A

Project Report

On

Campus Placement Portal

Submitted By

THANUSREE NAGALADINNE (R200958)

SAKE ARCHANA (R200999)

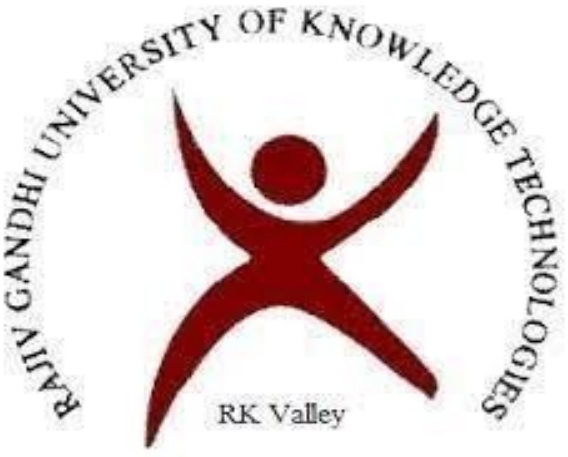
Under the guidance of

Mrs. GANGADHARI SWAPNA

Assistant Professor , MTech

Department of Computer Science and Engineering,

IIIT RK Valley, RGUKT-AP



Rajiv Gandhi University of Knowledge Technologies

(Catering the Educational Needs of Gifted Rural Youth of A.P)

R.K Valley, Y.S.R Kadapa (Dist.)-516330, Andhra Pradesh.

CERTIFICATE

This is to certify that the report entitled "**Campus Placement Portal**" submitted by **THANUSREE NAGALADINNE** bearing ID **R200958** ,**SAKE ARCHANA** bearing ID **R200999** in partial fulfilment of the requirements for the award of **(B.TECH) Bachelor of Technology in Computer Science and Engineering (CSE)** is a bonafide work carried out by her during the academic session 2024 - 2025 at **RGUKT- RK VALLEY** under my guidance and supervision.This report has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

PROJECT INTERNAL GUIDE

Mrs. Gangadhari Swapna

Assistant professor, MTech

Department of CSE

HEAD OF THE DEPARTMENT

Dr.CH.Ratna Kumari,

P.hD, Mtech, Assistant Proffessor

HOD, Department of CSE

Signature of External Examiner

DECLARATION

I, **THANUSREE NAGALADINNE** and **SAKE ARCHANA**, here by declare that the project report entitled " **CAMPUS PLACEMENT PORTAL**" done under the Guidance of **Mrs. GANGADHARI SWAPNA** madam is submitted in partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering during the session 2024 – 2025 at **RGUKT RKValley.** I also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. To the best of my knowledge, the results embodied this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

With Sincere Regards,

THANUSREE NAGALADINNE (R200958)

SAKE ARCHANA (R200999)

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude & respect to all those people behind the screen who guided, inspired and helped me crown all my efforts with Success.

I wish to express my gratitude to **Mrs. Gangadhari Swapna** madam advice,constructive suggestions, supportive attitude and continuous encouragement, without which it would not be possible to complete this project.

I would also like to extend our deepest gratitude & reverence the Director RGUKT, RK Valley **Prof . Kumara Swamy Gupta** sir and HOD of Computer Science and Engineering **Dr. CH.Ratna Kumari** for their constant encouragement.

Last but not least I express my gratitude to my parents for their constant source encouragement and inspiration for me to keep my morals high.

TABLE OF CONTENTS

ABSTRACT

The Campus Placement Portal is a web-based application designed to streamline and digitize the entire campus recruitment process for students, companies, and administrators. This project aims to simplify job application procedures, manage student and company data efficiently, and provide a centralized platform for all placement-related activities.

The system allows students to register, log in securely, view company listings, and apply for jobs relevant to their qualifications. Each student has a personalized dashboard to track applications, update profiles, and view academic statistics through graphical representations. Companies can post job openings, set eligibility criteria, and view applicant details. The admin panel provides tools to manage users, companies, and monitor placement activities.

The platform is developed using the MERN stack (MongoDB, Express.js, React, Node.js), ensuring a scalable and dynamic user experience. Security measures such as authentication and role-based access are implemented to ensure data integrity. Additionally, the system includes features like resume upload, email notifications for interview calls, and a statistics page to visualize student progress.

This project serves as a comprehensive solution to automate and manage the placement process effectively, reducing manual work, improving communication, and enhancing the overall experience for all stakeholders involved.

MODULE 1

INTRODUCTION

1.1 Problem Statement

In most educational institutions, the campus placement process is a critical phase that determines students' transition from academics to the professional world. However, the traditional methods used for managing campus placements are often outdated, relying heavily on manual records, physical notices, and inefficient communication channels. These methods lead to various challenges such as difficulty in tracking student eligibility, inconsistent updates on job postings, scheduling conflicts during interviews, and a lack of real-time communication between placement officers, students, and recruiters.

Moreover, managing a large volume of student data, resume verification, application tracking, and result dissemination becomes cumbersome without a centralized system. These inefficiencies can cause delays, loss of opportunities, and dissatisfaction among stakeholders.

The Campus Placement Portal seeks to address these issues by developing a comprehensive web-based solution that automates and integrates all aspects of the campus recruitment process. The system will provide modules for:

- Student registration and profile management
- Job posting and application tracking
- Resume uploads and eligibility checks
- Interview scheduling and result updates
- Communication tools for announcements and reminders

By digitizing and automating the workflow, this system will enhance transparency, improve efficiency, and ensure a seamless experience for all parties involved — students, placement coordinators, and recruiters.

1.2 Purpose of the Project

The purpose of this project is to design and develop a Campus Placement Portal that automates and simplifies the placement process for educational institutions. The system serves as a centralized platform for students, companies, and the placement cell to interact effectively and manage all recruitment-related activities online. It aims to eliminate the need for traditional paperwork, reduce manual errors, and streamline the overall recruitment process by enabling digital tracking of job applications, student profiles, company job postings, and interview schedules.

This system empowers students to explore job opportunities, apply for jobs based on eligibility, and track their placement status. It also provides the placement cell with administrative tools to manage data efficiently and helps companies to shortlist candidates and conduct recruitment drives with ease.

1.3 Motivation

In most colleges and universities, the placement process is still managed manually using spreadsheets, emails, or paperwork. This traditional approach often results in data mismanagement, loss of important information, time delays, and lack of transparency for students. The idea for this project was motivated by the need for a more efficient and transparent system that enhances communication between students, placement coordinators, and recruiters.

With the increasing use of technology in education and administration, a web-based placement portal can significantly improve user experience, data security, and time management. This project not only fulfills academic requirements but also provides a real-world solution that can be used by institutions to modernize their placement process.

1.4 Overview

Campus placement portal are platforms used by educational institutions to connect students with potential employers. These systems facilitate various tasks such as collecting student data, publishing job openings, managing company profiles, tracking applications, shortlisting candidates, and sending interview notifications.

Traditional campus placement systems are either fully manual or semi-digital with limited features. Modern systems aim to automate the end-to-end recruitment process with added features such as dashboards, analytics, eligibility filters, automated emails, and resume storage. This project seeks to bridge the gap between traditional and advanced systems by developing a user-friendly, fully functional web-based portal using modern technologies like MongoDB, Express.js, React, and Node.js (MERN Stack).

1.5 Scope of the System

The Campus Placement Portal is designed to cater to the needs of students, placement officers, and recruiters. The scope includes:

- ✔ Student Functionality:

- ✔ Login/Register securely

- ✔ Update personal and academic profiles
- ✔ Upload resumes
- ✔ View available job listings and company details
- ✔ Apply for eligible jobs
- ✔ Track application status
- ✔ View SGPA graphs and academic trends
- ✔ Admin Functionality:
- ✔ Manage student records
- ✔ Add/update/delete job postings
- ✔ View applications for each job
- ✔ Send interview calls and notifications via email
- ✔ View analytics and system stats
- ✔ Company Functionality:
- ✔ Post new job openings
- ✔ View list of applicants
- ✔ Download resumes
- ✔ Filter candidates based on criteria

The system is scalable and can be extended with more features such as interview scheduling, resume builders, real-time notifications, and analytics dashboards.

MODULE 2

LITERATURE SURVEY / EXISTING SYSTEM

2.1. Research on Existing Systems

Several institutions have implemented digital or semi-digital placement systems aimed at streamlining the campus recruitment process. These systems often include basic functionalities such as student registration, job postings, and email notifications. Some reputed platforms used by top universities include:

- **Superset** – A cloud-based platform that automates placements and internships in campuses.
- **T&P Portals (custom-built)** – Many colleges have their in-house Training & Placement portals for internal use.
- **ERP Modules** – Some educational ERPs have modules for placement tracking.

While these systems serve their purpose to an extent, many are either expensive, not customizable, or lack user-friendly interfaces. Moreover, they may not scale well for mid-sized institutions or lack real-time communication tools.

2.2. Traditional Methods

In colleges still using traditional methods, several issues are commonly observed:

- **Inefficient Communication:** Announcements about company visits or criteria are shared via emails or notice boards, leading to delayed or missed updates.
- **Manual Eligibility Checks:** Verifying student eligibility (CGPA, branch, etc.) for each drive is time-consuming and error-prone.
- **Cumbersome Resume Collection:** Collecting and sorting resumes manually increases the workload for placement officers.
- **Lack of Centralized Data:** Interview results, feedback, and application status are scattered and not easily traceable.

- **No Real-Time Updates:** Students remain unaware of real-time changes in interview schedules or job postings.

- **Limited Interaction:** No dedicated space for student queries, drive discussions, or chat support.

2.3. Advantages

The proposed Campus Placement Portal addresses these issues by offering:

- **Centralized Dashboard:** All placement activities accessible from a single portal for students, recruiters, and coordinators.

- **Automated Eligibility Checks:** Students are automatically filtered based on academic criteria defined by the recruiter.

- **Resume Upload and Sorting:** Students can upload resumes directly to their profiles; TPOs can sort them based on filters.

- **Real-Time Notifications:** Instant alerts for new jobs, interview updates, or results via email or in-app notifications.

- **User Roles and Access:** Customized views for students, TPOs, and recruiters with role-specific functionality.

- **Data Analytics:** Reports and statistics about placements, student performance, and recruiter visits.

- **Interactive Tools:** Chat or query modules for student support and recruiter communication.

This solution not only simplifies the process but also increases transparency, minimizes delays, and provides a professional and efficient approach to campus placements.

MODULE 3

SYSTEM REQUIREMENTS

The Campus Placement Portal is a web-based application that requires both client-side and server-side resources. The system is designed for students, companies, and administrators to interact seamlessly through a responsive interface. Below are the detailed system requirements needed to support development, deployment, and use.

3.1. Hardware Requirements

To ensure smooth performance and responsiveness of the application during development and execution, the following hardware specifications are recommended:

COMPONENT	SPECIFICATION
Processor	Intel Core i3 or above
RAM	Minimum 4 GB
Hard Disk	Minimum 100 GB free space
Monitor	Standard HD display
Internet Connection	Required for hosting and testing

- Processor: A multi-core processor (Intel Core i5 or higher) is essential for running development environments, servers, and handling backend processes.
- RAM: A minimum of 8 GB RAM is recommended for development tasks and running servers locally without lag.
- Storage: At least 250 GB of available disk space for storing code files, dependencies, databases, and user-uploaded content such as resumes.
- Display: A screen resolution of 1366x768 or higher is advised for optimal display of UI components.
- Internet Connectivity: Required for accessing cloud services, database APIs, and deploying the application.

3.2. Software Requirements

The system depends on several software components for both the frontend and backend. These include:

SOFTWARE COMPONENT	SPECIFICATION
Operating System	Windows 10 / Ubuntu 20.04 or later
Web Browser	Chrome / Firefox (latest version)
Code Editor	Visual Studio Code / Sublime Text
Backend Language	JavaScript (Node.js)
Frontend Language	HTML, CSS, JavaScript, React.js
Database	Mongo DB / MySQL
Server Platform	Node.js with Express.js framework
Version Control	Git & GitHub

- Operating System: Cross-platform development is supported (Windows 10/11, Ubuntu 20.04+, or macOS).
- Code Editor: Visual Studio Code is preferred due to its extensions for React and Node.js development.
- Browser: Modern browsers like Google Chrome or Mozilla Firefox are required for testing and usage.
- Database Server: MongoDB Community Server for local development; MongoDB Atlas for cloud-hosted databases.
- Runtime Environment: Node.js (v14 or later) to run the backend application and manage packages.
- Package Manager: npm (Node Package Manager) or Yarn for installing required dependencies.
- Version Control: Git is used for tracking code changes and collaborating via platforms like GitHub.

3.3. Special Tools and Libraries

The application utilizes several frameworks and libraries that facilitate rapid development and efficient user experience:

TOOL / LIBRARY	PURPOSE
Node.js	Server-side runtime environment
Express.js	Backend web application framework
Mongo DB	No SQL database for storing user data
Mongoose	OMD for Mongo DB
React.js	Frontend library for building UI
Bootstrap / Tailwind CSS	Styling and responsive design
JWT / Bcrypt.js	Authentication and password encryption
Postman	API testing tool

- Frontend Framework: React.js — for building a fast, dynamic, and responsive user interface.
- Backend Framework: Node.js with Express.js — for handling HTTP requests, APIs, and database communication.
- Database: MongoDB — a flexible NoSQL database for storing user data, job listings, applications, and statistics.
- Styling Tools: Tailwind CSS or Bootstrap — for creating modern, responsive designs.
- Email Integration: Nodemailer — used to send confirmation, selection, or rejection emails to students.
- Data Visualization: Recharts or Chart.js — to graphically represent SGPA statistics and placement data.

MODULE 4

SYSTEM DESIGN

The system design outlines the architecture, functional behavior, and flow of data in the Campus Placement Portal. It includes diagrams and explanations to visualize how users interact with the system and how various modules work together.

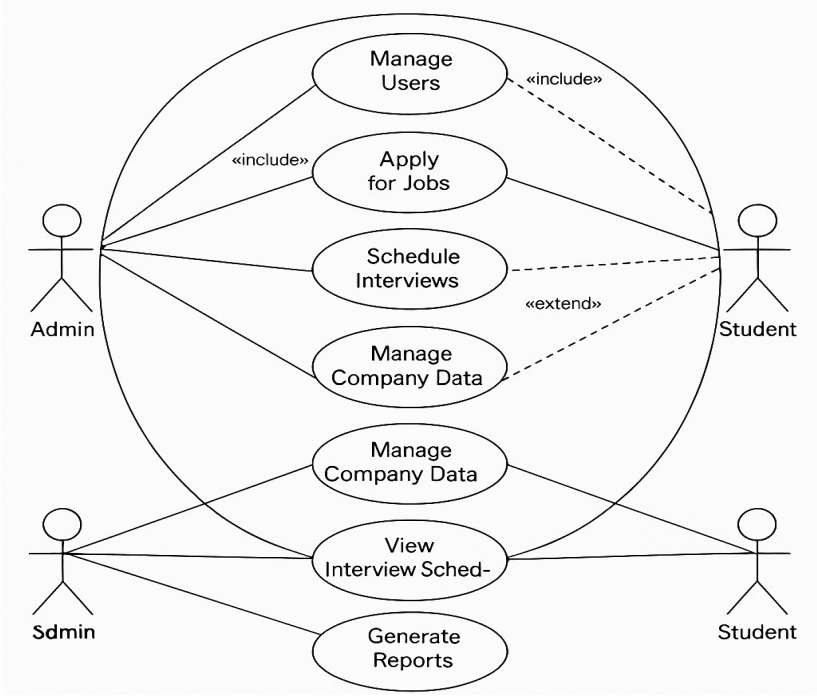
4.1. Use Case Diagram

The use case diagram represents the interactions between the system and its users (Actors). The main actors involved are:

- **Admin** – manages users, jobs, and company data,posts job openings, shortlists candidates.
- **Student** – registers, applies for jobs, views interview schedules.

Use Cases include:

- Register/Login
- Post/View Jobs
- Apply for Jobs
- Schedule Interviews
- Generate Reports



4.2. Architecture Diagram

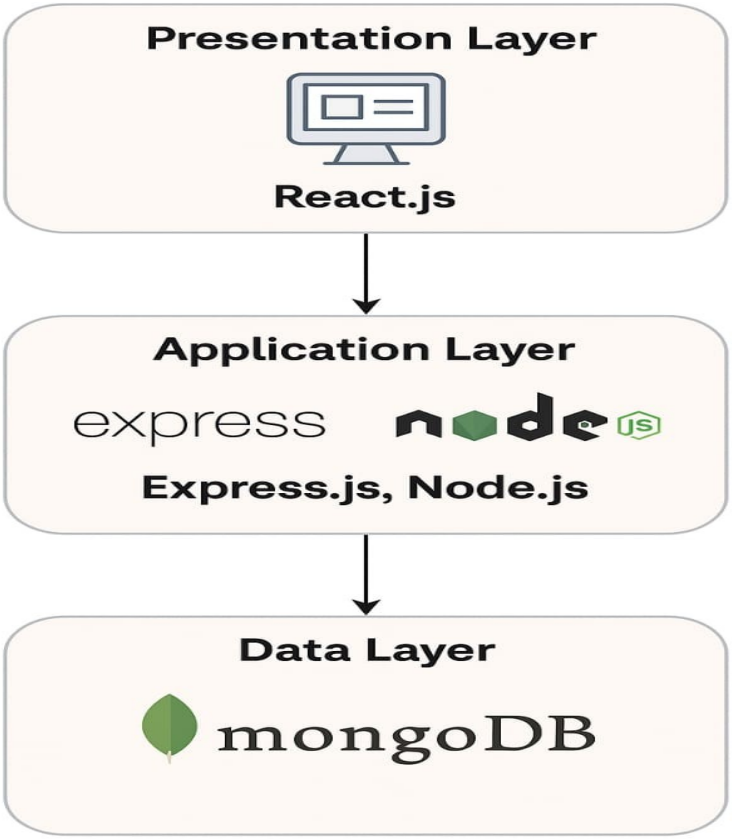
The system follows a 3-tier architecture:

1. Presentation Layer – The front-end UI developed using React.js for user interaction.

2. Application Layer – The server-side logic built using Node.js and Express.js.

3. Data Layer – The MongoDB database used for storing persistent data like user profiles, job postings, and applications.

This architecture ensures separation of concerns, scalability, and maintainability.

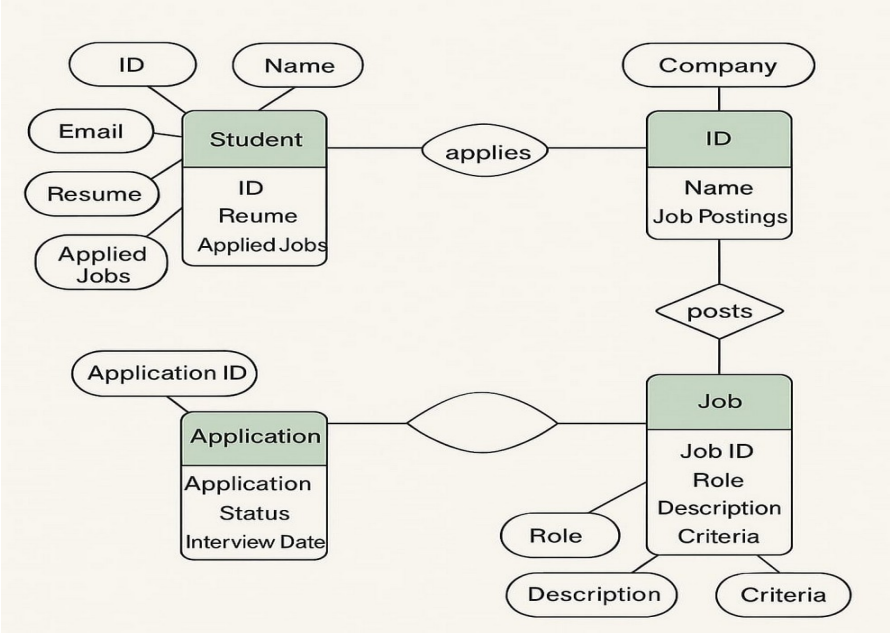


4.3. ER Diagram / Data Flow Diagram (DFD)

- ER Diagram describes the entities (e.g., Student, Company, Job, Application) and relationships among them.
- DFD shows the flow of information within the system, including input, processing, and output stages.

Entities and Attributes (for ERD):

- Student: ID, Name, Email, Resume, Applied Jobs
- Company: ID, Name, Job Postings
- Job: Job ID, Role, Description, Criteria
- Application: Application ID, Status, Interview Date



4.4. Brief Description of the Design Approach

The design of the Campus Placement Portal is modular and user-centric. It emphasizes:

- Scalability – Components are designed to handle increasing numbers of users and job posts.

- Security – User authentication and secure data handling are implemented using JWT and bcrypt.
- Responsiveness – The UI adapts to different devices using CSS frameworks.
- Efficiency – RESTful APIs ensure smooth communication between frontend and backend.

The design ensures a seamless experience for students, recruiters, and administrators, enabling efficient placement process management.

MODULE 5

IMPLEMENTATION

This section covers the technologies used and a module-wise breakdown of the system's functionalities. Each module is built with a focus on simplicity, efficiency, and user experience.

5.1 Technology Stack Used

- **Frontend:** React.js – for building a dynamic and responsive user interface.
- **Backend:** Node.js with Express.js – for handling server-side logic and APIs.
- **Database:** MongoDB – for storing data in a flexible, NoSQL format.
- **Authentication:** JWT (JSON Web Tokens) – for secure login sessions.
- **Styling:** CSS3 / Tailwind CSS – for styling and responsiveness.
- **Data Visualization:** Chart.js or D3.js – for visual analytics (SGPA trends, job stats).
- **File Handling:** Multer – for resume uploads and management.

5.2 Module-Wise Explanation

- **Student Module**

  Features:

  - User registration and login with secure authentication.
  - Dashboard showing available jobs based on eligibility.
  - Option to apply for jobs and view application status.
  - Resume upload and profile management.
  - View personal SGPA trends and qualification analytics.

- **Admin Module**

  Features:

  - Oversee all student and company registrations.
  - Approve/reject companies and validate job postings.
  - Manage placement timelines and notifications.
  - Generate analytical reports like job success rate, number of placed students, etc.
  - Secure login for recruiters/HR.
  - Post new job openings with eligibility criteria (branch, SGPA, etc.).
  - View list of student applicants per job.
  - Schedule interviews and update selection status.
  - Manage and edit existing job listings.

- **Job Application & Tracking**

  How it works:

  - Students can view job listings filtered by their eligibility.

- Apply with one click and track application status (Applied, Shortlisted, Interviewed, Selected).
- Notifications for updates and deadlines.
- Backend maintains all applications in MongoDB for quick querying.

➢ **Resume Upload**

Functionality:

- Resume upload in PDF format via Multer.
- Stored securely and linked to student profile.
- HRs can view/download resumes directly from the job application dashboard.
- Includes validation to check file size and format.

➢ **SGPA Stats Visualization**

Purpose:

- Helps both students and recruiters understand academic performance visually.
- Students can see their semester-wise SGPA in graph format.
- Admins can analyze performance patterns across departments
- SGPA data is entered or imported per student.
- Graphs are rendered using Chart.js.
- Options to view trends, averages, and comparisons across branches.

MODULE 6

SCREENSHOTS & OUTPUT

This section presents visual evidence of the functionality implemented in the Campus Placement Portal . Screenshots are included to demonstrate the user interface, features, and overall design of various modules of the application.

6.1 Login Page :

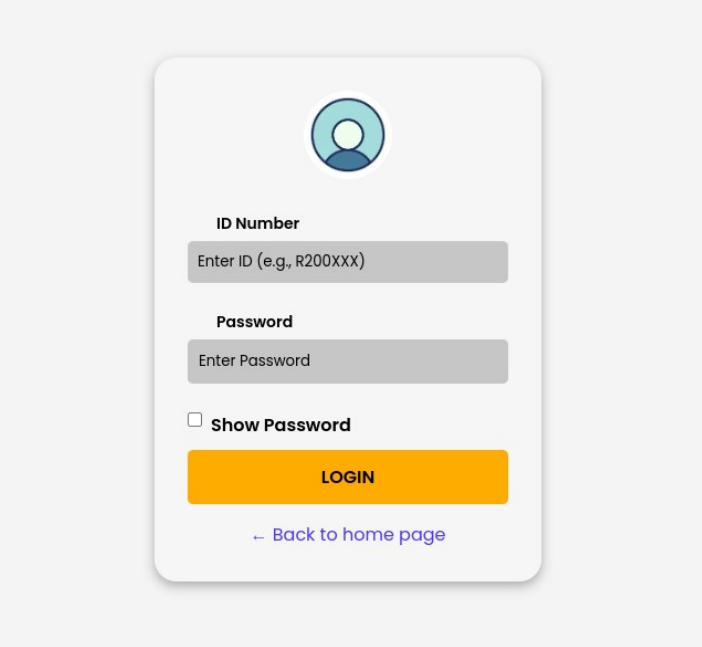


Fig 1.1: Login Page for Student

Description:

The login page allows access for three types of users: Students, Companies, and Admins. It includes fields for email and password, with proper error handling and validation messages.

6.2 Student Profile Page :

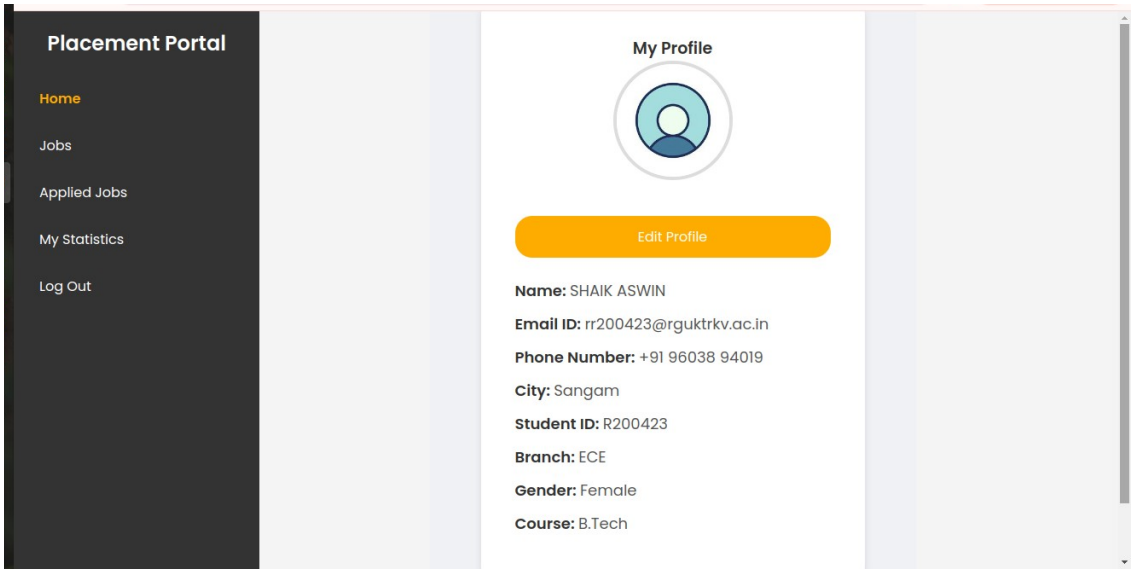


Fig 1.2: Student Profile Page

Description:

After logging in, students can access their personal dashboard. This page includes details such as name, registration number, branch, CGPA, and resume upload status.

6.3 Job Listings Page :

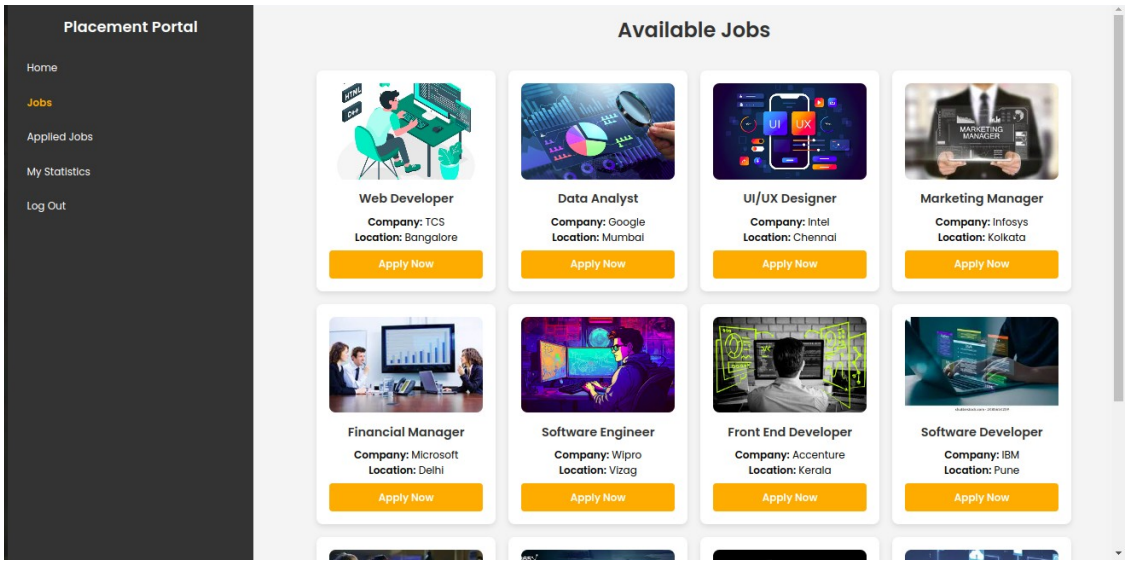


Fig 1.3: Job listing page

Description:

Displays active job opportunities with relevant filters. Each listing shows company name, position, eligibility criteria, and application deadline, with an option to apply directly.

6.4 Admin Panel :

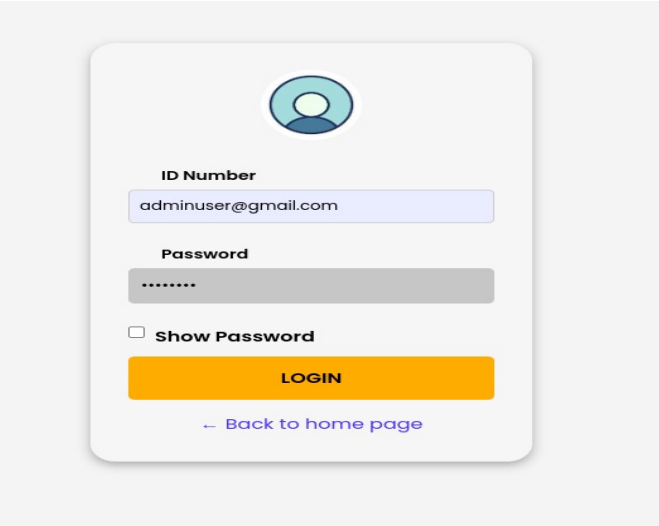


Fig 1.4: Admin Panel Login

Description:

The admin dashboard provides control over all system users and data. Admins can approve company registrations, manage job listings, and view system-wide statistics.

6.5 SGPA Stats Graph :

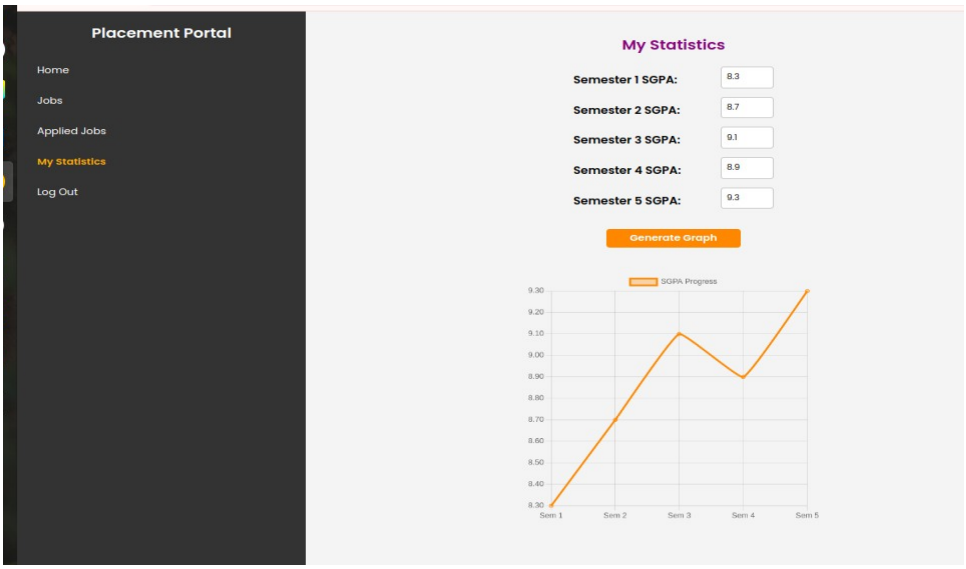


Fig 1.5: student graph

Description:

Visualizes a student's semester-wise SGPA using a graph. Helpful for quick academic performance tracking. Charts can be created using Chart.js or similar libraries.

6.6 Resume Upload Interface :

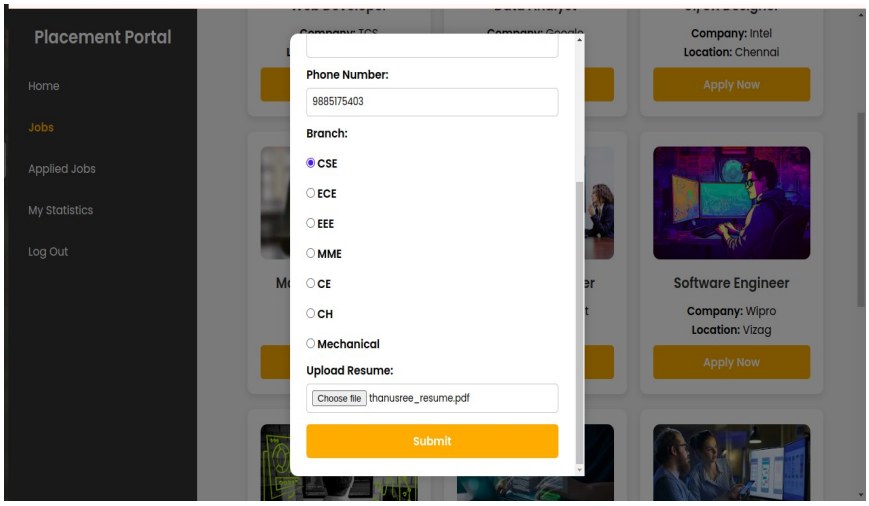


Fig 1.6: Resume upload

Description:

Enables students to upload their resumes in PDF format. The system validates file type and displays a preview link after upload.

6.7 Email Notification Interface :

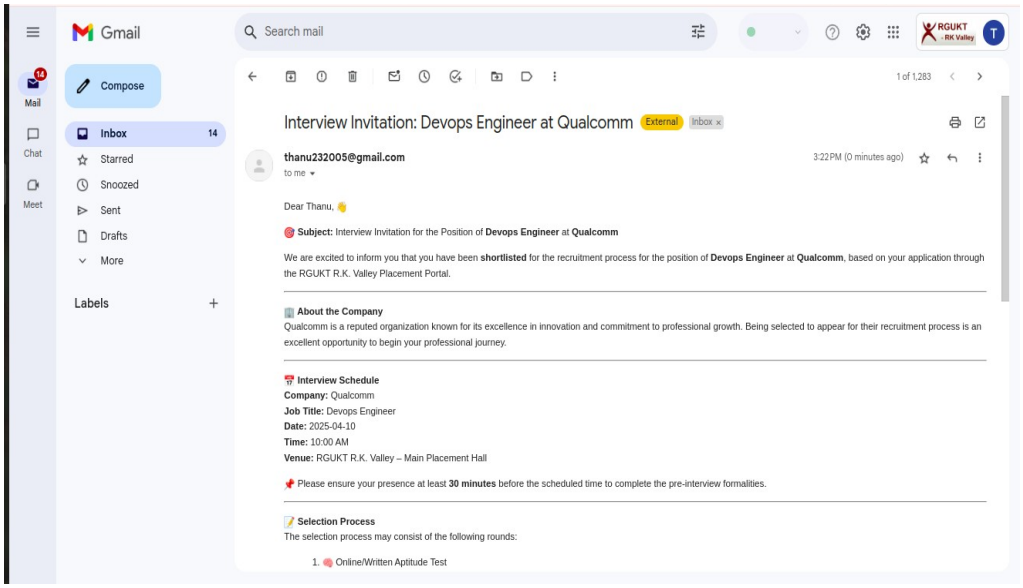


Fig 1.7: Email notification

Description:

The system sends automated emails for application status updates, interview invites, and account verification.

6.8 Database :

These screenshots show how data is structured and stored in the backend using MongoDB.

Student Collection: Contains user details like name, email, role.

- StudentId
- Password
- Name
- Branch
- Course
- Email
- Gender
- City
- Number

Applications Collection: Tracks which student applied to which job and current status.

- StudentId
- Name
- Email
- PhoneNumber

- JobTitle
- CompanyName
- Branch
- Resume
- Status

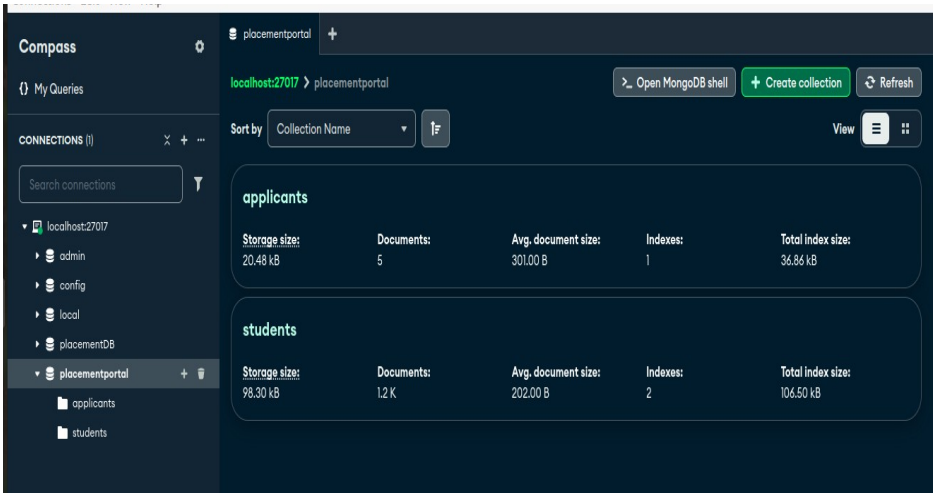


Fig 1.8: Database

MODULE 7

TESTING AND VALIDATION

7.1 Testing Approach

To ensure the reliability and performance of the Campus Placement Portal, a combination of manual and automated testing techniques was used. The system was tested module-wise to verify the functionality, integration, and overall flow. Focus was placed on usability, input validation, responsiveness, and compatibility across different devices and browsers.

- Unit Testing: Each individual component (e.g., login, job apply) was tested in isolation.
- Integration Testing: Interactions between modules like student-job application or admin-job posting were tested.
- System Testing: The entire system was tested end-to-end in a production-like environment.
- User Acceptance Testing (UAT): Conducted with a sample group of users to ensure it meets requirements.

7.2 Test Cases and Results

Test Case ID	Description	Input	Expected Result	Actual Result	Status
TC01	Student Login	Valid username/password	Redirect to dashboard	As Expected	Passed
TC02	Job Application	Click on "Apply" button	Show confirmation message	As Expected	Passed
TC03	Resume Upload	Upload PDF resume	Resume uploaded successfully	As Expected	Passed
TC04	Admin Add Job	Fill job form	Job listed on portal	As Expected	Passed
TC05	Invalid Login	Wrong credentials	Show error message	As Expected	Passed

7.3 Error Handling

Proper error handling has been implemented throughout the application to manage exceptions and invalid inputs gracefully. For instance:

- Invalid Form Input: Users are shown validation errors in real-time.
- Server Errors: Custom error pages (e.g., 404, 500) guide users back to the homepage.
- Upload Errors: If the file is not in the correct format, a user-friendly message is displayed.

Logging and debugging tools were used during development to capture and trace unexpected behaviors.

MODULE 8

FUTURE ENHANCEMENTS

To further improve the functionality and usability of the Campus Placement Portal, the following enhancements are proposed:

8.1. Advanced Admin Dashboard

- A more interactive and data-rich dashboard for the admin.
- Graphs and visual analytics showing student applications, placement rates, most active companies, and department-wise stats.
- Real-time updates and alerts on new job postings, application counts, and company responses.

8.2. Email Notifications

- Automatic email notifications to students and companies on key activities like job postings, application status updates, and interview schedules.
- Customizable templates for acceptance, rejection, shortlisting, and other alerts.
- Email reminders for upcoming deadlines or interview dates.

8.3. Resume Review and Management

- Allow admins and recruiters to accept or decline resumes based on predefined criteria.
- Resume viewer module to preview resumes directly within the dashboard without downloading.
- Feature to leave feedback or comments on resumes visible to students for improvement.

8.4. Resume Database and Search

A searchable resume database categorized by department, skills, or academic performance.

- Filters for recruiters to find candidates with specific qualifications quickly.

8.5. Interview Scheduling Tools

- Integration with calendar APIs to manage and schedule interviews efficiently.
- Notify both students and companies about scheduled slots via email and dashboard alerts.

8.6. Enhanced Role-Based Access

- More refined role management to give specific permissions to department heads, coordinators, and recruiters.
- Admin can monitor activities of all users for transparency and control.

MODULE 9

CONCLUSION

The **Campus Placement Portal** was developed to streamline and enhance the entire campus recruitment process for students, companies, and administrators. Through a centralized web portal, the system effectively bridges the communication gap between stakeholders, offering modules for student registration, job listings, resume submission, application tracking, and real-time notifications. The use of modern technologies like React.js, Node.js, and MongoDB ensures scalability, responsiveness, and a user-friendly experience.

This project has provided us with valuable hands-on experience in full-stack web development, database management, system design, and real-world problem solving. We gained practical insights into how enterprise systems are structured and how different modules interact within a single platform. Moreover, working on features like admin dashboards, SGPA-based analytics, and email automation has deepened our understanding of software integration and user-centric design.

Looking ahead, this system holds potential for further enhancement, such as integrating AI-based resume screening, chat-based support, and auto-scheduling interviews, making it a more robust and intelligent platform. This project has not only strengthened our technical skills but also our ability to collaborate and manage software development lifecycles efficiently.

MODULE 10

REFERENCES

10.1. References

1. MDN Web Docs – https://developer.mozilla.org

(For HTML, CSS, and JavaScript references)

2. React.js Documentation – https://reactjs.org/docs

(For component structure and state management)

3. Node.js Documentation – https://nodejs.org/en/docs

(Used for backend setup and API handling)

4. MongoDB Documentation – https://www.mongodb.com/docs

(Used for understanding NoSQL database operations)

5. W3Schools – https://www.w3schools.com

(For basic web development tutorials)

6. GeeksforGeeks – https://www.geeksforgeeks.org

(For system design ideas and algorithm concepts)

7. YouTube Tutorials –

- Traversy Media – MERN Stack Projects

8. Stack Overflow – https://stackoverflow.com

(For resolving bugs and clarifying implementation queries)