Personalized Event Schedule Application

Test Plan Report by: Arvind Nataraj

Test Plan for CreateEvent Form Component

1. Test Plan Identifier

TP-CEF – (Test Plan-Create Event Form)

2. Introduction

This test plan outlines the testing strategy for verifying the functionality of the **CreateEventForm** component in the personal event scheduling application. The feature enables users to create new events by entering event details such as **Event title**, **Attendees**, **Date**, **Time** and **Description**. The testing will ensure proper form validation, data persistence, UI behavior, and email notifications.

3. Test Items

- **Creating an Event**: Users should be able to enter event details and successfully submit the form.
- **Form Validations**: Ensuring that all required fields are properly validated.
- **Event Data Persistence**: Event details should be stored correctly after submission.
- **Email Notifications**: Emails should be sent to attendees upon event creation.

4. Features to be Tested

- **Event Creation** – Users can enter and submit an event form.
- **Validation Handling –** Display error messages when required fields are missing.
- **Data Storage –** Ensure event details persist in Redux state or database.
- **UI Interactions –** Verify proper rendering of input fields, buttons, and alerts.
- **Email Notification Triggering –** Ensure email notifications are triggered upon successful event creation.

5. Features Not to be Tested

- Editing and deleting events (covered separately).
- User authentication and login/logout flow.
- Backend API functionality beyond Redux state management.
- Email delivery mechanism (only verifying function call execution, not actual delivery).

6. Approach

Testing will be conducted using:

- **Unit Testing (Jest & React Testing Library) –** To verify UI component rendering and Redux actions.

- **Integration Testing –** To check form submission and event persistence.
- **Automation Testing (Cypress) –** To test UI behavior in an end-to-end environment.
- **Manual Testing –** To verify UI responsiveness and error handling.

7. Item Pass/Fail Criteria

Pass Criteria

- Users can successfully create an event when all required fields are filled.
- Alerts display correctly for missing required fields.
- Event details persist after submission.
- Emails are triggered upon event creation.

Fail Criteria

- Events are not saved in Redux state or database.
- UI elements do not render or function correctly.
- Email notifications are not triggered.

8. Suspension Criteria and Resumption Requirements

- Testing will be paused if there are major Redux store failures.
- Testing resumes once the issue is resolved or an alternative test setup is provided.

9. Test Deliverables

- **Test cases document** (Jest & Cypress test scripts).
- **Test execution report** (Pass/Fail status for each test).
- **Bug report** (For any identified issues).

10. Testing Tasks

- **Write unit tests** for event creation and validation errors.
- **Implement Cypress tests** for UI form behavior and event persistence.
- **Perform manual testing** for UI responsiveness and user interactions.
- **Log and report bugs** for any unexpected behavior.

11. Environmental Needs

- **Browser**: Google Chrome (latest version).
- **Tools**: Jest, React Testing Library, Cypress.
- **Test Data**: Sample user accounts and predefined event entries.

12. Responsibilities

- **Frontend Developer**: Ensures UI and Redux state updates correctly.
- **QA Tester**: Runs Jest & Cypress tests and reports bugs.
- **Team Lead**: Reviews test results and approves fixes.

13. Staffing and Training Needs

- Knowledge of **Jest & Cypress** for test implementation.
- Familiarity with **Redux state testing** if required.

14. Schedule

Task	**Duration**	**Assigned To**
Write unit tests	1 day	Developer
Implement Cypress tests	2 days	QA Tester
Manual Testing	1 day	Tester
Bug Fixes & Review	1 day	Team Lead
Final Approval	1 day	Team Lead

15. Risk and Contingencies

Risk	**Impact**	**Mitigation**
Redux store issues	High	Ensure correct Redux implementation
UI does not update correctly	Medium	Verify state updates in event handling
Email notifications fail to trigger	Medium	Debug email function calls

16. Approvals

- **Project Lead**: Arvind Nataraj
- **QA Tester**: Arvind Nataraj
- **Professor**: Prof Peter Dillinger