iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80541

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Design and Implementation of a Role-Based Internship and Job Portal System

Prof. Sujata Tripude[1], Swaraj Koyande[2], Ganesh Kanojiya[3], Ashish Sarkate[4]
[1]*Assistant Professor, Department of Computer Engineering*
[2, 3, 4]*Students, Department of Computer Science and Information Technology Bharat College of Engineering, Badlapur, University of Mumbai*

Abstract: ***In today's competitive employment landscape, students and fresh graduates often face difficulty in finding suitable internship and job opportunities, while companies struggle to manage applicants efficiently. The Internship and Job Portal is proposed as a centralized, role-based web platform designed to simplify the end-to-end hiring process. The system enables students to register, build their profiles, upload resumes, and apply for opportunities. Companies can post job and internship listings, manage applications, and schedule interviews from a unified dashboard. An administrator panel provides complete oversight of users, job listings, and platform activities. The platform is developed using Node.js, Express.js, MySQL, and HTML/CSS/JavaScript, following a RESTful architecture with bcrypt-based authentication and role-based access control. Three distinct user roles are supported: Student, Company, and Admin. Key features include job and internship listings with filters, real-time application status tracking, interview scheduling notifications, company verification, and a responsive multi-theme interface. This system aims to reduce the communication gap between job seekers and recruiters by providing an organized and automated recruitment ecosystem.***

Keywords: Online Recruitment System, Internship Management, Applicant Tracking System (ATS), Role-Based Access Control (RBAC), Web Application, Node.js, MySQL.

I. INTRODUCTION

In recent years, internships and entry-level employment have become essential milestones for students and fresh graduates. However, the existing process of discovering, applying for, and tracking opportunities is highly fragmented. Students typically rely on general-purpose job boards, institutional placement cells, and manual email applications, while companies depend on spreadsheets and disconnected tools to manage recruitment. This lack of integration results in delays, missed candidates, and inefficient communication for all parties involved.

The Internship and Job Portal addresses these challenges by providing a centralized, web-based platform where students can explore relevant opportunities and apply with ease, and where companies can manage their complete recruitment pipeline in one place. The system integrates role-based dashboards, real-time application tracking, interview scheduling, and administrative oversight to create a seamless end-to-end hiring experience.

From a technical standpoint, the system is built using Node.js and Express.js for the backend REST API, MySQL for relational data storage, and HTML/CSS/JavaScript for the frontend. User authentication is secured through bcrypt password hashing combined with token-based session management, ensuring data protection at every level.

A. Existing System

Current recruitment processes rely heavily on manual effort and disconnected tools. Students depend on generic platforms such as Naukri and Indeed, or on college placement cells that do not offer real-time integration with employer data. Job applications are typically submitted via email, and follow-ups are tracked manually. Companies, on the other hand, manage applicant resumes through email inboxes or Excel sheets and coordinate interviews using separate calendar applications. There is no unified platform that handles job posting, application collection, candidate tracking, and interview scheduling in one place.

Limitations of Existing Systems

- Manual Application Tracking: Companies manage applicants through spreadsheets, which is error-prone and time- consuming.
- Fragmented Communication: Interview scheduling requires separate tools such as email and calendar apps, leading to coordination delays.

©IJRASET: All Rights are Reserved

- No Role Separation: General platforms provide the same interface for all users, with no customized experience for students or companies.
- Absence of Verification: Unverified companies can post listings, which reduces applicant trust.
- Limited Search Filters: Most free platforms offer only basic search options with no salary, skill, or deadline-based filtering.

B. Proposed System

The proposed Internship and Job Portal introduces a fully integrated, role-based web platform that eliminates the manual bottlenecks present in traditional recruitment workflows. The system consists of a Node.js/Express.js REST API backend, a MySQL relational database, and a multi-page HTML/CSS/JavaScript frontend, each organized into distinct interfaces for Students, Companies, and Admins.

Students interact with a personalized dashboard to browse filtered job listings, submit applications with resumes, track status updates in real time, and receive interview notifications. Companies access an applicant tracking dashboard to post listings, review candidates, update application statuses, and schedule interviews with meeting links. Admins manage all registered users, verify company accounts, and monitor live platform statistics.

Benefits of the Proposed System:

- Unified Platform: All hiring activities including posting, applying, tracking, and scheduling occur within a single system.
- Role-Based Access Control: Students, Companies, and Admins each have secure and tailored interfaces with appropriate permissions.
- Automated Application Pipeline: Applications progress through structured stages: Applied, Reviewed, Shortlisted, and Hired/Rejected.
- Interview Scheduling: Companies schedule interviews with date, time, and meeting link; students receive instant notifications.
- Company **Verification:** Admin-controlled verification badges increase applicant trust and platform credibility.

II. LITERATURE SURVEY

We studied several existing research papers and real-world recruitment platforms to understand current systems and identify their limitations. The following studies and platforms were examined:

1) Professional networking platforms such as LinkedIn have demonstrated that personalized job recommendations and network-based referrals significantly increase placement rates. The importance of skill-based candidate matching highlighted in this study inspired the recommendation approach considered in our platform.

2) We looked at internship platforms like Internshala that focus on students. These platforms have a feature where you can upload your resume with one click. This makes it really easy and fast to apply for internships. In fact it can cut down the application time by 60 percent. This is why we added an apply feature that supports resumes for students.

3) We also checked out job platforms like Indeed and Naukri.com. These platforms have filters that help people find jobs that are a good fit. They let you filter jobs by things, like location, type of job, salary and skills. We used these filters as an example when we built our filtering system.

4) Many studies on Applicant Tracking Systems like Greenhouse show that a structured application process makes recruiters more efficient and reduces the number of candidates who drop out. Our portals application status system, which includes Applied, Reviewed, Shortlisted and Hired stages was inspired by these findings on ATS.

5) Research on automated scheduling tools, such as Google Meet and Calendly finds that integrated scheduling reduces interview no-shows and saves time on coordination. This research helped us develop our interview scheduling module, which allows users to schedule interviews and captures the date, time and meeting link within our platform.

6) Studies on employer transparency platforms like Glassdoor show that verified company profiles make applicants more trusting and improve engagement rates. This is why we added a company verification badge that can be controlled by admins in our system.

7) REST API design principles, as established in literature confirm that resource-based RESTful APIs are the standard for scalable web applications. By using HTTP methods and status codes we ensure reliable communication between the frontend and backend of our system, which is based on these principles.

©IJRASET: All Rights are Reserved

8) Security guidelines from OWASP emphasize the importance of password hashing and secure session management to protect user credentials. Following these guidelines we use bcrypt with 10 salt rounds and token-based session handling in our platform to ensure security.

9) Research on relational database management systems shows that normalized schemas with key constraints improve data integrity in applications with multiple entities. This research directly informed our MySQL database design, which connects users, jobs, applications and interviews through relational tables.

10) Software engineering literature on the Model-View-Controller pattern demonstrates its effectiveness in building modular and maintainable web applications. Our project follows this methodology by separating the HTML/CSS frontend, The Node.js API layer and the MySQL data layer, in an organized way.

III. METHODOLOGY

The development of the Internship and Job Portal follows a Full Stack MVC (Model-View-Controller) architectural approach, emphasizing modularity, security, and role-based access control. The project separates concerns across three core layers: the Presentation Layer (Frontend), the Application Logic Layer (Backend API), and the Data Services Layer (MySQL Database). The overall development followed a structured lifecycle from requirement analysis to final integration testing.

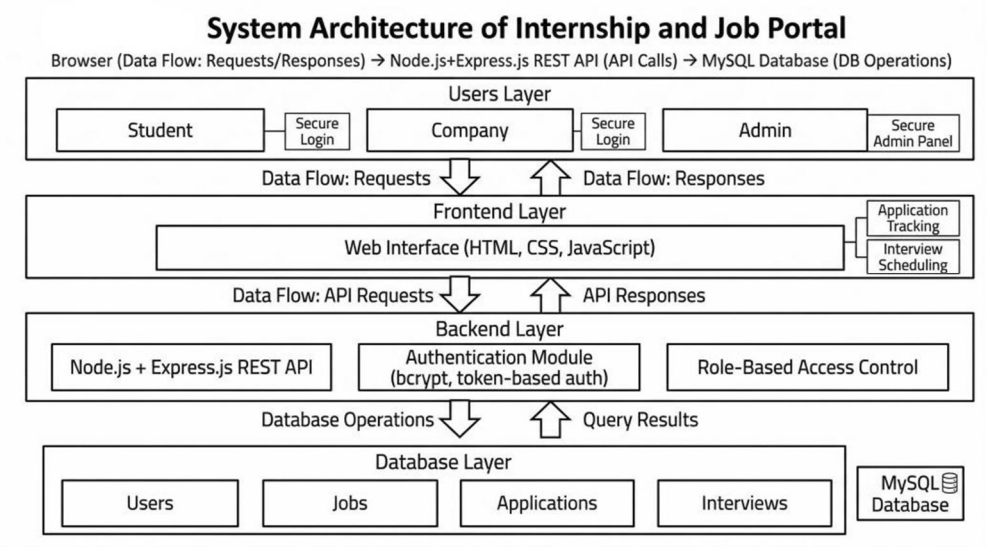


Fig. 2. System Architecture of the Internship and Job Portal

A. Backend - REST API (Node.js + Express.js)

The backend is implemented as a single server.js module that exposes all API endpoints following standard REST conventions using HTTP methods GET, POST, PUT, and DELETE. An authentication of middleware validates bearer tokens on every protected route before processing requests. Role-based authorization checks prevent cross-role data access, ensuring that company users cannot access student endpoints and vice versa. The bcryptjs library is used for password hashing with 10 salt rounds, so no passwords are ever stored in plain text. A shared api.js helper on the frontend side manages all HTTP requests, token storage, and automatic session expiry redirects.

B. Database - MySQL (Relational Schema)

The database consists of four core tables: users, jobs, applications, and interviews, connected through foreign keys with ON DELETE CASCADE behavior. The users table serves both Students and Companies, differentiated by a role column. A UNIQUE KEY constraint on the combination of job_id and student_email in the applications table prevents duplicate submissions. The server automatically creates all required tables on first startup using CREATE TABLE IF NOT EXISTS statements, eliminating the need for manual database configuration. MySQL connection pooling with a limit of 10 connections handles concurrent user requests efficiently.

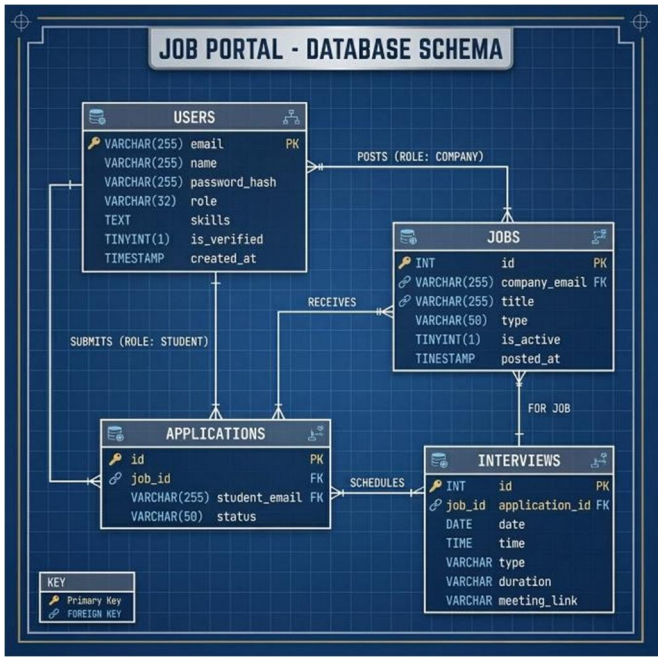


Fig. 3. ER Diagram of the Database Schema

C. Frontend - Multi-Page HTML/CSS/JavaScript

The frontend is structured into separate page directories for Home, Login, Registration, Student, Company, and Admin interfaces, each containing its own HTML, CSS, and JavaScript files. A shared api.js file is included on every page to provide consistent API communication, token handling, and authentication state management. The UI supports multiple CSS themes including Void Dark, Paper Light, Neon Jungle, and Molten, implemented using CSS custom properties for one-click switching. Skeleton loaders are displayed during API calls to provide a smooth user experience while data is being fetched.

D. Development Lifecycle

1) Requirement Analysis: User roles (Student, Company, Admin) and their functional requirements were identified.
2) Database Design: A normalized schema with foreign key constraints was designed to ensure data integrity.
3) API Development: All REST endpoints were implemented and individually tested in server.js.
4) Frontend Development: Role-specific pages were built using HTML/CSS/JS, consuming the REST API.
5) Integration Testing: End-to-end flows were verified: Register → Post Job → Apply → Schedule Interview →
6) Admin Verify.
7) UI Refinement**:** Themes, skeleton loaders, animations, and responsive styling were applied for a polished user experience.

IV. RESULTS AND DISCUSSION

The Internship and Job Portal was tested across all three user roles to evaluate its functionality, performance, and usability. The system demonstrated reliable behavior across the complete recruitment workflow from registration to interview scheduling.

Student users were able to successfully register, build profiles, and apply for listings using the one-click apply feature. Application statuses updated correctly as companies reviewed and shortlisted candidates. Interview notifications were received in real time once companies scheduled sessions.

Company users could post job and internship listings with full control over deadlines, eligibility, and job type. The ATS-style dashboard allowed companies to review applicants by position, move them through pipeline stages, and schedule interviews with associated meeting links. The admin verification badge feature was also tested; only verified companies appeared with the trust badge visible to student users.

Admin functionality was verified to provide complete oversight of all registered users and listings, with the ability to verify or deactivate accounts and monitor platform statistics.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

In terms of performance, the server handled multiple concurrent API requests without significant delay during local testing, owing to MySQL connection pooling. Page load times were reduced through the use of skeleton loaders, which improved perceived performance during data fetching. The bcrypt authentication mechanism added approximately 100–150 milliseconds to login operations due to hashing, which is within acceptable limits for security purposes.

The multi-theme interface was tested across desktop and mobile screen sizes, confirming responsive behavior. Users were able to switch between themes (Void Dark, Paper Light, Neon Jungle, Molten) without page reloads, using CSS custom properties.

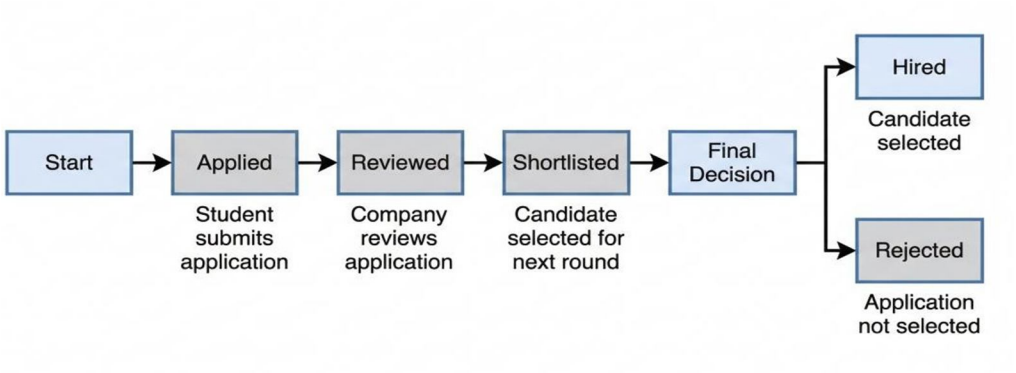


Fig. 4. Application Pipeline: Applied → Reviewed → Shortlisted → Hired/Rejected

V. ADVANTAGES

1) Unified Hiring Ecosystem: All recruitment activities including posting, applying, tracking, and scheduling take place within a single platform, eliminating tool fragmentation.
2) Role-Based Security: Tailored interfaces for Students, Companies, and Admins ensure that each user accesses only their relevant features and data.
3) Automated Pipeline Management: A structured ATS workflow replaces manual tracking, moving candidates through defined stages automatically.
4) Trust and Verification: Admin-controlled company verification badges improve platform credibility and applicant confidence.
5) Cost-Efficiency: The system is built entirely with open-source technologies including Node.js and MySQL, making it a self-hosted alternative to expensive commercial ATS tools.
6) Data Security: User credentials are protected through bcrypt password hashing with 10 salt rounds and token-based session management.

VI. FUTURE SCOPE

The Internship and Job Portal presents several opportunities for future enhancement. Integration of Artificial Intelligence (AI) and Machine Learning (ML) algorithms could enable personalized job recommendations based on a student's skills, preferences, and past application history, significantly improving matching accuracy. Additionally, the application of Natural Language Processing (NLP) techniques for automated resume parsing and skill extraction would further streamline profile creation and recruiter shortlisting. A real-time messaging or chat feature between students and recruiters could reduce communication delays during the hiring process. Extending the platform to a mobile application for Android and iOS would increase accessibility and enable users to track applications and receive notifications on the go. Cloud-based storage integration would support secure resume uploads, document verification, and data backup. From an administrative perspective, analytics dashboards displaying hiring trends, user engagement metrics, and platform performance data would support better decision-making. Features such as video interview integration, online assessments, and coding challenges could eventually make the platform a comprehensive end-to-end recruitment solution.

VII. CONCLUSION

This paper presented the design and implementation of an Internship and Job Portal, a centralized, role-based web platform aimed at simplifying the recruitment process for students, companies, and administrators. Traditional hiring workflows involve significant manual effort through email-based applications, spreadsheet tracking, and disconnected scheduling tools, creating friction for all stakeholders. The proposed system addresses these challenges by integrating every stage of the recruitment lifecycle into a single automated platform.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

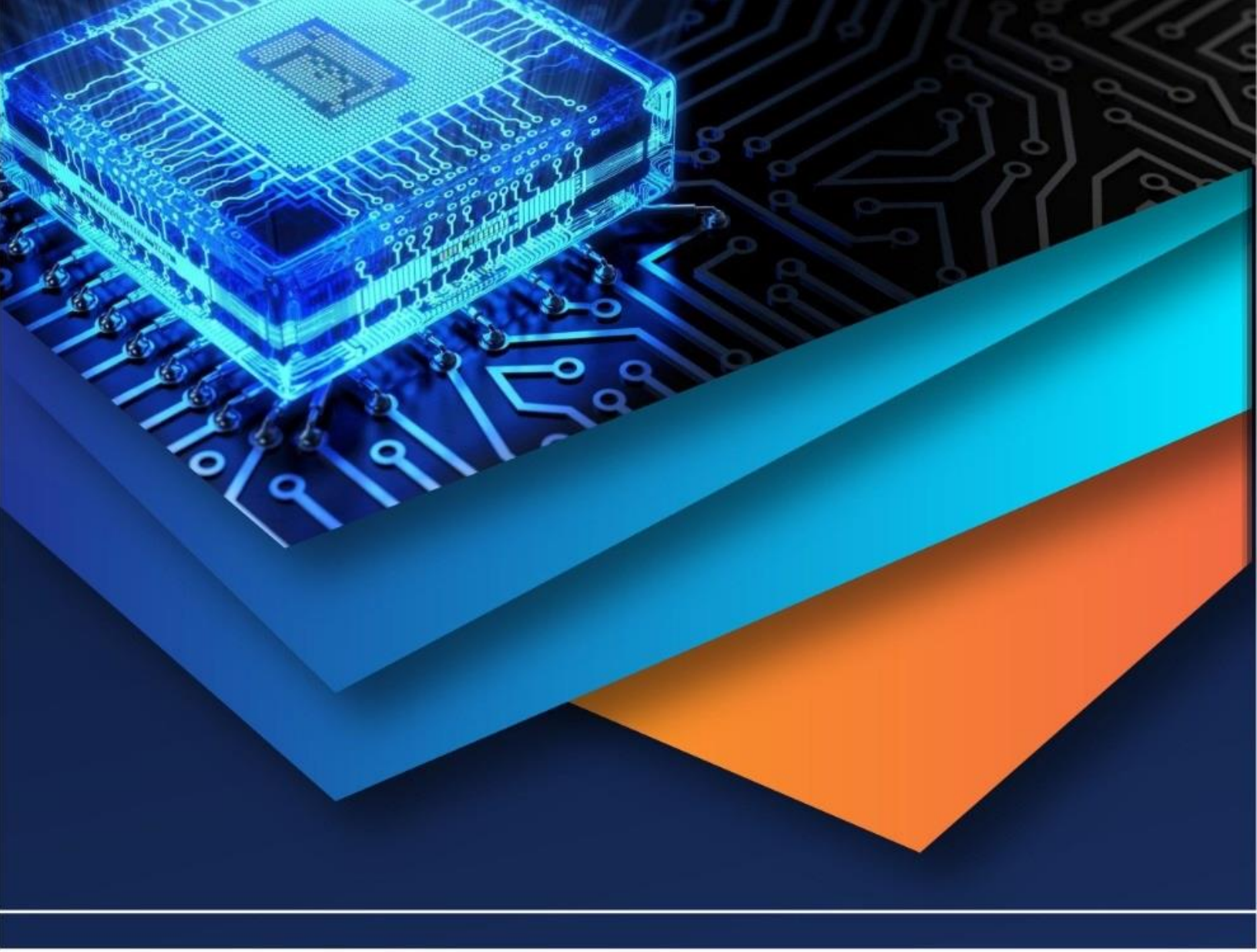
Students benefit from personalized dashboards, one-click applications, real-time status tracking, and interview notifications. Companies gain structured applicant management through an ATS-style pipeline and company verification for enhanced trust. Admins maintain complete oversight through user management and platform statistics. The system was developed using Node.js, Express.js, MySQL, and standard web technologies, following REST API conventions and MVC architecture principles.

Testing confirmed that the system performs reliably across all user roles and devices. The project demonstrates how thoughtfully designed web technology can bridge the gap between emerging talent and employment opportunity. Future work may extend the platform through AI-based recommendations, mobile application development, and advanced analytics capabilities.

REFERENCES

[1] LinkedIn Engineering Blog. (2023). Scaling personalized job recommendations using graph-based models. Retrieved from https://engineering.linkedin.com
[2] Internshala. (2023). Platform Report: Student Internship Trends in India 2023. Retrieved from https://internshala.com
[3] Naukri.com Research Division. (2022). Impact of Advanced Search Filters on Job Discovery Quality. Retrieved from https://www.naukri.com
[4] Greenhouse Software. (2022). Applicant Tracking System Design Patterns and Recruiter Efficiency. Retrieved from https://www.greenhouse.io
[5] Google Workspace. (2023). Automated Scheduling and Calendar Integration with Google Meet. Retrieved from https://workspace.google.com
[6] Glassdoor Research Team. (2023). Company Transparency and Its Effect on Application Rates. Retrieved from https://www.glassdoor.com
[7] Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. Doctoral Dissertation, University of California, Irvine.
[8] OWASP Foundation. (2023). Authentication Cheat Sheet — OWASP Cheat Sheet Series. Retrieved from https://cheatsheetseries.owasp.org
[9] MySQL Documentation. (2024). MySQL 8.0 Reference Manual. Oracle Corporation. Retrieved from https://dev.mysql.com/doc/
[10] Freeman, E., & Robson, E. (2020). Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software. O'Reilly Media.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)