



**COLLEGE CODE : 9222**

**COLLEGE NAME: Theni Kammavar Sangam College Of Technology**

**DEPARTMENT: B.Tech(IT)**

**STUDENT NM-ID : 7BC751C4446D12EB2464A8DB59FABE1A**

**ROLL NO: 23IT017**

**DATE : 03/10/2025**

**Completed the project named as Phase 3 TECHNOLOGY**

**PROJECT NAME : JOB APPLICATION TRACKER**

**SUBMITTED BY,**

**NAME : S.Kishor**

**MOBILE NO :7708887782**

# MVP Implementation

TITLE:IBM-NJ-JOB APPLICATION TRACKER

## 1. Project Setup

- Choose tech stack (e.g., React for UI, Node/Express for backend, SQLite/MongoDB for DB).
- Initialize frontend and backend projects.
- Setup version control (Git + GitHub repo).
- Define folder structure (e.g., /frontend, /backend).
- Install dependencies (UI framework, API libraries, DB connector)

## 2. Core Features Implementation

- Application Management
  - Add new application form (company, role, status, date).
  - Edit application details/status.
  - Delete application.
  - View application list (table/list UI).
- Status Tracking
  - Dropdown for statuses (Applied, Interview, Offer, Rejected).
  - Dashboard summary (counts per status).
- Organization & Search

- Search bar (by company/job title).
- Status filter.
- Reminders (Basic)
- Add follow-up/interview date.
- View upcoming reminders list.

### **3. Data Storage**

- Option A (MVP Quick Start): Local State
- Store data in frontend state (React state or Redux).
- Persist with localStorage for refresh safety.
- Option B (Scalable): Database
- Setup backend with REST API.
- Store applications in a database (SQLite for lightweight / MongoDB for flexible schema).
- Expose endpoints (/applications, /summary, reminders).

### **4. Testing Core Features**

#### **Unit tests for:**

- Adding, editing, deleting applications
- Status updates reflected in dashboard.
- Search & filter functionality.

#### **Integration tests:**

- API endpoints return correct data
- UI reflects backend changes.

- Manual QA:
- Test workflows (add edit filter delete).

## **5. Version Control (GitHub)**

- Initialize Git repo.
- Create branches per feature (feature/add-application, feature/dashboard).
- Use pull requests for code reviews.
- Commit often with clear messages (feat: add application form, fix: update reminder sorting).
- Setup README.md with project details + setup instructions.