Job Portal App - Full MERN Stack (with Email Verification)

1. Overview

This document outlines the complete project scope, architecture, and flow for a full-stack Job Portal Application using the MERN stack. The project includes three user roles — Job Seeker, Recruiter, and Admin. It supports email verification during registration, JWT authentication, an admin dashboard using EJS, and a React.js frontend for users and recruiters.

2. Tech Stack

1 Backend: Node.js + Express.js
2 Database: MongoDB + Mongoose
3 Frontend: React.js (Vite or CRA)
4 Admin Dashboard: EJS
5 Authentication: JWT + bcrypt + Email Verification (Nodemailer)
6 File Upload: Multer + Cloudinary
7 Deployment: Render / Vercel / MongoDB Atlas

3. Core Features

1 User Registration with Email Verification
2 JWT-based Authentication
3 Job CRUD Operations (Post, Edit, Delete, View)
4 Job Application System with Resume Upload
5 Recruiter Dashboard (React)
6 Admin Dashboard (EJS) for managing users, jobs, and applications
7 Protected Routes for all user roles

4. Roles and Functionalities

1 Job Seeker: Register, verify email, login, browse jobs, apply to jobs, track applications.
2 Recruiter: Register, verify email, login, post jobs, manage listings, review applications.
3 Admin: Login via EJS dashboard, manage users, jobs, and applications, view analytics.

5. Email Verification Flow

1 User registers → account created with isVerified=false.
2 Backend generates a verification token and emails the user a verification link.
3 User clicks link → verification route validates token → updates isVerified=true.
4 Only verified users can login and access the platform.

6. API Endpoints Summary

1 POST /api/auth/register – Register new user with verification email
2 GET /api/auth/verify/:id/:token – Verify email link
3 POST /api/auth/login – Login (only for verified users)

4 GET /api/jobs – List all jobs with filters

5 POST /api/jobs – Create new job (Recruiter only)

6 POST /api/apply/:jobId – Apply to a job (Seeker)

7 GET /admin/dashboard – Admin EJS dashboard

7. Database Models (Simplified)

1 User: name, email, password, role, resume, company, isVerified

2 Job: title, description, skills, salaryRange, location, recruiter, createdAt

3 Application: job, applicant, resume, status, appliedAt

4 Token: userId, token, createdAt (expires in 1 hour)

8. Project Development Roadmap

1 1. Backend setup (Express + MongoDB + MVC structure)

2 2. Add User & Token models (for email verification)

3 3. Implement Registration + Verification flow

4 4. Implement Login + JWT authentication

5 5. Add Job and Application APIs

6 6. Build Admin Dashboard using EJS

7 7. Develop React frontend (seeker/recruiter portals)

8 8. Integrate APIs and authentication flow

9 9. Test and deploy on Render/Vercel