Project Title	Online Sports Booking & Tournament
Technologies	MERN
Domain	Sports
Project Level	Hard

Table

Table of Contents

1. Problem Statement:

Design a web application “Online Sports Booking & Tournament” to allow users to book / make reservations for sports events.

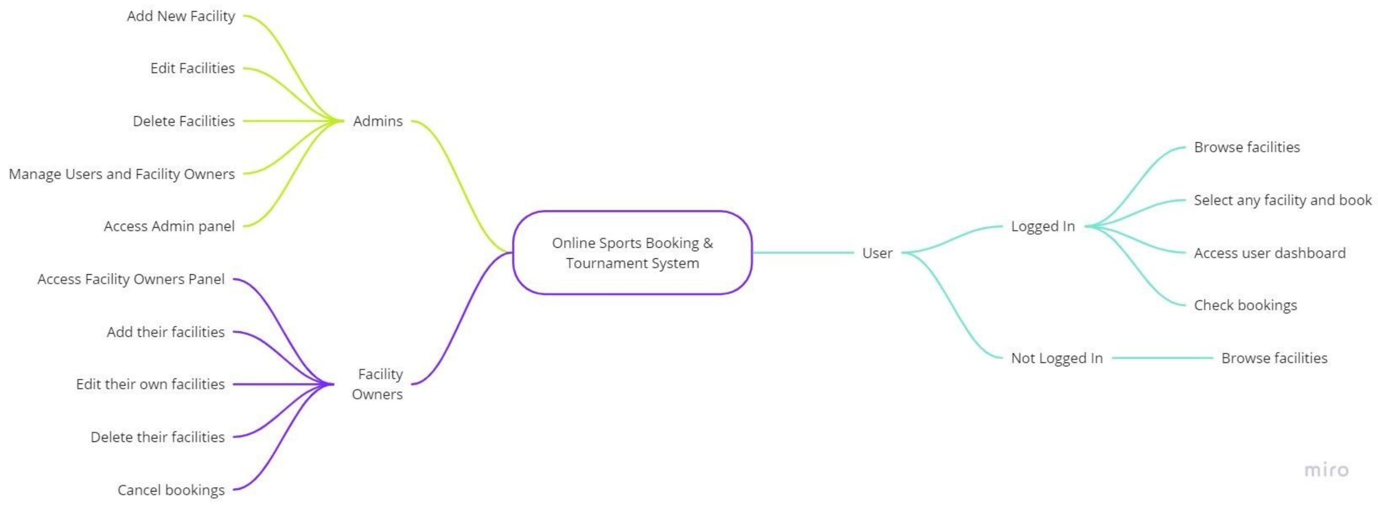
1.1. What is a Sports Booking & Tournament System?

A Sports Booking & Tournament System is an online tool which helps both the sports users and facility owners to connect to each other. Here the facility owners can list their sports facilities on the website and the sports enthusiasts can check the facility and if they like it, they can directly go ahead and book the facility from the convenience of their house. Once the booking is confirmed they can go to the facility on the booking day and play their specified sport for the time period that they have booked the facility for.

You might have used or heard about **playo.co** where one can go and book any facility for any sport they like.

Now that you understand what a Sports Booking & Tournament System is, let’s discuss some of the functionality of the “**Online Sports Booking & Tournament System**” that you will design.

1. To create a web-based system that will enable sports lovers to register and book any available facility of their choice from the list along with the name of the sport they would like to play.
2. There has to be a user dashboard for logged-in users and an admin dashboard for admins/managers.
3. Users must be able to see the available facilities and book any of the available facilities after they log in and fill in their necessary details.
4. The dashboard should also show them the booking period that the user has chosen while booking the facility.
5. The admins/facility owners can add/edit/delete any of the facilities available for booking.
6. All non-logged-in users must be able to browse the facilities available for booking and see the relevant info for any specific facility.
7. Facility owners should be able to either accept the booking or reject a booking (with reason).
8. Users should be able to host a tournament and other players can check and join any tournament they like. Tournament owner should be able to approve or reject any player that applied to join the tournament.
9. The project should be very easy to use with a clean UI.



Low-Level system flow

1.2. Project Objective

- To build a system which minimizes the manual task of going to a facility to enquire about the availability and as well as save time.
- To make the facility owners life easy by keeping record of available facilities and upcoming bookings in a clean and nice UI.
- The program functions as an always-open office.
- It improves the management's effectiveness in providing customers with high-quality services.

1.3. Scope of The Project

1. The online sports booking & Tournament system maintains detailed data of the facilities and the clients, including the duration of time for which the facility is rented.

2. Sports enthusiasts can check the facilities online, book the facility online, select the duration specifying how long they want the facility for and directly go to the facility on the booking day.

3. Tournaments can be hosted and different teams can compete against each other in any sport of their choice. Even players can join any tournament that they like.

1.4. Functional and Non-Functional Requirements: -

1.4.1. Functional Requirements

Admin:

- Can manage all the data.
- Can approve Tournaments and Players.
- Can approve or reject facilities.

Player:

- Player can browse the available facilities and can even book after registration.
- Players can either book their own facility for a predefined time or the players can also join the tournaments hosted by tournament owners.
- Players can view the details of all the facilities and the tournaments.
- Players can view the match schedule.

Tournament Owner:

- Tournament owner can register for a new tournament.
- Tournament owner needs to fill the details about the tournament (ex: Start date, end date, prizes, among others).
- Tournament owner can manage the tournament schedule.
- Tournament owner can approve/reject the players who apply for their tournament.

UI:

- There should be a dashboard for admins, tournament owners, and registered users.
- Frontend should be clean and minimal which shows the available facilities and the upcoming tournaments.
- Register/Login forms and a search bar for easier navigation.

1.4.2. Non-Functional Requirements

1. **Security:** The system should have a certain level of security such that not anybody can have access to sensitive information and the passwords should be properly encrypted.

2. **Robustness:** If the user's system crashes, a backup of the user data must be stored on remote database servers to enable recovery.

3. **Performance:** The application must be lightweight and the UI should be fast and responsive.

4. **Authentication Requirements:** Each user should have different privileges and there should be an authentication system in place which checks the credentials each time a user logs in.

1.5. Modules List

Admin:

- Login
- Add Inventory o Item, Return Date, Status
- Update/Delete Inventory
- Add Booking Slot o Slot Time, Facility, Approve / Cancel Booking o View booking request from user **User:**

- Login/Register
- View inventory
- View Booking slot o Book slot o View status of request (Approved/Rejected)
- Book a facility o Number of Players, Category of Game, Contact details
- Update/Delete Game
- Add Tournament o Sport type, Number of teams, time slot o List Team

     Player’s registration

        - Selected or Rejected (By Host)

     Tournament Schedule

    - List of Time table (Update/Delete) o Team 1 vs Team 2 – Ground, Date, Time – Results o Team 3 vs Team 4 – Ground, Date, Time – Results
    - Match Results
    - My Profile
    - Logout

2. Project Evaluation metrics:

2.1. Code:

- You are supposed to write code in a modular fashion ● Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system).
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.

- Follow the coding standards.

2.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.4. Deployment:

Deploy the application on your preferred service.

2.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Lowlevel Document (LLD), and Wireframe documents.

2.6. System Architecture:

You must submit a system architecture design in your wireframe and architecture documents.

2.7. Optimization of solutions:

Try to optimize your solution on the code level, and architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

3.8. The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.