COLLEGE CODE : 9201

COLLEGE NAME : BHARATH NIKETAN ENGINEERING COLLEGE

DEPARTMENT : ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

STUDENT NM-ID : 75083df473583c68e1747054c40ec351

ROLL NO : 920123243045

DATE : 05-10-2025

Completed the project named as

Phase 5
TECHNOLOGY PROJECT NAME:

JOB APPLICATION TRACKER

SUBMITTED BY,

NAME : M.SURYAPRAKASH

MOBILE NO: 98946 66534

Phase 5 — Project Demonstration & Documentation

Project Title: Job Application Tracker

1. Final Demo Walkthrough

The Job Application Tracker project has reached its final implementation stage. This web-based application helps users manage and monitor all their job applications in one place. During the final demonstration, the following features were showcased:

1. User Registration & Login:

Secure authentication system allowing new users to sign up and existing users to log in.

Password encryption ensures data safety.

2. Dashboard Overview:

Displays all job applications with essential details such as company name, job title, application date, and current status (Applied, Interview, Offer, or Rejected).

Users can filter and sort applications based on their progress or date.

3. Add / Edit / Delete Applications:

Allows users to easily add new job applications, update statuses, or remove entries.

Real-time updates ensure all actions are reflected instantly.

4. Status Tracking and Notes Section:

Each job entry includes a notes area where users can record interview details or feedback.

Color-coded status labels for better visibility.

5. Search and Filter Functionality:

Helps users quickly locate specific applications using keywords or filters.

6. API Integration:

REST APIs were used for CRUD operations to handle job data efficiently.

Demo Outcome:
The final demonstration successfully highlighted the project's working features, clean UI, and smooth data flow. It proved the system's capability to simplify job application tracking for individuals and professionals.

---

2. Project Report

The Job Application Tracker project was developed using a modern full-stack approach. The backend manages secure data storage and APIs, while the frontend provides a user-friendly interface.

Key Objectives:

Centralize all job applications in one platform.

Provide real-time tracking and updates.

Enable easy management through filtering, editing, and notes.

Tech Stack Summary:

Frontend: React.js / HTML / CSS / JavaScript

Backend: Node.js with Express.js

Database: MongoDB

Version Control: Git & GitHub

Deployment: Render / Netlify

Outcome:
The project achieved all planned features of the MVP and demonstrated reliability, efficiency, and a clean user experience. It can be further scaled by adding features like resume uploads, reminders, and analytics dashboards.

---

3. Screenshots / API Documentation

Screenshots:

1. Login Page: Secure user authentication interface.

2. Dashboard: Displays all job applications with filters and sorting options.

3. Add New Application Form: Input fields for company name, position, status, and notes.

4. Application Details View: Shows full information and tracking updates.

5. Edit / Delete Actions: Demonstrating CRUD functionalities.

API Documentation:

Endpoint	Method	Description
api/jobs	GET	Fetch all job applications for the logged-in user
api/jobs/:id	GET	Retrieve details of a specific job application

/api/jobs	POST	Add a new job application
/api/jobs/:id	PUT	Update existing job details
api/jobs/:id	DELETE	Remove a job entry

Each API request is secured with JWT authentication and error-handled to ensure system stability.

4. Challenges & Solutions

Challenge	Description	Solution Implemented
Data synchronization	Real-time updates were not reflecting immediately in the dashboard.	Implemented state management using React hooks and optimized API calls.
Authentication issues	JWT tokens were expiring early, causing logout problems.	Adjusted token expiration and added auto-refresh functionality.
UI responsiveness	Some pages were not mobile-friendly.	Applied responsive CSS and tested across devices.
API Error Handling	Unhandled exceptions during API requests caused crashes.	Added proper backend validation and try-catch blocks.
Git Conflicts	Team collaboration led to merge conflicts.	Used Git branching strategy and frequent commits to maintain clean code.

These challenges were overcome with continuous testing, debugging, and team collaboration, resulting in a stable and efficient final product.

---

5. GitHub README & Setup Guide

GitHub README Overview:

The project's GitHub repository contains:

Project Overview & Features

Screenshots

API Documentation

Tech Stack Used

Installation & Setup Instructions

Setup Guide:

To run the project locally:

1. Clone the repository:

git clone https://github.com/yourusername/job-application-tracker.git

2. Navigate to the project folder:

cd job-application-tracker

3. Install dependencies for both backend and frontend:

npm install

4. Set up the environment variables (.env file) for database URI and JWT secret.

5. Start the backend server:

npm run server

6. Start the frontend:

npm start

7. Open the application in your browser at http://localhost:3000.

---

6. Final Submission (Repo + Deployed Link)

GitHub Link:
https://github.com/romonsuryaprakash-sketch/Job-application-tracker-

Final Deliverables:

Fully functional job tracking web application.

GitHub repository with source code and documentation.

Deployed working link demonstrating complete project functionality.

Project report and final documentation (this document).

Conclusion

The Job Application Tracker project demonstrates a complete software development cycle — from problem identification to deployment. Through this project, essential skills such as UI design, API integration, database management, and deployment were successfully implemented. The system serves as a practical and efficient tool for job seekers to stay organized and informed throughout their job search process.