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.2026.81272

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

A Study on Internship and Job

Prof. Namrata Bhalerao[1], Hemant Naik[2], Shani Bind[3], Avnish Mistra[4], Digambar Lokhande[5]
[1]*Assistant Professor,* [2, 3, 4, 5]*Students, Department of Computer Engineering, Bharat College of Engineering, Badlapur, University of Mumbai*

Abstract: ***In today's competitive job market, students and fresh graduates often find it difficult to secure suitable internships and job opportunities. At the same time, companies struggle to manage large numbers of applicants efficiently using traditional methods. To address this issue, the Internship and Job Portal is designed as a centralized, role-based web platform that simplifies the recruitment process. The system allows students to create profiles, upload resumes, and apply for opportunities in a convenient way. Companies can post job or internship listings, review applications, and manage candidates through a structured dashboard. An administrator panel provides full control over users, job postings, and overall platform activities. The platform is developed using Node.js, Express.js, MySQL, and standard web technologies like HTML, CSS, and JavaScript. It follows a RESTful architecture and implements secure authentication using bcrypt along with role-based access control. The system supports three main user roles: Student, Company, and Admin.***
Key features include advanced job filtering, real-time application status tracking, company verification, and a responsive multi-theme user interface. The main goal of this system is to reduce the gap between job seekers and recruiters by providing a more organized, efficient, and reliable recruitment platform.
Keywords: Online Recruitment System, Internship Management, Applicant Tracking System (ATS), Role-Based Access Control (RBAC), Web Application, Node.js, MySQL

I. INTRODUCTION

Internships and entry-level jobs play a crucial role in shaping the careers of students and fresh graduates. However, the current process of finding and applying for these opportunities is often scattered and inefficient. Students typically rely on multiple job portals, college placement cells, or manual email applications. On the other hand, companies use spreadsheets, emails, and separate tools to manage applicants. This lack of a unified system leads to delays, missed opportunities, and poor communication between candidates and recruiters. The Internship and Job Portal aims to solve these problems by providing a centralized web-based platform. Students can easily explore opportunities and apply for them, while companies can manage applications in a structured and organized manner. The platform includes role-based dashboards, real-time application tracking, and administrative control, making the entire process smoother and more efficient.
From a technical perspective, the system uses Node.js and Express.js for backend development, MySQL for database management, and HTML/CSS/JavaScript for the frontend. Security is ensured through password hashing and token-based session management.

A. Existing System

The current recruitment process heavily depends on manual methods and disconnected tools. Students use general platforms like Naukri and Indeed or rely on college placement systems that lack real-time integration. Applications are often submitted via email, and tracking them becomes difficult. Similarly, companies manage resumes through email inboxes or Excel sheets, which is inefficient and prone to errors. There is no single system that combines job posting, application management, and candidate tracking in one place.

B. 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.
- 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.

©IJRASET: All Rights are Reserved

C. Proposed System

The proposed system introduces a fully integrated, role-based platform that removes the inefficiencies of traditional recruitment methods. It consists of a backend API built with Node.js and Express.js, a MySQL database, and a multi-page frontend built using HTML, CSS, and JavaScript.

Students can browse job listings, apply with their resumes, and track their application status in real time. Companies can post listings, review applications, and manage candidates through an organized dashboard. Admins have full control over users, verify company accounts, and monitor system activity.

D. Benefits of the Proposed System

- Unified Platform: All recruitment activities are handled in one system
- Role-Based Access: Separate dashboards for Students, Companies, and Admins
- Structured Application Pipeline: Applications move through stages like Applied, Reviewed, Shortlisted, and Final Decision
- Company Verification: Improves trust and authenticity of job listings

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:

To design this system effectively, we studied various research papers and existing platforms.

Professional platforms like LinkedIn show how personalized job recommendations improve hiring success. Internship platforms such as Internshala demonstrate the importance of quick and simple application processes.

Job portals like Indeed and Naukri provide advanced filters that help users find relevant opportunities. These ideas were incorporated into our filtering system.

Applicant Tracking Systems like Greenhouse highlight how structured workflows improve recruiter efficiency. This inspired our application status pipeline.

Security practices recommended by OWASP guided the implementation of password hashing and session management. Database design principles and MVC architecture were also studied to ensure scalability and maintainability.

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.

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.

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.

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.

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.

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

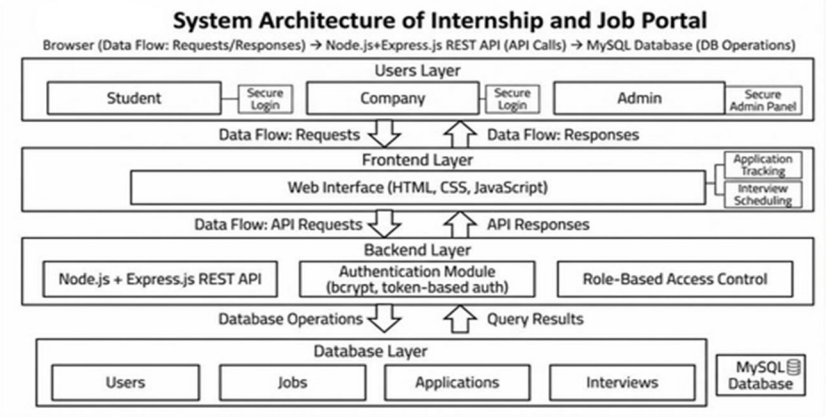


Fig. 2. System Architecture of the Internship and Job Portal

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

The backend is implemented using REST APIs with standard HTTP methods. Authentication middleware ensures that only authorized users can access protected routes. Role-based access control prevents unauthorized data access.. 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 includes four main tables: users, jobs, applications, and interviews (used structurally but not actively for scheduling). These tables are connected using foreign keys to maintain data integrity.

Constraints are used to prevent duplicate applications. The system automatically creates required tables, reducing setup complexity. Connection pooling ensures efficient handling of multiple users.

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.

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

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

IV. RESULTS AND DISCUSSION

The system was tested for all three user roles and performed reliably across different scenarios.

Students were able to register, create profiles, and apply for jobs easily. Application statuses updated correctly as companies reviewed candidates.

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

Companies successfully posted job listings and managed applicants through the dashboard. The verification system ensured only trusted companies were highlighted.

Admins were able to control users, verify companies, and monitor platform activity effectively.

The system handled multiple requests efficiently due to connection pooling. UI performance was improved using skeleton loaders, and authentication remained secure with minimal delay.

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.

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.

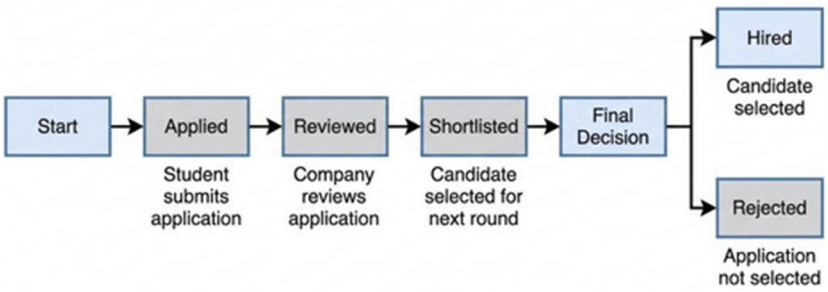


Fig. 4. Application Pipeline: Applied → Reviewed → Shortlisted → Hired/Rejected

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 system can be further improved by adding AI-based job recommendations, resume parsing using NLP, and mobile app support. Other enhancements could include chat systems, analytics dashboards, and advanced recruitment tools like coding tests or assessments. 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.

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

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

The Internship and Job Portal successfully demonstrates how a centralized system can simplify the recruitment process for students and companies. Traditional methods rely heavily on manual work and disconnected tools, which create inefficiencies.

This platform provides a structured and user-friendly approach, allowing students to apply easily and companies to manage candidates effectively. Admin controls ensure platform reliability and trust.

Overall, the project shows how modern web technologies can improve the hiring process and create better opportunities for both job seekers and recruiters.

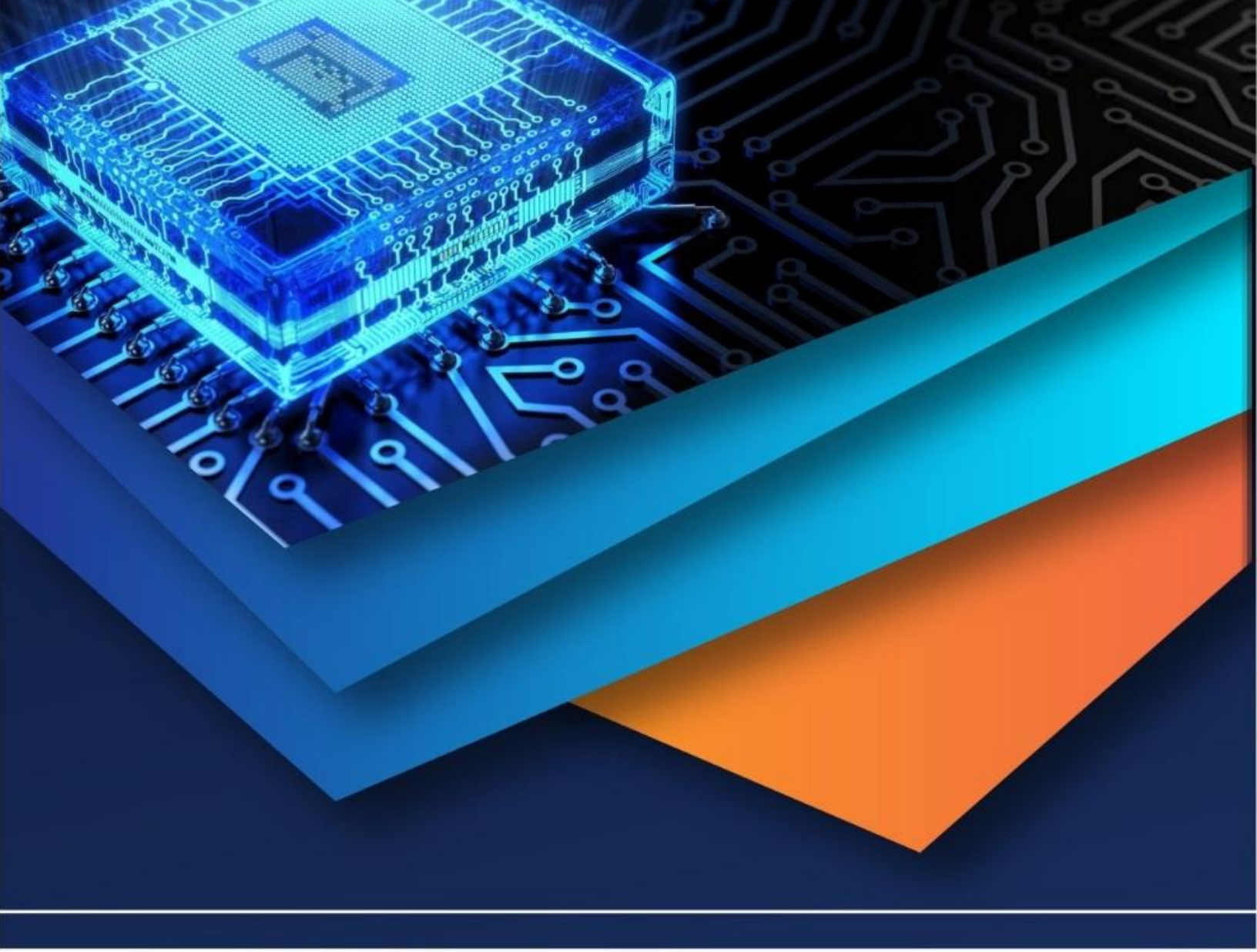
Students benefit from personalized dashboards, one-click applications, real-time status tracking, and interview notifications. Companies gain structured applicant management through an ATS-style pipeline and company verification for enhanced trust. Admins maintain complete oversight through user management 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] Glassdoor Research Team. (2023). Company Transparency and Its Effect on Application Rates. Retrieved from https://www.glassdoor.com
[6] Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. Doctoral Dissertation, University of California, Irvine.
[7] OWASP Foundation. (2023). Authentication Cheat Sheet — OWASP Cheat Sheet Series. Retrieved from https://cheatsheetseries.owasp.org
[8] MySQL Documentation. (2024). MySQL 8.0 Reference Manual. Oracle Corporation. Retrieved from https://dev.mysql.com/doc/
[9] 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)