Centralized Platform for Streamlining Campus Placement

[1]Prof. Amol Jadhav, [2]Dipraj Vasav, [3]Akshay Raut, [4]Omraje Patil, [5]Vineet Sakpal
[1,2,3,4,5] *Zeal College of Engineering and Research, Pune*

***Abstract*— Centralized Platform For Streamlining Campus Placement is a centralized web-based platform designed to enhance communication, collaboration, and resource sharing within educational institutions. By integrating features such as announcements, event management, discussion forums, and resource repositories, the platform aims to bridge the gap between students, faculty, and administration. Leveraging modern web technologies, Centralized Platform For Streamlining Campus Placement provides a user-friendly interface that facilitates real-time interactions and fosters a connected campus community. The system's modular design ensures scalability and adaptability to various institutional needs, promoting efficient information dissemination and collaborative learning environments.**

***Index Terms*— Placement, Recruitment, Dashboard, Authentication, Resume Parsing, Student Portal, Admin Panel, Job Application, MERN Stack, Deployment**

I. INTRODUCTION

This application is a comprehensive web-based platform designed to revolutionize and digitize the traditional campus placement process in educational institutions. The current placement system in many colleges, including ours, relies heavily on manual processes—placement coordinators distribute and collect physical forms, manage spreadsheets, and manually coordinate with companies, making the entire process tedious, error-prone, and inefficient. Recognizing the need for modernization, It was conceptualized as a centralized solution to address these challenges by automating the workflow and bringing transparency, efficiency, and real-time access to all stakeholders involved in campus recruitment. The application offers distinct modules for administrators and students, each equipped with features tailored to their roles. Admins can manage placement drives, track student applications, post notices, and oversee alumni interactions, while students can view and apply for drives, access learning resources, receive instant notifications, and manage their profiles. Built using modern technologies like React.js for the frontend, Node.js/Express and Python APIs for the backend, and MongoDB Atlas for cloud-based storage, the system adopts an agile development approach and modular architecture to ensure scalability and maintainability. Deployed on Render for easy accessibility, Campus Connect not only simplifies the placement process but also enhances the overall experience for users through a secure, responsive, and intuitive interface.

II. LITERATURE SURVEY

The landscape of campus placement management is evolving, with a growing focus on automation and advanced technologies to streamline processes and enhance student opportunities. This essay explores various research studies and their findings related to placement management systems, predictive analytics, and web-based solutions, detailing their limitations, implementations in our project, and potential improvements.

In another relevant study, Prasad Khalkar and colleagues (2023) discuss a web-based TnP portal that uses machine learning to recommend suitable companies to students. The system's scalability and dependency on administrators limit student flexibility. [1]

Shreya Khale and her co-authors (2024) present a comprehensive web-based campus placement portal. The study highlights challenges such as high maintenance costs and limited adoption by institutions. Our project aims to develop a full-featured portal with user-friendly interfaces for both students and administrators while focusing on reducing maintenance complexity through efficient backend systems and minimizing scaling costs.[3]

Moreover, the study by Ms. Sarita Byagar, Dr. Ranjit Patil, and Dr. Janardan Pawar (2024) proposes using machine learning models to boost campus placements. One significant limitation is the requirement for extensive data and technical expertise. Our project will integrate machine learning to forecast student placement outcomes based on

historical data while simplifying model deployment. We aim to automate model retraining as new data becomes available, reducing the need for ongoing technical expertise.[4]

K. Mahalakshmi and R. Sathish Kumar (2024) discuss the creation of a centralized web application for managing placement cells, facilitating interactions between students, placement officers, and recruiters. Challenges include user engagement issues and technology access difficulties, particularly for students unfamiliar with digital platforms. Our platform will also centralize placement activities and integrate interactions among administrators and recruiters, focusing on improving user experience and accessibility for all students, especially those with limited digital literacy.[5]

Similarly, Ashish Nanotkar and his colleagues (2023) describe the creation of a web-based application designed to automate essential placement tasks, including student training and application processes. This application is limited to institutions willing to adopt the technology. Our project will also automate processes such as student data management and application tracking, with enhancements like data visualization and real-time notifications for improved interaction between students and administrators.[6]

We plan to develop a similar MERN-based application that manages placement activities while providing students with greater autonomy over their applications and ensuring streamlined oversight for administrators. Fiza Kousar and her team (2022) developed a MERN-based placement portal aimed at simplifying processes and ensuring data security. However, their reliance on digital platforms may exclude students with limited access to technology. Our project will also employ the MERN stack but will focus on optimizing the portal for low-bandwidth environments, enhancing accessibility for students who may have limited internet access.[7]

In their 2022 research, Geeta Kesavaraj and Manjula Pattnaik evaluate the effectiveness of campus recruitment processes, particularly in the IT sector. The findings may not be applicable to other industries, representing a significant limitation. Insights from this study will inform our placement tracking system, which will be designed to be industry-agnostic, supporting placement drives across diverse sectors and providing a broader range of opportunities for students.[8]

Caesar Jude Clemente and Myungjae Kwak (2022) focus on using data science tools and machine learning algorithms for predicting campus placements. A key limitation of their research is the reliance on data quality, making it complex to scale. In our project, we will leverage historical student placement data to employ machine learning models for job placement predictions. We also plan to enhance the model's accuracy through feature selection and hyperparameter tuning, ensuring scalability and ease of implementation across various institutions.[9]

Additionally, Imran and Almusharraf (2024) present a study on Google Gemini as a multimodal AI system with applications in learning and data processing. Their discussion on Gemini's ability to parse multimodal inputs—such as text and PDFs—aligns with our project's use of Gemini API in the resume parsing module. Our application integrates Gemini's capabilities to extract information from student resumes and compare them with job descriptions, enabling an AI-powered ATS scoring system that provides detailed insights into resume quality and missing skills. This contributes to personalized feedback and enhanced student preparation. [10]

In their 2022 study, Alfiya Banu and Dr. Manju Bargavi S. K investigate ways to enhance placement management systems through automation, particularly in candidate data collection. However, this research is limited to specific institutions and regions. Our project aims to adopt the systematic framework proposed in their study to manage student data while automating similar tasks. Additionally, we plan to extend the reach of our system to a wider range of institutions and incorporate advanced machine learning techniques for predicting placements, which goes beyond mere data collection. [11]

III. SYSTEM ARCHITECTURE

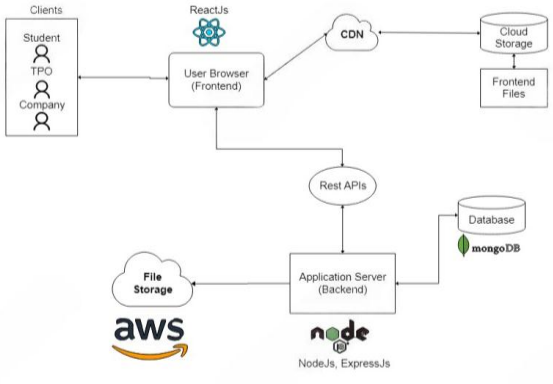


Fig -1: *System Architecture*

Clients (User Roles)

- Student: Students create and manage their profiles, upload resumes, and apply for placement drives.
- TPO (Training and Placement Officer): TPOs manage placement drives, review student applications, and oversee system operations.
- These users access the system via a web browser interface on desktop or mobile devices.

Frontend (React.js)

- The user interface is built using React.js, providing a fast and responsive single-page application experience.
- Students and TPOs interact with various modules like the dashboard, drive application pages, and resume score viewer.

Content Delivery and Hosting

- Render is used for deploying the full-stack application, enabling seamless integration between frontend and backend.
- Static frontend files are served via CDN to reduce latency and improve performance.

REST APIs (Node.js)

- The frontend communicates with the backend via RESTful APIs.
- APIs handle tasks like user authentication, drive creation, fetching drive lists, and resume evaluation requests.

Backend (Node.js + Express.js + Python Flask)

- Node.js (with Express.js) handles core business logic like user sessions, data validation, and database transactions.
- Python Flask is used for the ATS resume parsing module, which processes uploaded resumes, compares them with job descriptions, and returns an ATS score and missing skills.

Database (MongoDB)

- MongoDB Atlas stores student profiles, drive details, TPO information, and application history.
- The flexible schema allows easy modification and expansion of the data structure as needed.

File Storage (AWS S3 or equivalent)

- Resume files and related documents are uploaded and stored securely using cloud storage solutions.
- Files are retrieved during resume parsing and application viewing processes.

Flow of Interaction

- Users access the system through the web browser, triggering requests handled by the React.js frontend.
- React sends requests to the backend API hosted on Node.js, which may also invoke the Flask-based resume parser if needed.
- The backend queries the MongoDB database and retrieves/stores data accordingly.
- Resume files are fetched or stored using AWS S3 (or another integrated storage service).
- The final responses are rendered back to the frontend, providing a seamless and fast user experience.

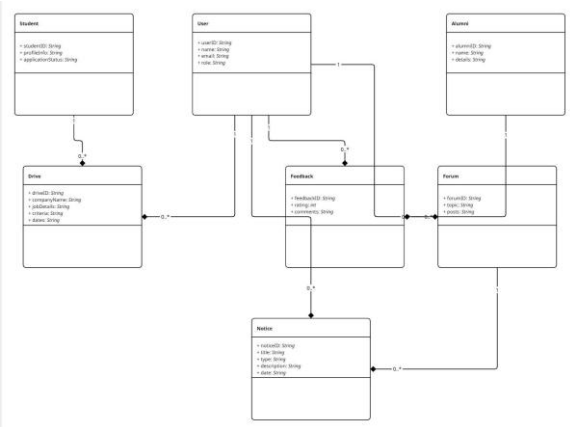


Fig -2: *UML Class Diagram*

1. User
    - Attributes: userID, name, email, role
    - Represents all users of the system including students, admins, and alumni.
    - Acts as a parent entity to Student and Alumni.

2. Student
    - Attributes: studentID, profileInfo, applicationStatus
    - Represents students participating in campus drives.
    - Has a one-to-many relationship with Drive.

3. Alumni
    - Attributes: alumniID, name, details
    - Represents alumni users.
    - Participates in forums for guidance and discussion.

4. Drive
    - Attributes: driveID, company Name, job Details, criteria, dates
    - Represents placement drives conducted by companies.
    - Linked to multiple students.
    - Can receive feedback from users.

5. Feedback
    - Attributes: feedbackID, rating, comments
    - Collects feedback from users (students or alumni).
    - Related to multiple other entities: User, Drive, Forum.

6. Forum
    - Attributes: forumID, topic, posts
    - Facilitates discussion between students and alumni.
    - Can receive feedback.
    - Each forum is hosted by a single alumni.

7. Notice
    - Attributes: noticeID, title, type, description, date
    - Represents public announcements regarding events, drives, or system updates.
    - Issued by users and viewable by all roles.

- A User can be linked to:
    - Multiple Feedback entries.
    - Multiple Notice entries.

- A Student:
    - Is associated with one User.
    - Can apply to multiple Drives.

- A Drive:
    - Can have feedback from multiple Users.
    - May be associated with multiple Students.

- A Forum:
    - Is hosted by one Alumni.
    - Can receive feedback from multiple Users.

- A Notice:
    - Is created by one User.
    - Multiple users may read a notice (though not shown explicitly).

- Feedback:
    - Is a bridge entity allowing users to provide input on Drives and Forums.

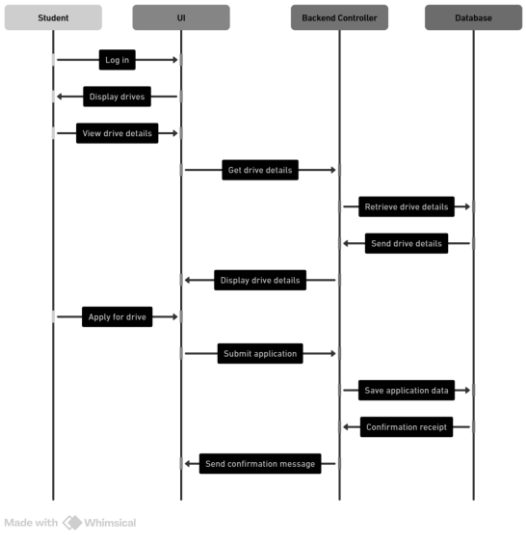


Fig -3: *UML Sequence Diagram*

1. Login & Display Drives
    - Student logs into the system via the UI.
    - UI fetches and displays available job drives.

2. View Drive Details
    - Student selects a drive to view more details.
    - UI sends a request to the backend controller to fetch specific drive details.

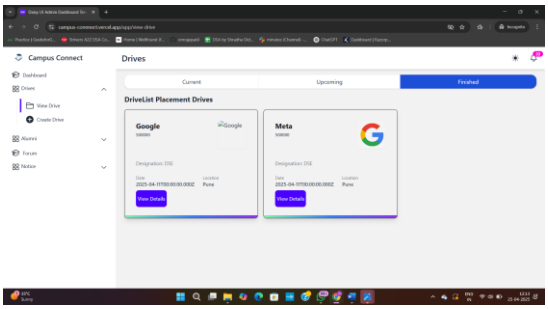
- Backend controller queries the database to retrieve the required details.
- Retrieved drive details are sent back to the backend and then displayed on the UI.

3. Apply for Drive
   - Student clicks "Apply for Drive".
   - UI sends the application data to the backend controller.

4. Submit Application
   - Backend controller receives the application.
   - It stores the application data in the database.
   - Database sends a confirmation receipt.

5. Confirmation Message
   - Backend sends the confirmation to the UI.
   - UI displays a confirmation message to the student indicating successful application.

IV. IMPLEMENTATION



Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). This applies to papers in data storage. For example, write ―15 Gb/cm^2 (100 Gb/in^2).‖ An exception is when English units are used as identifiers in trade, such as ―3½ in disk drive.‖ Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

The SI unit for magnetic field strength H is A/m. However, if you wish to use units of T, either refer to magnetic flux density B or magnetic field strength symbolized as μ0H. Use the center dot to separate compound units, e.g., ―A·m^2.‖

V. CONCLUSION

This application simplifies and automates the campus placement process through a centralized web platform. It integrates resume parsing with AI (Gemini API), drive management, and student applications under one system. Future enhancements include mobile app development, AI-based job recommendations, real-time communication tools, and stronger security measures. The platform is scalable, user-friendly, and ready for broader institutional deployment.

VI. FUTURE SCOPE

1. Advanced Resume Matching: Integrate AI models to recommend personalized job roles based on student skills and history.
2. Mobile Application Development: Build a mobile app with push notification support for placement management.
3. Real-time Communication Integration: Add chat and video conferencing features for better interaction between students, TPOs, and companies.
4. Predictive Analytics for Placement Trends: Use historical data to forecast hiring trends and improve training alignment.
5. Enhanced Security with 2FA: Implement two-factor authentication and stronger encryption for better data security.

REFERENCES

[1] P. Khalkar, A. Muthal, S. Khopade, L. Ghule, and T. Rane, “TnP Portal Using Web Development and Machine Learning,” *Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET)*, 2023.

[2] A. Kale, S. Joshi, T. Rajyaguru, I. Borgaonkar, C. Jadhav, and P. Sakunde, “Enhancing Student Placement with Cross-Platform Application,”

Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET), 2024.

[3] S. Khale, M. Grover, and S. Kadam, "A Dynamic, Full Featured Portal for Campus Placement," *Int. J. Res. Eng. Sci. Manage.*, 2024.

[4] S. Byagar, R. Patil, and J. Pawar, "Maximizing Campus Placement through Machine Learning," *J. Adv. Zool.*, 2024.

[5] K. Mahalakshmi and R. Sathish Kumar, "Online Placement Cell Management System," *Int. J. Res. Publ. Rev.*, 2024.

[6] A. Nanotkar, N. Jamgade, P. Tiwari, and S. Ninawe, "Development of Web Application to Automate Training & Placement Process," *Int. J. Novel Res. Dev. (IJNRD)*, 2023.

[7] F. Kousar, G. Hegde, and S. Saravanan, "Design and Development of Department Placement Portal using MERN Technology," *Int. Adv. Res. J. Sci. Eng. Technol.*, 2022.

[8] G. Kesavaraj and M. Pattnaik, "A Study on the Effectiveness of Campus Recruitment and Selection Process in IT Industries," *ResearchGate*, 2022.

[9] C. J. Clemente and M. Kwak, "Utilizing Data Science and Analytics in Predicting Campus Placement," 2022.

[10] A. Banu and M. Bargavi, "A Research on Placement Management System," *Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET)*, 2022.

[11] M. Imran and N. Almusharraf, "Google Gemini as a Next Generation AI Educational Tool: A Review of Emerging Educational Technology," *Smart Learn. Environ.*, vol. 11, no. 22, 2024, doi: 10.1186/s40561-024-00310-z.