Evan Christianson
Aaliyah Kurrelmeyer

Reservation Maker

Reservation Maker (name not final) is a web application designed to streamline the reservation process for diners and facilitate efficient reservation management for restaurant staff. The system will be built using Node.js for the backend, React for the frontend, and will leverage RESTful APIs for communication. Docker and microservices architecture will be employed for scalability and maintainability. Our initial focus will be on the restaurant industry, but the fundamentals of this application can be applied to several industries (clinics, hair stylists, dental appointments, etc.).

Key Features:

User Authentication:
- Users can create accounts or log in using their credentials.
- Two user roles: customers and restaurant staff.

Reservation Management:
- Customers can make, view, and cancel reservations.
- Real-time availability updates to prevent overbooking.
- Reservation details include username, real name, phone number, party size and reservation time.

User Profiles:
- Users can update their profiles, including contact information.
- Preferences and past reservations can be stored for a personalized experience.

Dashboard for Restaurant Staff:
- Restaurant staff can log in to a dedicated dashboard.
- Comprehensive view of all reservations with search and filter options.
- Ability to manage reservation statuses and update availability.

Notifications:
- Automated email or SMS notifications for users upon successful reservation.
- Reminders for upcoming reservations.

Scalability with Docker:
- Containerization of services using Docker for easy deployment and scalability.

Technology Stack:

- **Frontend:**
  - React.js for dynamic and responsive user interfaces.
- **Backend:**
  - Node.js for server-side logic and API development.
- **APIs:**
  - RESTful APIs for communication between frontend and backend.
- **Database:**
  - MongoDB for storing user data and reservations.
- **Authentication:**
  - Implement secure authentication using JSON Web Tokens.
- **Microservices:**
  - Divide functionalities into microservices for modularity and scalability.
- **Containerization:**
  - Docker for containerizing the application components.

Development Phases:

Backend Development:
- User authentication and profile management.
- Reservation CRUD (create, read, update, delete) operations.

Frontend Development:
- User interfaces for registration, login, and reservation management.
- Integration with backend APIs.

Dashboard for Restaurant Staff:
- Backend logic for staff dashboards.
- Frontend implementation for staff views.

Notifications:
- Implement automated notifications for users and staff.

Testing:
- Unit testing, integration testing, and end-to-end testing.

Deployment:
- Dockerize microservices for deployment in a production environment.

Documentation:
- Create comprehensive documentation for future maintenance and updates.

Conclusion:

The Restaurant Reservation System aims to provide a seamless and user-friendly experience for customers while offering efficient reservation management tools for restaurant staff. The use of modern technologies and microservices architecture ensures scalability and adaptability to evolving business needs. The fundamentals of this application can be further applied to non-restaurant reservation needs in the future.