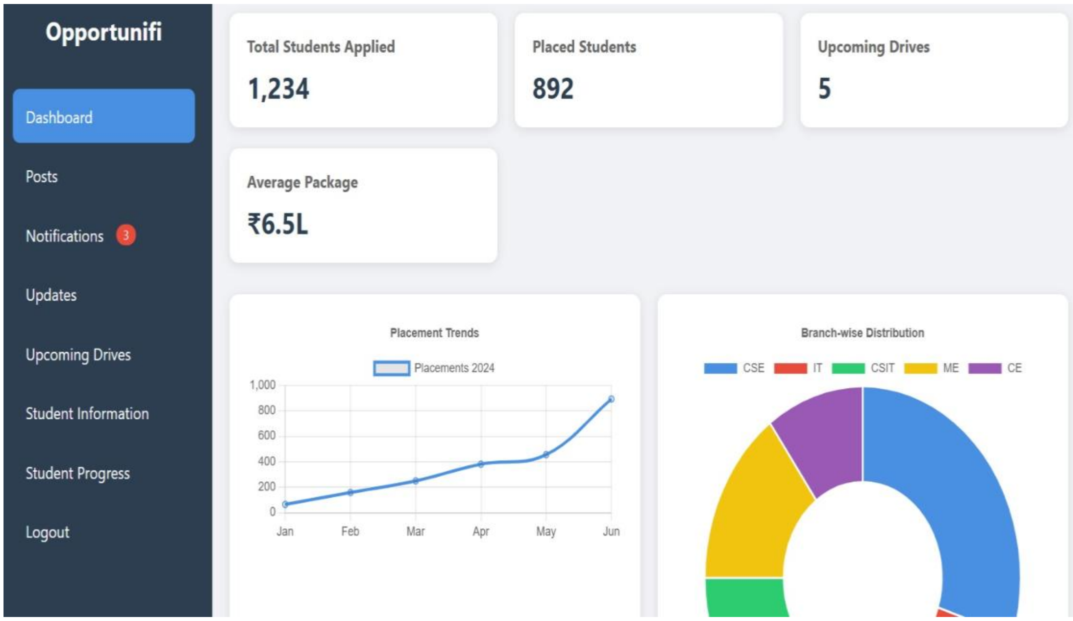
Attendance

90%

27 of 30 days present

Performance Overview

Fig.6.2 Student Dashboard



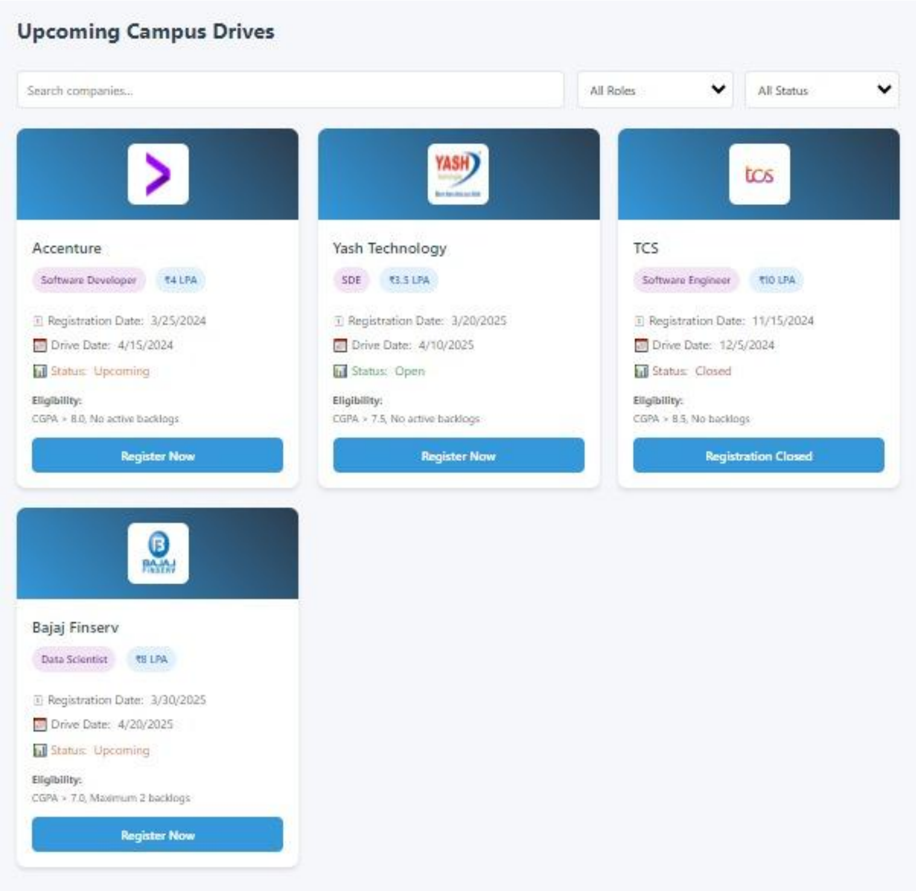
Total Students Applied
1,234

Placed Students
892

Upcoming Drives
5

Average Package
₹6.5L

Fig 6.3 Coordinator Dashboard



Upcoming Campus Drives

Accenture
Software Developer ₹4 LPA
Registration Date: 3/25/2024
Drive Date: 4/15/2024
Status: Upcoming
Eligibility:
CGPA > 8.0, No active backlogs

Yash Technology
SDE ₹3.5 LPA
Registration Date: 3/20/2025
Drive Date: 4/10/2025
Status: Open
Eligibility:
CGPA > 7.5, No active backlogs

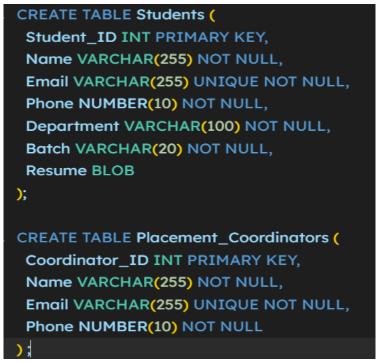
TCS
Software Engineer ₹10 LPA
Registration Date: 11/15/2024
Drive Date: 12/5/2024
Status: Closed
Eligibility:
CGPA > 8.5, No backlogs

Bajaj Finserv
Data Scientist ₹8 LPA
Registration Date: 3/30/2025
Drive Date: 4/20/2025
Status: Upcoming
Eligibility:
CGPA > 7.0, Maximum 2 backlogs

Fig 6.4 Upcoming Campus Drives

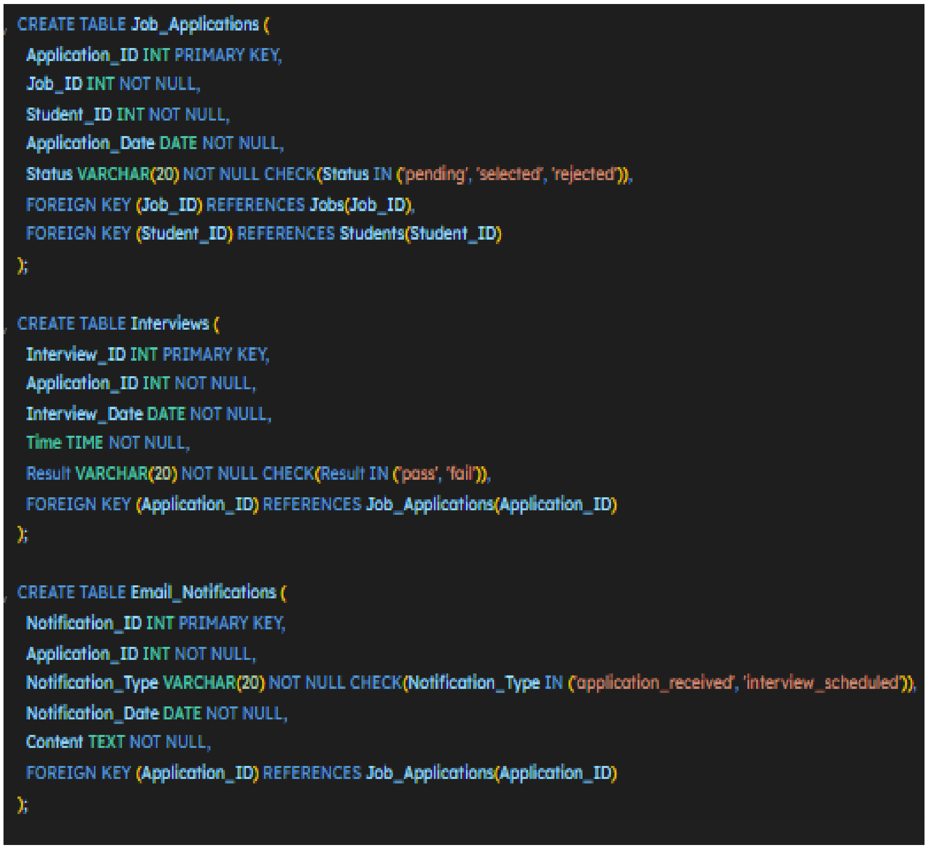
3.6 Data Design

3.6.1 Schema Definitions



```
CREATE TABLE Students (
  Student_ID INT PRIMARY KEY,
  Name VARCHAR(255) NOT NULL,
  Email VARCHAR(255) UNIQUE NOT NULL,
  Phone NUMBER(10) NOT NULL,
  Department VARCHAR(100) NOT NULL,
  Batch VARCHAR(20) NOT NULL,
  Resume BLOB
);

CREATE TABLE Placement_Coordinators (
  Coordinator_ID INT PRIMARY KEY,
  Name VARCHAR(255) NOT NULL,
  Email VARCHAR(255) UNIQUE NOT NULL,
  Phone NUMBER(10) NOT NULL
);
```



```
CREATE TABLE Job_Applications (
  Application_ID INT PRIMARY KEY,
  Job_ID INT NOT NULL,
  Student_ID INT NOT NULL,
  Application_Date DATE NOT NULL,
  Status VARCHAR(20) NOT NULL CHECK(Status IN ('pending', 'selected', 'rejected')),
  FOREIGN KEY (Job_ID) REFERENCES Jobs(Job_ID),
  FOREIGN KEY (Student_ID) REFERENCES Students(Student_ID)
);

CREATE TABLE Interviews (
  Interview_ID INT PRIMARY KEY,
  Application_ID INT NOT NULL,
  Interview_Date DATE NOT NULL,
  Time TIME NOT NULL,
  Result VARCHAR(20) NOT NULL CHECK(Result IN ('pass', 'fail')),
  FOREIGN KEY (Application_ID) REFERENCES Job_Applications(Application_ID)
);

CREATE TABLE Email_Notifications (
  Notification_ID INT PRIMARY KEY,
  Application_ID INT NOT NULL,
  Notification_Type VARCHAR(20) NOT NULL CHECK(Notification_Type IN ('application_received', 'interview_scheduled')),
  Notification_Date DATE NOT NULL,
  Content TEXT NOT NULL,
  FOREIGN KEY (Application_ID) REFERENCES Job_Applications(Application_ID)
);
```

Fig.7.1 Schema Definition

3.6.2 E-R Diagram

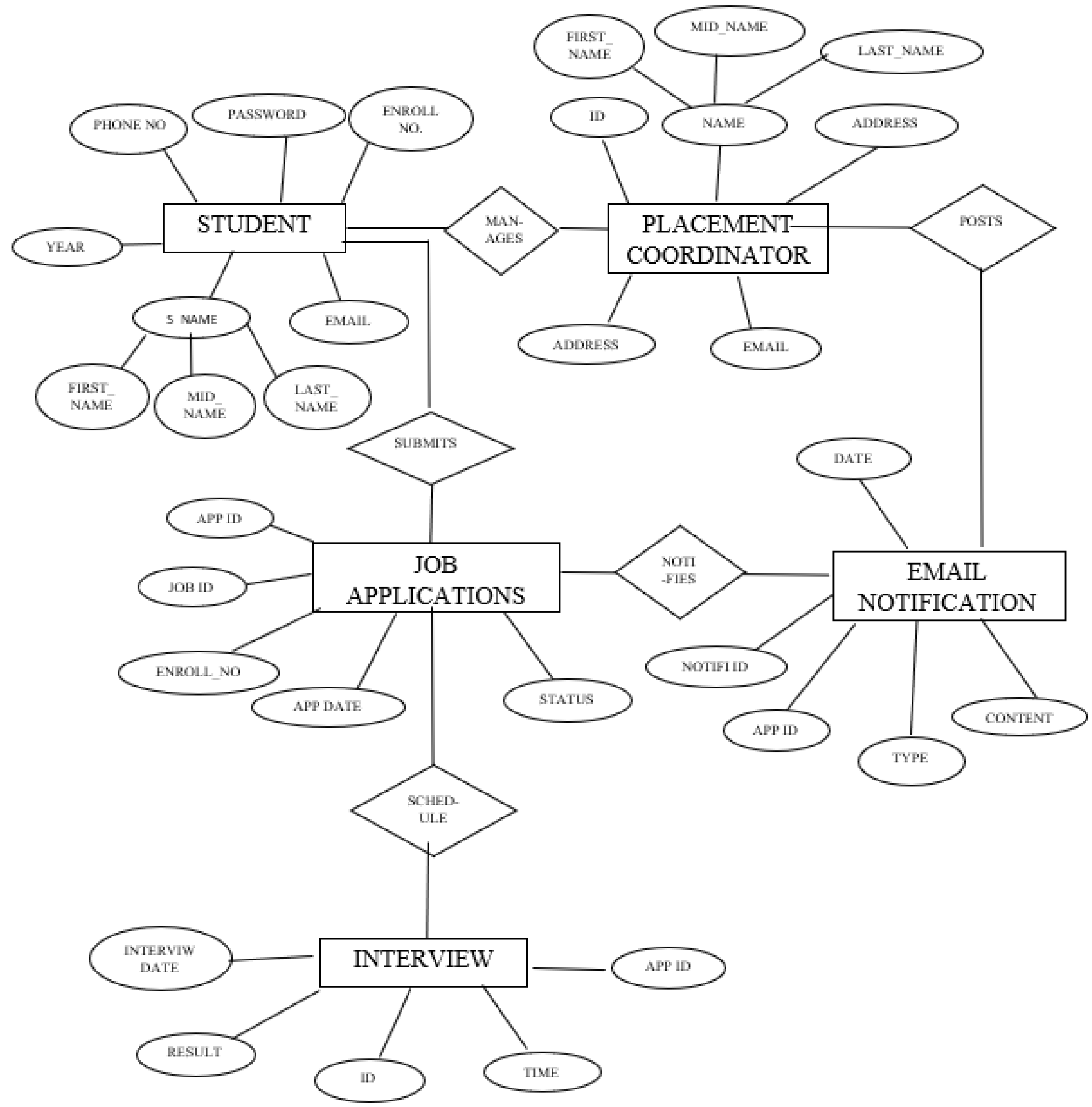


Fig.7.2 E-R Diagram

CHAPTER 4: IMPLEMENTATION & TESTING

4.1Methodology

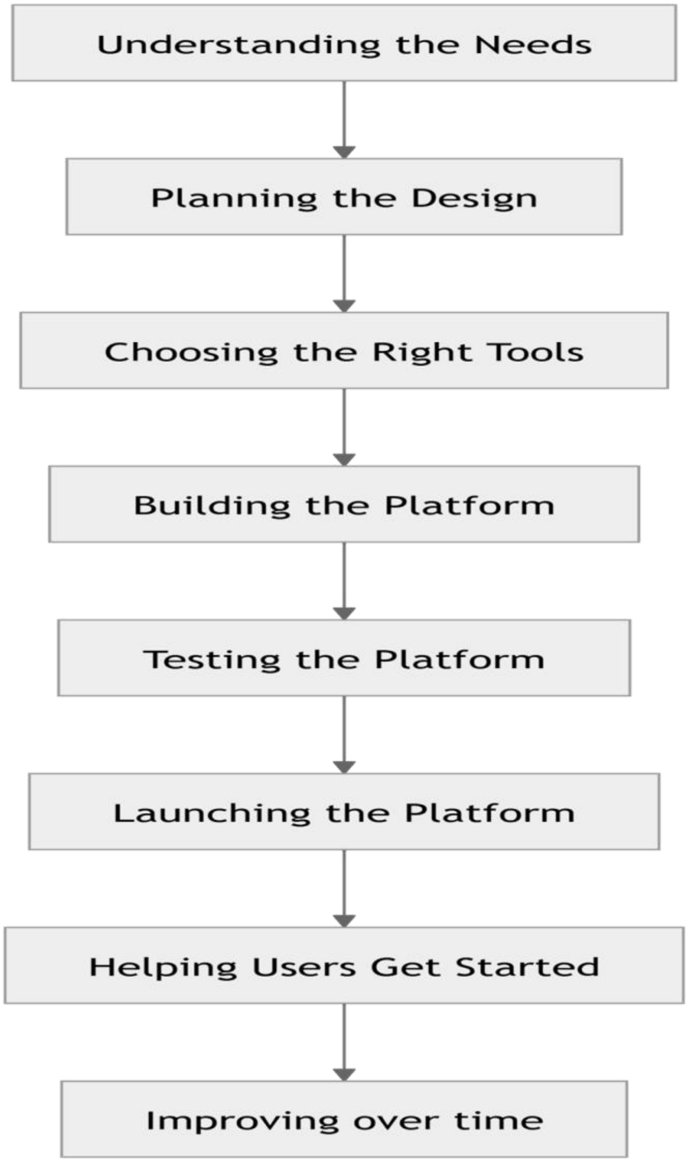
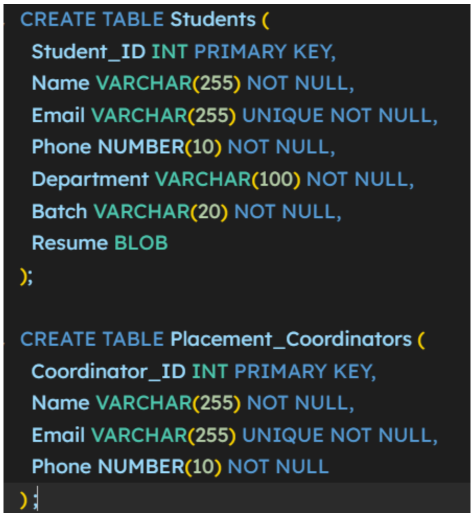


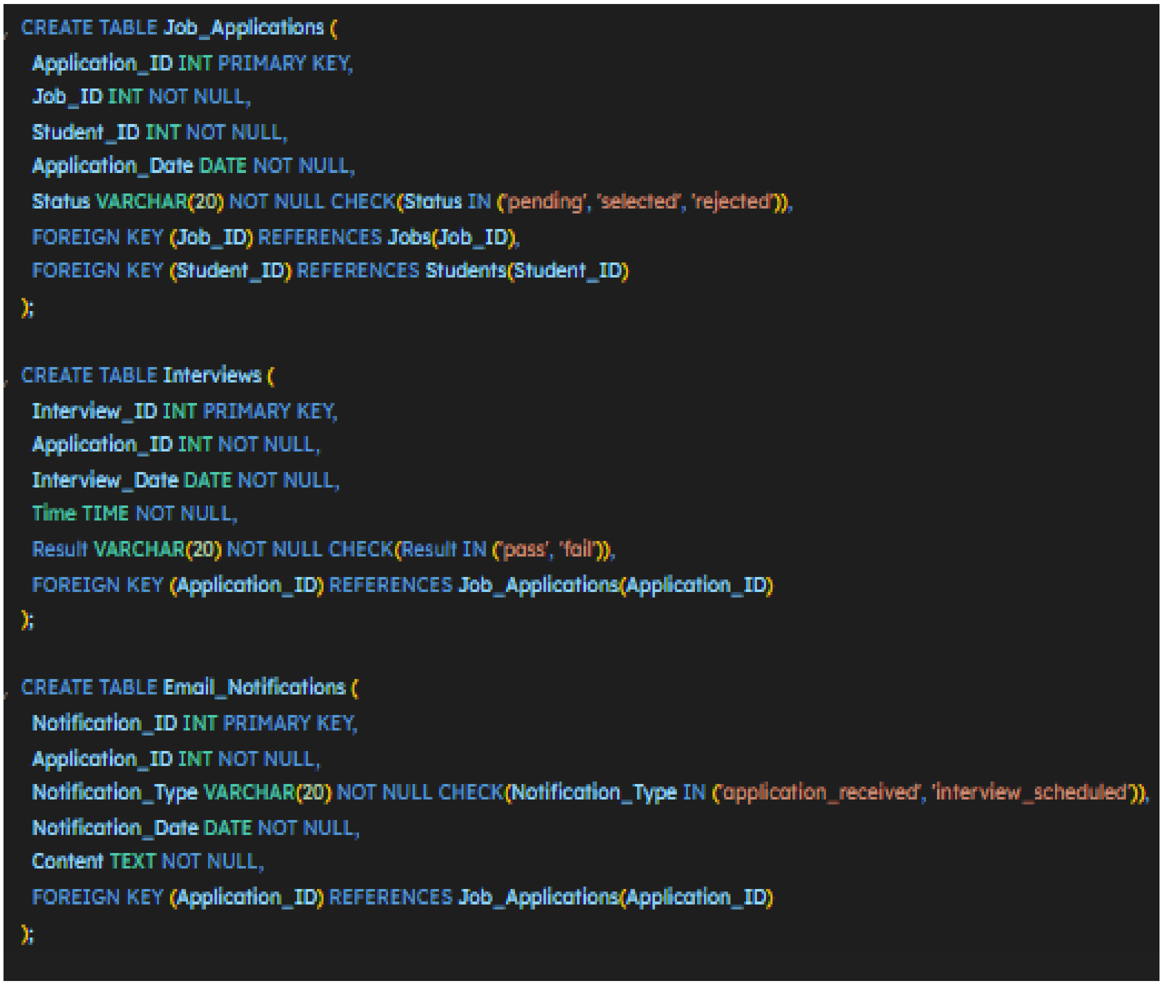
Fig.8 Methodology

4.1.1 Proposed Algorithm



```
CREATE TABLE Students (
  Student_ID INT PRIMARY KEY,
  Name VARCHAR(255) NOT NULL,
  Email VARCHAR(255) UNIQUE NOT NULL,
  Phone NUMBER(10) NOT NULL,
  Department VARCHAR(100) NOT NULL,
  Batch VARCHAR(20) NOT NULL,
  Resume BLOB
);

CREATE TABLE Placement_Coordinators (
  Coordinator_ID INT PRIMARY KEY,
  Name VARCHAR(255) NOT NULL,
  Email VARCHAR(255) UNIQUE NOT NULL,
  Phone NUMBER(10) NOT NULL
);
```



```
CREATE TABLE Job_Applications (
  Application_ID INT PRIMARY KEY,
  Job_ID INT NOT NULL,
  Student_ID INT NOT NULL,
  Application_Date DATE NOT NULL,
  Status VARCHAR(20) NOT NULL CHECK(Status IN ('pending', 'selected', 'rejected')),
  FOREIGN KEY (Job_ID) REFERENCES Jobs(Job_ID),
  FOREIGN KEY (Student_ID) REFERENCES Students(Student_ID)
);

CREATE TABLE Interviews (
  Interview_ID INT PRIMARY KEY,
  Application_ID INT NOT NULL,
  Interview_Date DATE NOT NULL,
  Time TIME NOT NULL,
  Result VARCHAR(20) NOT NULL CHECK(Result IN ('pass', 'fail')),
  FOREIGN KEY (Application_ID) REFERENCES Job_Applications(Application_ID)
);

CREATE TABLE Email_Notifications (
  Notification_ID INT PRIMARY KEY,
  Application_ID INT NOT NULL,
  Notification_Type VARCHAR(20) NOT NULL CHECK(Notification_Type IN ('application_received', 'interview_scheduled')),
  Notification_Date DATE NOT NULL,
  Content TEXT NOT NULL,
  FOREIGN KEY (Application_ID) REFERENCES Job_Applications(Application_ID)
);
```

4.2 Implementation Approach

- Define project scope, requirements, and plan.
- Implement user registration, job posting, and notification functionality using React, and Material-UI.
- Implement user authentication, API endpoints, and data storage using Node.js and Express.js.
- Set up database using MongoDB and implement schema and data modeling.
- Integrate frontend and backend, conduct testing, and ensure security using Jest, Enzyme, and Cypress.
- Deploy 5 application to production environment using Docker and Kubernetes.
- Assemble a team of frontend, backend, database, QA, and project management professionals.
- Utilize tools and technologies such as React, Node.js, MongoDB, and Docker.

4.2.1 Introduction to Languages, IDEs Tools and Technologies

4.2.1.1 Languages:

- JavaScript : Utilized for both front-end and back-end development. JavaScript played a crucial role in creating dynamic user interactions and implementing server-side logic.
- HTML/CSS : Used for structuring and styling web pages. HTML provided the semantic structure, while CSS ensured a visually appealing and responsive design across devices.

4.2.1.2 Frameworks and Libraries:

- React.js : Employed for front-end development, React.js provided a component-based architecture, enabling modular and maintainable UI design.
- Node.js with Express.js : Used for back-end development, Node.js ensured high performance for server-side operations, while Express.js simplified API creation and request handling.

4.2.1.3 Database:

- Firebase : Chosen for its real-time database capabilities, Firebase facilitated efficient data synchronization, secure storage, and user authentication, ensuring a smooth experience for large-scale user interactions.

4.2.1.4 Version Control:

- Git : Used for source code management, Git streamlined team collaboration, enabling effective tracking of changes and ensuring seamless integration across development stages.

4.2.1.5 IDE Tools and Technologies :

- Visual Studio Code : The primary integrated development environment (IDE) for writing and debugging code, ensuring developer productivity.
- Postman : Utilized for testing APIs, Postman enabled validation and debugging of server-side logic during development.

4.3 Testing Approaches

Testing is a crucial part of the Opportunifi-where talent meets technology application to ensure that the application functions as expected, with minimal errors and optimized performance. Both Unit Testing and Integration Testing were conducted to validate individual components and their interaction within the system. Below are the approaches used for these testing phases:

4.3.1 Unit Testing

4.3.1.1 Test Job Posting

- Test that a coordinator can post a job successfully.
- Test that a coordinator cannot post a job with invalid details (e.g., missing fields, invalid dates).

4.3.1.2 Test Application Tracking

- Test that a coordinator can view and track student applications successfully.
- Test that a coordinator cannot view or track applications for a job that is not posted.

4.3.1.3 Test Communication with Companies

- Test that a coordinator can send emails to companies successfully.
- Test that a coordinator cannot send emails to companies with invalid email addresses.

4.3.1.4 Test New Job Opportunity Email

- Test that an email is sent to students when a new job is posted.
- Test that the email contains the correct job details.

4.3.1.5 Test Application Tracking

- Test that a coordinator can view and track student applications successfully.
- Test that a coordinator cannot view or track applications for a job that is not posted.

4.3.1.6 Test Communication with Companies

- Test that a coordinator can send emails to companies successfully.
- Test that a coordinator cannot send emails to companies with invalid email addresses.

4.3.1.7 Test New Job Opportunity Email

- Test that an email is sent to students when a new job is posted.
- Test that the email contains the correct job details.

4.3.1.8 Test Interview Invitation Email

- Test that an email is sent to students when they are invited for an interview.
- Test that the email contains the correct interview details.

4.3.1.9 Test Status Update Email

- Test that an email is sent to students when their application status is updated.
- Test that the email contains the correct status update.

4.3.1.10 Test that students receive real-time updates on placement activities.

- Test that students can view the latest updates on their dashboard.

4.3.1.10 Test Coordinator Updates

- Test that coordinators receive real-time updates on placement activities.
- Test that coordinators can view the latest updates on their dashboard.

a. Test Cases

TC-001: Successful Login

- Preconditions: Valid username and password
- Steps: Enter credentials, click login
- Expected Results: User logged in, redirected to dashboard

TC-002: Failed Login

- Preconditions: Invalid username and password
- Steps: Enter credentials, click login
- Expected Results: Error message displayed, user not logged in

TC-003: Job Posting Creation

- Preconditions: User logged in, necessary permissions
- Steps: Enter job posting details, click create
- Expected Results: Job posting created, displayed on job board

TC-004: Job Application Submission

- Preconditions: User logged in, job posting to apply to
- Steps: Enter job application details, click submit
- Expected Results: Job application submitted, displayed on job application page

4.3.2 Integration Testing

4.3.2.1 User Authentication and Authorization: Test the integration of user authentication and authorization with the rest of the application, verifying successful login, access control, and user profile management.

4.3.2.2 Job Posting and Application Management: Test the integration of job posting and application management, verifying the creation of job postings, application submission, and job application status updates.

4.3.2.3 Automated Email Notification: Test the integration of automated email notification, verifying the sending of email notifications for new job postings, job application submissions, and job application status updates.

4.3.2.4 Real-Time Update: Test the integration of real-time update, verifying the updating of the dashboard in real-time when job postings or applications are updated.

4.3.2.5 Database and Server: Test the integration of the database and server, verifying the storage and retrieval of data and the server's response to requests.

4.3.2.6 Error Handling: Test the integration of error handling, verifying the handling of server errors, database errors, and network errors.

b. Test Cases

➢ User Authentication and Authorization

- Test login with valid credentials.
- Test login with invalid credentials**.**
- Test access to protected pages with authorized user.

➢ Job Posting and Application Management

- Test creation of job posting with valid data.
- Test creation of job posting with invalid data.
- Test application submission with valid data.
- Test application submission with invalid data.

➢ Automated Email Notification

- Test sending of email notification for new job posting.
- Test sending of email notification for job application submission.

➢ Real-Time Update

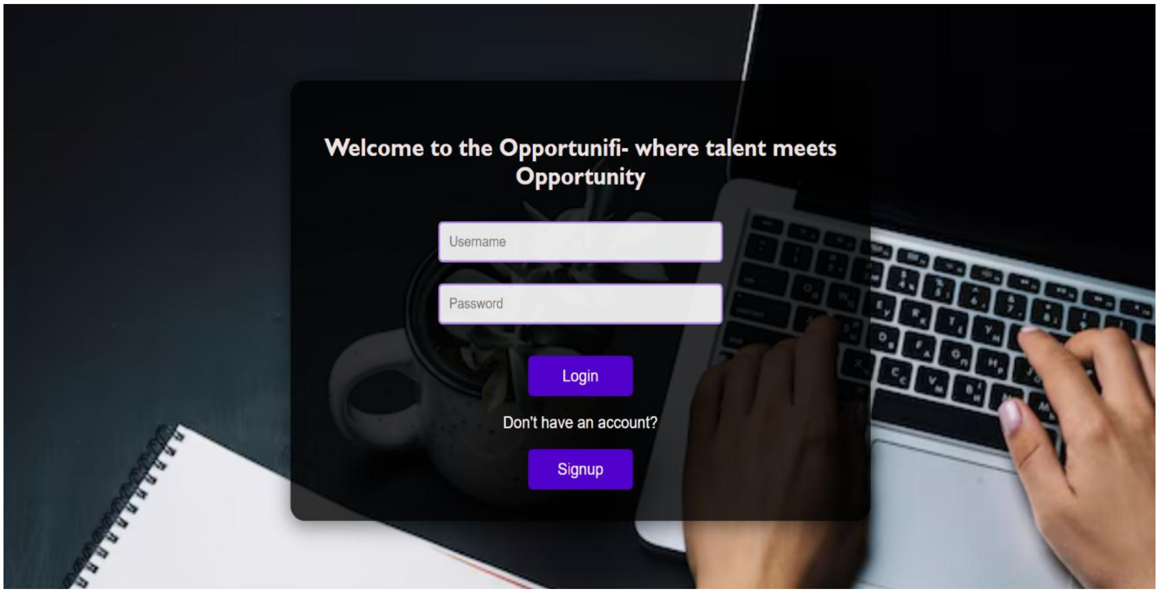
- Test updating of dashboard in real-time when job posting is updated
- Test updating of dashboard in real-time when job application is updated.

- Database and Server

- Test storage of data in database.
- Test retrieval of data from database.
- Test server response to requests.

Chapter 5: Results & Discussion

5.1 User Interface Representation



Welcome to the Opportunifi- where talent meets Opportunity

5.1.1 Brief Description of Various Modules

5.1.1.1 User Management Module

- Purpose: Manage user accounts for students and coordinators.
- Features: Registration, profile management, and role-based access.

5.1.1.2 Student Profile Module

- Purpose: Maintain comprehensive student profiles.
- Features: Profile creation, resume upload, and application tracking.

5.1.1.3 Job Postings Module

- Purpose: Facilitate job opportunity postings.
- Features: Job creation, searchable listings, and application management.

5.1.1.4 Application Management Module

- Purpose: Manage student job applications.
- Features: Application submission, status updates, and feedback mechanism.

5.1.1.5 Notification Module

- Purpose: Keep users informed about updates.
- Features: Automated notifications, custom messages, and notification history.

5.1.1.6 Reporting and Analytics Module

- Purpose: Provide insights into placement activities.
- Features: Dashboard, custom reports, and data export options.

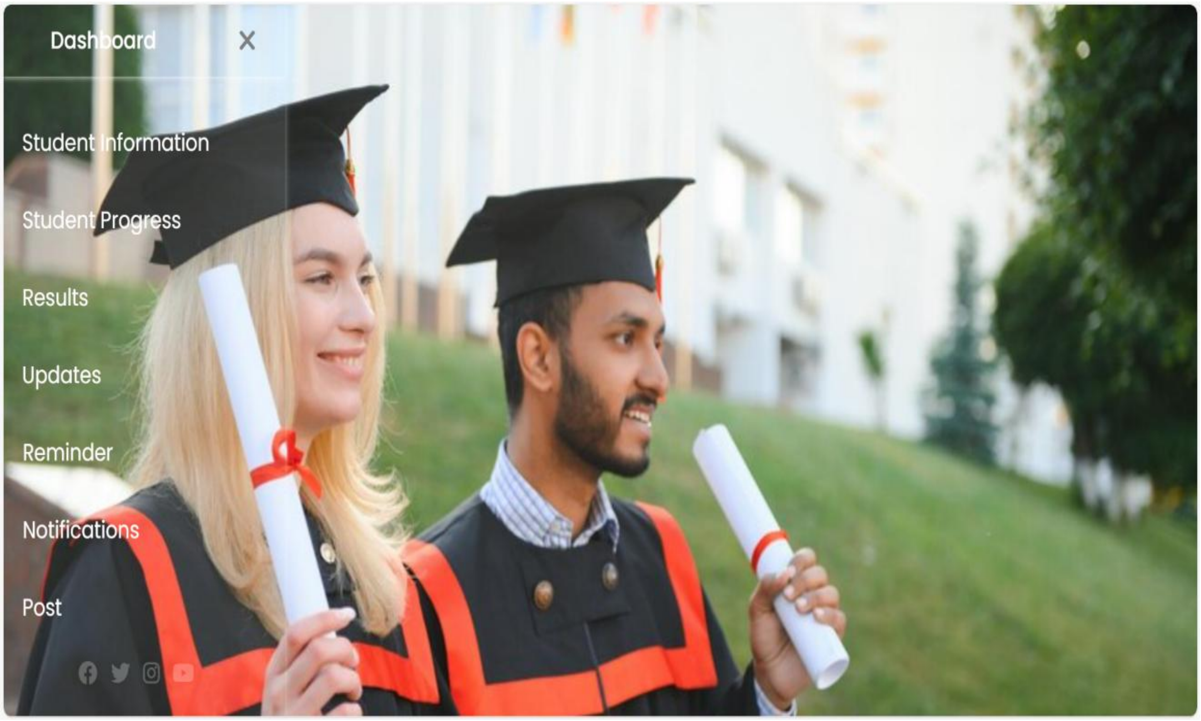
5.1.1.7 . Feedback and Evaluation Module

- Purpose: Gather feedback on the placement process.
- Features: Post-interview feedback, coordinator evaluations, and surveys.

5.1.1.8 Help and Support Module

- Purpose: Assist users with inquiries and issues.
- Features: FAQs, contact support, and user guides.

5.1.2 Snapshot of System with Brief Description



The Automated Placement Portal is a centralized platform designed to enhance communication and streamline the placement process for students and coordinators.

5.1.2.1 Student Profiles:

- Students can create profiles, upload resumes, and apply for jobs, showcasing their qualifications effectively.

5.1.2.2 Coordinator Dashboard:

- Coordinators manage job postings, track applications, and communicate with recruiters from a single interface.

5.1.2.3 Automated Email Notifications:

- The system sends automatic emails to students about new job opportunities, interview invitations, and application status updates, ensuring timely communication.

5.1.2.4 Real-Time Updates:

- Both students and coordinators receive instant updates on placement activities, keeping everyone informed and engaged.

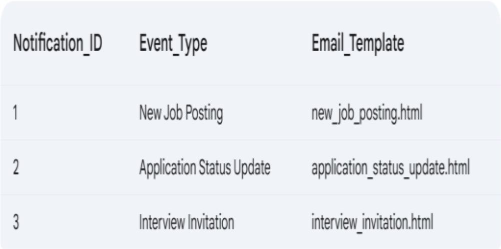
5.3 Database Description

The Automated placement portal utilizes a relational database management system (RDBMS) to store and manage data. The database schema consists of several tables, including:

- Student Profiles: stores student information, resume, and job preferences
- Job Postings: stores job details, company information, and application deadlines
- Applications: tracks student applications, including status updates and interview schedules
- Coordinator Dashboard: stores placement coordinator information and job posting management data
- Email Notifications: stores email templates and notification settings for various events
- Real-Time Updates: stores updates on placement activities, including job openings, interview schedules, and application status.

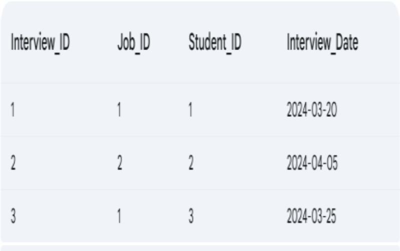
5.3.1 Snapshot of Database Tables with Brief Description

Email_Notifications Table:



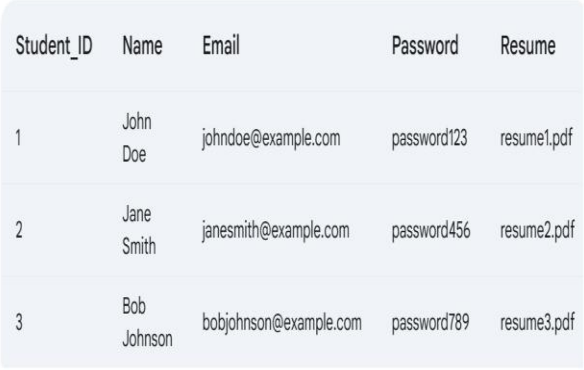
Notification_ID	Event_Type	Email_Template
1	New Job Posting	new_job_posting.html
2	Application Status Update	application_status_update.html
3	Interview Invitation	interview_invitation.html

Interview_Schedules Table:



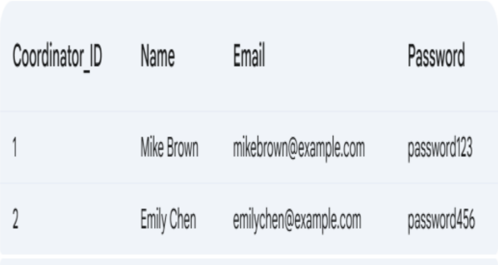
Interview_ID	Job_ID	Student_ID	Interview_Date
1	1	1	2024-03-20
2	2	2	2024-04-05
3	1	3	2024-03-25

Students Table:



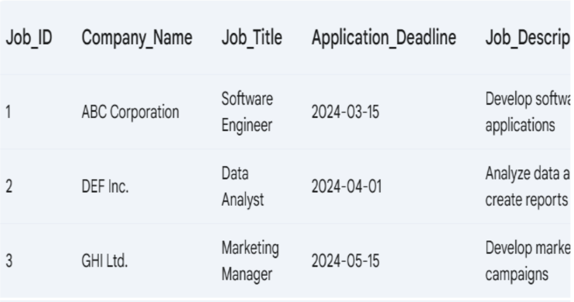
Student_ID	Name	Email	Password	Resume
1	John Doe	johndoe@example.com	password123	resume1.pdf
2	Jane Smith	janesmith@example.com	password456	resume2.pdf
3	Bob Johnson	bobjohnson@example.com	password789	resume3.pdf

Coordinators Table:



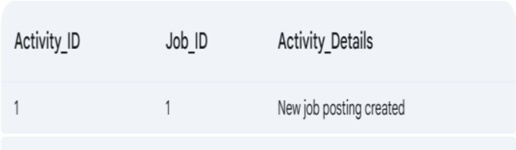
Coordinator_ID	Name	Email	Password
1	Mike Brown	mikebrown@example.com	password123
2	Emily Chen	emilychen@example.com	password456

Job_Postings Table:



Job_ID	Company_Name	Job_Title	Application_Deadline	Job_Descrip
1	ABC Corporation	Software Engineer	2024-03-15	Develop softwa applications
2	DEF Inc.	Data Analyst	2024-04-01	Analyze data a create reports
3	GHI Ltd.	Marketing Manager	2024-05-15	Develop marke campaigns

Placement_Activities Table:



Activity_ID	Job_ID	Activity_Details
1	1	New job posting created

Fig Database Tables and Description

The Automated Placement portal is designed to streamline placement activities for students and coordinators. It consists of five main tables:

5.3.1.1 Students: Stores student profiles, including names, contact info, and resumes for job applications.

5.3.1.2 Coordinators: Contains details of placement coordinators, facilitating communication with students and recruiters.

5.3.1.3 Job Postings: Holds job opportunity details, including titles, descriptions, and the posting coordinator.

5.3.1.4 Placement Activities: Tracks student applications and their statuses (e.g., Pending, Shortlisted).

5.3.1.5 Notifications: Manages automated email alerts for students about job openings and application updates.

5.3.1.6 Interview Schedule: Stores interview, job_ID, Student_ID and Interview Date

5.4 Final Findings

5.4.1 Enhanced Efficiency: The centralized database streamlines management of profiles, job postings, and applications, saving time for coordinators.

5.4.2 Improved Communication: Automated notifications keep students updated on job opportunities and application statuses, boosting engagement.

5.4.3 User Satisfaction: Positive feedback highlights the portal's user-friendly interface, contributing to a satisfying experience for both students and coordinators.

5.4.4 Data Accuracy: The structured database format enhances data integrity, ensuring reliable information through regular updates.

5.4.5 Increased Job Applications: The portal has led to more job applications from students, thanks to easy browsing and application processes.

5.4.6 Scalability: The database can grow with future needs, allowing for new features like analytics and reporting tools.

5.4.7 Areas for Improvement: Opportunities exist to add features like post-interview feedback and employer profiles for better matching.

CHAPTER 6. CONCLUSION & FUTURE SCOPE

6.1 Conclusion

The platform brings all placement activities into one place, making the process smoother and less time-consuming for everyone involved.Automated email notifications keep students informed about job openings, interviews, and results, ensuring they never miss important updates.Its simple and user-friendly design makes it easy for both students and coordinators to navigate and use effectively.The cloud-based setup ensures the platform can handle many users at once without any interruptions.Feedback from students and coordinators shows that the system has made the placement process more organized and stress-free.

6.2 Future Scope

6.2.1 Integration with Other University Systems: Integrate with student information systems and alumni networks.

6.2.2 Mobile App Development: Develop a mobile app for on-the-go access.

6.2.3 Advanced Analytics and Reporting: Implement analytics and reporting tools.

6.2.4 Artificial Intelligence and Machine Learning: Incorporate AI and ML for enhanced matching and recommendations.

6.2.5 Expansion to Other Institutions: Expand to other institutions for a broader network.

6.2.6 Enhanced Security and Compliance: Implement additional security measures and ensure regulatory compliance.

6.2.7 Continuous Improvement: Regularly update and refine the portal.

REFRENCES

1. A. Kumar et al., "Design and Development of a Web-Based Placement Portal," IJSDR, vol. 28, no. 4, pp. 89-100, 2023.

2. S. Patel, "Student-Placement Portal System," Educational Technology Conference Proceedings, 2022.

3. L. Davis and M. Brown, "Secure and Scalable Web Applications with Node.js," Journal of Web Development, vol. 18, no. 2, 2021.

4. J. Lee and H. Kim, "Job Recommendation Systems for Placement Portals," AI and Education Journal, vol. 12, no. 3, pp. 155-165, 2022.

5. A. Zhao, "Real-Time Notifications in Placement Systems," Journal of Computer Science and Technology, vol. 24, no. 5, 2020.

6. P. Wilson et al., "User-Centered Design for Placement Portals," HCI Conference Proceedings, 2019.

7. M. Thomas and K. Johnson, "Cloud-Based Deployment for Placement Portals," Cloud Computing Journal, vol. 10, no. 1, 2022.

8. S. Gupta and R. Sinha, "AI and Machine Learning in Placement Systems," Journal of AI, vol. 21, pp. 140-150, 2023.

AppendixA:https://github.com/Mauli178/opportunifi/blob/main/Synopsis_Opportunifi.pdf

AppendixB:https://github.com/Mauli178/opportunifi/blob/main/Logbook_Opportunifi.pdf

Appendix C: https://github.com/Mauli178/opportunifi/blob/main/User%20Manual.txt

Appendix D: Git/GitHub Commits/Version History

- **GitHub Repository:** https://github.com/Mauli178/opportunifi
- **Commits History:** https://github.com/Mauli178/opportunifi/commits/main/