Technical Specification Document (TSD)

# Project Name

JobGenie – AI-Powered Resume Analyzer & Job Matcher

# 1. Introduction

1.1 Objective:  
To build a free, full-stack AI-powered platform where users can upload resumes (PDF/DOCX), extract skills/experience, and receive job recommendations.  
  
1.2 Target Users:  
- Job Seekers: Get job matches based on AI analysis of their resumes.  
- Recruiters/Admins: Manage job postings and view user analytics.

# 2. Architecture Overview

[Client: React + MUI] ←→ [Server: Node.js + Express] ←→ [MongoDB]  
 ↓  
 [MinIO]  
 ↓  
 [AI Microservice: FastAPI (Python)]

# 3. Tech Stack

Frontend: React.js, MUI  
Backend: Node.js, Express (No TypeScript)  
Storage: MinIO (S3-compatible)  
AI: FastAPI, spaCy, scikit-learn  
Database: MongoDB  
Deployment: Docker, GitHub Actions  
Orchestration (Optional): Docker Compose or k3s  
Hosting: Render, Railway, Fly.io, or Azure free tier

# 4. Key Features

User:  
- Sign up / Log in  
- Resume upload  
- View parsed resume  
- View job matches  
- Apply to jobs  
  
Admin:  
- Manage job listings  
- View uploads, analytics  
  
AI Engine:  
- Resume parsing with spaCy  
- Skill extraction  
- Resume-job similarity using sentence embeddings

# 5. Folder Structure

jobgenie/  
├── client/ (React)  
├── server/ (Node.js)  
├── ai-service/ (FastAPI)  
├── minio/ (MinIO Docker setup)  
└── docker-compose.yml

# 6. MongoDB Schema

User Collection:  
{ name, email, resume\_url, parsed\_data: { skills, experience }, matches: [{jobId, score}] }  
  
Job Collection:  
{ title, company, description, skills\_required, location }

# 7. DevOps Setup

CI/CD: GitHub Actions  
Deployment: Docker Compose / Render  
Storage: MinIO  
Reverse Proxy: NGINX  
Logging: Console/File Logs

# 8. APIs

POST /api/upload: Upload resume  
GET /api/matches/:userId: Get matches  
POST /api/admin/jobs: Create job  
GET /api/jobs: List jobs  
POST /api/ai/parse: Parse and match resume

# 9. Resume Parsing Flow

1. Upload resume  
2. Save to MinIO  
3. Call FastAPI service  
4. Parse + match with jobs  
5. Save result to MongoDB

# 10. Future Enhancements

- LinkedIn import  
- PDF download  
- Admin analytics  
- GPT resume rewrite suggestions