GAME-THEORY-PROJECT-REPORT

PROJECT

Sports Booking Application

CONTENTS OF THE REPORT

INTRODUCTION

The **Sports Booking Application** is a web-based platform designed for managing court bookings across multiple sports centers. The goal is to provide a seamless experience for operations teams, preventing double bookings and offering real-time booking management. This project uses the **MERN stack** (MongoDB, Express.js, React.js, Node.js) along with **Vite** for frontend development and **TailwindCSS** for responsive styling. Hosting is managed using **Netlify** for the frontend and **Render.com** for the backend.

DESIGN DECISIONS

The following design choices were made to enhance the application's efficiency:

1. **Vite for Frontend**: Chosen for its fast development experience and optimized builds.
2. **TailwindCSS for Styling**: To ensure a responsive and clean UI with minimal custom CSS.
3. **MongoDB Atlas**: Selected for cloud-based storage, eliminating the need for local databases.
4. **Backend Hosting on Render.com**: Render.com offers easy deployment and management of web services.
5. **Frontend Hosting on Netlify**: Netlify provides fast, continuous deployment with GitHub integration.
6. **React Context API**: Used for state management, avoiding the overhead of Redux for a small-scale app.

IMPLEMENTATION DETAILS

Technologies Used

- **Frontend**:
  - **Vite**: Fast and efficient build tool for React projects.
  - **React.js**: Component-based frontend framework.
  - **TailwindCSS**: Utility-first CSS framework.
  - **Axios**: HTTP client for API communication.
- **Backend**:
  - **Node.js**: JavaScript runtime for server-side logic.
  - **Express.js**: Web framework for building APIs.
  - **MongoDB Atlas**: Cloud-based NoSQL database.
  - **Mongoose**: ODM for MongoDB schema management.

Challenges and Solutions

1. **CORS Issues During Development**:
    - **Problem**: Cross-origin resource sharing (CORS) errors when the frontend and backend were served from different origins.
    - **Solution**: Used Vite's proxy configuration during development and properly configured CORS middleware in Express for production.
2. **Deployment Delays on Render.com**:
    - **Problem**: Initial deployment on Render.com failed due to missing environment variables.
    - **Solution**: Ensured all environment variables were correctly set before deployment.
3. **Handling Duplicate Bookings**:
    - **Problem**: Preventing the same court from being booked for overlapping time slots.
    - **Solution**: Added a unique index to the booking model in MongoDB based on court, date, and time slot.

Future Improvements

1. **User Authentication**: Implement JWT-based authentication for secure access to booking features.
2. **Role Management**: Add roles such as admin, manager, and user to control access.
3. **Notifications**: Integrate email or SMS notifications for booking confirmations.
4. **Calendar View**: Provide a calendar-based view for easier booking management.
5. **Analytics**: Add booking analytics to identify trends and resource utilization.

Conclusion

The **Sports Booking Application** provides an efficient platform for managing court bookings across multiple centers. By leveraging the MERN stack and cloud hosting, the project ensures scalability and reliability. Future improvements will focus on adding user authentication, notifications, and analytics to enhance the application's functionality.