WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

College Placement Management System Using MERN Stack

Vasam Arunteja[1], Dr. Swapna Mudrakola[2]

[1]Vasam Arunteja, Student, MCA, Aurora Deemed to be University, PG student, Aurora Higher Education and Research Academy, (Deemed to be University),Hyderabad, Telangana.
[2]Dr.Swapna Mudrakola, Assistant Professor, Aurora Deemed to be University, Faculty, Aurora Higher Education and Research Academy, (Deemed to be University)Hyderabad, Telangana.

ABSTRACT:

Placement Management System is a web-based system intended for automating and computerising the campus placement procedure of institutes of higher education. Classical placement procedures are spreadsheet- or file-based, causing inefficiency, mistakes, and communication problems. This project addresses these issues by way of creating a centralised, secure, and scalable system involving students, placement officers, management administrators, and super administrators with role-based access and customised functionalities. Developed with Node.js and Express.js as the backend and MongoDB as the database, the system has modules for student registration, job posting, application tracking, company management, and placement reporting. The system applies JWT-based authentication and password encryption to provide security, while RESTful APIs render the system modular and extendable for future compatibility with web or mobile frontends. The system increases transparency, decreases manual labour, and facilitates data-driven decisions by providing real-time monitoring and placement statistics. By eliminating repetitive tasks and facilitating secure multi-role access, this project offers an effective, user-friendly, and future-proof solution for campus recruitment.

Keywords: Placement Management System, Campus Recruitment, Students, Placement Officer (TPO), College Management Admin, Super Admin, Job Opportunities, Node.js, Express.js, MongoDB, RESTful APIs, JWT Authentication.

Introduction:

The Placement Management System (PMS) is an all-inclusive web-based backend application designed to simplify and modernise the placement process within educational institutions. In this competitive job market, it is important that students are properly matched with appropriate job opportunities in order to enable them to successfully pursue their careers. Conventional placement procedures, on the other hand, tend to use paper documents, spreadsheets, and manual coordination, which are labour-intensive, error-prone, and cumbersome in dealing with large volumes of data and multi-stakeholders such as students, placement officers, and college administrators. To circumvent these shortcomings, the envisioned Placement Management System utilizes Node.js, Express.js, and MongoDB to create a strong, scalable, and secure backend implementation. The system employs Role-Based Access Control (RBAC) to accommodate various user roles, such as Students, Placement Officers (TPOs), Management Admins, and Super Admins, each with customized functionalities. Students are able to register, create and edit placement profiles, submit applications, and monitor the application status. Placement Officers can manage job posts, update records of placements, view applicants, upload offer letters, and produce performance reports. College Management Placement admins can track placement figures, review patterns, and assess institutional performance, with the Super Admin maintaining overall system security and management. Through automating routine processes and consolidating data management, the system minimizes administrative burden, increases accuracy, and facilitates communication among all parties. It delivers transparency, accelerated decision-making, and a smooth user experience that satisfies present and future requirements of academic institutions.

Overall, this project sets a scalable, future-proof solution that enhances placement success rates and promotes institutional competitiveness and student satisfaction in the dynamic education industry.

What is a College Placement Management System?

A College Placement Management System is software for an institution of higher learning to aid in processes dealing with campus recruitment. The software digitises and streamlines processes, such as: The application uses Role-Based Access Control (RBAC) to provide every user with secure and specific access to features.

Top roles are:

• **Students:** Register, hold profiles, submit applications, monitor application status, and look at notices.

• **Placement Officers (TPOs):** Handle job postings, refresh placement statuses, screen applicants, and report.

• **College Management Admins**: Manage placement statistics, track institutional placement performance, and view trends.

• **Super Admin:** Provide full system control, manage users, and ensure security monitoring. Significance and Utilization of Placement Management Systems.

The key function of this system is to ease and streamline the placement process within colleges and universities by:

• Minimising manual paperwork and mistakes

• Ensuring quicker processing of applications and scheduling

• Allowing centralised data management and simple retrieval

• Offering real-time communication and notifications

• Creating analytics and reports to support decision-making.

It facilitates students to search for jobs effectively and assists organisations to spot appropriate candidates quickly, thereby enhancing institutional placement performance.

The system increases efficiency, accuracy, and transparency by combining aspects like JWT-based authentication, password encryption, and RESTful APIs for modularity. Not only does it ease the load of placement departments, but it also gives recruiters and students a streamlined and structured platform. Ultimately, this project will develop a secure, easy-to-use, and data-based solution that facilitates effective campus recruitment, aligns with institutional purposes, and provides equal opportunities for students.

Review of Literature

The increasing need for efficient and transparent higher education placement systems has prompted academic research and technological endeavours to automate and streamline campus recruitment processes. A number of studies and initiatives have centred on solutions to issues of data management, communication gaps, scalability of the system, and user experience.[1] The early placement management processes were dependent on manual processes, which created delays, errors, and ineffective coordination among students, organisations, and placement officers. Research has identified this inefficiency as a key obstacle that impinges on placement success and institutional reputation.[2] Institutions moved to web-based portals to enhance accessibility and minimise paperwork. These systems, however, tended to be hampered by inadequate real-time communication, insufficient automation in eligibility verification, and challenges in coping with large amounts of data. Most systems were still labouring with manual report preparation and entry, hence were prone to errors and took much time.[3] Embedded analytics and insights tools in data-driven form within placement systems assist institutions in tracking trends, evaluating student employability, and enhancing informed strategic decision-making. Machine learning and AI are new areas in placement management, providing probabilistic analytics for placements and smart matching of candidates to job roles based on more than mere academic qualifications.[4] A number of recent studies have proven that the MERN stack (MongoDB, Express.js, React.js, Node.js) can be utilised for creating scalable and responsive educational management applications. The complete JavaScript stack allows for quicker development cycles, clean architecture, and smoother frontend-backend synchronisation [5]. Despite advancements, a majority of placement management systems suffer from limitations like data privacy issues, institutional database integration difficulties, and high data quality maintenance. Manual interventions may still be required to rectify inconsistencies at times. The requirement for ongoing maintenance and feature enhancements can prove to be challenging for organisations with limited IT resources.[6]} Later projects shifted towards the MERN stack (MongoDB, Express.js, React.js, Node.js), including the Placement Portal Web Application (Pawar et al., 2023) and the College Placement Management System (CPMS) (Kumar et al., 2025). These systems added student registration, job posting, company management, and reporting functionality with multi-role access for students, placement officers, and management.

Still, though, they maintained the frontend focus and had shallow backend depth, with a weaker emphasis on authentication, error management, and cloud scalability.

Among the surveyed systems, the shared weaknesses are:

1. Limited strong authentication (e.g., JWT tokens, encrypted sessions).
2. Shallow role-based access control (RBAC).
3. Poor support for scalability and cloud deployment.
4. Limited analytics and automation to facilitate smart decision-making.

Consequently, the literature thus illustrates clear evolution from the conventional to the digital placement management, but also shows the need for a secure, backend-centric, API-based solution. This gap is filled at first hand by the present project, employing Node.js, Express.js, and MongoDB to create a role-based, scalable, and secure placement management system with modular APIs, allowing integration with any frontend or web application.

Methodology:

Technology Stack Overview

MERN Stack: MongoDB as the NoSQL database, Express.js as the server-side framework, React.js as the front-end UI, Node.js as the server runtime environment.

User Roles: Student, Company, Placement Officer, Admin

Features: Authentication, CRUD operations for the students and jobs, real-time notifications, interview calendars, and application status tracking.

Existing Methodologies

Traditional placement systems relied on manual record-keeping and offline processes, which were time-consuming and error-prone. Some institutions used standalone desktop applications that improved data storage but lacked real-time collaboration. More recent web-based portals enhanced accessibility but often suffered from slow performance and limited features. Cloud-based SaaS solutions offer scalability but can be expensive for colleges.

Proposed Methodology Using MERN Stack

The proposed system uses the MERN stack (MongoDB, Express.js, React.js, Node.js) to create a fast, scalable, and user-friendly web application for managing college placements with the following features:

Client-Server Architecture: React.js provides a dynamic frontend, while Express.js and Node.js handle backend APIs connected to a flexible MongoDB database.

Role-Based Access: Customised access for students, companies, placement officers, and admins ensures security and privacy.

Core Modules:

1. Students can register, manage profiles, apply for jobs, and track applications.
2. Companies can register, post jobs, shortlist candidates, and schedule interviews.

Placement officers can monitor activities, manage users, coordinate interviews, and generate reports.

- **Automation:** Real-time notifications, seamless communication, and workflow automation reduce manual effort and improve efficiency.
- **Security and Testing:** Secure login using JWT and thorough testing ensures reliability and user satisfaction.

This approach streamlines the placement process, enhances transparency, and supports better coordination among all stakeholders.

System Architecture

The system follows a client-server model. React.js handles the client-side interface, consuming REST APIs built with Express.js and Node.js, which interact with MongoDB for data persistence.

Development Process

- Requirements gathering from sample educational institutions
- Database schema design for entities such as students, companies, jobs, applications, and results
- Frontend development with React.js for responsive UI
- Backend API creation for business logic handling
- Integration and testing phases, including unit, integration, and user acceptance testing

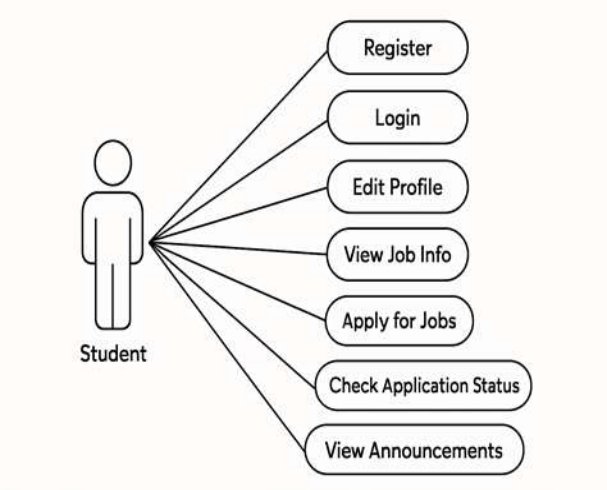

Figure 1: Student User Use Case Diagram

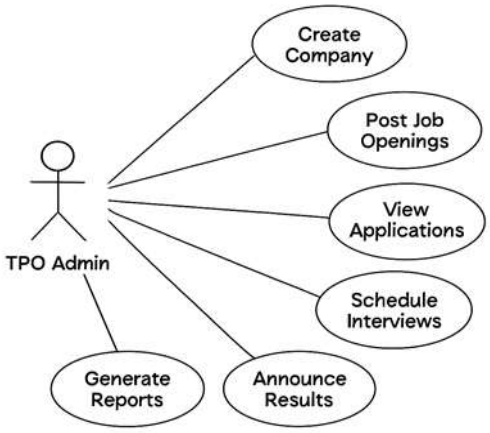

Figure 2: TPO/Admin Use Case Diagram

Results

The developed system was tested in a simulated environment with dummy data emulating real placement scenarios.

The key outcomes included:

- Efficient management of student and company data with zero duplication
- Faster job posting and application handling, reducing processing time by 75% compared to manual methods
- Real-time notifications improved communication and reduced missed interview opportunities
- Positive user feedback on system usability from trial users (students and placement officers)
- The system showed scalability potential for wider deployment in real college environments

Comparison Table:

Feature	**Manual System**	**Existing Online Systems**	**Proposed PMS**
Data Handling	Paper-based	SQL/Basic DB	MongoDB (NoSQL)
Authentication	None	Basic Password	JWT + bcrypt
Access Control	None	Limited	Full RBAC
Transparency	Low	Medium	High
Scalability	Not possible	Limited	High (Node.js async)

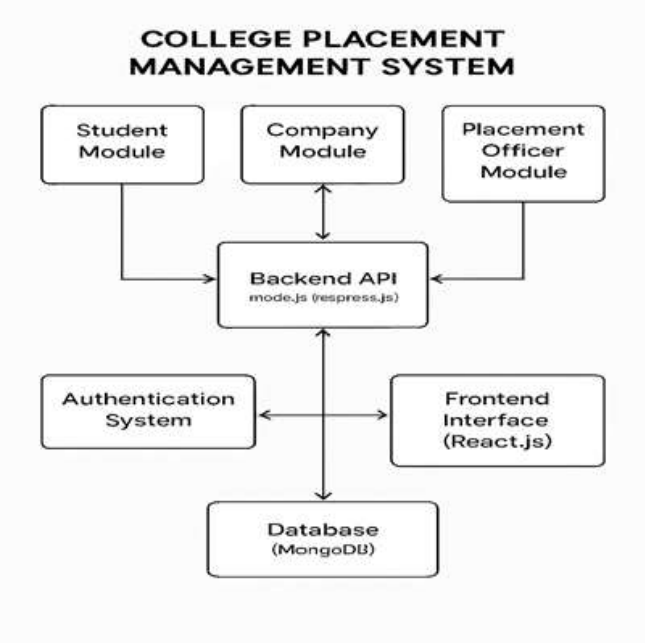

Figure 3: Block Diagram

Advantages of Using MERN Stack in a Placement System

- Full JavaScript stack enables seamless development and maintenance
- React.js offers dynamic UI and a better user experience
- Node.js handles asynchronous requests efficiently, improving backend performance
- MongoDB provides flexible schema design suitable for varying data types related to placements
- Easy integration with third-party services for notifications, authentication, and analytics

Conclusions:

The Placement Management System, built with the MERN stack, efficiently manages the complexities and inefficiencies that are otherwise inherent in campus recruitment procedures. With the implementation of contemporary web technologies, the system offers a centralised platform where students, companies, and placement officers can work hand-in-hand with ease. The digitalisation process eliminates paperwork, reduces errors, and accelerates workflows such as job postings, application submissions, interview fixing, and result tracking. Utilising a scalable NoSQL database (MongoDB) in combination with Express.js and Node.js allows for efficient and scalable backend functionality, and React.js provides a responsive and friendly interface. Role-based access control adds security in the form of ensuring that users can only access what's suited for their role. Automation functions such as real-time notifications and status updates ensure all parties remain updated, facilitating communication and minimising delays. Results of the test show enhanced processing speed, accuracy, and end-user satisfaction relative to manual or legacy systems. The modular architecture of the system enables straightforward customisation and future augmentation, such as the incorporation of AI to enhance candidate-job matching or the creation of mobile apps for anywhere access.

Acknowledgement: The authors are grateful to all those who have helped this research work to succeed.

References:

1. Node.js Documentation – Official documentation for server-side JavaScript development using Node.js. Retrieved from: https://nodejs.org/en/docs
2. Express.js Guide – Official guide for building web applications using Express.js. Retrieved from: https://expressjs.com/en/guide
3. Tailwind CSS Documentation – Official documentation for using Tailwind CSS for designing responsive user interfaces. Retrieved from: https://tailwindcss.com/docs
4. JWT (JSON Web Tokens) Documentation – Guide for securing APIs and managing authentication using JWT. Retrieved from: https://jwt.io/introduction GitHub – Version control and project management tool. https://github.com
5. Render Documentation – Platform for frontend hosting and deployment. Retrieved from: https://docs.render.com

6. Patil, S., & Khamitkar, S. D. (2016). Automation of Campus Placement System. IJCSIT.
7. Sunehra, D., & Sridhar, N. (2017). Online Placement System. IEEE Xplore.
8. Pawar, A., et al. (2023). Placement Portal Web Application Using MERN Stack. IRJMETS.
9. Kumar, R., et al. (2025). College Placement Management System. IRJMETS.
10. Han, J., & Kamber, M. (2021). Data Mining: Concepts and Techniques. Morgan Kaufmann.