NCW Event Organizer - Project Documentation

1. Testing Documentation

Testing is a crucial part of software development to ensure functionality, reliability, and security. Below are the different types of testing conducted for our project.

1.1 Unit Testing

Objective: To test individual components or functions to ensure they work as expected.

Implementation:

- We performed unit testing on authentication functions, such as user signup, login, and logout.
- Testing was done using console logs and debugging tools in the browser.
- Mock data was used to validate expected outputs for each function.

📌 **Screenshot Required:** Add a screenshot of unit test results for authentication.

1.2 Integration Testing

Objective: To verify the interaction between different modules.

Implementation:

- We tested the integration of Firebase Authentication with Firestore Database.
- Checked whether user data is correctly stored and retrieved after login.
- Ensured that event data entered by users is saved and displayed correctly.

📌 **Screenshot Required:** Include screenshots showing successful login and data retrieval.

1.3 Validation Testing

Objective: To confirm the correctness of inputs and ensure data integrity.

Implementation:

- Form validation was implemented for user registration (valid email, strong password, required fields).

- Event form validation checks for correct date format and prevents past event entries.
- Error messages are displayed for incorrect inputs.

📌 **Screenshot Required:** Add screenshots of form validation errors and success messages.

---

2. Table Design (ER Diagram & Data Flow Diagram)

The database structure is designed to store users, events, and authentication details efficiently. Below is an outline of the table design:

2.1 Users Table

Field	Type	Description
user_id	String	Unique identifier for users
name	String	User's full name
email	String	Email address
password	String	Hashed password

2.2 Events Table

Field	Type	Description
event_id	String	Unique identifier for events
event_name	String	Name of the event
event_date	String	Date of the event
organizer_id	String	Reference to user_id (event creator)

📌 **Diagram Required:** ER Diagram illustrating the relationships between Users and Events.

---

3. Modules Overview

Our project consists of different modules for better maintainability and scalability.

3.1 Main Modules

- **Authentication Module** (Signup, Login, Logout)

- **Event Management Module** (Create, Edit, Delete Events)
- **User Dashboard** (View personal events, manage settings)

3.2 Sub-Modules

- **Form Validation Module** (Validating user inputs)
- **Database Interaction Module** (Fetching and storing data in Firestore)
- **Calendar Integration Module** (Displaying events in a calendar format)

---

4. Testing Approach

We used **Black Box Testing** methodology, where testing was conducted without looking at the internal code structure.

Why Black Box Testing?

- Focuses on user experience and system behavior.
- Helps identify real-world issues without needing to modify the source code.

Techniques Used:

- **Equivalence Partitioning** (Testing different sets of valid and invalid inputs).
- **Boundary Value Analysis (**Testing edge cases like maximum and minimum values).

📌 **Additional Notes:** A report on manual testing scenarios and test case results can be attached if required.

Conclusion

This document provides a detailed view of our testing strategies, database structure, project modules, and testing methodology. With well-defined testing processes and a structured database, our project ensures reliability, efficiency, and a seamless user experience.

📌 **Next Steps:** Finalize testing results, integrate the calendar module, and prepare the final submission package.