



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Event Buddy: Event Management System

A Software Quality and Testing Project Submitted
By

Semester: Spring_24_25				Section: D	Group No: 01
SL	SN	Student Name	Student ID	Individual Contribution (in %)	Total Marks: 50
					Earned Marks:
A		MD ASSADULLA AL GALIB	22-46697-1	31	
B		MD ASIF CHOWDHURY	22-46734-1	31	
C		MEHERAF HASAN KHAN	22-46729-1	31	
D		IMRAN HOSSAIN		7	

The project will be Evaluated for the following Course Outcomes

EVALUATION CRITERIA	Total Marks (50)	
Revision History, Test Plan Identifier, Reference Materials, Problem Background, Solutions	[10 Marks]	
Requirements Specification (System feature, Quality Attributes, System Interface, Project Requirements)	[10 Marks]	
Item Not to be tested, Testing approach (Testing levels, tools, meetings), Test cases	[10 Marks]	
Item pass/fail criteria, Test deliverables, Staffing and Training, Responsibilities, Scheduling, Risk	[10 Marks]	
Approval, Format, Submission, and Defense	[10 Marks]	

Software Test Plan

for

Event Buddy: Event Management System

Version 1.0 approved

Prepared by:

Md Asif Chowdhury

Md Assadulla al Galib

Meheraf Hasan Khan

Imran Hossain

American International University-Bangladesh

June 19, 2025

Revision History	4
1. Test Plan Identifier: AT-TP01.3	5
2. Reference Materials History	5
3. Introduction	5
3.1 Background to the Problem	5
3.2 Solution to the Problem.....	5
4. Requirement Specification	7
4.1 System Features	7
4.2 System Quality Attributes.....	8
4.3 System Interface	10
4.4 Project Requirements	15
5. Features Not to Be Tested	16
6. Testing Approach	17
6.1 Testing Levels	17
6.2 Test Tools	18
6.3 Meetings	18
7. Test Cases / Test Items	19
8. Item Pass/Fail Criteria	44
9. Test Deliverables	45
10. Staffing and Training Needs	46
11. Responsibilities	48
12. Testing Schedule	49
13. Planning Risks and Contingencies	50

Revision History

Revision	Date	Updated by	Update Comments
0.1	2025.06.01	MD ASSADULLA AL GALIB	First Draft
0.2	2025.06.03	MD ASIF CHOWDHURY	Second Draft
0.3	2025.06.05	MEHERAF HASAN KHAN	Third Draft
0.4	2025.06.07	IMRAN HOSSAIN	Fourth Draft
0.5	2025.06.09	MD ASSADULLA AL GALIB	Fifth Draft
0.6	2025.06.10	MD ASIF CHOWDHURY	Sixth Draft
0.7	2025.06.12	MEHERAF HASAN KHAN	Seven Draft
0.8	2025.06.14	MD ASSADULLA AL GALIB	Eight Draft
0.9	2025.06.16	MD ASIF CHOWDHURY	Ninth Draft
1.0	2025.06.19	MD ASIF CHOWDHURY	Final Draft

1. TEST PLAN IDENTIFIER: EB-TP01.0

2. REFERENCE MATERIALS:

- Figma Design of Event Buddy (UI/UX Design)
- Software Requirement Specification (SRS) – based on the Figma and assumed scope
- Software Quality and Testing course materials
- Sample test plan: *Digital Document Artifacts Archive System*

3. INTRODUCTION

3.1 Background to the Problem

A. In academic institutions, managing various events such as seminars, workshops, cultural festivals, and competitions is a frequent activity. However, the current method of organizing these events often remains manual or partially digitized. Common practices include using social media posts, printed posters, emails, or word-of-mouth communications to inform students about upcoming events. These approaches are not only time-consuming but also unreliable, resulting in critical issues such as:

- i. Lack of timely communication regarding event details.
- ii. Overlapping schedules due to decentralized planning.
- iii. Inefficient registration processes using Google Forms or Excel sheets.
- iv. Difficulty in maintaining accurate attendance and feedback records.
- v. Inadequate event promotion and participant engagement.

B. The root cause of these problems is the absence of a centralized, interactive, and automated platform specifically designed to manage all aspects of academic or institutional event planning. Without such a system, event organizers face increased workloads and reduced efficiency, while students and participants often miss out on opportunities due to poor information flow.

The significance of addressing this problem lies in its direct impact on success and overall organizational reputation. Without proper coordination, valuable educational and extracurricular opportunities may go unnoticed, resulting in low participation and engagement.

3.2 Solution to the Problem

A. To resolve these issues, we propose the development of “*Event Buddy*”, an event management system tailored for institutional use. The system is designed to automate and streamline the entire event lifecycle from event creation and approval to registration, notification, attendance tracking, and feedback collection.

The proposed platform will offer the following core functionalities:

- **Role-Based Access Control:** Separate access and functionalities for admins, event organizers, and participants.
- **Event Creation and Approval Workflow:** Organizers can create events which are reviewed and approved by administrators before publication.
- **Smart Registration System:** Participants can view upcoming events and register with a single click. Email confirmations and in-app notifications will be provided.
- **Live Notifications and Updates:** Participants receive automatic reminders about registered events, including any changes to the schedule or location.
- **Analytics Dashboard:** Admins and organizers can view graphical data on attendance, registration trends, and event feedback.
- **Feedback and Review System:** After events, participants can rate and review, which helps in assessing event success and future improvements.

Event Buddy is a practical and effective solution for improving how events are managed in an institutional environment. It's built using modern and scalable tools like Next.js for the frontend, NestJS for the backend, and PostgreSQL for the database. These technologies make the system fast, reliable, and easy to maintain. By cutting down on manual work, it helps institutions use their resources better and run more smoothly. Moreover, its clean, mobile-friendly interface makes it easy for users to get on board, which is key to making sure the system is used and makes a real impact.

B. Event Buddy is a smart event management system built to make handling institutional events easier and more organized. Instead of relying on scattered manual processes, it brings everything together on a digital platform. This helps save time, improves the experience for participants, and makes it easier to analyze data for better planning in the future. The main goals of Event Buddy are to automate repetitive tasks, increase event sign-ups, and keep participants happy. In the long run, the aim is to make Event Buddy the go-to platform for managing all institutional events in a more efficient, engaging, and data-driven way.

C. Several platforms such as Eventbrite, Meetup, and Facebook Events are available on the market. However, they are not specifically designed for internal academic event management. These platforms:

- Lack integration with student authentication systems.
- Are mostly commercial and not suitable for internal campus use.
- Do not support academic analytics such as participation tracking or academic credits.

Event Buddy fills this gap by providing a solution custom-built for educational institutions, with lightweight design, essential tools, and easy integration into existing campus IT infrastructure.

4. REQUIREMENT SPECIFICATION

4.1 System Features

The following system features have been derived strictly from the Figma screenshot provided for the Event Buddy application. These features represent the functional aspects clearly visible in the interface design and do not include any assumed or future additions.

1. User Login

Functional Requirements

- 1.1. The system shall allow users to log in using their email or username along with a password.
- 1.2. A "Forgot Password" link shall be displayed on the login screen, allowing users to initiate a password reset if needed.
- 1.3. Login shall be submitted and verified when the user clicks the Login button.

Priority Level: High

Precondition: User must be previously registered in the system.

2. User Registration

Functional Requirements

- 2.1. The system shall allow new users to register using a registration form.
- 2.2. The registration form shall include fields for the user's name, email, phone number, and password.

Priority Level: High

Precondition: User must not have an existing account.

3. Homepage with Event List

Functional Requirements

- 3.1. After logging in, users shall be redirected to the homepage.
- 3.2. The homepage shall display a list of upcoming events in a card-like format.
- 3.3. Each event card shall show the event name, date, time, and location.

Priority Level: High

Precondition: User must be logged in.

4. Event Details View

Functional Requirements

- 4.1. The system shall allow users to click on an event to view its full details.
- 4.2. The event details view shall display the title, date, time, location, and optionally a "Join" or "Register" button.

Priority Level: High

Precondition: User must be logged in to access detailed event information.

5. Event Creation Form

Functional Requirements

- 5.1. The system shall provide a visible event creation form for users with organizer privileges.
- 5.2. The event creation form shall include input fields for event title, date and time, description, and location.

Priority Level: High

Precondition: User must be logged in as an event organizer.

6. Event Edit Form

Functional Requirements

- 6.1. The system shall allow users with organizer privileges to open an existing event in edit mode.
- 6.2. The event edit form shall allow modifications to fields such as title, date and time, description, and location.
- 6.3. Upon submission, the updated event shall reflect the changes in the event list.

Priority Level: High

Precondition: User must be logged in as the event organizer and have access to the event.

4.2 System Quality Attributes

QA1 - Usability: A first-time user shall be able to register, log in, and browse events without needing external guidance, with clear field labels, buttons, and form layouts.

- Actions like “Join Event” or “Create Event” shall be intuitive and easy to locate.

Priority Level: High

Precondition: N/A

QA2 - Accessibility: The system shall be fully responsive across mobile, tablet, and desktop screens.

- Buttons, forms, and text shall remain readable and usable on all supported devices.
- Color contrast and font sizing shall meet minimum WCAG standards for readability.

Priority Level: High

Precondition: N/A

QA3 - Performance: The system shall be fully responsive across mobile, tablet, and desktop screens.

- Backend API responses shall return within 500ms for common queries like event listing.

Priority Level: Medium

Precondition: Server and database are running optimally.

QA4 - Security: The system shall be fully responsive across mobile, tablet, and desktop screens.

- Unauthorized users shall not be able to access event creation or admin routes.
- User sessions shall expire after a period of inactivity (e.g., 30 minutes).

Priority Level: High

Precondition: User authentication module must be configured.

QA5 - Maintainability: The system shall be fully responsive across mobile, tablet, and desktop screens.

- Codebase shall include comments and be structured to support future feature updates with minimal refactoring.

Priority Level: Medium

Precondition: Project follows a version-controlled repository and uses consistent coding standards.

4.3 System Interface

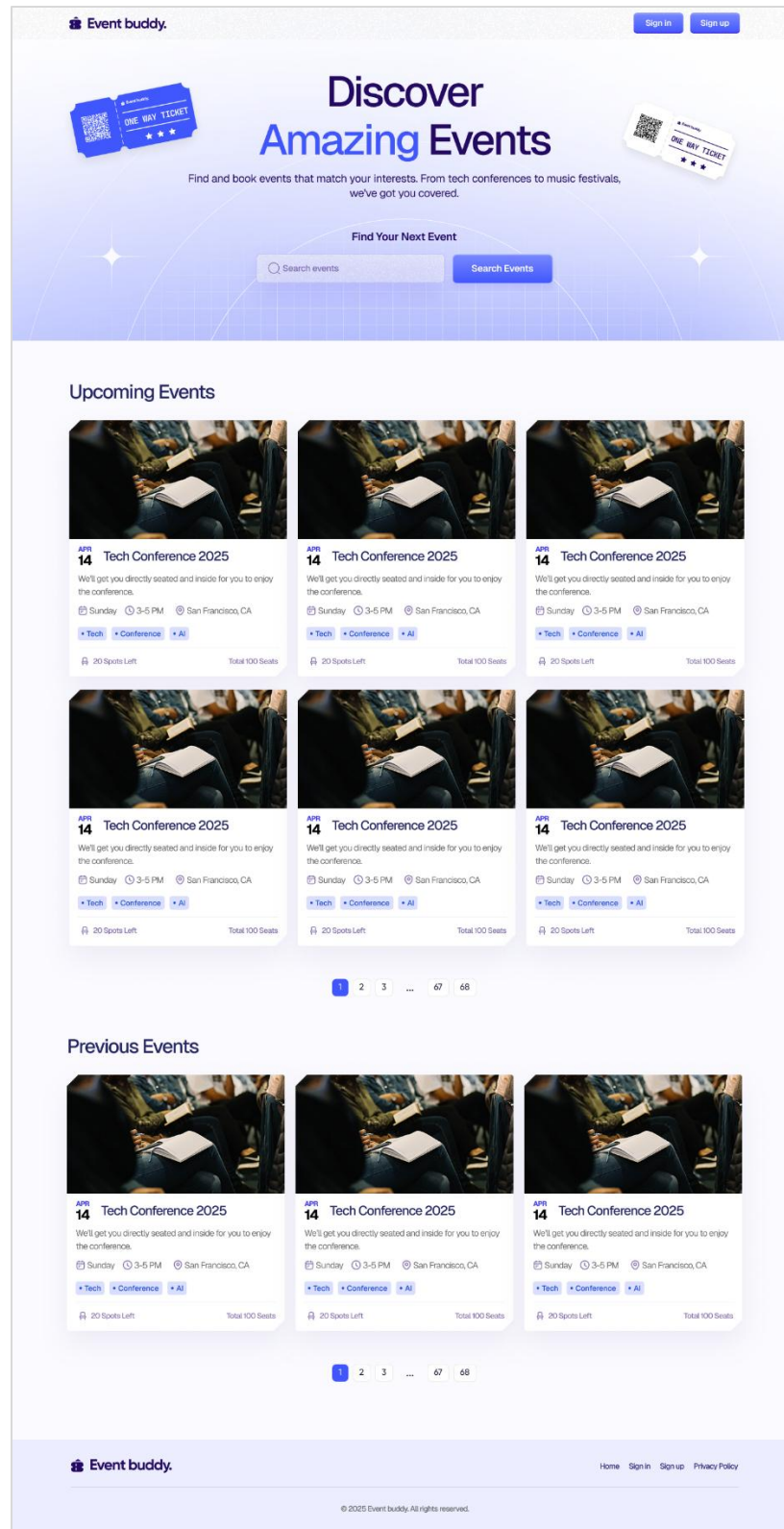



Figure 1: Home Page with event cards



Sign in


New User? [Create an account](#)

Email

Password

Sign In

Figure 2: Sign in Page with Form Inputs



Sign Up

Already have an account? [Sign in](#)

Full Name

Email

Password

Sign up

Figure 3: Registration Page with Form Inputs

[Sign in](#)
[Sign up](#)

[Back to event](#)

[Tech](#)
[Conference](#)
[AI](#)

Tech Conference 2025

Date
Sunday, 14 April, 2025

Time
03:00 - 05:00 PM

Location
San Francisco, CA

Select Number of Seats

1
Seat

2
Seats

3
Seats

4
Seats

[Book 1 Seat](#)

About this event

Join us for Tech Future Expo 2025, an immersive one-day technology event bringing together developers, startups, and industry leaders to explore the future of software, AI, blockchain, and cloud computing.

This event will feature:

- Keynote talks from industry pioneers
- Live demos of upcoming tech products
- Startup pitching sessions Hands-on coding workshops
- Networking lounge for professionals and students
- Whether you're an aspiring developer, a seasoned engineer, or just curious about what's next in tech, this event offers something for everyone.

Reserve your seat today and be part of tomorrow's innovation. Limited seats available. Advance booking required.

20 Spots Left (2000 registered)

[Home](#)
[Sign in](#)
[Sign up](#)
[Privacy Policy](#)

© 2025 Event buddy. All rights reserved.

Figure 4: Event Details Page

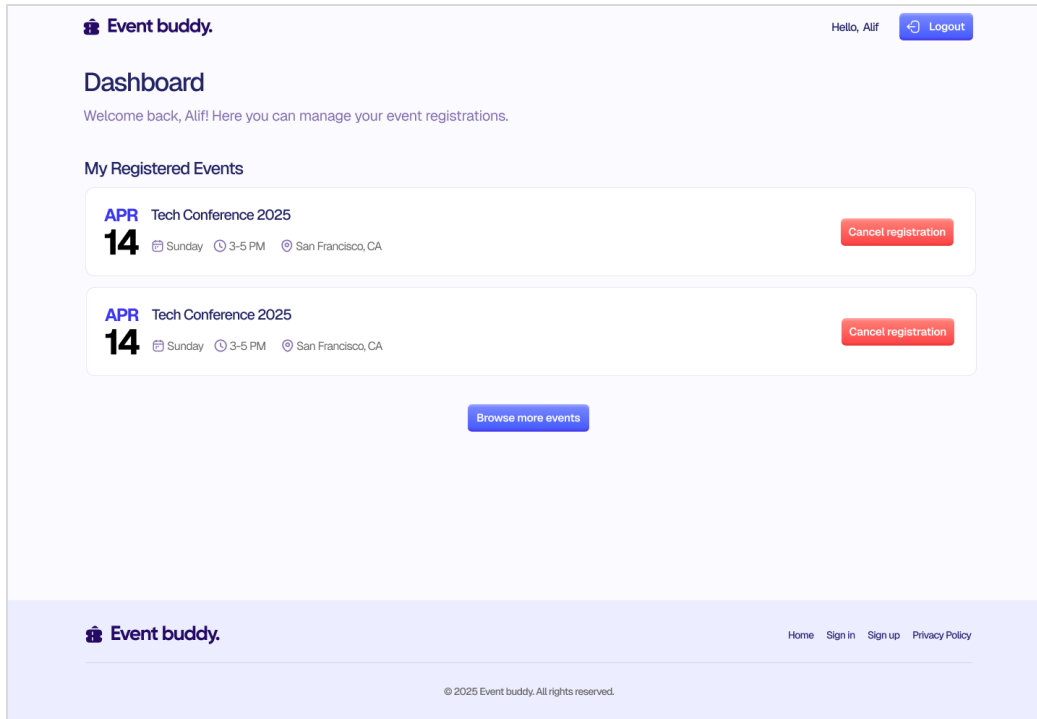


Figure 5: User Dashboard

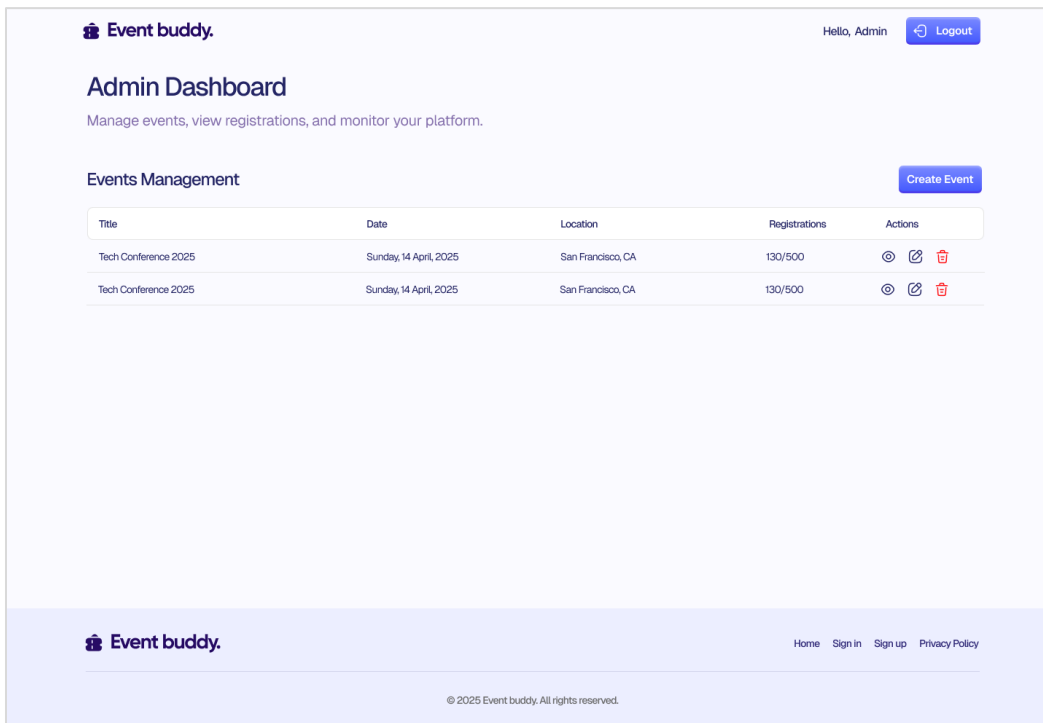


Figure 6: Admin Dashboard

Create New Event

×

Title

Date

Time

Description

Event Location

Capacity

Tags (comma separated)

Image

Drag or [upload](#) the picture here
 Max: 5MB | JPG, PNG

Browse

Cancel

Create Event

Figure 7: Create Event Form

Edit Event

×

Title

Date

Time

Description

Event Location

Capacity

Tags (comma separated)

Image

Drag or [upload](#) the picture here
 Max: 5MB | JPG, PNG

Browse

Cancel

Update

Figure 8: Edit Event Form

4.4 Project Requirements

A. Project Management Constraints & Goals: The primary goal of this project is to deliver a working software solution within the scheduled timeline, allocated budget, and the expected level of quality. To achieve this, the project must carefully balance the key constraints: Time, Cost, Scope, Resources and Environment.

- Timely delivery is essential, missed deadlines may impact on deployment opportunities.
- The scope must include all core features: user registration/login, event listing, event creation, and detailed views.
- Resources, including skilled personnel and hardware, must be managed efficiently.
- If each constraint is handled properly, the project is expected to succeed with a usable product.

Defined Requirements:

- A functional and practical version of the application must be ready within 14 weeks (3.5 months).
- The installed application should not exceed 100 MB in size.
- Preferred development tool is Visual Studio Code, but other IDEs/editors may also be used.
- Git will be the version control system, and the codebase will be hosted on GitHub for collaborative development.
- Figma will be used for UI/UX prototyping.
- Selenium will be used for unit testing and automated test scenarios.
- The total project budget is estimated at 4,50,000 BDT.

B. Time Estimation: Using a basic effort estimation approach, the total effort is broken down as follows:

Table 1: Effort-Breakdown

Phase	Estimated Hours
UI/UX Prototyping	100 hours
Development	880 hours
Revision & Refactor	80 hours
Testing & Debugging	220 hours
Total	1,280 hours

- **Daily Workload:** 12 hours/day.
- **Estimated Duration:** $1,280 / 12 = \sim 107$ days (~ 14 weeks)

C. Resources:

i. Human Resources

- 3 App Developers
- 3 Software Testers

ii. Hardware & Network

- 5 Custom-Built PCs
- 6 Android Smartphones
- 5 LAN Connections

iii. Tools & Platform

- Version Control: Git + GitHub
- Prototyping: Figma

D. Technology Stack:

- **Programming Languages:** Next.js, NestJS
- **Database:** PostgreSQL
- **Development Environment:** Visual Studio Code or equivalent IDEs
- **Workspace Environment:** Office space with stable power and internet connectivity

5. FEATURES NOT TO BE TESTED

The following features of the Event Buddy system are considered out of scope and will not be tested in the current testing cycle. These exclusions are based strictly on the features visible in the current UI design as presented in the Figma interface. Any functionality not directly accessible or represented in the current prototype is omitted from testing.

i. Admin Event Approval Panel

There is no visible administrative dashboard or event approval interface. As a result, features involving event review, approval, or rejection by an admin will not be tested.

ii. Notification System

The system does not include any form of notifications (e.g., email alerts, push notifications, or reminders). Notification-related features are excluded from this test cycle.

iii. Attendance Check-In System

There is no implementation of user check-in, QR code scanning, or attendance tracking features. Therefore, attendance verification will not be part of testing.

iv. Feedback or Review Submission

No feedback forms, rating components, or comment sections are available in the design. Thus, features related to collecting or displaying event feedback will not be tested.

v. Admin Dashboard and Analytics

The current design does not include any analytics or admin dashboards displaying event stats or user metrics. These features are outside the scope of this version's testing.

vi. Password Reset Workflow

While the login page shows a "Forgot Password?" link, the actual password reset flow (e.g., email/OTP verification, new password submission) is not implemented in the current design and will not be tested.

vii. User Profile and Settings Management

There is no visible option for viewing or updating user profile details or account settings. These functions are not part of the testing scope.

viii. Calendar or Scheduling Integration

The system shows no integration with external tools such as Google Calendar or iCalendar. Calendar syncing/exporting features are excluded.

ix. Offline Usage and Connectivity Handling

The design assumes a fully online system. There is no indication of support for offline mode, cached pages, or poor network handling, and thus these will not be tested.

6. TESTING APPROACH

The testing approach for the Event Buddy system has been designed to ensure that all the core functionalities visible in the provided interface are tested thoroughly through different levels of testing. This section outlines the structure, tools, and coordination strategy that will be followed during the testing phase.

6.1 Testing Levels

The testing process will be conducted in four distinct levels to ensure both individual component integrity and system-wide functionality.

A. Unit Testing

- Unit testing will be performed by the developers during the development phase.
- The primary objective is to verify that individual modules such as the login, registration, and event creation forms function correctly.
- Each input field and form component will be validated using both valid and invalid data.
- Sample validations include empty field handling, proper format checking (e.g., email and phone), and button functionality.

B. Integration Testing

- Integration testing will be performed to verify that different modules work together as intended.
- For example, after a successful registration, the user should be able to log in and access the home page containing the list of events.
- This level ensures that data flows properly from one module to another and that navigation across screens is consistent and functional.

C. System Testing

- System testing will be conducted by the quality assurance team after the development of all major features is complete.
- The full system will be tested against the user interface and flow as shown in the Figma design.

- The focus will be on ensuring that users can successfully register, log in, view events, and create new events without encountering any errors.

D. Acceptance Testing

- Acceptance testing will be performed in the final phase of the project.
- Test cases will simulate real user scenarios to validate that the application meets the expected behavior based on its visible interface.
- The system will be reviewed and tested by team members and, if applicable, external evaluators before final submission.

6.2 Test Tools

As the Event Buddy system is being tested based on a user interface prototype and form-based flow, the following tools will be used to facilitate the testing process:

1. **Figma** – To cross-check implemented features with the UI design.
2. **Manual Testing** – Core form submissions and navigation will be tested manually.
3. **Browser Developer Tools** – Used to inspect input validations, console logs, and UI behaviors.
4. **Google Sheets or Excel** – To maintain the list of test cases, expected vs. actual results, and overall testing status.

No automated or scripting tools are used in this version due to the early design stage and the limited functional scope.

6.3 Meetings

Every week, the quality assurance team leader will schedule a meeting to assess the progress made on the application. The team will also regularly perform code reviews and walk through the codebase to detect errors and bugs as early as possible. Each week, the project manager will meet with the QA team leader to discuss testing progress and project status. Additionally, every two weeks, all team members involved in the project will participate in an inspection session to review overall quality and identify any major issues or roadblocks.

7 .TEST CASES/TEST ITEMS

Project Name: Event Buddy: Event Management System			Test Designed by: Galib	
Test Case ID: TC_01			Test Designed date: 28 May 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Galib	
Module Name: Login			Test Execution date: 7 June 2025	
Test Title: verify login with valid username and password				
Description: Test the login session module for expected behavior.				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Go to the website Enter email Enter password Click submit	Email: user@example.com Password: correctPass	User should log in successfully	As expected,	Pass
Post Condition: User session is initiated				

Table 2: Test Case for Login

Project Name: Event Buddy: Event Management System		Test Designed by: Asif		
Test Case ID: TC_02		Test Designed date:29 May 2025		
Test Priority (Low, Medium, High): High		Test Executed by: Asif		
Module Name: Login		Test Execution date: 8 June 2025		
Test Title: verify login with incorrect credentials				
Description: Test the login session module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Go to the website Enter Invald username / email Enter wrong password Click submit	Email: wrongUser@gmail.com Password: wrongPass	Login should fail with error message	As expected	Pass
Post Condition: User is not logged in				

Table 3: Test Case for Registration

Project Name: Event Buddy: Event Management System			Test Designed by:Meheraf	
Test Case ID: TC_03			Test Designed date:29 May 2025	
Test Priority (Low, Medium, High): High			Test Executed by:Mereraf	
Module Name: Login			Test Execution date:9 June 2025	
Test Title: verify login with empty fields				
Description: Test the login session module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Go to the website Leave username / email blank Leave password blank Click submit	Username: Password:	System should prompt for missing credentials	As expected	Pass
Post Condition: User remains on login page				

Table 4: Test Case for Login

Project Name: Event Buddy: Event Management System			Test Designed by: Imran	
Test Case ID: TC_04			Test Designed date: 29 May 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Imran	
Module Name: Login			Test Execution date: 7 June 2025	
Test Title: verify Forgot Password link visibility				
Description: Test the login session module for expected behavior.				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Go to the login page Check if 'Forgot Password' link is visible		Link should be visible	As expected,	Pass
Post Condition: User can initiate password reset				

Table 5: Test Case for Password Reset

Project Name: Event Buddy: Event Management System			Test Designed by: Galib	
Test Case ID: TC_05			Test Designed date: 29 May 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Galib	
Module Name: User Registration			Test Execution date: 7 June 2025	
Test Title: verify registration with valid data				
Description: Test the user registration module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Go to registration page Enter valid details Click submit	Name: John Doe Email: john@example.com Password: Test@123	User should be registered successfully	As expected,	Pass
Post Condition: User account is created				

Table 6: Test Case for Registration

Project Name: Event Buddy: Event Management System			Test Designed by: Asif	
Test Case ID: TC_06			Test Designed date: 30 May 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Asif	
Module Name: User Registration			Test Execution date: 8 June 2025	
Test Title: verify registration with existing email				
Description: Test the user registration module for expected behavior.				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Go to registration page Enter duplicate email Click submit	Email: existing@example.com	System should reject duplicate registration	As expected,	Pass
Post Condition: User is not registered				

Table 7: Test Case for Registration

Project Name: Event Buddy: Event Management System			Test Designed by: Meheraf	
Test Case ID: TC_07			Test Designed date: 31 May 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Meheraf	
Module Name: User Registration			Test Execution date: 6 June 2025	
Test Title: verify registration with invalid email format				
Description: Test the user registration module for expected behavior.				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Go to registration page Enter malformed email Click submit	Email: invalid-email	System should show email format error	As expected,	Pass
Post Condition: User remains on registration page				

Table 8: Test Case for Registration

Project Name: Event Buddy: Event Management System			Test Designed by: Galib	
Test Case ID: TC_08			Test Designed date: 31 May 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Galib	
Module Name: User Registration			Test Execution date: 8 June 2025	
Test Title: verify password strength validation				
Description: Test the user registration module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Go to the registration page Enter weak password Click submit	Email: existing@example.com Password: 123	System should reject weak password	As expected,	Pass
Post Condition: User remains on registration page				

Table 9: Test Case for Registration

Project Name: Event Buddy: Event Management System			Test Designed by: Asif	
Test Case ID: TC_09			Test Designed date: 1 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Asif	
Module Name: Dashboard			Test Execution date: 10 June 2025	
Test Title: verify redirection to dashboard after login				
Description: Test the dashboard module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Log in with valid credentials	Email: john@example.com Password: Test@123	User should be redirected to dashboard	As expected,	Pass
Post Condition: User lands on dashboard				

Table 10: Test Case for Dashboard

Project Name: Event Buddy: Event Management System			Test Designed by: Meheraf	
Test Case ID: TC_10			Test Designed date: 1 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Meheraf	
Module Name: Homepage			Test Execution date: 9 June 2025	
Test Title: verify event cards render correctly				
Description: Test the homepage module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
View homepage		Events should be shown in card format	As expected,	Pass
Post Condition: User sees list of events				

Table 11: Test Case for Homepage

Project Name: Event Buddy: Event Management System			Test Designed by: Galib	
Test Case ID: TC_11			Test Designed date: 2 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Galib	
Module Name: Event Details			Test Execution date: 8 June 2025	
Test Title: verify clicking event opens detail view				
Description: Test the event details module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Log in Click on event card	Email: john@example.com Password: Test@123	Event details should be displayed	As expected,	Pass
Post Condition: User is on event detail page				

Table 12: Test Case for Event Details

Project Name: Event Buddy: Event Management System			Test Designed by: Asif	
Test Case ID: TC_12			Test Designed date: 3 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Asif	
Module Name: Event Details			Test Execution date: 10 June 2025	
Test Title: verify Join/Register button functionality				
Description: Test the event details module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Open event detail page Click Join/Register		User should be joined or confirmed for event	As expected,	Pass
Post Condition: User is marked as participant				

Table 13: Test Case for Event Details

Project Name: Event Buddy: Event Management System			Test Designed by: Meheraf	
Test Case ID: TC_13			Test Designed date: 4 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Mereraf	
Module Name: Event Creation			Test Execution date: 10 June 2025	
Test Title: verify organizer accesses creation form				
Description: Test the event creation module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Log in as organizer Go to create event page	Email: admin@example.com Password: admin@123	Event form should be accessible	As expected,	Pass
Post Condition: Form loads				

Table 14: Test Case for Event Creation

Project Name: Event Buddy: Event Management System			Test Designed by: Galib	
Test Case ID: TC_14			Test Designed date: : 4 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Galib	
Module Name: Event Creation			Test Execution date: 11 June 2025	
Test Title: verify event creation with valid data				
Description: Test the event creation module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Fill form with valid inputs Submit form	Title: Workshop Date: 2025-07-01 Location: Room 101 Capacity: 30	Event should be created successfully	As expected,	Pass
Post Condition: Event appears on homepage				

Table 15: Test Case for Event Creation

Project Name: Event Buddy: Event Management System			Test Designed by: Asif	
Test Case ID: TC_15			Test Designed date: 5 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Asif	
Module Name: Event Creation			Test Execution date: 11 June 2025	
Test Title: verify form submission with missing fields				
Description: Test the event creation module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Leave required fields blank Submit form		System should prompt to fill missing fields	As expected,	Pass
Post Condition: Event is not created				

Table 16: Test Case for Event Creation

Project Name: Event Buddy: Event Management System			Test Designed by: Meheraf	
Test Case ID: TC_16			Test Designed date: 5 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Meheraf	
Module Name: Event Creation			Test Execution date: 8 June 2025	
Test Title: verify access control for non-organizers				
Description: Test the event creation module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Log in as regular user Try to access event creation form	Email: user@example.com Password: User@123	Access should be denied	As expected,	Pass
Post Condition: User sees access restriction				

Table 17: Test Case for Event Creation

Project Name: Event Buddy: Event Management System			Test Designed by: Galib	
Test Case ID: TC_17			Test Designed date: 6 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Galib	
Module Name: Event Edit			Test Execution date: 14 June 2025	
Test Title: verify organizer opens edit form				
Description: Test the event edit module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Log in as organizer Click edit on an event	Email: admin@example.com Password: admin@123	Edit form should appear	As expected,	Pass
Post Condition: Form pre-fills with existing data				

Table 18: Test Case for Event Edit

Project Name: Event Buddy: Event Management System			Test Designed by: Asif	
Test Case ID: TC_18			Test Designed date: 6 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Asif	
Module Name: Event Edit			Test Execution date: 16 June 2025	
Test Title: verify successful event edit submission				
Description: Test the event edit module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Modify event details Submit changes	Title: Updated Title	Events should update and reflect new info	As expected,	Pass
Post Condition: Event updated				

Table 19: Test Case for Event Edit

Project Name: Event Buddy: Event Management System			Test Designed by: Meheraf	
Test Case ID: TC_19			Test Designed date: 8 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Meheraf	
Module Name: Event Edit			Test Execution date: 14 June 2025	
Test Title: verify edit form validation				
Description: Test the event edit module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Leave required fields empty Submit		System should prompt for required fields	As expected,	Pass
Post Condition: Edit is not saved				

Table 20: Test Case for Event Edit

Project Name: Event Buddy: Event Management System			Test Designed by: Galib	
Test Case ID: TC_20			Test Designed date: 8 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Galib	
Module Name: Event Edit			Test Execution date: 15 June 2025	
Test Title: verify non-organizer cannot access edit				
Description: Test the event edit module for expected behavior				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Log in as regular user Try to access edit form	Email: user@example.com Password: User@123	Access should be denied	As expected,	Pass
Post Condition: User sees access restriction				

Table 21: Test Case for Event Edit

Non-Functional Test Cases

Project Name: Event Buddy: Event Management System			Test Designed by: Asif	
Test Case ID: NF_TC_01			Test Designed date: 9 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Asif	
Module Name: Security - Login			Test Execution date: 15 June 2025	
Test Title: verify Password Encryption in Database				
Description: Confirm that passwords are stored encrypted, not as plain text.				
Precondition (If any): Admin DB access required				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Inspect password field	User credentials	Password stored as hashed	As expected,	Pass
Post Condition: User data is securely stored.				

Table 22: Non-Functional Test Case for Login

Project Name: Event Buddy: Event Management System			Test Designed by: Meheraf	
Test Case ID: NF_TC_02			Test Designed date: 9 June 2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Meheraf	
Module Name: Registration Page			Test Execution date: 15 June 2025	
Test Title: Check Mobile Responsiveness				
Description: Ensure the registration page adjusts correctly on mobile screens.				
Precondition (If any): Open registration on a mobile browser				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Open registration form on mobile	User credentials	All fields/buttons fit properly, no horizontal scroll	As expected,	Pass
Post Condition: User can register easily on mobile.				

Table 23: Non-Functional Test Case for Registration

Project Name: Event Buddy: Event Management System			Test Designed by: Galib	
Test Case ID: NF_TC_03			Test Designed date: 10 June 2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Galib	
Module Name: HomePage			Test Execution date: 16 June 2025	
Test Title: Verify Homepage Load Time Under Normal Network Conditions				
Description: Ensure the homepage loads within 3 seconds under normal network conditions.				
Precondition (If any): Stable internet; browser cache cleared				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Open homepage in browser		Homepage loads fully within 3 seconds	As expected,	Pass
Post Condition: User sees full homepage content without delay.				

Table 24: Non-Functional Test Case for Homepage

Project Name: Event Buddy: Event Management System			Test Designed by: Asif	
Test Case ID: NF_TC_04			Test Designed date: 11 June 2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Asif	
Module Name: Event Details			Test Execution date:16 June 2025	
Test Title: Confirm Page Rendering Across Browsers				
Description: Ensure consistent design and functionality on Chrome, Firefox, Safari, and Edge.				
Precondition (If any): Latest version of all browsers installed				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Open page in each browser	Event details page	Layout and elements render correctly	As expected,	Pass
Post Condition: Consistent user experience across major browsers.				

Table 25: Non-Functional Test Case for Event Details

Project Name: Event Buddy: Event Management System			Test Designed by: Meheraf	
Test Case ID: NF_TC_04			Test Designed date: 11 June 2025	
Test Priority (Low, Medium, High): High			Test Executed by: Meheraf	
Module Name: Event Edit			Test Execution date: 17 June 2025	
Test Title: Confirm Save Operation Performance on Edit Event Page				
Description: Ensure that updates to event details are saved within acceptable response time				
Precondition (If any): User must be logged in and have permission to edit event				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Edit and save changes to event	Updated event name, time, location	Changes saved within 2 seconds	As expected,	Pass
Post Condition: Event details updated quickly and reflected in system.				

Table 26: Non-Functional Test Case for Event Edit

8. ITEM PASS/FAIL CRITERIA

This section defines the standards that will be used to determine whether a test item has passed or failed. The pass/fail criteria apply to the functional modules that are clearly present in the current Figma interface of the Event Buddy system. A test case is considered passed when the actual result matches the expected result without any critical errors.

8.1 General Pass Criteria

A test item will be marked as **"Pass"** if all the following conditions are met:

1. Input Validation:

- a. All required fields accept correct data formats (e.g., valid email, phone number, date).
- b. Errors are shown appropriately for invalid or empty fields.

2. Navigation Flow:

- a. After form submissions (e.g., login or registration), the user is redirected to the correct screen to the expected screen (e.g., homepage or login page).

3. UI Display:

- a. UI elements such as buttons, forms, and event cards render correctly on the screen.
- b. Event details display accurately without layout issues.

4. Expected Functional Behavior:

- a. Clicking "Join Now" successfully completes the expected action (e.g., registration confirmation or join success).
- b. Submitting the Create Event form results in the event appearing in the event list.

5. No Errors or Crashes:

- a. No console errors or unexpected behaviors occur during testing.
- b. Application remains stable throughout the user flow.

8.2 General Fail Criteria

A test item will be marked as **Fail** if any of the following occur:

1. Mismatch in Expected and Actual Results:

- a. The output does not match the defined expected behavior in the test case.

2. Form Validation Failures:

- a. Valid inputs are rejected, or invalid inputs are accepted without warning.

3. Navigation Errors:

- a. User is not redirected properly after a successful login, registration, or event creation.

4. UI Elements Not Working:

- o Buttons, input fields, or event cards are non-functional or unresponsive.

5. System Crashes or Errors:

- a. The application crashes, reloads unexpectedly, or displays visible system errors.

6. Duplicate Actions Allowed:

- a. The same user can join the same event multiple times without restriction.

8.3 Priority-Based Evaluation

Each failed test case will be evaluated based on its priority

- **High Priority:** Core system functionalities. These must be resolved before final delivery.
- **Medium Priority:** Important but non-blocking features. These should be fixed before the presentation.
- **Low Priority:** Minor or cosmetic issues. These may be deferred for future updates.

8.4 Criteria for Acceptance

The system will be considered acceptable for delivery if:

- All High priority test cases have passed.
- At least 90% of Medium priority test cases have passed.
- Any Low priority failures are non-blocking and do not affect core usage.

9. TEST DELIVERABLES

The following test deliverables will be produced during and after the testing phase of the Event Buddy system. These documents and artifacts serve as evidence of the testing process and its outcomes, and they will be submitted as part of the final project documentation.

All deliverables will focus solely on the functionalities shown in the Figma interface: login, registration, homepage with event cards, event joining, and event creation.

9.1 Test Plan Document

This document includes the testing strategy, scope, schedule, approach, environment, responsibilities, and risk analysis. It is the primary planning document for all testing activities.

9.2 Test Case Documents

A detailed list of test cases designed to cover the following system modules:

- Login
- Registration
- Event Listing
- Event Joining
- Event Creation

Each test case includes:

- Test Case ID
- Module Name
- Test Steps
- Input Data
- Expected vs. Actual Results
- Status (Pass/Fail)

9.3 Test Execution Report

A record of the actual testing results, showing the status of each executed test case. This report will highlight:

- Total number of test cases executed
- Number of passes and failures
- Summary of defects identified
- Retest outcomes (if any)

9.4 Bug/Defect Report

A defect tracking document listing any bugs found during testing. It includes:

- Bug ID
- Module name
- Severity (High, Medium, Low)
- Description

- Steps to reproduce
- Status (Open, In Progress, Fixed, Closed)

9.5 Final Acceptance Report

After completion of all testing and resolution of major issues, a final acceptance report will be prepared. This report confirms that:

- All high-priority features are working as expected
- The system is stable and ready for submission or presentation

10. STAFFING AND TRAINING NEEDS

1. A horizontal team structure will be followed, where all testers are equally responsible for their assigned modules.
2. The project group members will act as the testing team.
3. Each tester will be assigned different parts of the system: login, registration, event list, and event creation.
4. No formal recruitment is required, as testing will be conducted by internal team members.
5. All testers will be oriented before testing begins, covering the following areas:
 - i. Understanding the UI flow based on the Figma design
 - ii. Writing manual test cases in Excel
 - iii. Executing test cases and logging results manually
6. One internal knowledge-sharing session will be conducted to ensure all testers are aligned.
7. The documentation lead will help maintain consistent formatting and assist with logging and tracking any defects found during testing.

11. RESPONSIBILITIES

Table 27: Roles and Responsibilities in Testing Activities

(X = Responsible or Actively Involved)

Activity	Project Manager (PM)	Dev Team	Test Team	Client
Acceptance Test Documentation & Execution	X		X	X
System/Integration Test Documentation & Execution	X	X	X	
Unit Test Documentation & Execution		X		
System Design Reviews	X	X	X	
Detailed Design Reviews	X	X	X	
Define Test Procedures and Rules	X		X	
Screen & Report Prototype Reviews	X			X
Change Control and Regression Testing	X	X	X	

12. TESTING SCHEDULE

The testing schedule for the Event Buddy system is presented below in the form of a Gantt chart. It outlines the timeline of key testing activities, including test planning, test case design, execution, bug fixing, and final review. The chart includes estimated start and end dates, task durations, and dependencies between testing phases.

The schedule was prepared using Microsoft Project and includes assignments of key personnel involved in each phase. The Project Manager, in coordination with the development and testing leads, will ensure that resources are allocated and activities are carried out according to the timeline.

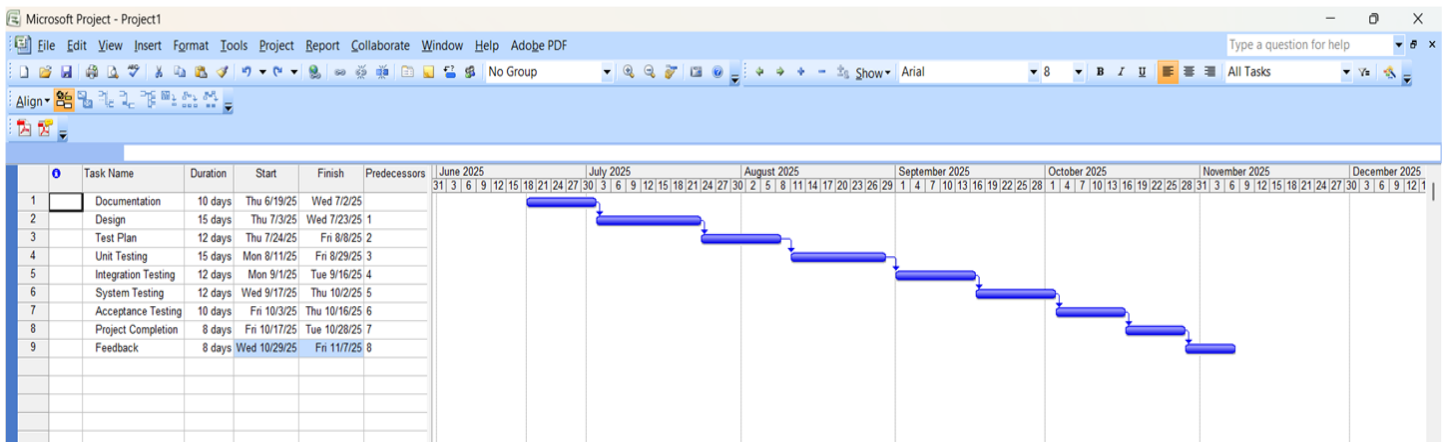


Figure 9: Gantt Chart

13. PLANNING RISKS AND CONTINGENCIES

Table 28: Risk Mitigation Plan for Testing

S/N	Risk Description	Probability	Impact	Mitigation Plan
1	Unrealistic time estimation	40%	Delays overall test schedule by 1–2 weeks	Use buffer periods and validate timeline with multiple estimation methods
2	Missing or incomplete UI modules	50%	Certain test cases cannot be executed	Track module completion closely; exclude incomplete features from test scope
3	Core feature bug (e.g., Login fails)	30%	Blocks all related test cases	Prioritize fixing critical bugs; pause dependent tests until fixed and retested
4	Lack of proper form field validation	60%	Test case failures; false positives	Validate forms using both valid and invalid inputs during test case execution
5	Loss of test data	20%	Delays or inconsistent test reports	Store test data on Google Drive; maintain version-controlled Excel sheets
6	Internet downtime	30%	Halts remote collaboration and testing	Use mobile hotspot or campus lab with backup connection
7	Power failure	15%	Loss of testing time, unsaved progress	Use laptops with battery; relocate to a lab with backup power
8	Missing team member during execution	25%	Delays test case coverage	Reassign tasks to available team members quickly
9	Figma design updated mid-testing	35%	Invalidates current test cases	Pause affected test cases; sync with updated design and revise cases accordingly
10	Test document corruption	10%	Loss of written test artifacts	Daily backups on Google Drive; optionally export to PDF regularly