SportsHub

Testing Plan

---

1. Objective

The goal of this testing plan is to validate the correctness, stability, and user experience of the Community Sports Facility Management System through a mix of **unit**, **integration**, and **end-to-end (E2E)** tests. The testing strategy includes **mocking external dependencies**, rendering components in test environments, and using JavaScript tools and testing libraries to simulate real user interactions.

---

2. Testing Strategy

2.1. Approach

We utilize the following approach for frontend testing:

- **Mock user types (roles)** to simulate the access control layer.
- **Mock data and services** such as PDF generation (e.g., html2canvas, jsPDF) and Firestore interactions.
- Render the component/page being tested using testing utilities like **React Testing Library**.
- Interact with the rendered component using simulated DOM events (clicks, form fills).
- Use assertions (e.g., expect) to verify expected behavior, DOM updates, and error handling.

2.2. Testing Types

Test Type	Tools/Frameworks Used	Description
Unit Testing	Jest	Test isolated logic and utility functions

Component Testing	React Testing Library	Render React components in isolation and test UI behavior
Integration Testing	React Testing Library + Firebase Mocks	Validate how multiple components/services work together
E2E Testing	Cypress (Optional)	Simulate complete user workflows across the system

---

🔧 3. Test Environment

- **Testing Library**: React Testing Library
- **Mocking Framework**: Jest mocks and msw (Mock Service Worker) for API
- **PDF mocking**: jest.mock() for libraries like html2canvas and jsPDF
- **User roles**: Mocked using provider wrappers or context values
- **Test IDs**: data-testid attributes placed in the DOM where appropriate

---

👤 4. User Role Simulation

User roles are mocked using either:

- React **Context Providers**
- Jest spies/mocks for Auth context
- Manual overrides during test setup

Roles:

- Resident
- Facility Staff
- Admin

□ 5. Mocking Tools & Libraries

Library	Use in Testing
html2canvas	Mocked to avoid actual PDF rendering in test env
jsPDF	Mocked to verify that PDF generation is called without rendering files
firebase/firestore	Mocked Firestore functions to simulate reads/writes
react-router-dom	Mocks for navigation and route rendering
Custom API calls	Mocked using Jest or msw

---

6. Test Implementation Structure

Tests are implemented as:

```
/tests
  └── components/
     └── bookingForm.test.js
  └── pages/
     └── homePage.test.js
  └── utils/
     └── pdfGenerator.test.js
```

---

7. Sample Testing Workflow

Example: Testing Booking Confirmation Page

1. **Setup**
   - Mock Firebase Auth to simulate a logged-in Resident
   - Mock Firestore responses for existing bookings
2. **Render**
   - Use `render(<BookingConfirmationPage />)` with necessary providers
3. **Simulate Interaction**
   - Use `fireEvent.click(getByTestId('confirm-booking-button'))`
4. **Assertions**
   - Check that the success message appears:
     `expect(getByText(/Booking confirmed/i)).toBeInTheDocument();`
   - Confirm PDF function was triggered:
     `expect(jsPDF).toHaveBeenCalled();`

---

8. Acceptance Criteria for Test Cases

Each test must:

- Simulate real-world user interactions
- Work in isolation and not depend on external services
- Use clear, maintainable selectors (`data-testid`)
- Be repeatable across environments
- Pass CI on push via GitHub Actions

---

9. Reporting and Coverage

- **Test reports** generated using `jest --coverage`
- **Code coverage thresholds**:

    - Statements: 80%
    - Branches: 75%
    - Functions: 80%
    - Lines: 80%
- **Failed tests** block PR merges via GitHub CI checks

🔄 10. CI/CD Integration

Every commit triggers:

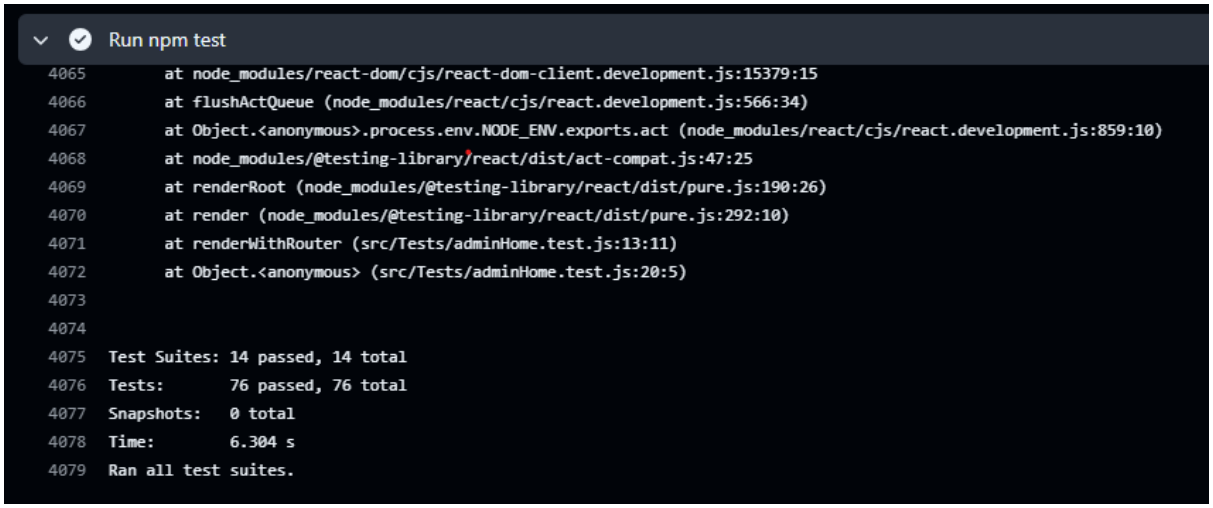
1. **Lint checks**
2. **Run tests**
3. **Generate coverage**
4. **Build & deploy if tests pass**

🔁 11. Continuous Improvement

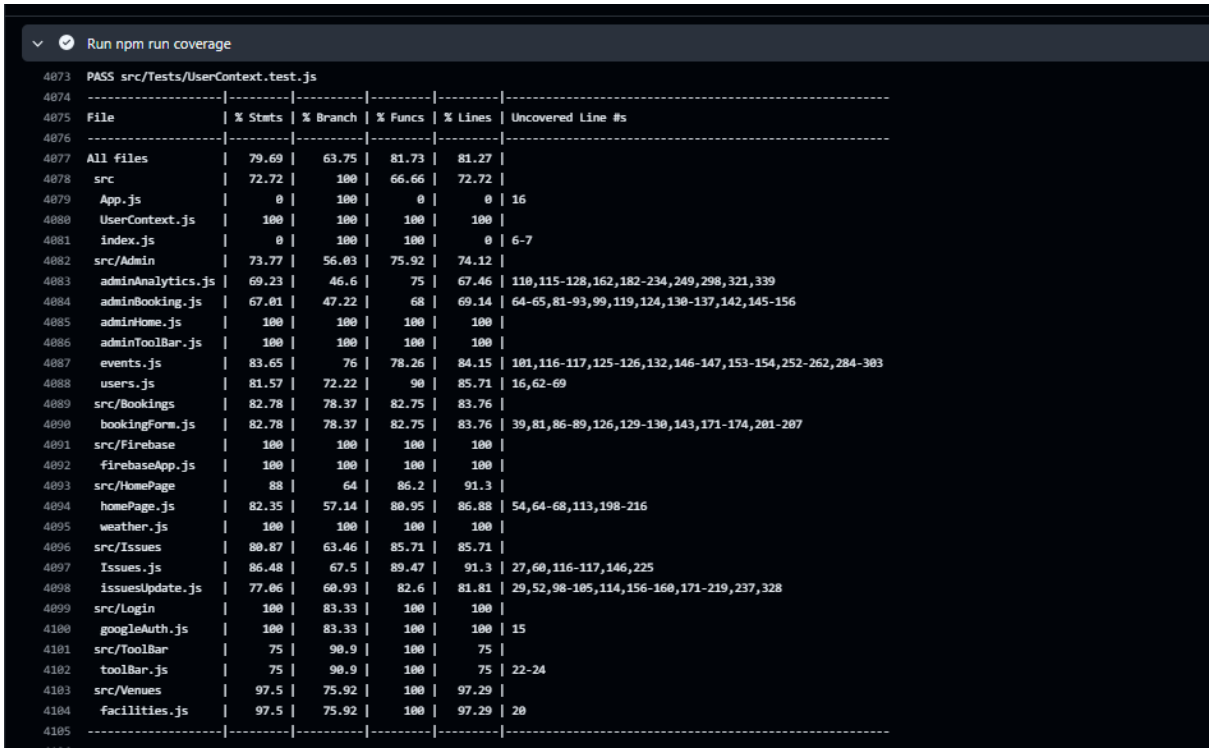
- Sprint retrospectives include a review of test performance and gaps.
- Bugs found in production are turned into new automated tests.
- Feedback from stakeholders is used to improve test cases and UX testing.

---

12. Images



Testing



```
Run npm run coverage

PASS src/Tests/UserContext.test.js
--------------------|---------|----------|---------|---------|-------------------------------------------------------------------
File                | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
--------------------|---------|----------|---------|---------|-------------------------------------------------------------------
All files           |   79.69 |    63.75 |   81.73 |   81.27 |
 src                |   72.72 |      100 |   66.66 |   72.72 |
  App.js            |       0 |      100 |       0 |       0 | 16
  UserContext.js    |     100 |      100 |     100 |     100 |
  index.js          |       0 |      100 |     100 |       0 | 6-7
 src/Admin          |   73.77 |    56.03 |   75.92 |   74.12 |
  adminAnalytics.js |   69.23 |     46.6 |      75 |   67.46 | 110,115-128,162,182-234,249,298,321,339
  adminBooking.js   |   67.01 |    47.22 |      68 |   69.14 | 64-65,81-93,99,119,124,130-137,142,145-156
  adminHome.js      |     100 |      100 |     100 |     100 |
  adminToolBar.js   |     100 |      100 |     100 |     100 |
  events.js         |   83.65 |       76 |   78.26 |   84.15 | 101,116-117,125-126,132,146-147,153-154,252-262,284-303
  users.js          |   81.57 |    72.22 |      90 |   85.71 | 16,62-69
 src/Bookings       |   82.78 |    78.37 |   82.75 |   83.76 |
  bookingForm.js    |   82.78 |    78.37 |   82.75 |   83.76 | 39,81,86-89,126,129-130,143,171-174,201-207
 src/Firebase       |     100 |      100 |     100 |     100 |
  firebaseApp.js    |     100 |      100 |     100 |     100 |
 src/HomePage       |      88 |       64 |    86.2 |    91.3 |
  homePage.js       |   82.35 |    57.14 |   80.95 |   86.88 | 54,64-68,113,198-216
  weather.js        |     100 |      100 |     100 |     100 |
 src/Issues         |   80.87 |    63.46 |   85.71 |   85.71 |
  Issues.js         |   86.48 |     67.5 |   89.47 |    91.3 | 27,60,116-117,146,225
  issuesUpdate.js   |   77.06 |    60.93 |    82.6 |   81.81 | 29,52,98-105,114,156-160,171-219,237,328
 src/Login          |     100 |    83.33 |     100 |     100 |
  googleAuth.js     |     100 |    83.33 |     100 |     100 | 15
 src/ToolBar        |      75 |     90.9 |     100 |      75 |
  toolBar.js        |      75 |     90.9 |     100 |      75 | 22-24
 src/Venues         |    97.5 |    75.92 |     100 |   97.29 |
  facilities.js     |    97.5 |    75.92 |     100 |   97.29 | 20
--------------------|---------|----------|---------|---------|-------------------------------------------------------------------
```

Code coverage