

4. Tech Stack & Architecture Planning

FRONTEND

- React.js + TailwindCSS (UI Layer)
- Next.js (SSR and routing)
- Zustand or Redux (State management)

BACKEND

- Node.js + Express (REST APIs)
- Optional: NestJS (for scalable architecture)
- RESTful API or GraphQL (to be finalized)

DATABASE

- PostgreSQL (Relational)
- Redis (Cache layer for fast lookups)
- MongoDB (For unstructured job post data - optional hybrid)

FILE STORAGE & HOSTING

- AWS S3 (File uploads like resumes)
- Vercel or Netlify (Frontend deployment)
- AWS/GCP (Backend hosting)
- Cloudflare (CDN + Security layer)

API GATEWAY & SERVICES

- API Gateway for routing & throttling
- Microservices approach (jobs, users, feed, metrics, analytics)

FOLDER STRUCTURE (Frontend Example)

- /components
- /pages
- /services (API clients)
- /store (Zustand or Redux)
- /utils
- /public/assets