**Table Booking: Streamlined Table Booking for Restaurants**

TableBooking is a comprehensive Full Stack web application designed to simplify the process of reserving tables at restaurants. The platform offers a convenient and user-friendly interface for customers to book tables, ensuring a seamless and hassle-free dining experience.

**Purpose of TableBooking**

The primary aim of TableBooking is to enhance the process of booking tables at restaurants by providing a centralized platform that simplifies reservations for diners. It aims to:

* **Facilitate Easy Access**: Allow diners to effortlessly find and reserve tables at various restaurants without the need for phone calls or long waiting times.
* **Improve Management**: Help restaurants manage table reservations and seating arrangements more effectively, ensuring an improved experience for all patrons.
* **Enhance Customer Experience**: Offer a reliable and intuitive service that enhances the overall dining experience for customers.

**Key Features**

**Restaurant Listings**

* **Detailed Information**: Comprehensive details about each restaurant, including location, operating dates and hours.

**Table Reservation**

* **View Available Slots**: Real-time availability of tables for different dates and times.
* **Easy Booking Process**: Simple and quick reservation process with just a few clicks.

**User Authentication**

* **Secure Registration and Login**: Ensure secure access to booking features through user authentication.

**Booking Confirmation**

* **Instant Confirmation**: Immediate confirmation of reservations.
* **E-Ticket Generation**: Generation of e-tickets for booked tables.

**Payment Integration**

* **Secure Payment Gateway**: Seamless and secure payment process for online reservations.

**Admin Dashboard**

* **Administrative Access**: Restaurant management access to update table availability.
* **Booking Management**: View and manage all reservations and customer interactions.
* **Analytics and Reports**: Insights on reservation trends and customer activity.

**Technologies Used**

**Backend**:

* **Node.js**: JavaScript runtime environment for building scalable server-side applications.
* **Express.js**: Web application framework for Node.js. Simplifies the creation of robust APIs and web servers.
* **MongoDB**: NoSQL database for storing and managing application data.

**Frontend**:

* **React**: JavaScript library for building user interfaces.

**Additional Tools and Libraries**:

* **Axios**: Promise-based HTTP client for making API requests. Simplifies communication between frontend and backend.

**Development and Build Tools**:

* **Nodemon**: Utility that automatically restarts the Node.js server on file changes during development.

**Prerequisites**

**Development Environment:**

* **Node.js**: JavaScript runtime environment required to run server-side code. [Download Node.js](https://nodejs.org/).
* **npm (Node Package Manager):** Package manager for Node.js. Command: `npm install`.
* **MongoDB:** NoSQL database for storing application data. [Download MongoDB](https://www.mongodb.com/try/download/community).
* **MongoDB Compass**: GUI for managing MongoDB databases. [Download MongoDB Compass](https://www.mongodb.com/products/compass).

**Code Editor:**

* **Visual Studio Code (VS Code):** Popular code editor with support for JavaScript and Node.js. [Download VS Code](https://code.visualstudio.com/).

**Development and Build Tools:**

* **Nodemon:** Utility that automatically restarts the Node.js server on file changes.

**Project Structure**

**Backend (Node.js & Express.js):**

* **node.js:** Entry point of the backend server.
* **config/:** Configuration files (e.g., database connection).
* **models/:** Mongoose models.
* **routes/:** API routes.
* **controllers/:** Request handling logic.

**Frontend (React):**

* **src/:** Main source directory.
* **components/:** Reusable UI components.
* **styles/**: CSS or styled-components.

**API Documentation**

**Authentication**:

* **POST/api/auth/register**: Register a new user.
* **POST /api/auth/login**: User login.

**User**:

* **GET /api/users**: Get all users.
* **GET /api/users/:id**: Get user by ID.
* **PUT /api/users/:id**: Update user by ID.
* **DELETE /api/users/:id**: Delete user by ID.

**Acknowledgement**

Lastly, we acknowledge all the open-source projects and developers whose work has inspired us and provided a foundation for the development of Tablebooking. Your contributions to the developer community are greatly appreciated.

Thank you all for your support and contributions to the TableBooking project!