SportSync (IIT2021146)

Introduction

This project is a SportSync that allows users to book sports courts at various centers. It supports viewing available courts, managing bookings, and handling different sports for specific centers.

The backend is built using Node.js, Express, and MongoDB, while the frontend (if included) can be built using frameworks like React or Angular. The backend handles APIs for managing centers, courts, sports, and bookings.

Design Decision

- Center Model: Represents each center (e.g., Indiranagar, Koramangala) that the company operates.
- Sport Model: Represents the various sports offered across centers, such as badminton or squash.
- User Model: Represents both customers and operations staff (managers), with role-based access for different functionalities.
- Court Model: Represents a court or resource available for a sport within a specific center.
- CenterSports Model: Maps which sports are available at each center, enabling flexibility in associating centers with multiple sports.
- Booking Model: Represents individual bookings made by users for specific courts at designated times.

  By having a separate CenterSports model, we maintain a clean and normalized database structure where we avoid repetitive data, making the system more efficient and scalable.

Implementation Details

Technologies Used: The application is built with Node.js and Express for the backend, MongoDB as the database, and Mongoose as an ORM to manage data relationships. These technologies were chosen for their scalability, flexibility, and compatibility with JavaScript, which streamlines the development process.

Rationale: Node.js and Express provide a robust environment for handling asynchronous operations required for real-time booking management. MongoDB's document-oriented structure makes it ideal for representing hierarchical data, such as centers, sports, and courts

Challenges and Solutions

- Challenge: Managing complex relationships between centers, sports, and courts required careful data modeling to avoid redundancy and ensure efficient querying.
  Solution: I created dedicated models for CenterSports and Court to maintain a clean and scalable database schema. This approach simplifies future additions of new centers or sports and enables flexible querying of available resources per sport at each center.

- Challenge: Conflict Prevention
  Solution: I implemented validation at the booking level to check for overlapping bookings within the same slot and court. This helps maintain data integrity and avoid double-booking.

Future Improvements

- Flexible Slot Booking: Instead of restricting bookings to fixed 60-minute slots, allow customers to book courts for variable durations (e.g., 30 minutes, 90 minutes, etc.), depending on availability.
- Booking Updates and Cancellations: Allow users to modify existing bookings (e.g., change the time, court, or sport) or cancel their bookings directly through the app. Managers could have the ability to update or cancel bookings on behalf of customers if needed.
- Payment Integration: Integrate a payment system to allow customers to pay for bookings within the app.
- Analytics for the Operations Team: Provide an analytics dashboard where center managers can view insights like peak booking times, court utilization rates, and

customer preferences.