Project Documentation

Table of Contents

1. Project Overview

Project Name :
Task Management App

Description :
A full-stack task management application with user authentication, task management, filtering and sorting, and audit tracking. Users can create, assign, view, update, and delete tasks, as well as track changes. Also there is a simple analytic dashboard with metrics.

2. Project Setup

Prerequisites :

- Node.js v20
- Angular CLI v18
- PostgreSQL v17
- Git (for version control)

Cloning the Repository :
git clone https://github.com/charithJayawardene/task-management-app.git

3. Environment Configuration

Backend (Create .env in the root of the task-management-server folder) :

```
DB_USER=your_database_user
DB_HOST=localhost
DB_NAME=your_database_name
DB_PASSWORD=your_database_password
DB_PORT=5432
JWT_SECRET=your_jwt_secret_key
```

4. Running the Application Locally

Database
Run the sql queries in the db_scripts/queries.sql file in task-management-server folder

Backend
Navigate to the task-management-server folder :

```
cd task-management-server
```

Install dependencies :

```
npm install
```

Start the backend server :

```
npm start
```

Frontend
Navigate to the task-management-client folder:

```
cd task-management-client
```

Install dependencies:

```
npm install
```

Start the Angular development server:

```
ng serve
```

Open the browser and go to http://localhost:4200

5. Usage Guide

User Interface

1. Login / Signup: Users authenticate with JWT tokens.
2. Home : Home page.
3. Tasks List : View all tasks with filtering ans sorting.
4. Create/Edit/Delete Task : Users can create, edit or delete a task.
5. View Task Audit : Clicking view audit button on a task displays tracking of the task.
6. View Audit Details : User can view the task details after corresponding change to the task.
7. Dashboard: Displays a summary of tasks.

6. Technologies Used

Frontend : Angular, Angular Material
Backend : Node.js, Express.js
Database : PostgreSQL
Authentication : JSON Web Tokens (JWT)

7. API Documentation

Endpoint	Method	Description
/api/auth/signup	POST	Signup a new user
/api/auth/login	POST	Login user
/api/tasks/getTasks	GET	Retrieve all tasks
/api/tasks/addTask	POST	Add a new task
/api/tasks/updateTask/:id	PUT	Update an existing task
/api/tasks/deleteTask/:id	DELETE	Delete a task
/api/tasks/audit/:id	GET	Get audit log for a specific task
/api/users/getUserss	GET	Retrieve all users