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)

Submission Components:

Link to Submission GitHub:

https://github.com/Weppman/SD-Project Submission-SportsHub.git

1. Screen Recording Video:

https://youtu.be/Q5WyuA0Crb0

2. Publicly Hosted Application:

https://lemon-moss-0b8334303.6.azurestaticapps.net/

3. GitHub Code Repository:

https://github.com/Weppman/SD-Project-SportsManagement.git

4. Project Artifacts:

4.1 Scrum Methodology

Documents Shown:

Product Backlog

- A full list of prioritized features, enhancements, and fixes.
- Items are written as **User Stories**, each assigned estimated **story points**.

™ Burndown Charts

- Visual charts representing the remaining effort (story points) vs. time for each sprint.
- Show a downward trend indicating progress and completion of sprint goals.

Story Point Estimation

Folder Shown:

Sprints: Contain

- Sprint 1
- Sprint 2
- Sprint 3
- Sprint 4

All four Sprints contain the following:

Sprint Backlogs

 Each sprint includes a detailed backlog with user stories, tasks, and story point estimations.

Q Sprint Retrospective Reports

- Summary documents from each sprint retrospective.
- Highlight what went well, areas for improvement, and actionable changes for upcoming sprints.

① Daily Stand-Up Summaries

- Brief notes from about 20 Daily Stand-Up meetings (+/- 5 per sprint).
- Include blockers, tasks in progress, and updates shared by team members.

Sprint Review Demonstrations

- Evidence of sprint reviews where completed work was demoed to stakeholders.
- May include screenshots, meeting notes.

4.2 Additional Artifacts

Folders Shown:

Project Plan

Project Plan

- High-level roadmap with major deliverables, deadlines, and milestones.
- Includes Sprint schedule and development phases.

Architecture

Architecture Diagram

- Visual representation of the application's multi-tier architecture, including:
 - Frontend (React.js)
 - API Layer (Firebase Cloud Functions)
 - Database (Firestore)
 - Hosting (Azure)

• Design

□ Design Documents

• UI, wireframes, and other supporting design diagrams

• Testing Plan

□ Test Plan and Results

- Comprehensive testing strategy with mock-based and component-level tests.
- Includes test scripts, expected outcomes, and screenshots of automated test results.