MMIL Recruitment Website — Backend Documentation

Purpose of Backend

1. The backend system acts as an Applicant Tracking System (ATS) for the MMIL Society Recruitment Portal.
2. It manages candidate accounts, tracks recruitment rounds, stores technical submissions, resumes, and GitHub links, and allows admins to shortlist candidates.

Recruitment Flow Logic

1. Aptitude Round: Conducted on an external platform (Unstop). Admin updates result in system.
2. Technical Round: Candidates submit project, GitHub, and resume on the website.
3. HR Round: Conducted offline. Admin updates final decision.

Backend Folder Structure

1. config/ - Database configuration
2. controllers/ - Route logic
3. middleware/ - Authentication and file upload middlewares
4. models/ - MongoDB schemas
5. routes/ - API route definitions
6. uploads/ - Stores resumes
7. utils/ - Utility functions
8. server.js - Entry point

Database Design - User Model Fields

1. name, email, password, role
2. githubLink, resume
3. aptitudeStatus, technicalStatus, hrStatus

Database Design - Technical Submission Fields

1. userId, projectTitle, projectDescription
2. projectLink, githubRepo, submittedAt

Authentication APIs

1 POST /api/auth/register

2 POST /api/auth/login

Candidate APIs

1 GET /api/user/profile

2 PUT /api/user/profile

Technical Round API

1 POST /api/round/technical

Admin APIs

1 PUT /api/admin/aptitude/:id

2 PUT /api/admin/technical/:id

3 PUT /api/admin/hr/:id

System Workflow

1 1. Candidate registers

2 2. Candidate logs in

3 3. Candidate uploads resume and GitHub

4 4. Candidate attempts aptitude

5 5. Admin updates aptitude result

6 6. Qualified candidates submit technical project

7 7. Admin shortlists for HR

8 8. HR interview offline

9 9. Admin marks final result

How to Run Backend

1 npm install

2 npm run dev

3 Environment variables required: PORT, MONGO_URI, JWT_SECRET