Game Theory Task Submission

Submitted by: Devangi Gajjar IIB2021037

Introduction

The Sports Facility Booking App is a comprehensive web-based solution designed to streamline the process of booking sports facilities across multiple locations. This project aims to provide an efficient and user-friendly platform for customers to reserve sports facilities and for managers to oversee bookings at their respective centers.

The primary objectives of this project include:

1. Facilitating easy booking of sports facilities for customers
2. Providing a management interface for center managers
3. Implementing a robust backend system
4. Creating an intuitive frontend interface

Design Decisions

Several key design decisions were made to ensure the effectiveness and scalability of the application:

1. **User Types**: The system incorporates two distinct user types - Customers and Managers. This separation allows for tailored functionalities and access controls based on user roles.
2. **Center Management**: Each sports center is assigned a dedicated manager, enhancing localized oversight and management. Manager accounts were pre-populated in the database to ensure immediate functionality upon deployment.
3. **Location-based Approach**: The application is designed to work with 7 specific locations, based on real-life Game Theory centers. This approach allows for a more realistic and immediately applicable system.
4. **Sports Variety**: The system currently supports 5 different sports, providing a diverse range of options while maintaining simplicity. This design allows for easy expansion in the future if needed.
5. **Booking System**: Users can search for and book available slots based on date, center, and sport. This granular approach offers flexibility to users while allowing efficient resource allocation.

6. **Separate Login Systems**: Distinct login pathways for customers and managers enhance security and provide role-specific user experiences.
7. **Dynamic Resource Allocation**: To mirror real-world scenarios, the number of courts available per game at each center is randomly generated between 1 and 4. This adds an element of variability and realism to the system.

Implementation Details

The implementation of the Sports Facility Booking App leverages modern web technologies and best practices:

1. **Backend Technology**: The backend is built using Node.js with Express.js, providing a robust and scalable server-side solution. This choice allows for efficient handling of concurrent requests and easy implementation of RESTful APIs.
2. **Database**: PostgreSQL was selected as the database management system due to its reliability, ACID compliance, and excellent performance for complex queries.
3. **ORM (Object-Relational Mapping)**: Prisma ORM is utilized to streamline database operations. This decision reduces the need for writing manual SQL queries, enhances type safety, and improves overall development efficiency.
4. **Authentication**: JSON Web Tokens (JWT) are employed for secure authentication. This stateless approach is scalable and well-suited for RESTful APIs.
5. **Authorization**: A custom role authorization middleware ensures that users can only access endpoints appropriate for their role, enhancing security and maintaining proper access control.
6. **Database Schema**: The schema includes models for User, Manager, Centre, Court, and Slot, with appropriate relationships and constraints to maintain data integrity.

   Below is the database schema:

   User

   - `userId`: Int (Primary Key, Auto-increment)
   - `name`: String
   - `email`: String (Unique)
   - `password`: String
   - `role`: Role (Enum, Default: CUSTOMER)
   - `mobileNumber`: String
   - `slots`: Relation to Slot model

   Manager

   - `managerId`: Int (Primary Key, Auto-increment)
   - `name`: String
   - `email`: String (Unique)

- `password`: String
- `role`: Role (Enum, Default: MANAGER)
- `centre`: Relation to Centre model
- `centreId`: Int (Foreign Key)

Centre

- `centreId`: Int (Primary Key, Auto-increment)
- `name`: String
- `location`: String
- `courts`: Relation to Court model
- `managers`: Relation to Manager model

Court

- `courtId`: Int (Primary Key, Auto-increment)
- `centre`: Relation to Centre model
- `centreId`: Int (Foreign Key)
- `sport`: Sports (Enum)
- `courtNumber`: Int
- `slots`: Relation to Slot model

Slot

- `slotId`: Int (Primary Key, Auto-increment)
- `court`: Relation to Court model
- `courtId`: Int (Foreign Key)
- `isOccupied`: Boolean (Default: false)
- `time`: Int
- `date`: DateTime
- `user`: Relation to User model (Optional)
- `userId`: Int (Foreign Key, Optional)

Enums

`Sports`: BADMINTON, SWIMMING, TABLE_TENNIS, CRICKET, FOOTBALL

`Role`: CUSTOMER, MANAGER

7. **API Design**: RESTful API endpoints are implemented for various operations including user authentication, slot booking, and management functions.

Below are the API endpoints:

Authentication

- `POST /customer/login`: Customer login
- `POST /customer/register`: Customer registration
- `POST /management/login`: Manager login

Customer Operations

- `GET /customer/slots`: View customer's booked slots (requires authentication and CUSTOMER role)
- `POST /customer/book`: Book a slot (requires authentication and CUSTOMER role)
- `DELETE /:slotId`: Delete a booked slot (requires authentication)
- `POST /customer/available`: View available slots (requires authentication and CUSTOMER role)

Manager Operations

- `POST /manager`: View centre slots (requires authentication and MANAGER role)

8. **Frontend**: React is used for the frontend, leveraging its component-based architecture for building a responsive and interactive user interface. Axios is employed for seamless integration with the backend API.
9. **Hosting**: The application uses a distributed hosting approach:
    - Database: Hosted on Aiven for reliable and scalable database services
    - Backend: Deployed on Render for flexible and efficient server hosting
    - Frontend: Hosted on Vercel, providing fast content delivery and easy deployment

Challenges and Solutions

Throughout the development process, several challenges were encountered and addressed:

1. **Challenge**: Populating the database with realistic slot data for all locations and sports.

   Solution: A script was developed to pre-populate centers and their resources for each game, ensuring a realistic and engaging user experience from the outset.

2. **Challenge**: Ensuring each center has a dedicated manager.

Solution: A script was created to pre-populate manager accounts with standardized email formats ({centre name}@gmail.com) and a default password. This approach ensures immediate functionality while maintaining consistency.

3. **Challenge**: Separating customer and manager functionalities.

   Solution: Implemented a role authorization middleware to enforce access controls based on user roles, ensuring that users can only access appropriate endpoints.

4. **Challenge**: Time constraints prevented full implementation of frontend authentication.

   Solution: As a temporary measure, authorization headers were hardcoded for both customer and manager API requests in the frontend. This allows for functional testing and demonstration of the system's capabilities while awaiting full frontend auth implementation.

Future Improvements

Given additional time and resources, several enhancements could be made to the system:

1. **Complete Frontend Authentication**: Implement a full authentication flow in the frontend, including secure token storage and refresh mechanisms.
2. **Notification System**: Integrate email or SMS notifications using libraries like Nodemailer or Twilio. This would allow for booking confirmations and reminders to be sent to customers.
3. **Admin User Role**: Introduce a system administrator role with the ability to create and manage manager credentials, providing an additional layer of system oversight.
4. **Enhanced Booking Features**:
   - Implement a waitlist system for popular time slots
   - Allow for recurring bookings
   - Introduce a rating system for facilities
5. **Payment Integration**: Implement a secure payment gateway for processing booking fees.