

SVISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JNANA SANGAMA”, BELAGAVI - 590 018



DBMS MINI PROJECT REPORT
on
“PLACEMENT MANAGEMENT SYSTEM”

Submitted by

JITHESH P SHETTY
DHEERAJ K B

4SF22CS085
4SF22CS059

In partial fulfillment of the requirements for the V semester

BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE & ENGINEERING

Under the Guidance of

Mrs. Ashwini CS

Department of CS&E

at



SAHYADRI

College of Engineering & Management
(An Autonomous Institution)

MANGALURU

2024 - 25

Abstract

The Placement Management System is a modern web application designed to digitize and streamline the campus placement process in academic institutions. Developed using the MERN stack (MongoDB, Express.js, React.js, Node.js), the system provides an efficient, user-friendly, and scalable solution for managing student placements. By eliminating manual procedures, the platform enhances transparency, real-time tracking, and operational accuracy.

This system caters primarily to students and placement officers, facilitating smooth coordination and management of placement drives. Core functionalities include CGPA-based eligibility verification, drive application tracking, and automated attendance management for placement events. The use of MongoDB allows for flexible and efficient data storage, while Express and Node.js handle robust server-side logic. The React-based frontend delivers an interactive and responsive user interface.

Key backend features include the use of Mongoose schemas, validation middleware, and custom APIs for seamless communication between components. The system also supports modular code design for maintainability and role-based access control to ensure secure interactions.

By leveraging full-stack JavaScript development, the Placement Management System reduces administrative overhead, improves the accuracy of placement records, and provides a centralized platform for placement-related activities. Its scalable architecture sets the foundation for future integrations such as AI-driven analytics, real-time notifications, and third-party recruitment platform connectivity, making it an essential tool for modern placement ecosystems.

Table of Contents

Abstract	i
Table of Contents	ii
1. Introduction	
2. Problem Formulation	
2.1. Problem Statement	
2.2. Objective	
3. Methodology	
3.1. Requirement analysis	
3.2. Development	
3.3. Role-based system	
3.4. Maintainance and Future Enhancements	
4. Results and Discussion	
4.1. Results	
5. Conclusion	

Chapter 1

Introduction

The Placement Management System is a comprehensive digital platform developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) to streamline the placement process in academic institutions. It is designed to address the increasing challenges faced by students and placement cells in coordinating campus recruitment activities. The system bridges the gap between key stakeholders by offering a centralized, user-friendly solution for managing placement operations with improved transparency, accountability, and efficiency.

The primary objective of the system is to automate and simplify the management of placement drives, student participation, and recruiter interactions. Institutions often manage large volumes of student data and multiple concurrent placement drives, each with unique eligibility criteria. Manual handling of these processes is time-consuming and error-prone. This system solves these challenges by automating critical workflows and offering role-based features tailored to students and placement officers.

For students, the platform offers a seamless experience to register, manage their profiles, and view available placement drives. They can verify their eligibility in real time based on CGPA and department criteria, apply for drives, and track their application status. The system ensures students stay informed about upcoming opportunities through a centralized dashboard, eliminating the need for manual updates or paper-based notices.

Placement officers benefit from an advanced dashboard that enables them to create and manage placement drives, configure eligibility criteria, add company details, and oversee student participation. These tools help reduce administrative workload and support data-driven decision-making.

From a technological standpoint, the system uses MongoDB as the database for efficient and scalable data storage. Mongoose serves as the ODM layer, ensuring structured access and validation of data. The Node.js + Express.js backend handles business logic and API endpoints, while the React.js frontend, styled with Material-UI, provides a responsive and modern interface. JWT is used for secure role-based authentication, and bcrypt ensures encrypted password storage. The application is modularly built, allowing easy maintenance, testing, and future scalability.

By replacing traditional methods with an automated web-based system, this project enhances the placement process's accuracy, speed, and reliability. With planned enhancements such as email notifications, analytics dashboards, resume parsing, and mobile app support, the Placement Management System lays a strong foundation for evolving placement needs in modern academic institutions.

Key features of the Placement Management System include:

1. **Role-Based Access Control:** Distinct functionalities for students and placement officers ensure secure and relevant access.
2. **Placement Drive Management:** Placement officers can create, manage, and monitor multiple placement drives seamlessly.
3. **Eligibility Verification:** Automated CGPA-based eligibility checks streamline student shortlisting.
4. **Real-Time Updates:** Instant updates keep students and placement officers informed about placement activities.
5. **Application Tracking:** Students can track their application status and view their placement history.

By automating the placement process, the system minimizes errors, reduces administrative workload, and promotes a more organized and transparent approach to campus recruitment. It ensures that all eligible students have equal access to placement opportunities, empowering them to prepare effectively for their professional careers.

The Placement Management System is built with a modular and scalable MERN stack architecture, making it adaptable for institutions of all sizes. It lays a strong foundation for future enhancements such as machine learning-based predictive analytics, integration with external recruitment platforms, and advanced performance metrics for students and drives. Overall, this system serves as a vital tool for modern academic institutions aiming to upgrade their placement operations and support students in navigating an increasingly competitive job market.

Chapter 2

Problem Formulation

2.1 Problem Statement

Managing campus placements manually in educational institutions often results in inefficiencies, errors, and communication gaps. The lack of a centralized digital system poses challenges in organizing and maintaining large volumes of data, such as student profiles, placement drive details, and eligibility criteria. Core processes like verifying student eligibility, tracking applications, and monitoring participation become time-consuming and error-prone without automation. Furthermore, poor communication between students and placement officers can lead to missed opportunities and lack of transparency. These challenges highlight the need for a robust **Placement Management System** that automates critical tasks, enhances data management, improves communication, and delivers a streamlined and transparent placement experience for all users involved.

2.2 Objective

The objective of this project is to design and implement a robust Placement Management System using the MERN stack that automates and streamlines the campus placement workflow for educational institutions. The system aims to centralize data management, allowing placement officers to efficiently organize and manage placement drives, while enabling students to track their eligibility, applications, and status in real time. By leveraging modern technologies like MongoDB, Express.js, React.js, and Node.js, the system ensures enhanced operational efficiency, transparency, and secure communication. The ultimate goal is to deliver a scalable, secure, and user-friendly platform that simplifies placement operations and improves the overall experience for both students and placement administrators.

Chapter 3

Methodology

The Placement Management System employs a structured and systematic approach to design, development, and deployment, ensuring efficiency and reliability.

3.1 Requirement Analysis

The project starts with a thorough analysis of the requirements of key stakeholders, primarily students and placement officers. Essential functionalities identified include role-based access control to differentiate between these users and provide them with appropriate access and features. The system enables efficient management of placement drives, incorporating eligibility criteria and automating the verification of student eligibility and participation tracking. Real-time updates and notifications ensure that stakeholders stay informed throughout the placement process. To enhance the system's effectiveness, regular user feedback sessions are conducted to gather insights and refine the requirements. Additionally, a competitive analysis of existing placement management solutions is performed to identify industry best practices and incorporate unique, valuable features into the system.

3.2 Development

The development phase focuses on implementing the Placement Management System using the MERN stack. Node.js and Express.js are used for the backend server-side logic, MongoDB with Mongoose manages the database for flexible and scalable data storage, and React.js combined with Material-UI delivers a responsive and user-friendly frontend interface. The entire development is carried out following Agile methodologies, with Scrum practices enabling iterative progress and regular stakeholder feedback. To ensure code quality and maintainability, a rigorous code review process is established. Additionally, Git is employed as the version control system to facilitate effective collaboration and track changes throughout the development lifecycle.

3.3 Role-Based Functionalities

Each user role in the system has specific functionalities tailored to their needs. Students can manage their profiles, browse and apply for placement drives, verify their eligibility, and track their application status in real-time. Placement officers have comprehensive control to create and manage placement drives, set eligibility criteria, view eligible students, and monitor the progress of applications. Enhancements planned include customized dashboards for each user type to present relevant data and metrics clearly, along with advanced search and filtering capabilities to help users efficiently navigate placement drives and applications.

3.4 Maintenance and Future Enhancements

The system is architected for easy updates and scalability to accommodate growing user needs. Future enhancements include implementing email notifications to keep users informed about placement updates and developing advanced analytics dashboards to visualize placement statistics. Additionally, regular maintenance windows will be scheduled to apply updates and patches seamlessly. A continuous feedback loop with users will be established to gather suggestions for ongoing improvements. Integration of advanced data analytics will provide deeper insights into placement trends and overall performance, helping stakeholders make informed decisions.

This methodology guarantees that the Placement Management System remains reliable, efficient, and user-friendly while fulfilling its primary goal of automating and streamlining the placement process. It also provides flexibility to adapt to evolving requirements and supports future scalability and growth.

Chapter 4

Results and Discussion

The Placement Management System features a robust database management component that greatly improves the system's efficiency and overall functionality. This section highlights the project outcomes, emphasizing the database management aspects, and presents key findings and observations.

4.1 Results

1. Efficient Database Design and Management: The project began with designing a comprehensive database schema for the Placement Management System, focusing on collections for users (students and placement officers), placement drives, and departments. The database structure was carefully modeled using Mongoose schemas to ensure data consistency and minimize redundancy. Referential integrity was maintained through the use of object references between collections, enabling efficient queries and reliable relationships across the data.

2. Automated Eligibility Verification: A key feature of the system is the automated eligibility verification implemented in the backend. When students apply for placement drives, the system programmatically checks their CGPA and other eligibility criteria against the requirements of each placement drive. This ensures that only eligible students can submit applications, streamlining the application process and reducing manual oversight.

3. Role-Based Access Control: The system implements robust role-based access control to ensure that users can only access features and data relevant to their roles. This mechanism enhances security by restricting sensitive operations and information based on whether the user is a student or a placement officer, thereby maintaining data integrity and protecting user privacy.

4. Real-Time Data Updates: The system provides real-time updates and notifications to keep stakeholders promptly informed about placement activities. Efficient backend operations and frontend state management ensure that users receive timely information on placement drives, application statuses, and other relevant events. This feature enhances user engagement and delivers a seamless, interactive experience.

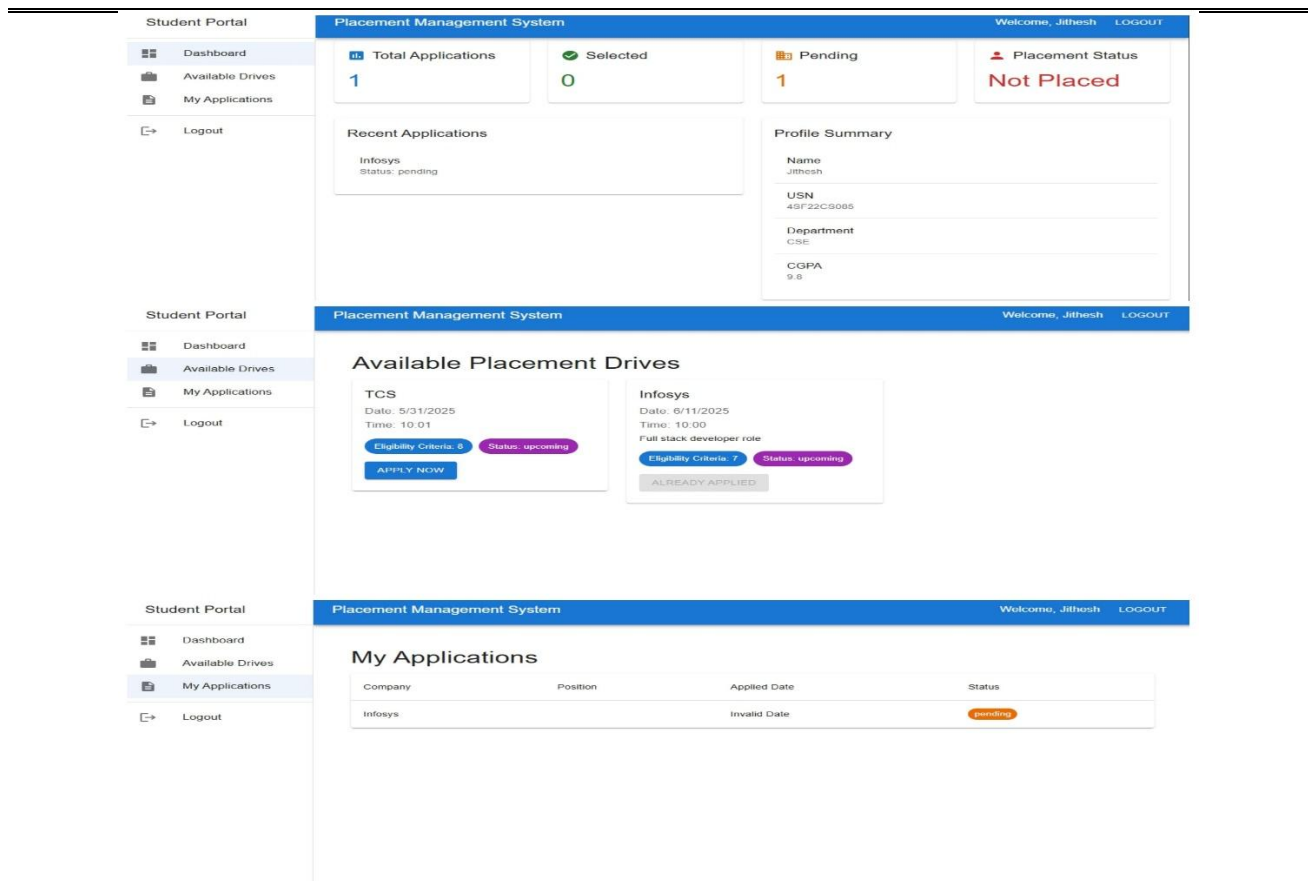


Fig-1: Student UI

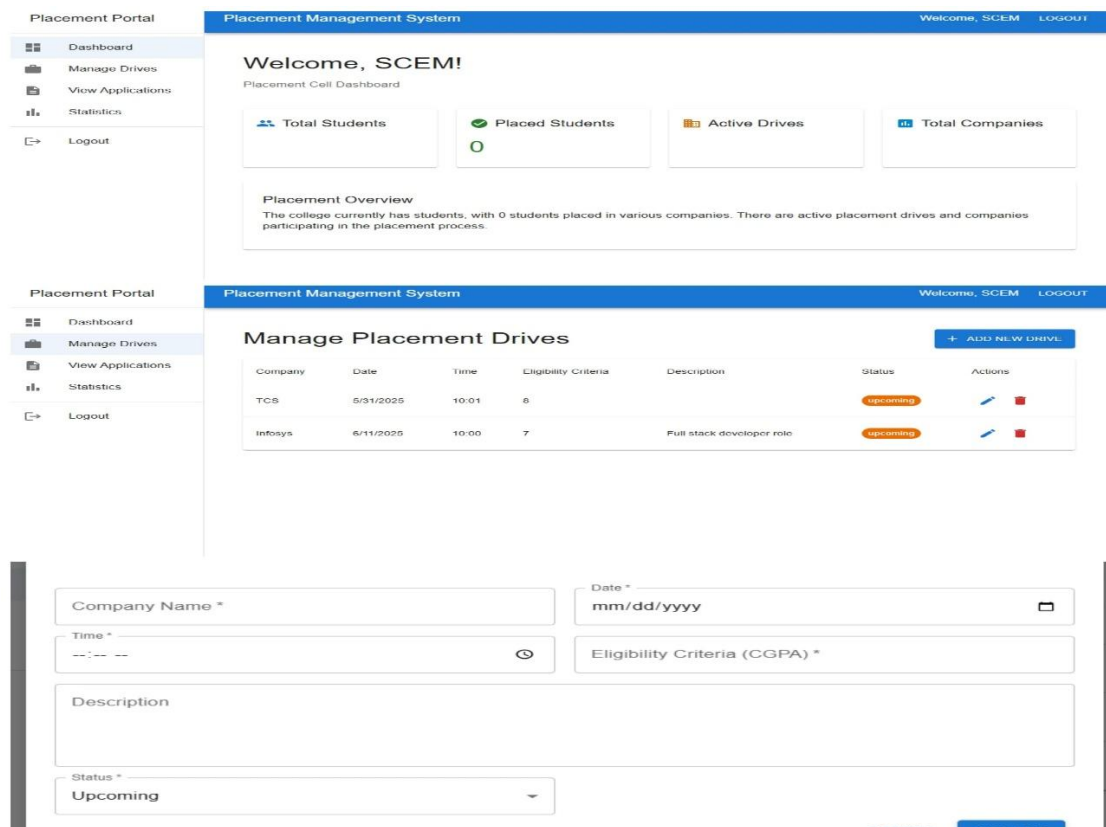


Fig-2: Placement Cell UI

View Applications

Student Name	Student USN	Student Department	Company	Drive Date	Drive Time	Applied At	Status
Dheeraj	4SF22EC077	ECE	Infosys	6/11/2025	10:00	6/3/2025, 8:36:46 PM	pending
Jithesh	4SF22CS085	CSE	Infosys	6/11/2025	10:00	6/3/2025, 8:25:32 PM	pending

Student Details:

Name: Dheeraj

Email: dheeraj@gmail.com

USN: 4SF22EC077

Department: ECE

CGPA: 7

Drive Details:

Company: Infosys

Date: 6/11/2025

Time: 10:00

Application Details:

Status: pending

Applied At: 6/3/2025, 8:36:46 PM

Update Status

Pending

Fig-3:Placement Cell UI

```
placement-management> db.users.find()
[
  {
    _id: ObjectId('683b1cc31652f3c1b85dfca1'),
    name: 'Jithesh',
    email: 'jitheshpshetty14@gmail.com',
    password: '$2a$10$vaGDIXkQYDALMIktmAFcD.yO.o.q9W10dVr7j7qQVYU1IyEmVnbce',
    role: 'student',
    department: 'CSE',
    usn: '4SF22CS085',
    cgpa: 9.8,
    createdAt: ISODate('2025-05-31T15:14:11.441Z'),
    updatedAt: ISODate('2025-05-31T15:14:11.441Z'),
    __v: 0
  },
  {
    _id: ObjectId('683b1f3e9d7b924e48ae0358'),
    name: 'SCEM',
    email: 'sahyadri@gmail.com',
    password: '$2a$10$vJCHVxOzQDiHhShPBLxo0Oes91naOj9F/JXj.UrtcDdj78Dz81Zha',
    role: 'placement_cell',
    department: '',
    usn: '',
    cgpa: null,
    createdAt: ISODate('2025-05-31T15:24:46.190Z'),
    updatedAt: ISODate('2025-05-31T15:24:46.190Z'),
    __v: 0
  },
  {
    _id: ObjectId('683b26929d7b924e48ae037b'),
    name: 'Dheeraj',
    email: 'dheeraj@gmail.com',
    password: '$2a$10$E6Wu2YHFFhIZH8GdsyiylenSKxXwbrCv4Xvhe/5nR04uI9T5k0gi',
    role: 'student',
    department: 'ECE',
    usn: '4SF22EC077',
    cgpa: 7,
    createdAt: ISODate('2025-05-31T15:56:02.587Z'),
    updatedAt: ISODate('2025-06-04T15:01:37.836Z'),
    __v: 0
  },
]
```

Fig-4:MongoDB terminal of Users Collection

```

placement-management> db.placementdrives.find()
[
  {
    _id: ObjectId('683b2f01d844f8b8d030fd18'),
    companyName: 'TCS',
    date: ISODate('2025-05-31T00:00:00.000Z'),
    time: '10:01',
    eligibilityCriteria: 8,
    description: '',
    status: 'upcoming',
    createdBy: ObjectId('683b1f3e9d7b924e48ae0358'),
    participants: [],
    createdAt: ISODate('2025-05-31T16:32:01.232Z'),
    updatedAt: ISODate('2025-05-31T16:32:01.232Z'),
    __v: 0
  },
  {
    _id: ObjectId('683f064af12197d75b741506'),
    companyName: 'Infosys',
    date: ISODate('2025-06-11T00:00:00.000Z'),
    time: '10:00',
    eligibilityCriteria: 7,
    description: 'Full stack developer role',
    status: 'upcoming',
    createdBy: ObjectId('683b1f3e9d7b924e48ae0358'),
    participants: [
      {
        student: ObjectId('683b1cc31652f3c1b85dfca1'),
        status: 'pending',
        _id: ObjectId('683f0ce442d9b52f0241967e'),
        appliedAt: ISODate('2025-06-03T14:55:32.721Z')
      },
      {
        student: ObjectId('683b26929d7b924e48ae037b'),
        status: 'selected',
        _id: ObjectId('683f0f8642d9b52f02419884'),
        appliedAt: ISODate('2025-06-03T15:06:46.476Z')
      }
    ],
    createdAt: ISODate('2025-06-03T14:27:22.906Z'),
    updatedAt: ISODate('2025-06-04T15:01:37.820Z'),
    __v: 2
  },
]

```

Fig-5:MongoDB terminal for placementdrives Collection

Chapter 5

Conclusion

The Placement Management System was developed to address the challenges and inefficiencies in managing placement processes within educational institutions. By automating essential tasks such as eligibility verification and participation tracking, the system has significantly reduced manual errors and administrative overhead. The integration of role-based access control and real-time updates has enhanced both user experience and data security, ensuring that only authorized users access sensitive information.

The database design, structured using Mongoose with clear relationships between collections, ensures efficient data storage and retrieval, forming a strong foundation for the application. Backend logic effectively replaces traditional stored procedures and triggers, automating complex operations and maintaining data consistency. The system's modular architecture and scalability-focused development allow it to accommodate growth in users and data volume without compromising performance.

Positive feedback from users has confirmed the system's effectiveness and usability, underscoring its potential for wider deployment. Overall, the project has met its primary objectives while offering key insights into building modern, efficient, and user-centric placement management solutions.

