**FOOD ORDERING WEBSITE DESIGN DOCUMENT**

**Table of Contents**

1. Introduction

2. Project Overview

3. Objectives

4. Scope

5. Requirements

6. Architecture

7. User Interface Design

8. Data Model

9. Testing Strategy

10. Deployment Plan

11. Maintenance Plan

12. Glossary

**1.INTRODUCTION**

In the fast-paced world of dining and culinary experiences, a seamless and efficient food ordering system is essential for both customers and restaurant owners. Traditional methods often fall short in providing a user-friendly and robust solution. The Food Ordering Website project aims to address this gap, utilizing modern web technologies to create a streamlined platform for customers to order food online.

**2.PROJECT OVERVIEW**

The Food Ordering Website is designed to offer a convenient and user-friendly interface for customers to browse restaurant menus, place orders, and track deliveries. Employing the MERN (MongoDB, Express.js, React.js, Node.js) stack, the project integrates various features to enhance the overall online food ordering experience.

**3. Objectives**

**1. User-Friendly Menu Browsing:**

* + Customers can easily navigate and browse restaurant menus.
  + Detailed information, including dish descriptions and prices, is provided.

**2. Efficient Order Placement:**

* + Customers can add items to their cart and place orders seamlessly.
  + Options for customization and special requests are incorporated.

**3. Real-Time Order Tracking:**

* + Integration with GPS technology allows customers to track the status of their orders in real-time.

**4. User Account Management:**

* + Customers can create accounts, manage personal details, and view order history.
  + Secure login ensures privacy and data protection.

**5. Restaurant Management Portal:**

* + Restaurant owners have access to an admin portal to manage menus, view orders, and update their restaurant details.

**4. Scope**

* **Efficiency:**
  + Streamlines the food ordering process, saving time for both customers and restaurants.
* **Accuracy:**
  + Real-time order tracking reduces errors and ensures accurate delivery.
* **Transparency:**
  + Provides customers with visibility into their orders and expected delivery times.
* **Communication:**
  + Facilitates communication between customers and restaurants for order modifications or inquiries.

**5. Requirements**

*Functional Requirements:*

**Customer Portal:**

1. Browse menus, place orders, and track deliveries.

2. View and update personal details.

**Restaurant Portal:**

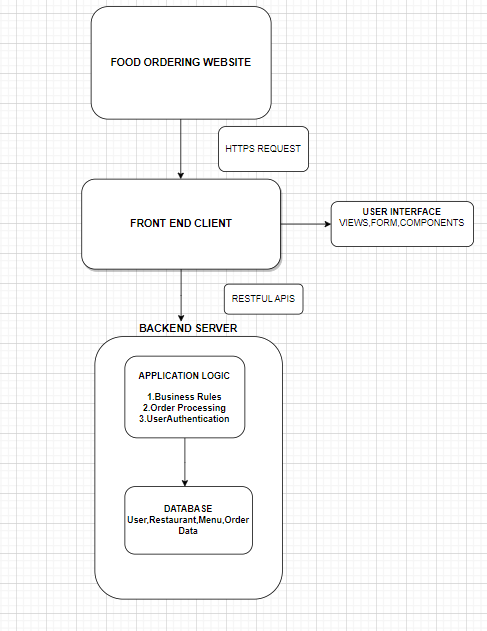
1. Manage menus, view and process orders.

2. Update restaurant information.

**6. Architecture**

The system architecture comprises:

* Frontend: React.js for customer and restaurant portals.
* Backend: Node.js and Express.js for API development.
* Database: MongoDB for storing user and order data.
* Integration with GPS for real-time tracking.

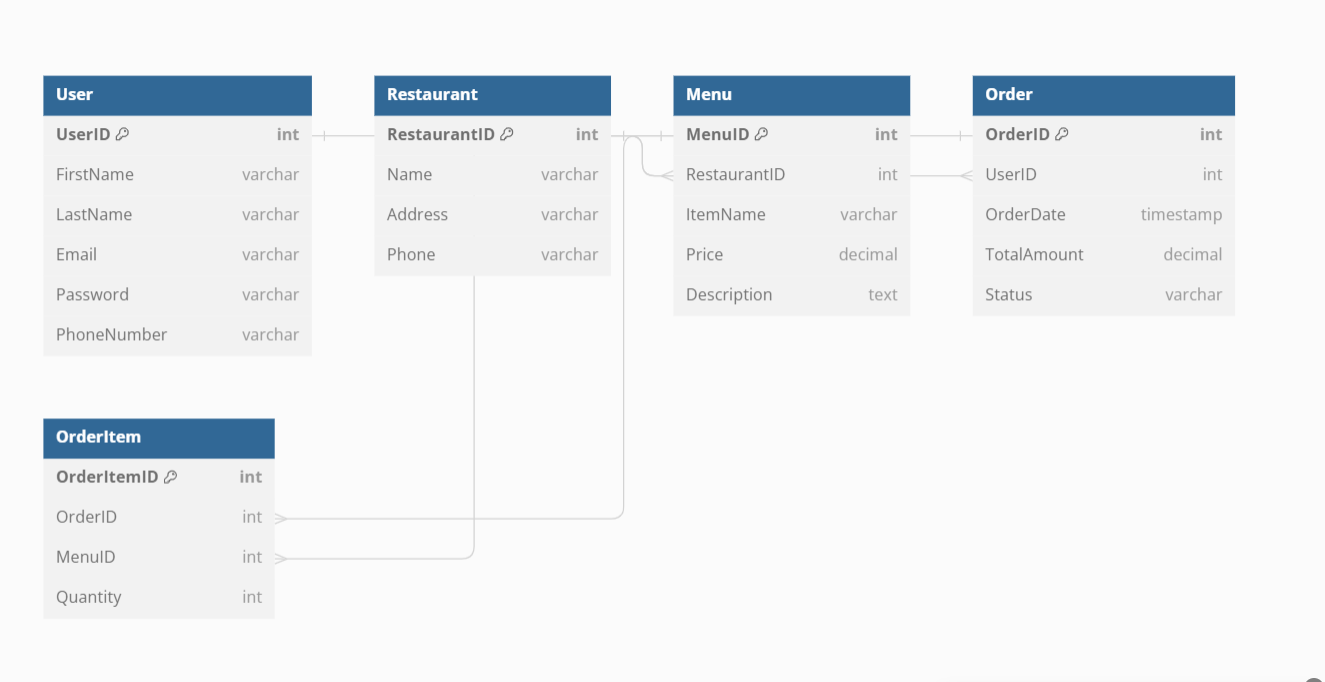


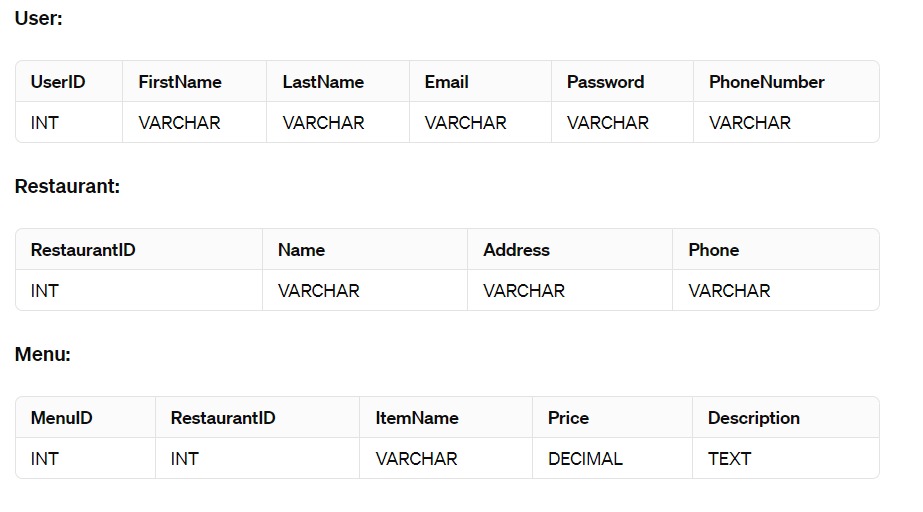
**7. User Interface Design**

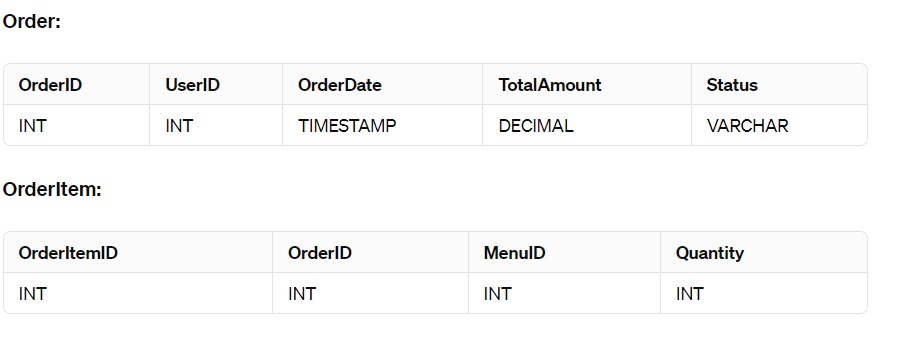
The user interface design includes:

* Customer Portal: Intuitive design for easy menu browsing and order placement.
* Restaurant Portal: Dashboard for managing menus and processing orders.
* Responsive design for accessibility across devices.

**8. Data Model**







**9. Testing Strategy**

* Unit testing for frontend and backend components.
* Integration testing for order processing and tracking.
* User acceptance testing for overall functionality.

**10. Deployment Plan**

* Deploy frontend and backend on a cloud platform like Heroku.
* Configure GPS integration for real-time order tracking.

**11. Maintenance Plan and API Used**

* **Payment Gateway API:**
  + Stripe API
  + PayPal REST API
  + Braintree API
* **Location and Mapping APIs:**
  + Google Maps API
  + Mapbox API
* **Food Menu and Restaurant Data APIs:**
  + Zomato API
  + OpenTable API
* **SMS and Email Notification APIs:**
  + Twilio API
  + SendGrid API
* **User Authentication and Authorization APIs:**
  + Firebase Authentication API
  + OAuth APIs: Google Sign-In, Facebook Login
* **Order Management APIs:**
  + Square API
  + Uber Eats API
* **Push Notification APIs:**
  + OneSignal API
  + Firebase Cloud Messaging (FCM)
* **Inventory and Stock Management APIs:**
  + TradeGecko API
  + QuickBooks API
* **Reviews and Ratings APIs:**
  + Yelp Fusion API
  + Trustpilot API

**12. Glossary**

**MERN Stack: MongoDB, Express.js, React.js, Node.js**

* A set of technologies used for building modern, scalable web applications.