Event Management

IIHT

Time To Complete: 10 to 12 hr

CONTENTS

1 Project Abstract

In the vibrant world of event management, there's a growing demand for modern platforms that streamline event planning and organization. The CEO of an event management startup, Mr. X, challenges a team of developers to create an Event Management Application using Angular.

Your Task is to develop a digital solution that empowers users to create, manage events and guests, facilitating seamless event planning and execution.

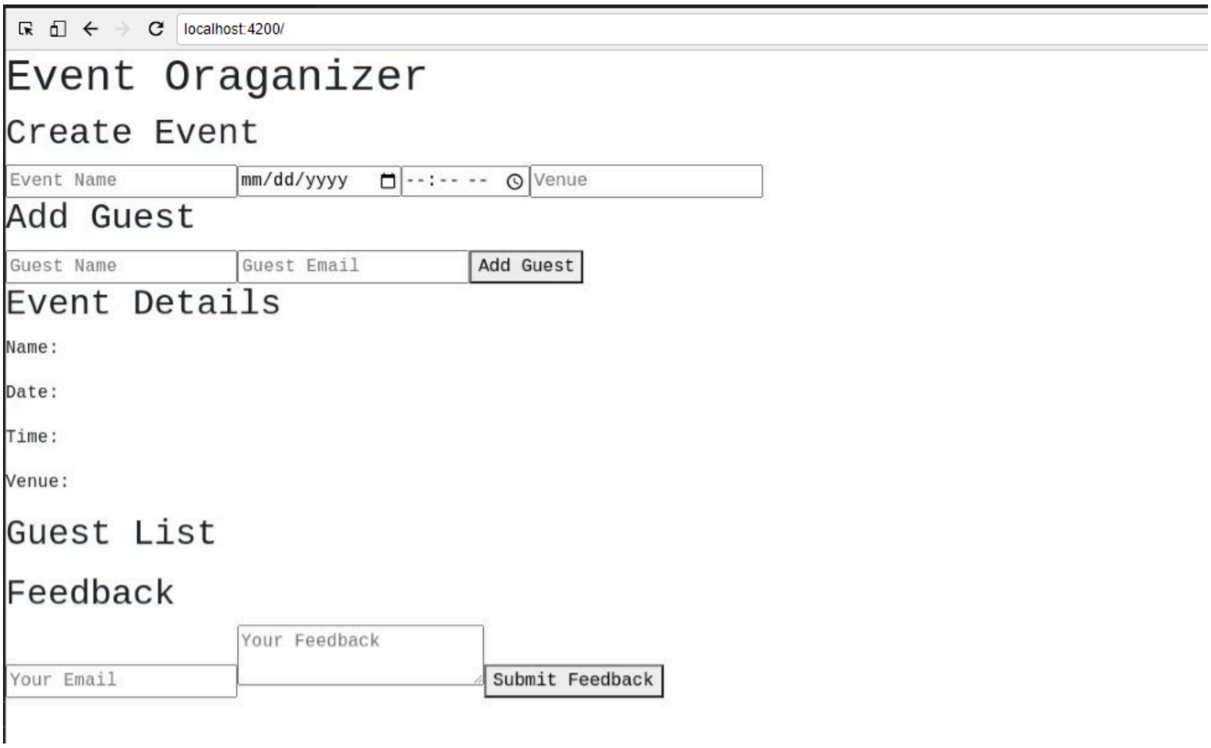
2 Problem Statement

"Event Management" is a Single Page Application (SPA) that empowers users with an interactive interface for creating events, managing guest lists and tracking the guests who are Attending / Not Attending / Maybe.

3 Proposed Event Management Application Wireframe

UI needs improvisation and modification as per given use case and to make test cases passed.

3.1 Welcome Page



localhost:4200/

Event Oraganizer

Create Event

Add Guest

Event Details

Name:

Date:

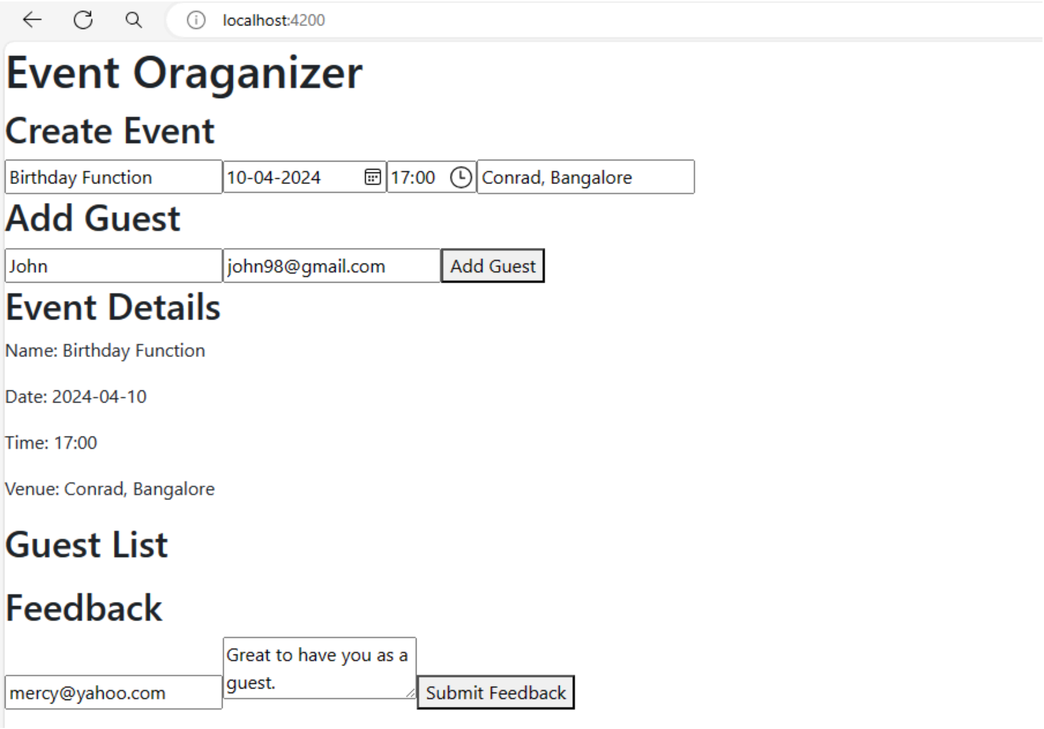
Time:

Venue:

Guest List

Feedback

3.2 ScreenShots



localhost:4200

Event Oraganizer

Create Event

Birthday Function | 10-04-2024 | 17:00 | Conrad, Bangalore

Add Guest

John | john98@gmail.com

Event Details

Name: Birthday Function

Date: 2024-04-10

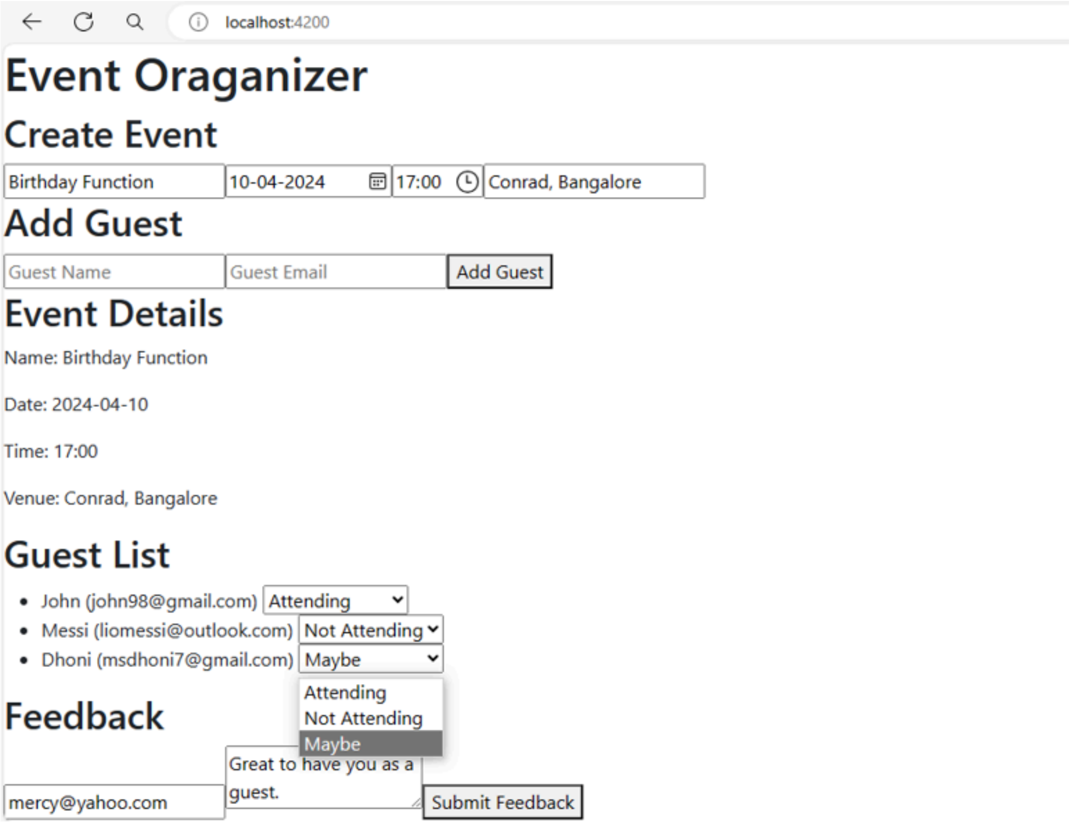
Time: 17:00

Venue: Conrad, Bangalore

Guest List

Feedback

mercy@yahoo.com | Great to have you as a guest.



localhost:4200

Event Oraganizer

Create Event

Birthday Function | 10-04-2024 | 17:00 | Conrad, Bangalore

Add Guest

Guest Name | Guest Email

Event Details

Name: Birthday Function

Date: 2024-04-10

Time: 17:00

Venue: Conrad, Bangalore

Guest List

- John (john98@gmail.com) Attending
- Messi (liomessi@outlook.com) Not Attending
- Dhoni (msdhoni7@gmail.com) Maybe

Feedback

mercy@yahoo.com | Great to have you as a guest.

4 BUSINESS-REQUIREMENT:

As an application developer, develop the Event Management Application (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	As a user I should be able to visit the welcome page as the default page. **Acceptance criteria:** **AppComponent** User Story: As a user, I should be able to view the Event Organizer heading and access event-related functionality via a dedicated component. Acceptance Criteria 1. Display a main heading: "Event Organizer". 2. Include the \`app-event-management\` component directly under the heading. 3. This component acts as the host container, delegating actual functionality (like listing or adding events) to \`EventManagementComponent\`. Component Overview Selector: app-root Template Used: app.component.html Stylesheet: app.component.css Child Component Rendered: <app-event-management>

HTML Structure (app.component.html)

• Use a top-level <div> to wrap the page content.

• Add an <h1> tag with the heading: “Event Organizer”

• Include the child component: <app-event-management></app-event-management>

Template Summary

• Heading: Use an <h1> to label the page: “Event Organizer”

• Event Section: Insert the EventManagementComponent using its selector: <app-event-management></app-event-management>

Dynamic Behavior

• The AppComponent itself does not contain any logic or data bindings in this setup.

• It acts as the shell that displays the static title and delegates all interaction to the app-event-management component.

• Any event-related features will be implemented inside the child component.

EventManagementComponent

User Story

As a user, I should be able to create an event, add guests, manage RSVPs, and submit feedback for the event.

Acceptance Criteria

1. Display a section to create an event with inputs for:

   - Event Name

  - Date

  - Time

  - Venue

2. Display a section to add guests with inputs for:

  - Guest Name

  - Guest Email

  - A submit button labeled “Add Guest”

3. Display event details showing entered event data.

4. Show a guest list, where each guest:

  - Displays their name and email

  - Has an RSVP dropdown with options: Attending, Not Attending, Maybe

5. Include a feedback form with:

  - Guest Email input

  - Comments textarea

  - A submit button labeled “Submit Feedback”

Component Overview

Selector: app-event-management

Template Used: event-management.component.html

State & Data Models

Component Properties:

- event: Holds the current event’s details.

- guests: List of guest objects.

- newGuest: Temporary guest input for form.

- feedbacks: Array storing feedback entries.

Component Methods

- addGuest(): Adds guest and resets form.

- updateRSVP(email, status): Updates guest RSVP.

- addFeedback(email, comments): Adds feedback entry.

HTML Structure (event-management.component.html)

Create Event:

- <h2> “Create Event”

- Input fields: Event Name, Date, Time, Venue

Add Guest:

- <h2> “Add Guest”

- Input fields: Guest Name, Guest Email, and a button

Event Details:

- <h2> “Event Details”

- Display event data using <p> tags

Guest List:

- <h2> “Guest List”

- List using *ngFor

- Each guest has RSVP <select>

		Feedback Form: - <h2> “Feedback” - Input, textarea, and button inside a form Dynamic Behavior - Uses [(ngModel)] for two-way data binding - Guests added dynamically and managed with dropdown - Feedback form uses ngForm and resets after submission

5 Constraints

1. You should be able to press the “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.

6 Mandatory Assessment Guidelines

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to**

   Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.

3. **This editor Auto Saves the code.**
4. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
5. **This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**

   Note: The application will not run in the local browser

6. **You can follow series of command to setup Angular environment once you are in your project-name folder:**
    a. **npm install -> Will install all dependencies -> takes 10 to 15 min.**
    b. **npm run start -> To compile and deploy the project in browser. You can press the <Ctrl> key while clicking on localhost:4200 to open the project in the browser -> takes 2 to 3 min.**
    c. **npm run test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min.**
7. **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.**