Event Management App Priyanka M V

Sprint Planning:

Day 1-2: Project Setup and Component Creation

Day 1:

- Initialize the Angular project using Angular CLI.
- Set up the basic project structure, including configuration files and dependencies and json server too.

Day 2:

- Create essential components (‘Homepage Component’, ‘Admin Login Component’, ‘Change Password Component’, ‘Employees Component’, ‘Add Employees Component’, ‘View Employee Component’) using Angular CLI or manually.

Day 3-4: UI Design and Routing Setup

Day 3:

- Design UI templates for the created components.
- Implement basic styling using bootstrap and CSS for initial components.

Day 4:

- Set up routing configuration in the ‘app.module.ts’.
- Define routes for components and ensure navigation works between the Home and Admin Login pages.

Day 5-7: Component Logic and Functionality

Day 5-6:

- Write TypeScript logic for the Homepage and Admin Login components.

- Implement basic functionality like form submissions and basic authentication.

Day 7:

- Test functionality to ensure components work as expected.
- Debug any issues encountered during the initial implementation.

Day 8-10: Additional Components and Services

Day 8-9:

- Create and implement the ‘Change Password Component’, integrating it

-Day 10:

- Develop the ‘View Employees Component’ and ‘Add Employees Component’.
- Create or integrate services for managing employee data.

Day 11-13: Styling and Layout

Day 11-12:

- Enhance bootstrap and CSS styling for components to improve the visual appearance.
- Ensure consistency in design and layout across all components.

-Day 13:

- Perform a comprehensive review of the application's UI and make necessary refinements.

Day 14-15: Testing, Integration, and Deployment Preparation

Day 14:

- Test all components thoroughly, checking functionality, data manipulation, and navigation.
- Integrate with backend services or APIs if required.

Day 15:

- Prepare the application for deployment.
- Deploy the Angular application to a hosting server or platform.

Algorithm:

Step 1: Create Components

1. ‘App Component’

app.component.html:

- Contains the main layout with routing logic.
- Has a router outlet to display different components.

app.component.ts:

- Manages the main structure of the application.

2. ‘Homepage Component’

homepage.component.html:

- Displays initial information and navigation options.

homepage.component.ts:

- Manages the content displayed on the homepage.

3. ‘Admin Login Component’

admin-login.component.html:

- Contains a login form for the admin.

admin-login.component.ts:

- Handles admin authentication logic.

4. ‘Change Password Component’

change-password.component.html:

- Provides a form to change the admin's password.

change-password.component.ts:

- Manages password change functionality.

5. ‘Employees Component’

employees.component.html:

- Displays a list of employees with options to view details

employees.component.ts:

- Handles fetching and displaying employee data.

6. ‘Add Employees Component’

addemployees.component.html:

- Contains a form to add new employee details.

addemployees.component.ts:

- Handles adding new employee functionality.

7. ‘View Employee Component’

view-employee.component.html:

- Displays detailed information about a single employee.
- Provides options to edit or delete the employee.

view-employee.component.ts:

- Manages the display and editing of individual employee details.

Step 2: Implement Routing

'App Module (app.module.ts)'

- Define routes for each component.
- Map URLs to corresponding components.

Step 3: Structure and Connect

Configure 'App Component':

- Set up routing logic.
- Use router outlet to display components based on URL.

Configure Routing:

- Define routes for each component.
- Associate each route with its respective component.

Link Components:

- Connect components by using router links or navigation buttons.
- Ensure that navigation between components works correctly.

Step 4: Styling and Layout

- Apply bootstrap and CSS to enhance the visual appeal of each component.
- Design layouts, forms, buttons, etc., using bootstrap and CSS classes.

Step 5: Service Integration

Integrate services to fetch, update, or delete data (e.g., employeeservice).

Step 6: Verification

- Verify navigation between components.
- Verify functionality like adding, editing, or deleting employees.

Step 7: Run the Application

- Open the terminal for the json server, where the src folder is placed
- Type and run “json-server --watch db.json”
- Open the another terminal where the project folder is place
- Type “ng serve -o”
- A loalhost:4200 will open in the browser

Tools/technologies used:

- Angular
- Bootstrap
- HTML
- CSS
- Typescrpit
- Json server

Github: https://github.com/priyankamv279/Phase4-Event-Management-App.git