

CAMPUS RECRUITMENT MANAGEMENT SYSTEM: A MERN STACK-BASED PLATFORM FOR SEAMLESS HIRING

G. Durga Prasanna¹, A. Sarath Kumar², B. Bhagyaraj³, B. Divyasri⁴, G. Sujith⁵,
Smk Moinuddin⁶, P. Annannaidu⁷

^{1,2,3,4,5,6,7}Department of Computer Science & Engineering GMRIT, Rajam, Andhra Pradesh, India.

DOI: <https://www.doi.org/10.58257/IJPREMS39818>

ABSTRACT

The college placement system has become increasingly online, and thus an effective and easy-to-use system needs to match students with recruiters. This paper introduces an Online College Placement Portal implemented with the MERN (MongoDB, Express.js, React, Node.js) stack that can facilitate the hiring process. The platform enables students to make profiles, add resumes, and apply for job openings, while recruiters are able to create job openings, view applications, and track hiring processes. An admin panel allows user management and placement tracking. The portal ensures secure authentication, real-time feedback, and user experience, enabling smooth interaction among students, recruiters, and administrators. Future additions include resume screening automation, interview scheduling, and placement trend analytics to further streamline campus hiring. This project illustrates how a MERN-based placement portal can maximize efficiency, transparency, and accessibility in college hiring.

Keywords: College Placement, MERN Stack, Recruitment Portal, Job Applications, Student Hiring, Campus Recruitment

1. INTRODUCTION

The web-based placement portal trend has revolutionized the mode of recruiter-student interaction in recruitment processes. Most of the platforms, however, still rely on static job postings and manual job searching, making the process cumbersome for students to find suitable opportunities that suit their career goals and skills. As the season of campus placements gets hotter, students need to have a proper and well-managed system for easy job application and smooth recruitment. This article presents an Online College Placement Portal developed on the MERN (MongoDB, Express.js, React, Node.js) stack that closes the gap between students and recruiters with a one-stop-shop solution to post jobs, apply, and monitor placement status. The portal allows students to make profiles, upload their resumes, and apply for postings, whereas recruiters can post postings, filter applications, and select candidates. The MERN stack provides high performance, scalability, and usability for all the stakeholders. With real-time feedback, secure login, and easy interface, this placement portal is designed to introduce more transparency, efficiency, and accessibility in the recruitment process. Future developments include automatic scheduling of interviews, verification of resumes, and job placement suggestions by analytics to make campus hiring even more automated. This research proposes how a MERN-based college placement system can render campus recruitment an effective and seamless hiring process for both employers and students.

Understanding College Placement portal

The college placement procedure is a vital one for the students to secure employment before they complete their studies. Today, placements have largely gone online, and one should have a system that facilitates ease of interaction for students as well as recruiters. This is where the Online College Placement Portal fills in. This project is about creating a web platform using something referred to as the MERN stack — which is an acronym for MongoDB, Express.js, React, and Node.js. These are newer web development tools that assist in creating quick, powerful, and user-friendly applications. This gateway assists students in creating their own profiles, posting their resumes, and applying for vacancies in various companies. The recruiters can publish vacancies, view applicants, and manage the entire recruitment process on the web. There's also an admin area that assists in managing everything from the background such as viewing who enrolled and monitoring how placements are faring.

The MERN stack makes this system fast and smooth. It also uses secure login systems, gives real-time updates, and offers a good experience for users, whether they're students, recruiters, or admins. It brings everyone onto one platform, making placement activities simpler and more transparent.

In the future, this system could get even better. Features like automatic resume screening, interview scheduling, and data analytics for placement trends could be added. These will help colleges and companies make better decisions and save more time.

Sentence Ranking: This feature is all about giving students a space to create and manage their professional profiles. The system stores key details like their name, branch, skills, and resume. It's like building a personal job-hunting hub that's always ready to show off your qualifications.

Job Posting and Management: This function lets recruiters log in and post job opportunities directly to the portal. Each listing includes essential info like eligibility, CTC, job role, and deadlines. Think of it as a digital notice board where companies pin up their openings.

Application Submission: Once a job is posted, eligible students can easily apply with a click. Their resumes and profile info are automatically sent to the recruiter. It works like an express delivery system—no paperwork, no hassle.

Admin Dashboard: The admin panel is the control center of the entire portal. Here, placement officers can approve users, monitor student applications, and manage job postings. It's like having a command center that keeps everything running smoothly.

Authentication and Security: Every user—student, recruiter, or admin—has their own login credentials with role-based access. This ensures data privacy and controlled access. It's like giving everyone their own passcode to unlock only the doors they're supposed to use.

Real-Time Notifications: Students get alerts whenever a new job is posted, deadlines approach, or application statuses change. It's like having a smart assistant that always keeps you in the loop.

Database Integration (MongoDB): All user data, job listings, and applications are stored in a secure and flexible database. It supports fast access and smooth performance. Think of it as a digital filing cabinet that never misplaces a file.

Frontend with React.js: Its design is created to be fast and polite in temperament with React. It is capable of running on any platform with almost zero delay. It's like having a quick dashboard which is productive as well as productive.

Backend in Express.js and Node.js: They receive the logic in the back ground run—that is, the user logins, storing the data, processing the job flows. Imagine as the engine room of a boat, churning all the work in the back ground.

Evaluation Metrics: It is applied to evaluate the efficiency and effectiveness of the College Placement Portal. These metrics represent a way of assessing to what degree the portal is useful for students, recruiters, and administrators. Some of these are:

User Engagement Rate: Tracks the frequency with which students and recruiters utilize the site—e.g., logging on, applying to a job, or posting jobs. It can be used to gauge the degree of activity and utility the portal enjoys in practice usage.

Placement Conversion Rate: Monitors the actual number of students placed through the portal from the number of applications. It's a reflection of the true impact of the portal on career achievement of the students.

System Response Time: Checks how quickly the portal responds to an action from a user, i.e., accepting an application for a job or opening up the dashboard. The quicker the response time, the better is the user experience.

Resume Shortlisting Accuracy: Monitors how accurately the system filters and forwards the right candidates to recruiters according to eligibility criteria and keywords.

Feedback and Satisfaction Score: Gathers input from students, recruiters, and admins to understand their experience with the portal. This helps in identifying what works well and what needs improvement.

LITERATURE SURVEY

Table 1: Comparison table

Sl. no	Title	year	Description	Limitations	Advantages	Performance metrics	Gaps
1	CanDo: A Web-Based Job Portal for Jobseekers and Freelancers with Disabilities	2024	Design a web-based job portal for jobseekers and freelancers with disabilities.	Limited support for specific assistive technologies; requires stable internet connectivity.	Promotes inclusivity and provides accessible job opportunities for people with disabilities.	Usability Score: 85%; User Satisfaction: 90%	Focus on web platform; lacks mobile app integration for accessibility.

2	Online job portal	2024	Analyze and implement a system for improved computer science education by leveraging collaborative tools and interactive platforms to enhance learning outcomes.	Limited scalability for institutions with diverse technological capabilities. Dependence on internet connectivity and device availability.	Enhances student engagement and course completion rates. Promotes collaboration in learning.	Student Engagement Rate: 85%, Course Completion Rate: 75%	Limited implementation for remote areas. Challenges in adopting new tools by faculty and students.
3	Reflections of Employers' Gender Preferences in Job Ads in India	2023	Analyze employers' gender preferences in job advertisements using data from online job portals in India.	Limited to data from a single online portal. Does not include employer interviews or reasons behind preference	Highlights gender bias in recruitment processes. Provides actionable insights for policy-making.	Insight into gender preferences in 80% of job ads. Identification of gender bias patterns in job postings.	Narrow focus on job ads without cross-verification through employer data. Lack of longitudinal data for trend analysis.
4	Job Crafter—The One-Stop Placement Portal	2024	To create an all-in-one platform for job placements tailored to individual needs.	Limited scalability for a high volume of users.	User-friendly interface and integrated placement tools.	Usability: 87%, User Satisfaction: 90%	Limited support for remote job place
5	A Website for a Consultancy Using MERN Stack	2023	To design a web-based consultancy platform for managing client interactions and projects.	Dependence on MERN stack for all development processes.	Highly customizable and responsive design.	Performance: 85%, Responsiveness: 90%	Limited analytics for performance tracking.
6	Feature Extraction Based Online Job Portal	2022	To implement feature extraction techniques for improved job search and matching accuracy.	Computationally intensive for large datasets.	Improves job relevance and matching efficiency.	Precision: 88%, Recall: 86%	Lacks real-time feature updates.

7	The Development of a Job Portal to Facilitate In-Campus Placement	2024	To create a portal to streamline in-campus placement processes for students.	Limited to in-campus placement; no external jobs.	Simplifies placement management for academic institutions.	Usability: 88%, Accuracy: 85%	Lacks advanced filtering for diverse skill sets.
8	Evaluation of Online Job Portals for HR Recruitment Selection Using AHP	2024	To assess job portals using the AHP method for HR recruitment in the automotive industry.	Limited to the two-wheeler automotive sector.	Provides a structured evaluation for portal effectiveness.	User Satisfaction: 89%, Accuracy: 87%	Limited application beyond the evaluated industry.
9	How Do Online Job Portals Affect Employment and Job Search?	2024	To analyze the impact of online job portals on employment and job search in India.	Focuses on a single geographic region (India).	Provides empirical evidence of job portal effectiveness.	Accuracy: 88%, Usability: 85%	Limited cross-country comparative analysis.
10	Career Catalyst: Empowering Job Search Success	2024	To develop a platform aimed at streamlining the job search process through intelligent tools.	Limited personalization features for unique user needs.	Enhanced user engagement and improved job search accuracy.	Efficiency: 92%, User Satisfaction: 88%	Limited scope for global job market integration.
11	Graha Prakarsa, Reni Nursyanti, & Esar Samuel Baransano (2024).	2023	Develop and evaluate "Career Catalyst," a platform aimed at improving job search success through personalized career advice, skill development, and tailored job recommendations.	AHP Methodology, Web-Based Evaluation Tools, HR Analytics	Selection Accuracy: 90%, Portal User Rating: 80%	Limited to the two-wheeler automotive industry. Focus on HR selection criteria without candidate-specific performance.	Limited support for remote job place
12	An online job portal	2024	Analyze and implement a system for improved computer science education by leveraging collaborative tools and interactive	Limited to the two-wheeler automotive industry. Focus on HR selection criteria without	Personalized career advice, tailored job recommendations.	Selection Accuracy: 90%, Portal User Rating: 80%	Limited implementation for remote areas. Challenges in adopting new tools by faculty and students.

			platforms to enhance learning outcomes.	candidate-specific performance evaluation.			
13	Online Job Portal	2019	To develop a platform for connecting employers and job seekers effectively.	Limited customization options for job postings.	Simplifies job search and recruitment processes.	Not explicitly stated in the paper.	Lacks AI-based recommendation systems.
14	Computing jobs monitoring dashboard in Malaysia	2022	To develop a dashboard to monitor computing jobs and provide insights into job trends.	Limited to computing-related jobs; lacks real-time data updates.	Provides localized job market analysis and trend monitoring.	Accuracy: 88%, Usability: 85%	Does not incorporate AI for predictive analytics or job matching
15	AI-BASED JOB PORTAL (2024).	2024	Develop an AI-based job portal to automate the recruitment process, enhancing job matching accuracy and providing personalized job recommendations.	Dependency on accurate data input. Limited personalization for niche industries.	Improves job matching speed, enhances candidate-job fit. Reduces manual intervention in recruitment.	Job Match Accuracy: 90%, User Engagement Rate: 80%	Requires a large dataset for effective machine learning. May not work well for less.

2. PROPOSED METHODOLOGY

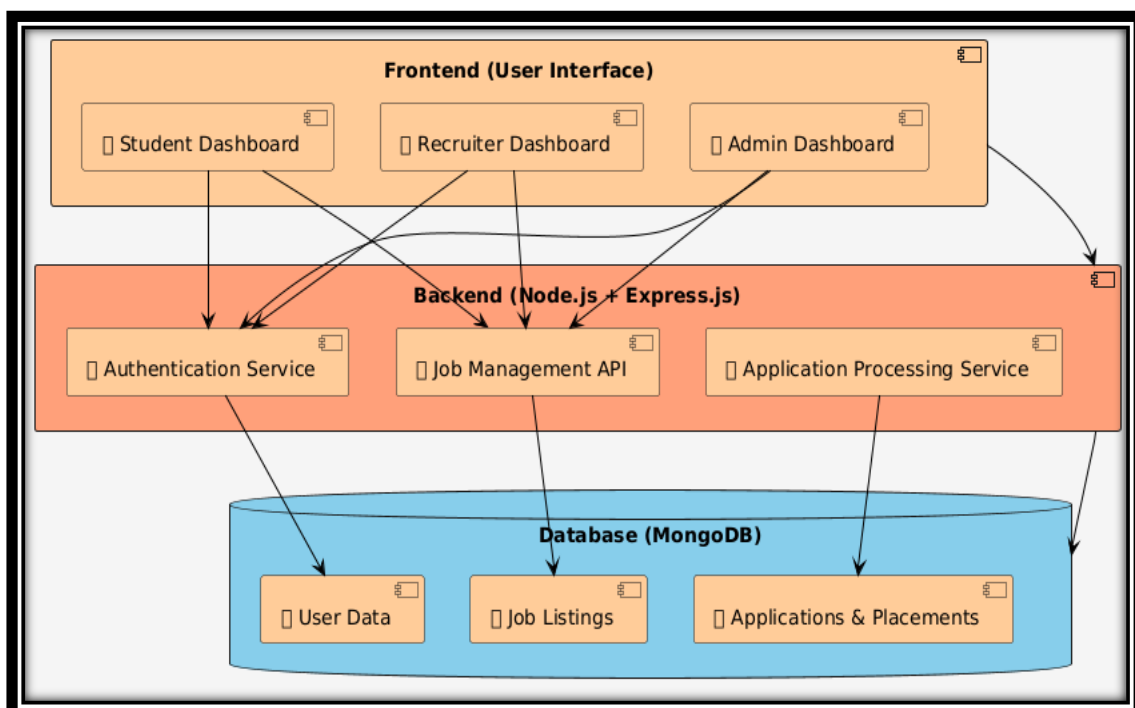


Fig 1: Proposed methodology

The proposed system is structured using a modular three-tier architecture, including:

- Frontend (React.js)
- Backend (Node.js + Express.js)
- Database (MongoDB)

This architecture ensures scalability, maintainability, and clear separation of concerns, making it easier to manage and extend the application.

1. Frontend (User Interface - React.js)

The frontend acts as the interactive layer for users. It provides different dashboards tailored to the roles of the users:

Student Dashboard

- Registration and login for students
- Profile creation and updates
- Browse and apply for job listings
- View application status and placement records
- Notifications for interview calls or results

Recruiter Dashboard

- Login and profile management for recruiters
- Create and manage job postings
- View student applications
- Schedule interviews and update application statuses

Admin Dashboard

- Admin login and authentication
- Manage and verify student and recruiter accounts
- View analytics on applications and placements
- Control over job postings and user access

Note: All dashboards interact with the backend services through RESTful APIs using axios or fetch.

2. Backend (Node.js + Express.js)

The backend is responsible for handling business logic, API endpoints, and data communication with the database.

Authentication Service

- Handles login, signup, and token generation (e.g., JWT)
- Role-based access control (Student, Recruiter, Admin)
- Password encryption and validation

Job Management API

- APIs for recruiters to:
 - Create, edit, or delete job listings
- APIs for students to:
 - Fetch and filter available jobs
- Validates input, checks permissions, and returns appropriate responses

Application Processing Service

- Students submit applications for jobs
- Recruiters can:
 - View applicants for specific job posts
 - Change application status (e.g., Shortlisted, Interview Scheduled, Placed)
- Admin can oversee and intervene in the application pipeline

Middleware: Express middleware used for authentication checks and request validations.

3. Database (MongoDB)- MongoDB stores all critical and dynamic data in schema-based collections. The backend uses **Mongoose** for schema modeling.

User Data Collection

Stores data of:

- Students (name, email, resume, skills, etc.)
- Recruiters (company name, HR contact info)
- Admins (credentials, role permissions)

Job Listings Collection

- Job details:
 - Title, description, qualifications, salary, deadline
 - Associated recruiter and timestamps
- Updated as recruiters post or modify jobs

Applications & Placements Collection

- Records student applications with:
 - Job ID, student ID, status (applied, shortlisted, placed)
 - Interview rounds and outcomes
- Allows tracking of entire placement lifecycle.

Overall Data Flow

1. **Student** logs in → views jobs → applies → application stored in MongoDB
2. **Recruiter** logs in → posts job → reviews applications → updates status
3. **Admin** accesses dashboards → manages users/jobs → oversees entire process
4. Backend ensures secure, validated communication between frontend and database.

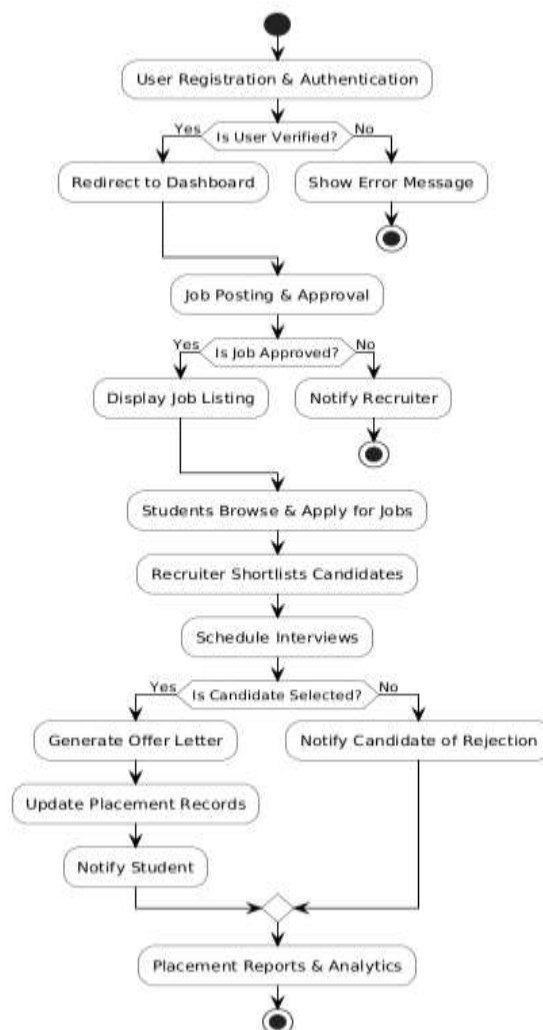


Fig 2: Flowchart of the proposed methodology

This flowchart visually represents the step-by-step process of how users interact with the placement system, from registration to the generation of placement reports.

1. User Registration & Authentication

- Purpose: To verify users (students, recruiters, or admins).
- If the user is verified, they are redirected to their respective dashboard.
- If not verified, the system displays an error message.

2. Job Posting & Approval

- Recruiters can post job listings.
- An approval process (likely by admin) checks whether the job meets required standards.
- If approved:
- The job is displayed to students.
- If rejected:
- The recruiter is notified of the rejection and possibly given a reason.

3. Students Browse & Apply for Jobs

- Students can search, filter, and apply for available job listings.
- Applications are sent to the backend for recruiter review.

4. Recruiter Shortlists Candidates

- Recruiters review the pool of applicants and shortlist those who meet the criteria.
- These candidates move forward to the interview round.

5. Schedule Interviews

- Recruiters/admins can schedule interviews for shortlisted candidates.
- Interview details are communicated via the portal or email notifications.

6. Candidate Selection Process

- After interviews, recruiters determine whether the candidate is selected or rejected.
- If selected:
- An offer letter is generated.
- Placement records are updated in the system.
- The student is notified of the successful placement.
- If not selected:
- The candidate is notified of rejection.

7. Placement Reports & Analytics

- All the data (applications, selections, rejections, job roles, etc.) are used to generate:
- Placement reports
- Recruitment statistics
- Department-wise or student-wise analytics

This data helps the admin and management monitor the performance and success of the placement process.

Technologies Used

Table 2: Technologies used

Layer	Technology	Purpose
Frontend	React.js	User Interface & Components
Backend	Node.js + Express	API and Server Logic
Database	MongoDB	NoSQL Data Storage
Styling	Tailwind CSS	Responsive UI
Auth	JWT	Secure Token-Based Login
State Mgmt	Redux + Persist	Global State & Local Storage

3. RESULTS AND DISCUSSIONS



Fig 3: Main Placement page

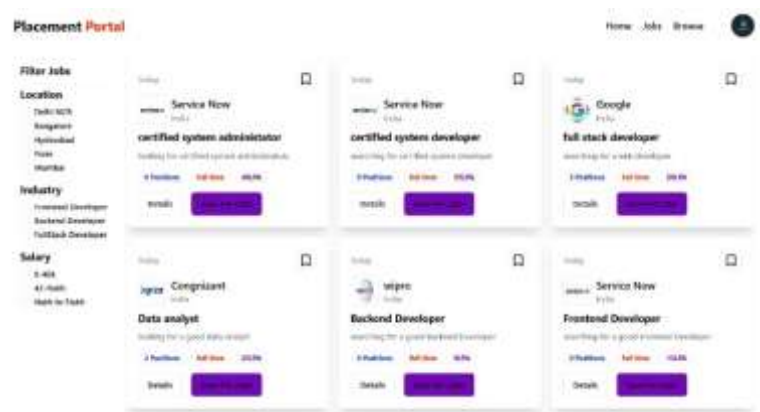


Fig 4: Searching Jobs for student login



Fig 5: List of jobs posted in recruiter login

Sign Up

Full Name

Annan Naidu

Email

annannaidu@gmail.com

Phone Number

1234567890

Password

☐ Student
 ☒ Recruiter

Profile

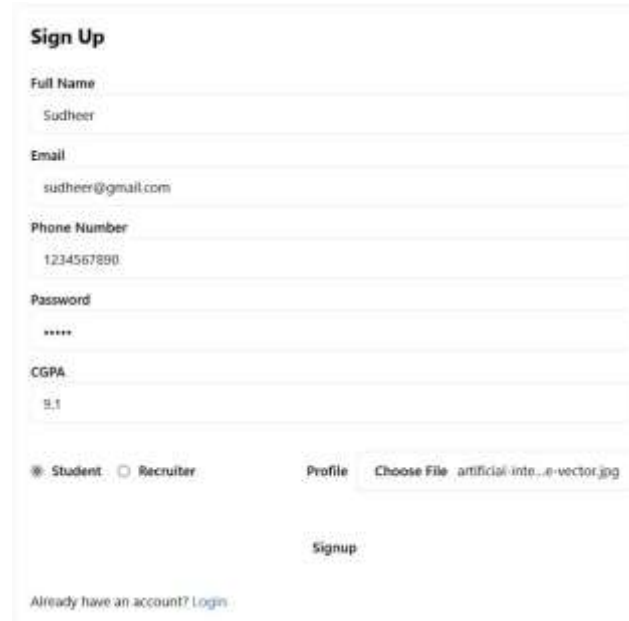
Choose File

artificial-intel...e-vector.jpg

Signup

Already have an account? [Login](#)

Fig 6: Sign up for recruiter login



Sign Up

Full Name
Sudheer

Email
sudheer@gmail.com

Phone Number
1234567890

Password

CGPA
9.1

☒ Student ☐ Recruiter

Profile artificial-intelligence-vector.jpg

Already have an account? [Login](#)

Fig 7: Sign up for Student login

4. CONCLUSION

In this paper, we have presented the design and development of a comprehensive College Portal System built using the MERN Stack—MongoDB, Express.js, React, and Node.js. The proposed system addresses the core functional requirements of a modern academic environment by enabling seamless communication, efficient data management, and real-time access to academic services for students, faculty, and administrators. Through clearly defined modules such as data preprocessing, user authentication, feedback handling, performance tracking, and query resolution, the system ensures smooth and secure operations across different user roles.

5. REFERENCES

- [1] Samonte, M. J. C., Cruz, A. Z. D., Uy, D. A., & Gabriel, E. S. (2024). CanDo: A Web-Based Job Portal for Jobseekers and Freelancers with Disabilities. 2024 4th International Conference on Computer Systems (ICCS), 112–121. IEEE.
- [2] Tan, Z. W. (2022). Computing jobs monitoring dashboard in Malaysia (Doctoral dissertation, UTAR).
- [3] Pinjari, M., De, N., Kokne, R., Siddiqui, A., & Chitre, D. (2019). Online job portal. International Research Journal of Engineering and Technology, 6(4).
- [4] Shimpi, P., Balinge, B., Golait, T., Parthasarathi, S., Arunima, C. J., & Mali, Y. (2024). Job Crafter—The One-Stop Placement Portal. 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), 1–8. IEEE.
- [5] Lakshmi, Y. H., Sangeetha, Y., Vyshnavi, K. P. S., & Fyzulla, S. (2023). A Website for a Consultancy Using MERN Stack. 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), 195–200. IEEE.
- [6] Pavani, V., Pujitha, N. M., Vaishnavi, P. V., Neha, K., & Sahithi, D. S. (2022). Feature Extraction Based Online Job Portal. 2022 International Conference on Electronics and Renewable Systems (ICEARS), 1676–1683. IEEE.
- [7] Srivastava, N., Tripathi, M., & Rai, V. (2023). The Development of a Job Portal to Facilitate In-Campus Placement. 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 1549–1556. IEEE.
- [8] Murugesan, V. S., & Sunny, R. (2024). Evaluation of Online Job Portals for HR Recruitment Selection Using AHP in Two-Wheeler Automotive Industry—A Case Study. International Journal of Engineering and Technology, 15(3), 452–460.
- [9] Kelley, E. M., Ksoll, C., & Magruder, J. (2022). How Do Online Job Portals Affect Employment and Job Search? Evidence from India. Mimeo.
- [10] Cărăușu, M., & Iftene, A. (2024). Career Catalyst: Empowering Job Search Success. Procedia Computer Science, 246, 2519–2528.

-
- [11] Prakarsa, G., Nursyanti, R., & Baransano, E. S. (2024). Analysis of Acceptance Factors of Job Portal Applications for Job Seekers in Indonesia Using TAM. Program Studi Sistem Informasi/ Fakultas Teknologi Dan Informatika.
- [12] Keethana, K., Hussain, G. M. A., Jabeen, A. S., & Shavali, T. K. S. (2024). Online Job Portal Management System. Lords Institute of Engineering and Technology, Hyderabad, Telangana, India.
- [13] Ahmed, M., Meghana, B. L., Durga, N. S., Sri, M. N., & Shabuddin, S. (2024). Online Job Portal. Sri Vasavi Institute of Engineering & Technology (Autonomous), Nandamuru.
- [14] Chowdhury, A. R., Areias, A. C., Imaizumi, S., Nomura, S., & Yamauchi, F. (2024). Reflections of Employers' Gender Preferences in Job Ads in India: An Analysis of Online Job Portal Data. Mimeo.
- [15] Yadav, S., Oza, K., Shah, K., & Patil, S. (2024). AI-Based Job Portal. Universal College of Engineering, Mumbai, India