Writeup:- Event Management App - Angular Project

Github link :- https://github.com/akshansha9/JAVA-FSD.git

Overview

The Event Management App is a web application designed for the administration team of an organization to efficiently manage employee details. The application is developed using Angular as the client-side framework, incorporating Bootstrap for styling and layout components. The backend development, including APIs for CRUD operations, has been outsourced, and a JSON server is set up locally for testing purposes.

Project Objectives

To build the Event Management App described in the statement, you can follow a step-by-step algorithmic approach. Below is a high-level algorithm that outlines the key steps involved in developing the application:

Algorithm:

1. Project Setup:

   - Set up a new Angular project using the Angular CLI.
   - Install Bootstrap for styling.

2. Admin Authentication:

- Create an admin login page with Angular forms for authentication.
- Implement logic to allow the admin to change the password after logging in.

3. Database Setup:

- Install JSON Server globally using `npm install -g json-server`.
- Create a `db.json` file with a sample dataset for employees.

4. Angular Components:

- Create components for the main app layout, login page, employee list, and employee management forms.

5. Routing:

- Set up Angular routing to navigate between different components.

6. Service Integration:

- Create an Angular service to interact with the JSON Server API.
- Implement methods for CRUD operations (GET all, GET by ID, POST, PUT, DELETE).

7. Employee List Component:

- Fetch and display the list of employees from the API.
- Provide a link to view details or manage each employee.

8. Employee Details Component:

- Create a component to display detailed information about a specific employee.
- Allow navigation back to the employee list.

9. Employee Management Forms:

- Implement forms for adding and editing employee details.
- Apply form validation to ensure data integrity.

10. CRUD Operations:

- Implement functionality for adding, updating, and deleting employees using the API.

11. Single Page Application (SPA):

- Ensure that the application behaves as a Single Page Application with smooth navigation.

12. Styling:

- Apply Bootstrap styles for a clean and responsive UI.

13. Testing:

- Test the application to ensure that all features work as expected.
- Perform user acceptance testing for admin functionality.

14. Documentation:

- Document the code for future reference.
- Provide a README file with instructions for running the application.

15. Review and Refinement:

- Review the code for any improvements or optimizations.
- Refine the UI and user experience based on feedback.

16. Final Deployment:

- Deploy the application to a hosting service if needed.

Sample URI for CRUD Operations

After starting the JSON Server, the following URIs are available for CRUD operations:

- GET /employees: Get all employees
- GET /employees/1: Get employee by ID
- POST /employees: Add employee
- PUT /employees: Update employee
- DELETE /employees/1: Delete employee by ID

==