Game Theory SDE Assignment Report

Introduction:

This project involves building a booking app for the operations team of a sports technology company. The app will allow managers to manage customer bookings for various sports at multiple centers. Each center offers different sports, each with multiple courts or resources, and bookings are made in 60-minute slots. The key objectives are to create an intuitive interface for viewing and creating bookings, ensure no double bookings occur, and provide a simple and functional backend and frontend solution.

Design Decisions:

a)Core Functionality:

1)**Create a Booking**:

- When a user arrives to book a court for a sport under a specific center, they are first prompted to select the **center**, **sport**, and a **desired date** for the booking.
- Based on the selected parameters, the system queries the available courts and displays time slots for the selected sport.
- Slots are dynamically generated based on the availability of courts for that sport on the given date.
- Once the user selects a time slot that has an available court, the system assigns the court to the user, and a booking is created under the user's name.
- This booking is then saved in the system, ensuring the court is reserved for that time slot.

2)**View Bookings**:

- The **center manager** can view all bookings by selecting a center, a sport, and a specific date.
- The system displays a list of both **reserved** and **available** courts for the selected sport and date.

- This allows the center manager to monitor the utilization of resources (courts) and manage any operational needs based on the reservation status of courts.

  b)**Design Considerations and Choices:**
- We implemented fixed 1-hour time slots for booking courts to standardize booking duration.
- Center Managers are allowed to see both available and reserved courts for a particular day and hour.
- To ensure data consistency and prevent double booking, the system checks for court availability before creating a booking.

Implementation Details:

a)**Backend**:

- The backend of the project is built using **Node.js**, ensuring a scalable and non-blocking architecture.
- **Express.js** is used as the framework for handling HTTP requests, providing a robust and flexible way to build RESTful APIs.

b)**Database**: **MongoDB** is used as the database for storing and managing data related to centers, sports, courts, and bookings.

c)**Data Models**:

- **Center Model**: Contains the details of each center, such as name, location, and the sports it offers.
- **Sport Model**: Represents the different sports available at each center, with attributes for the number of courts/resources available.
- **Booking Model**: Records each booking with details like customer name, court, sport, date, and time slot.
- **User Model**:Contains the details of the center manager who will allow booking of customers.

d)**JWT Authenciation:**

- JWT authentication was used to validate if the center manager are valid or not, also validation maximum session duration for a login.
- THe details present in JWT token were further used to retrieve other details based on the data related to centre manager.

e)**APIs**:

- **Create Booking API**: This API allows users to create a new booking. It checks for court availability by querying the `Booking` collection in MongoDB to prevent double bookings. If a court is available, the booking is confirmed and stored in the database.
- **View Bookings API**: This API retrieves all bookings for a specific center, sport, and date, allowing center managers to view both available and reserved slots.

f)**Front-end**:

- A simple web interface is created using **HTML**, **CSS**, and **JavaScript**.
- Center managers can use the interface to view bookings and manage court availability.

g)**Error Handling and Validation**:

- The application includes basic validation for inputs such as date, time, and customer information.
- It also handles edge cases like booking a slot that is already reserved and ensures proper error messages are returned to the user.

Challenges and Solutions:

1**.Challenge:** Ensuring that no double bookings occur for a specific court and time slot.

Solution: Created a query system in MongoDB that checks for existing bookings before confirming a new booking, ensuring courts are booked only once per time slot.

2.**Challenge**: Managing relationships between centers, sports, courts, and bookings in a scalable way.

Solution: Used MongoDB's document-based structure to store related data in a flexible format, making it easy to query and manage relationships.

3.**Challenge**: Ensuring that booking data is updated in real-time and displayed accurately to all users.

Solution: Used frequent database updates and server-side logic to refresh availability status after every booking action.

Future Improvements:

- Add real-time updates or notifications for customers on booking confirmation, cancellations, or court availability.
- Improve the booking system by adding filters for court types, time preferences, or price ranges.
- Extend the app’s functionality by building a mobile-friendly version for easier access and bookings.
- Implement a feedback or rating system for customers to review facilities and court conditions.