Project Report

# Job Portal Website

Team Member: Jay Prakash

# 1. Introduction

The Job Portal Website is a web-based platform developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) that enables efficient interaction between recruiters and job-seeking students. The application allows company administrators to register their organizations, post job vacancies, and manage job applications. Students can browse jobs, apply for them, and track their application status in real-time.  
  
The portal streamlines the hiring process by providing a centralized platform where recruiters can view applicants’ resumes and update their application status as Pending, Accepted, or Rejected.

# 2. Objectives

- To create a responsive and user-friendly job portal using MERN stack technologies.  
- To facilitate job postings and applications in an organized manner.  
- To provide recruiters with tools to manage applicants efficiently.  
- To provide students with an easy way to apply for jobs and track their application status.

# 3. Features

For Recruiters (Admins)  
- Company Registration: Admins can create a company profile.  
- Job Posting: Ability to create and manage job listings.  
- Application Management: View applications for each job.  
- Status Update: Update the application status (Pending, Accepted, Rejected).  
- Resume View: Access the applicant’s resume directly.  
  
For Students  
- User Registration & Login: Create an account and manage profile details.  
- Job Search & Apply: Browse jobs and submit applications.  
- Application Tracking: Check current status of applications in real time.

# 4. Technology Stack

- Frontend: React.js (with React Router, Axios, Context API for state management)  
- Backend: Node.js with Express.js  
- Database: MongoDB (Mongoose for schema management)  
- Authentication: JWT (JSON Web Tokens) and bcrypt for secure login.  
- File Upload: Multer for resume uploads.  
- Styling: CSS / Bootstrap / Tailwind (depending on your choice)  
- Hosting: (Specify if deployed, e.g., Heroku for backend, Netlify/Vercel for frontend)

# 5. System Architecture

1. Frontend (React): Renders UI, manages routing, sends API requests to backend.  
2. Backend (Express & Node): Handles API endpoints, authentication, and application logic.  
3. Database (MongoDB): Stores user details, job postings, and applications.  
4. Resume Storage: Stores resumes either in server storage or a cloud storage service.  
  
Flow:  
- Student/Admin logs in → Backend verifies credentials → Depending on role, relevant dashboard is displayed.  
- Admin posts a job → Stored in MongoDB → Students can view and apply.  
- Student applies for a job → Application stored with default status Pending.  
- Admin updates application status → Student sees status changes in their dashboard.

# 6. Modules

1. Authentication Module: Secure login for students and admins.  
2. Job Management Module: CRUD operations for job postings.  
3. Application Management Module: Handles submission, storage, and status updates.  
4. Resume Module: Upload and retrieval of resumes.  
5. User Dashboard Module: Displays jobs, applications, and statuses.

# 7. Application Status Workflow

- Pending: Default status when a student applies.  
- Accepted: Recruiter has approved the candidate for further process.  
- Rejected: Recruiter has declined the application.

# 8. Screenshots (Add images of your UI here)

- Homepage  
- Recruiter Dashboard  
- Student Dashboard  
- Job Listing Page  
- Application Details Page

# 9. Future Enhancements

- Email notifications for status updates.  
- Advanced search and filter for jobs.  
- Resume parsing and keyword matching.  
- Interview scheduling system.

# 10. Conclusion

The MERN Stack Job Portal successfully demonstrates a full-stack application where different user roles (recruiters and students) interact with a shared system. It offers a clean, functional, and scalable approach to bridging the gap between employers and job seekers.