

# Tech Stack Documentation

## 1. FRONTEND

- **Framework:** React.js (with Vite or CRA)
- **UI Library:** Tailwind CSS / Material-UI
- **State Management:** Redux Toolkit / Context API
- **Routing:** React Router
- **Form Handling:** React Hook Form / Formik
- **Authentication:** JWT token management (via localStorage or HttpOnly cookies)
- **Other Libraries:**
  - Axios (API calls)
  - React Query (data fetching and caching)
  - Chart.js / Recharts (analytics)
  - react-toastify (notifications)

## 2. BACKEND

- **Runtime:** Node.js
- **Framework:** Express.js
- **Authentication:** Passport.js / custom JWT middleware
- **Authorization:** Role-based access
- **Validation:** express-validator / Joi
- **Security:**
  - Helmet (HTTP headers)
  - Rate Limiting (express-rate-limit)
  - CORS configuration

### 3. DATABASE

- **Primary Database:** MySQL
- **ORM:** Sequelize / Prisma

### 4. APIS & LIBRARIES USED

- **External APIs:**
  - Email API (SendGrid / Mailgun)
  - Job Skill Extraction API (custom NLP / third-party like TextRazor)
- **Internal APIs:**
  - RESTful API built with Express.js
  - Authentication (login, register, password reset)
  - Job matching logic (recommendations based on skills, preferences)
  - Admin API for moderation and analytics

### 5. TOOLS TO BE USED FOR TESTING

- **Frontend Testing:**
  - Jest
  - React Testing Library
  - Cypress (E2E)
- **Backend Testing:**
  - Mocha / Jest
  - Supertest (for API routes)
- **Linting & Code Quality:**
  - ESLint
  - Prettier
- **CI/CD Integration:**

- GitHub Actions / GitLab CI

**6. TOOLS TO BE USED FOR TESTING**

**7. TOOLS TO BE USED FOR TESTING**

**8. TOOLS TO BE USED FOR TESTING**