Booking App for Sports Technology

Game Theory

ROLL NO => IIT2021214

Introduction

I created a booking application. This project's primary objective was to develop a system that would make it simple for centre administrators to browse and make reservations for various sports and resources at various locations. By providing a user-friendly interface, the application aims to streamline operations and ensure effective booking administration while preventing duplicate bookings.

Design Decisions

In designing the booking application, I made several key decisions to ensure that the app was both scalable and user-friendly. The system needed to handle multiple centres, sports, and resources efficiently while ensuring that bookings could be made without conflicts.

1. **Entity-Relationship Model**:

   To capture the connections between centers, sports, and courts/resources, I organized the data models. Each sport may have more than one court or set of facilities, and each center may host more than one sport. Bookings across various venues and sports may be managed with flexibility thanks to this approach. For instance, a single facility may include three squash courts and two badminton courts, all of which are controlled by the same system
2. **Slot-Based Booking System**:

   To keep things consistent, I choose to assign all reservations to a 60-minute window. Each court or resource is only available for one booking per slot, and this arrangement guarantees that there are no overlaps. Due to the uniform time schedule for all sports and courts, the slot-based method also facilitates large-scale booking management.
   - .

3. **Backend with RESTful APIs**:

   The backend was constructed with Express.js and Node.js. Express offers a lightweight, adaptable framework for creating APIs, which is why I went this route. Requests to view and create reservations are handled by the APIs, which also make sure that no duplicate reservations are made. The choice to use RESTful APIs makes it simple to integrate them with other interfaces or systems in the future.

4. **Frontend Simplicity**:

   I used HTML, CSS, and JavaScript to create a simple and useful user interface for the frontend. The operations staff could choose a center, sport, and view available slots with ease thanks to the interface's simplicity. The booking procedure is made as efficient as possible by the simple design, which reduces distractions.

5. **Booking Conflict Prevention**:

6. One important aspect was avoiding duplicate reservations. I created the system so that it looks up previous reservations for that court, resource, and time slot before confirming a new one. This guarantees that there are no disputes, giving the operations team a seamless experience

Data Models

I created a number of crucial data models that precisely depict the connections between centers, sports, courts/resources, and bookings in order to enable the booking application's main operation. Multiple entities within the system can be managed effectively thanks to the data models' scalability and flexibility. The main data models I developed are listed below:

1. **Centre Model**:
   - This model represents each sports centre within the company. I included fields such as **centre name** and **location** to differentiate between centres. Each centre is associated with multiple sports, creating a one-to-many relationship between the centre and its available sports.
   - .

2. **Sport Model**:

The sport model lists the different sports, including squash and badminton, that are available at each center. I filled in the spaces for the sport's name and the affiliation with the center. This permits several sports to exist under a single center and guarantees that the sports are appropriately connected to their respective centers.

3. **Court/Resource Model**:

The purpose of this model is to depict the distinct courts or resources for every sport. For instance, squash may have three courts, but badminton may have two. With features like court number and sport ID, I made sure that every court is connected to its corresponding sport. Because of this connection, the system can differentiate between various resources for every sport, guaranteeing that the right bo okings are applied to the right court.

   - .

4. **Booking Model**:

The main element that connects clients to particular courts at particular times is the booking model. I filled in the following fields: sport ID, court/resource ID, and booking time. In order to avoid duplicate bookings, this model guarantees that reservations are exclusive to a certain court, sport, and time window. Additionally, the booking model preserves the connection between the sport and the center, making it simple to retrieve all reservations associated with a specific sport or center.

By structuring the data models this way, I ensured that the relationships between centres, sports, and courts/resources are properly maintained, allowing for a smooth booking process and preventing conflicts across different entities. Additionally, while I did not implement user authentication, the data models are flexible enough to accommodate future expansion to include users, if necessary.

Implementation Details

In implementing the booking application, I chose technologies that would provide a robust, scalable, and easy-to-manage solution for both the backend and frontend. Below are the details of the technologies used and my rationale for selecting them.

1. **Backend Implementation**:
    - **Technologies Used**: I used **Node.js** with **Express.js** to build the backend of the application. Node.js is lightweight and provides high scalability, while Express.js offers a flexible framework for creating APIs. Additionally, I used **MongoDB** for the database, which is well-suited for handling the dynamic relationships between centres, sports, courts, and bookings.
    - **Data Models**: I created models for **booking**, **centre**, **customer**, and **sport** to represent the relationships between these entities. Each model captures essential information:
        - **Booking Model**: Tracks bookings with fields like booking time, court/resource ID, and sport ID.
        - **Centre Model**: Represents each centre with fields like name and location.
        - **Customer Model** (optional): Allows for future expansion to include user accounts and authentication.
        - **Sport Model**: Defines each sport available at a centre, linked to respective courts/resources.
    - **Routes**: I structured routes to handle CRUD operations for each model. Routes include:

        - **/bookings**: To view, create, and manage bookings.
        - **/centres**: To handle centre data.
        - **/customers**: To manage customer information (if implemented).
        - **/sports**: To retrieve and manage available sports.
    - **Controllers**: I organized logic into separate controllers for clean, modular code:
        - **BookingController**: Manages all booking-related actions, including viewing and creating bookings.
        - **CentreController**: Handles centre-specific tasks like listing and updating centres.
        - **CustomerController**: Optional for handling customer details and authentication.
        - **SportController**: Manages the sports offered at each centre.
2. **Frontend Implementation**:
    - **Technologies Used**: I built the frontend using **React.js**, as it provides a component-based architecture that makes it easy to create reusable UI components. This allows for a more maintainable and scalable frontend structure.
    - **Components**:
        - **BookingForm**: A form that allows users to create new bookings by selecting a time slot, sport, and court.
        - **CentreList**: Displays the list of available centres for the operations team to choose from.
        - **Navbar**: Provides easy navigation between different sections of the app.
        - **SportList**: Displays available sports for a selected centre, making it easy to select a sport for booking.
    - **Pages**:
        - **Booking Page**: A dedicated page that displays available bookings and allows the user to create new bookings.
        - **Homepage**: The landing page that shows an overview of centres and sports.
    - **Styling**: I used a **style.css** file to manage the overall design and layout of the application. The styling was kept simple to ensure the UI was clean and

functional for the operations team, focusing on usability rather than extensive design.

- **HTML File**: I also created a basic **index.html** file as the entry point for the application. This file is responsible for rendering the React components and setting up the initial page structure.

Challenges and Solutions

1. **Time Constraint - Building the Full Website in One Day**:
2. **Preventing Double Bookings**:
    - **Challenge**: Ensuring no double bookings (i.e., the same court being booked for the same time slot) was a tricky part of the project. I needed to validate the booking data efficiently to avoid overlapping reservations.
    - **Solution**: I resolved this by implementing server-side validation in the **BookingController**. Every booking request was cross-checked against existing bookings in the database to ensure there were no conflicts. This logic was embedded within the booking creation process to maintain consistency and prevent double bookings.
3. **Managing Data Relationships Between Centres, Sports, and Courts**:
    - **Challenge**: With multiple centres, each offering multiple sports, and each sport having multiple courts, maintaining data consistency across these relationships was critical.
    - **Solution**: I designed robust data models for centres, sports, courts, and bookings, ensuring that each entity was linked through unique identifiers in the MongoDB database. This allowed me to effectively manage the relationships between the entities using controllers like **CentreController**, **SportController**, and **BookingController**.
4. **Complex UI State Management**:
    - **Challenge**: Keeping the UI responsive while managing state transitions between different selections (centre, sport, booking) was a challenge, especially when the user was switching between centres or sports.
    - **Solution**: I used **React's state management** (useState and useEffect hooks) to handle the dynamic updates across the UI components such as **CentreList**, **SportList**, and **BookingForm**. This allowed me to manage the application's state efficiently, ensuring the UI stayed responsive as users navigated through the different sections.
5. **Handling Large Data Sets for Bookings**:

- **Challenge**: Displaying a large number of bookings across multiple centres and sports without slowing down the user experience required careful data handling.
- **Solution**: I optimized the booking views by implementing pagination on the backend. This allowed the data to be fetched in smaller sets, reducing the load on the UI. Additionally, I made use of **lazy loading** to ensure only the necessary data was rendered when required, keeping the interface fast and smooth.

Future Improvements

1. **User Authentication**:
   - I would like to implement user authentication features so that only authorized personnel can create or view bookings. This would involve creating a secure login system using JWT tokens or session management, ensuring that sensitive booking data remains protected.
2. **Notifications System**:
   - Adding a notification system would be beneficial. I envision implementing email or SMS notifications to inform customers about their booking confirmations, reminders, or any changes to their scheduled slots. This would enhance customer experience and keep users informed.
3. **Enhanced User Interface**:
   - I plan to improve the user interface by adding better styling and responsiveness. This includes creating a more visually appealing design and ensuring the application works seamlessly on mobile devices. Utilizing frameworks like Bootstrap or Material-UI could streamline this process.
4. **Advanced Booking Management**:
   - I would like to introduce advanced features for booking management, such as cancellation options and the ability to edit existing bookings. This would give users more flexibility and control over their reservations, improving overall usability.
5. **Reporting and Analytics**:
   - Incorporating reporting features would be valuable for the operations team. I would aim to develop dashboards that provide insights into booking trends, popular sports, and centre utilization rates, helping management make data-driven decisions.
6. **Search Functionality**:

- Adding a search function to quickly find available sports, centres, or bookings would enhance user experience. Users could type in their desired criteria and see available options without having to scroll through all the data.

7. **Integration with Payment Gateways**:
    - I would consider integrating a payment gateway to allow customers to make payments directly through the app when booking their slots. This would simplify the booking process and make it more convenient for users.
8. **Feedback System**:
    - Implementing a feedback mechanism where users can rate their booking experience would be beneficial. Gathering user feedback would help in continuously improving the application based on customer input.