Sports Field Booking System

Overview

This project is a sports field booking system that allows users to view available slots and make bookings for various sports centers. The system provides a user-friendly interface for the operations team to manage bookings efficiently. The backend APIs handle requests for viewing, creating, and managing bookings with input validation and conflict prevention.

Key Features

- **Real-time Booking Management**: Users can view available slots and book fields for specific sports in different centers.
- **Time-slot Handling**: Time slots are managed hourly, ensuring that bookings are organized and cannot overlap. used **Bull npm** to automatically make slots available after one hour from booking.
- **Error Handling**: All error messages are logged in the console for debugging, and user feedback is provided via alert pop-ups.
- **Assumptions**:
    - Static images are used in the frontend for simplicity.
    - The system assumes that slots are available every day, and each slot lasts one hour.
    - Due to time constraints, pagination or infinite scrolling for courts and sport center has been avoided, assuming fewer courts and sport center for now.

Requirements Satisfied

Backend

- **View Bookings API**: Successfully retrieves bookings based on the selected center, sport, and date.
- **Create Booking API**: New bookings are created, with logic to prevent double booking of the same time slot for a given court.
- **Data Handling**: Implemented robust logic to avoid slot conflicts, ensuring that no two users can book the same slot at the same time.
- **Error Handling**: All errors are captured and displayed in the console for debugging, with alert pop-ups providing user feedback in the frontend.

Frontend

- **User Interface**:
    - The frontend allows users to select a sports center and sport, view available time slots, and create bookings through a clean and intuitive interface.
- **View Bookings**: Users can see all booked and available slots for the selected day, sport, and court.

- **Create Bookings**: Users can easily select a time slot and court to make a booking, receiving alerts for successful bookings or conflicts.
- **Feedback & Usability**: The UI is designed to provide immediate feedback via alerts, ensuring an easy-to-use and responsive experience.

Assumptions

- **Time Slots**: Each booking slot is one hour long.
- **Limited Courts**: Since the number of courts is relatively low, no pagination or infinite scrolling is required.
- **Static Images**: For the sake of simplicity and faster loading times, hardcoded images are used throughout the app.

Error Handling

- **User Feedback**: Users are notified of any errors or conflicts via alert messages.
- **Console Logging**: All backend errors and exceptions are logged in the browser console for easy debugging.

Future Improvements

- **Pagination**: Implement pagination or infinite scrolling if the number of courts increases.
- **Dynamic Images**: Add support for dynamic images in the future for a more flexible UI.

How to Use

1. Clone the repository.
2. Run npm install to install dependencies.
3. Build the frontend with npm run build.
4. Run the backend with node index.js.
5. The system will be accessible on http://localhost:8000 or the deployed Vercel link.

Deployment

- The project is deployed on Vercel for easy access.
- The backend and frontend are integrated and served from the same domain for seamless interaction.

Tech Stack

Fronted- ReactJS
Backend-NodeJS , ExpressJS
Database-MongoDb.

Deployed Link- http://booking-app-nine-blue.vercel.app/

Conclusion

This system provides a streamlined and efficient way for users to manage sports field bookings. While certain features like pagination and dynamic images have been deferred due to time constraints, the system is functional and meets the core requirements with strong error handling and user feedback mechanisms.