# Sports Venue Booking App

Submitted By:
Amitesh Yadav (IEC2021034)

## ➢ *Introduction*

- This project is a web application designed to handle booking operations for a sports technology company as part of our assignment. It manages multiple centers, each offering various sports and different courts or resources for each sport. The system allows both customers and the operations team to create and manage 60-minute booking slots. The main objective is to help center managers efficiently oversee bookings while offering customers a smooth booking experience.

## ➢ *Design Decisions*

### ❖ Frontend Design:

- ***Custom Booking Table**:*

The booking table was entirely created from scratch without relying on externallibraries. It consists of **24 slots** per day, each corresponding to an hour.

- **Booked slots** are marked in **red**, while **available slots** are markedin **cyan**.
- Clicking on an available slot opens a **card** where the user mustenter their name to confirm the booking.

- ***Navbar and Calendar**:*

The interface includes a **navbar** with a **date picker**, **court selector**, and a **dropdown** for selecting games. Users can also use a **calendar** to navigatebetween dates for booking.

- ***Booking Validation**:*

The frontend includes validation checks to prevent overbooking of previously booked slots. The system verifies available slots before confirming bookings.

## ❖ Backend Design:

The backend was built using Node.js and MySQL (AivenConsole) to manage the operations related to booking and handling centers. Express.js was utilized to structure the backend, providing multiple routes to efficiently handle centers, sports, courts, and bookings. Below is an overview of the key models and controllers:

**Models:**

- `Center`: Manages details about each center (e.g.,Indiranagar, Koramangala).
- `Sport`: Manages different sports available in each center.
- `Court`: Manages the courts available for each sport.
- `Booking`: Handles customer bookings for specific court.

**Controllers and Routes:** Each resource (centre, sport, court, booking, anduser) has dedicated controllers and routes to handle CRUD operations, making the system modular and scalable.

**Integration of Frontend and Backend:**

- The frontend communicates with the backend REST APIs to dynamicallyfetch slot availability, court/game selections, and submit bookings.
- Both the frontend and backend have synchronized validation mechanismsto ensure the booking system remains accurate and prevents issues like double booking.

## ➢***Implementation Details***

**Technologies Used:**

- **Backend**:
  - **Node.js** with **Express.js** for building REST APIs.
  - **MySQL** as the database to store information about centres,sports, courts, and bookings.
  - **Cors** for managing cross-origin resource sharing.
  - **dotenv** for handling environment variables securely
  -
- **Frontend**:
  - **React.js** to build the user interface.

**Key Files:**

- `app.js`: The core file that connects different routes, middleware, anddatabase configurations.
- `booking.controller.js`, `center.controller.js`, etc.: Handlethe business logic for booking, centres, courts, and sports.

## ➢***Challenges and Solutions***

**Challenge 1: Real-Time Slot Management**

Ensuring that booking slots don’t overlap was crucial for the application.Implementing booking logic to avoid clashes was a priority.

- **Solution**: Applied server-side validation using the `booking.controller.js`, which checks the availability of a slotbefore creating a new booking, ensuring no conflicts.

**Challenge 2: Integration between Backend and Frontend**

Integrating a backend developed with Node.js with a frontend built using React.js can introduce communication issues, such as mismatched responseformats or CORS errors.

- **Solution**: Consistent use of **JSON** format for responses across all APIs ensured that the data flow between the frontend and backend remained smooth. Additionally, **Cors** middleware resolved cross-origin issues.

**Challenge 3: Meeting the Deadline**

Completing the entire project, including both backend and frontend integration,while ensuring everything worked as intended, was a difficult task under the project's deadline. The sheer volume of work required made time management critical.

- **Solution**: Regular commits, clear prioritization of tasks, and focusedsprints helped to meet the project's requirements and complete the integration on time.

## ➢***Future Improvements***

Given more time, the following features and enhancements could beimplemented:

- **Admin Dashboard**: Create a more robust operations managementdashboard with detailed reporting, analytics, and the ability to view activity logs for better oversight of booking patterns and resource utilization.
- **Payment Integration**: Implement a secure payment gateway for customers to make payments directly through the application, streamlining the booking process and improving user convenience.
- **Authentication with Role-Based Access Control**: Enhance the authentication system by implementing role-based access control (RBAC) to ensure that different types of users (e.g., customers, centres, managers, admins) have appropriate access to specific parts of the system.