PROJECT PLAN

Project Title:

SportsHub

Community Sports Facility Management System

Team:

Dewald Smal (2605677)
Joshua Weppelman (2591952)
Justin Perumal (2745139)
Nathan Le Roux (2717495)
Ibrahim Vorajee (2351311)
Mohammed Gori (2580962)

1. Introduction

1.1 Project Overview

Many communities have shared sports facilities such as soccer fields, basketball courts, swimming pools, and gyms. These facilities require scheduling, maintenance, and fair access management. Local governments or homeowner associations often struggle with coordinating reservations, tracking facility usage, and handling maintenance. This project aims to provide a web-based solution for managing community sports facilities. It will allow users to book slots, report maintenance issues, and receive relevant notifications.

1.2 Purpose and Objectives

Purpose:

To equip communities with a central platform for scheduling, maintaining, and managing access to shared sports facilities.

Objectives:

- Develop a responsive web application using Agile methodology.
- Incorporate Continuous Integration and Continuous Deployment (CI/CD) principles.
- Ensure usability and accessibility.
- Utilize a test-driven development (TDD) approach.

1.3 Scope

In Scope:

- **Design and Development:**
  Responsive application for managing sports facility reservations and maintenance. User interface tailored to roles: Resident, Facility Staff, Admin. CI/CD pipelines to automate testing and deployment.
- **Features and Functionalities:** Facility booking and time slot management. User onboarding and role management. Maintenance issue tracking and updates. Event creation and notifications. Usage and maintenance reporting.
- **Testing and Quality Assurance:** TDD approach. Unit integration, and user acceptance testing. Performance and security testing.

Out of Scope:

- Long-term maintenance and updates beyond initial release.
- Integration with third-party weather APIs (if not feasible within project timeline).

1.4 Stakeholders

- **Project Owner:** Lucky Nkosi and Mayuri Balakistan
- **Development and Design Team:** Dewald Smal, Joshua Weppelman, Justin Perumal, Nathan Le Roux, Ibrahim Vorajee, Mohammed Gori
- **End Users:** Community residents, facility staff, and admins

---

2. Project Requirements

2.1 Functional Requirements

- **User Verification:** Use 3rd party identity provider. Roles include Resident, Facility Staff, Admin.
- **User Management:** Admins can onboard members, revoke access, assign roles.
- **Booking System:** Residents reserve facilities for time slots. Admins can approve/block bookings.
- **Maintenance Reporting:** Users report issues. Staff update statuses and feedback.
- **Events & Notifications:** Admins create events (e.g., tournaments). Users get notifications on events and closures.
- **Reporting:** At least 3 reports: Usage trends, Maintenance (open/closed), Custom view. Export as CSV/PDF.

2.2 Non-Functional Requirements

- Performance: Fast load times and processing.
- Security: Data protection measures.
- Scalability: Support increasing users and bookings.
- Usability: Intuitive, accessible UI.

---

3. Project Timeline

3.1 Milestones and Deadlines

Initiation Phase:

- Duration: 2-3 days
- Deliverables: Requirement gathering, research, planning

Sprint 1:

- Duration: 1 week
- Deliverables: UI/UX design, user verification, role management, initial deployment

Sprint 2:

- Duration: 1 week
- Deliverables: Booking system, approval/rejection logic, UML diagrams, unit tests

Sprint 3:

- Duration: 2 weeks
- Deliverables: Maintenance reporting module, dashboards and export features

Sprint 4:

- Duration: 2 weeks
- Deliverables: Event creation, notification system, final polish

---

4. Project Resources

4.1 Team Roles and Responsibilities

- **Developers:** Web app development, CI/CD setup
- **Designers:** UI/UX design, responsiveness

4.2 Tools and Technologies

- **Project Management:** Notion
- **Design:** Creately
- **Development:** HTML, CSS, JavaScript, GitHub Actions
- **Testing:** Jest
- **Database:** Firebase

---

5. Budget

5.1 Cost estimates per resource allocation:

- Development Team: Majority effort
- Design Team: Significant effort

- Testing: Moderate effort

---

6. Risk Management

6.1 Risks:

- Academic workload conflicts
- Technical issues (e.g., identity provider integration)
- Miscommunication

6.2 Mitigation:

- Regular check-ins
- Buffer time in schedule
- Use of Discord, WhatsApp, and Notion for communication

---

7. Communication Plan

7.1 Meeting Schedule:

- Sprint Planning: Start of every sprint
- Daily Stand-ups: Quick Discord meetings
- Sprint Reviews: End-of-sprint progress sharing
- Retrospectives: Post-sprint process evaluation

7.2 Reporting:

- WhatsApp/Discord for real-time updates
- Notion for structured progress tracking

---

8. Quality Assurance

8.1 Testing Plan:

- Unit, integration, UAT
- Accessibility and performance testing

8.2 Acceptance Criteria:

- All requirements met
- No critical issues

- Approval from stakeholders

---

9. Launch Plan

9.1 Pre-Launch:

- Final content lock
- Full testing
- Backup of project

9.2 Launch Strategy:

- Internal rollout
- Public launch for evaluation

9.3 Post-Launch:

- Monitor usage
- Address initial issues
