

# Job Portal Web Application: A Secure Role-Based Recruitment Platform for Connecting Talent and Opportunities

Dr.rajendra Singh HOD, Assistant professor Mr.ravi Kumar, Sunkara Vidya Sagar Satya Vara Prasad  
Department of Computer Science and Engineering  
Raffles University, Neemrana

**Abstract-** With the growing demand for digital transformation in recruitment, traditional methods like email-based job applications and manual screening have become inefficient. This paper proposes a secure, scalable, and role-based Job Portal Web Application that connects job seekers with employers via a modern, full-stack architecture. The system provides separate dashboards for Admin, Employer, and Job Seeker roles, enabling smooth user onboarding, job posting, resume uploads, and application tracking. Technologies used include React.js, Node.js, Express.js, MongoDB, and JWT authentication, with Cloudinary integration for secure file handling. The platform emphasizes data validation, role-based access control, and responsive UI design. Results show efficient user interaction and robust job management. The application is successfully deployed using Render and Vercel for real-time access and scalability.

**Keywords:** Job Portal, MERN Stack, JWT, Cloudinary, Role-Based Access, Resume Upload, Recruitment System, Web Application.

## INTRODUCTION

In today's competitive market, an efficient, automated, and secure job portal system is critical for simplifying the hiring process. This paper introduces a full-stack job portal web application that allows job seekers to apply and employers to post and manage job listings. It reduces the reliance on traditional methods and enables real-time, role-specific functionality.

## II. RELATED WORK

Various recruitment systems like LinkedIn, Indeed, and Naukri offer job listing and application functionalities. However, many are commercial, limited in functionality for startups, or lack role-specific access control. This project draws inspiration from such systems while improving on real-time resume tracking and secure file handling.

## III. SYSTEM ARCHITECTURE AND METHODOLOGY

The application follows a modular MERN architecture:

- Frontend: React.js + Tailwind CSS
- Backend: Node.js + Express.js
- Database: MongoDB with Mongoose ODM
- Authentication: JWT
- Cloud Storage: Cloudinary

Each user role—Admin, Employer, and Job Seeker—has a separate dashboard and protected API access. Middleware is used for role validation and token checking.

## IV. IMPLEMENTATION

**The system includes:**

- Job creation/edit/delete by employers
- Resume upload by job seekers
- Application management by employers
- Admin dashboard for user moderation

Mongoose schemas manage data integrity, while secure API routes enforce access restrictions. Screens are responsive, and real-time feedback is provided through toast notifications.

## V. RESULTS AND ANALYSIS

The application was tested across all user flows:

- Authentication worked consistently with JWT
- Resume files were uploaded securely to Cloudinary
- Employers could see applicant details and download resumes
- Admins were able to delete jobs/users successfully

Page responsiveness and performance were satisfactory across desktop and mobile browsers.

## VI. CONCLUSION AND FUTURE WORK

The job portal successfully demonstrates a secure, scalable, and user-role-specific hiring platform. Future enhancements may include:

- AI-based job recommendations
- Real-time messaging
- Mobile application integration
- Multilingual support and push notifications.

## REFERENCES

1. React.js Documentation, <https://reactjs.org>
2. Node.js Official Docs, <https://nodejs.org>
3. Express.js Docs, <https://expressjs.com>
4. MongoDB Documentation, <https://mongodb.com>
5. Cloudinary Docs, <https://cloudinary.com>
6. JWT.io – <https://jwt.io>
7. Render.com – <https://render.com>
8. Vercel – <https://vercel.com/docs>