Project Report: Game Theory <> IIIT-A

Introduction

The objective of this project is to develop a booking application for a sports technology company's operations team. The company operates multiple centers, each offering several sports with multiple courts. The core functionality of the application focuses on enabling efficient management of customer bookings and allowing center managers to handle reservations for various courts and sports.

Design Decisions

The project emphasizes simplicity and modularity. The following design choices were made:

1. **Modular Structure**:
   The project is divided into components, APIs, and context providers to manage state efficiently.
2. **React for Frontend**:
   Chosen for its flexibility in developing interactive UIs. The `react-router-dom` library is used for navigation between views.
3. **Backend with APIs**:
   Separate APIs handle booking creation and retrieval to ensure clean separation of concerns.
4. **In-memory/Database Flexibility**:
   The app allows the use of any database or mock data, ensuring easy prototyping.
5. **Authentication as Optional**:
   User authentication is optional, allowing lightweight development with JWT-based security as a fallback.

Implementation Details

1. **Technologies Used**:
   - **Frontend**: React, React Router
   - **Backend**: Node.js, Express (optional APIs to manage bookings)
   - **Database**: MongoDB (optional, or in-memory data structures)
   - **Deployment**: Cloud platforms such as Vercel, Netlify, or AWS
2. **APIs Implemented**:
   - **View Bookings API**: Retrieves bookings for a specified center, sport, and date.

    - **Create Booking API**: Ensures that bookings are created without conflicts (prevents double booking).
3. **Frontend Features**:
    - Select a facility (center) and a sport.
    - View daily bookings for each court of the selected sport.
    - Create bookings by selecting a time slot and court.

Challenges and Solutions

1. **Double Booking Prevention**:
   Implemented backend logic to prevent multiple bookings for the same court and time slot.
2. **State Management**:
   Used React Context (`DataProvider`) to manage global state efficiently, ensuring consistency across the app.
3. **User Experience vs. Time Constraints**:
   Focused on core functionality over UI aesthetics to meet deadlines.
4. **Handling Dynamic Data**:
   The app supports multiple sports and centers, making data relationships complex. This was addressed by maintaining a modular data model.

Future Improvements

1. **User Authentication**:
   Add JWT-based authentication for better user management and secure access control.
2. **Enhanced UI/UX**:
   Improve the user interface with better styling and responsiveness using CSS frameworks like Tailwind.
3. **Advanced Features**:
    - Add search and filter functionalities for bookings.
    - Introduce email or SMS notifications for bookings.
4. **Performance Optimization**:
   Cache frequently accessed data, such as center information, to reduce API calls.

Conclusion

This project successfully provides a booking solution for the company's operations team, meeting the core requirements within the stipulated time. The application offers flexibility for future enhancements and has been developed with a focus on functionality and maintainability.

Links and Submission Details

- **GitHub Repository**: https://github.com/sona-123/Book-My-Slot
- **Deployed Application**: https://bookyoursportslot.netlify.app/
- **README**: https://github.com/sona-123/Book-My-Slot/blob/main/README.md