Food Ordering System

Software Requirements Specification

Version 1.0

24th September 2023

Ashok Kumar Ganji
Raajitha Vardhanapu
Jitender Singh
Chandra Priya Kavalapti
Raghupathi Reddy Rupanagudi
Harshitha Singamala
Sravanthi Thukivakam

Submitted in partial fulfilment
Of the requirements of
CSIS 44-691 Graduate Directed Project 1

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date

Table of Contents

Table of Contents Page Number

1.Introduction

1.1.Purpose

1.2.Scope

1.3.Definitions, Acronyms, and Abbreviations

1.4.References

1.5.Overview

2.General Description

2.1.Product Perspective

2.2.Product Functions

2.3.User Characteristics

2.4.General Constraints

2.5.Assumptions and Dependencies

3.Specific Requirements

3.1.External Interface Requirements

3.1.1.User Interfaces

3.1.2.Hardware Interfaces

3.1.3.Software Interfaces

3.1.4.Communications Interface

3.2. Functional Requirements

3.3. Use Cases

3.4. Class/Objects

3.5. Non-Functional Requirements

3.5.1. Performance

3.5.2. Reliability

3.5.3. Availability

3.5.4. Security

3.5.5. Portability

3.6. Inverse Requirements

3.7. Design Constraints

3.8. Logical Database Requirements

3.9. Other Requirements

3.10. Prototypes (for complete project)

3.11. Use Case Diagrams

4. Design

4.1. ER diagram

4.2. GUI

5.Analysis Models

5.1. Data Flow Diagram

5.2. Data Sequence Diagram

1. Introduction

1.1. Purpose

Food Ordering System is a food delivery and logistics platform that connects customers with local restaurants and food establishments. Its primary purpose is to facilitate the delivery of restaurant meals directly to customers' doorsteps. The company operates through a website where customers can place orders from a wide range of restaurants and stores in their area.

1.2. Scope

The scope of the Food Ordering System encompasses the design and development of a user-friendly, reliable, and efficient platform that facilitates the ordering and delivery of food items from local restaurants to customers. The application aims to provide an intuitive interface for customers to browse restaurant menus, place orders, track deliveries, and make payments. Additionally, it encompasses a user-friendly interface for restaurant partners to manage orders and coordinate deliveries.

1.3. Definitions, Acronyms, and Abbreviations

- Food Ordering System: The web application developed to enable customers to order food and track deliveries.
- Customer: The end user who utilizes the application to place orders for food.
- Restaurant Partner: The establishments that collaborate with the Food Ordering System to offer their menus and fulfill orders.
- Delivery Driver: The individuals responsible for delivering orders to customers.
- UI: User Interface
- UX: User Experience
- API: Application Programming Interface

1.4. References

- DoorDash Official Website (https://www.doordash.com)
- DoorDash API Documentation (https://developer.doordash.com)
- Web Application Design (https://developer.mozilla.org/en-US/docs/Web/Design)

1.5. Overview

The subsequent sections of this document provide detailed information about the general description of the Food Ordering System, including its product perspective, user characteristics, and constraints. It also outlines the specific requirements, including external interface requirements, functional requirements, use cases, and non-functional requirements such as performance, reliability, availability, and security.

2. General Description

2.1. Product Perspective

The Food Ordering System is designed as a comprehensive platform to connect customers, restaurant partners, and delivery drivers, facilitating the seamless ordering and delivery of food. It operates within the context of the broader food delivery and restaurant industry, serving as an intermediary between customers seeking restaurant-quality food delivery and restaurants eager to expand their reach to a wider audience. The application enhances the convenience of food delivery by providing an easy-to-use interface that streamlines the process from order placement to doorstep delivery.

2.2. Product Functions

- Customer Ordering: Customers can browse restaurant menus, place orders, customize their selections, and make payments.
- Restaurant Management: Restaurant partners can manage their menus, track incoming orders, and coordinate with delivery drivers.
- Delivery Coordination: Delivery drivers can accept and manage delivery requests, navigate to the destination, and provide real-time updates.

- Order Tracking: Customers can track the status of their orders, including preparation, pickup, and delivery.
- Payment Processing: The application securely processes payments from customers and facilitates payouts to restaurant partners and delivery drivers.
- Rating and Feedback: Customers can provide ratings and feedback on their dining experience.

2.3. User Characteristics

The DoorDash application caters to three primary user categories:

- Customers: Individuals seeking convenient food delivery options.
- Restaurant Partners: Establishments interested in expanding their reach through the DoorDash platform.
- Delivery Drivers: Individuals interested in earning income by delivering orders to customers.

2.4. General Constraints

The Food Ordering System operates within the constraints of network connectivity, device compatibility, and regional coverage. It relies on accurate menu information, order details, and real-time delivery tracking to provide a positive user experience. Furthermore, the application adheres to data privacy regulations to ensure user information security.

2.5. Assumptions and Dependencies

The successful development and operation of the Food Ordering System web application are based on the following assumptions:

- Availability of consistent and reliable internet connectivity for all users.
- Integration with trusted third-party payment gateways for secure transactions.
- Willingness of restaurant partners to collaborate and maintain up-to-date menus and order status.

- Cross-device compatibility to cater to users on various platforms and operating systems.

The application's functionality hinges on seamless integration between frontend user interfaces, backend server components, and external APIs responsible for location services and payment processing.

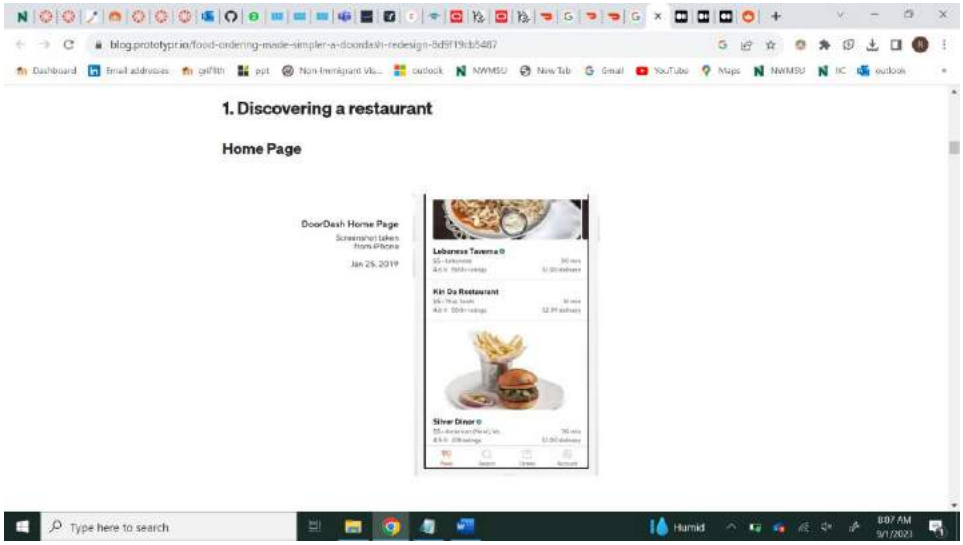
3.Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

- **Registration Form**
- **Login Page**
- **Home Screen:** This is where users can search for restaurants, browse menus, and place orders.
- **Restaurant Listings:** Displays a list of available restaurants, including their names, ratings, delivery times, and cuisine types.
- **Menu Screen:** Shows the menu items of a selected restaurant, with descriptions, prices, and options for customization.
- **Cart:** Allows users to review and modify their orders before proceeding to checkout.
- **Checkout:** The interface where users provide delivery details, payment information, and place their order.
- **Logout Page**

- For Reference



Link: https://blog.prototypr.io/food-ordering-made-simpler-a-doordash-redesign-8d9f19cb5487

https://myewati.wordpress.com/srs-2/

Driver Interface:

- **Order Acceptance:** Drivers see incoming orders with details such as the restaurant location, customer address, and order items.
- **Navigation:** Integrates with GPS for route guidance to the restaurant and customer's address.
- **Order Status:** Provides information on order status, including when the order is picked up and delivered.
- **Earnings and Reports:** Allows drivers to track their earnings, view order history, and access reports.

Restaurant Interface:

- **Menu Management:** Enables restaurants to create and manage their menus, including item descriptions, prices, and availability.
- **Order Management:** Displays incoming orders, allows confirmation, preparation, and marking orders as completed.
- **Account Management:** Provides access to restaurant profiles, settings, and payment details.

Admin Dashboard:

- **User Management:** Allows administrators to manage users, including customers, drivers, and restaurant partners.
- **Order Monitoring:** Provides an overview of orders, their status, and any issues that may arise during delivery.
- **Reporting and Analytics:** Offers insights into key performance metrics, sales, and trends.
- **Customer Support:** Integrates customer support tools for addressing user queries and issues.

3.1.2. Hardware Interfaces

Here are some of the key hardware interfaces and components involved in the application:

- **Smartphones and Tablets:** Customers and delivery drivers use their smartphones or tablets to access the app. These devices have touchscreens and cameras, which are used for placing orders, tracking deliveries, and capturing photos of food items.

- **GPS Receivers:** GPS (Global Positioning System) is crucial for application as it helps in tracking the location of delivery drivers in real-time. GPS receivers in smartphones provide accurate location information.

- **Internet Connectivity:** Reliable internet connectivity is essential for both customers and delivery drivers to access the app and its services.

- **Payment Terminals:** Restaurants and customers use payment terminals and card readers for processing payments made through credit/debit cards, digital wallets, or other online payment methods.

- **Restaurant Equipment:** Within restaurants, orders are received through a dedicated tablet or computer. These devices are used to manage and prepare orders for pickup or delivery.

- **Delivery Bags:** Delivery drivers use insulated delivery bags to keep food orders warm or cold during transportation. These bags are designed to maintain the quality of food items during transit.

- **Vehicles:** Delivery drivers may use bicycles, motorcycles, scooters, or cars to transport orders. The type of vehicle depends on the location and delivery preferences of the driver.

- **Thermal Printers:** Some restaurants and delivery hubs may use thermal printers to generate receipts and order tickets.

- **Smart Locks and Doorbell Cameras:** Some customers use smart locks and doorbell cameras to facilitate contactless deliveries. Delivery drivers may interact with these devices to leave orders securely at the customer's doorstep.

3.1.3. Software Interfaces

- Any window-based operating system with DOS support are primary requirements for software development.
- The systems must be connected to the internet.
- **Maps and Navigation:**
  Integration with mapping and navigation APIs (e.g., Google Maps, Mapbox) for route optimization and directions.
- **Order Processing:**
  Integration with order management systems for processing incoming orders, assigning drivers, and notifying restaurants.
- **Payment Gateway:**
  Integration with a secure payment gateway to handle transactions between customers, restaurants, and drivers.

3.1.4. Communications Interface

- **Phone and Messaging Services:** Relies on phone and messaging services for communicating with customers and drivers through calls, text messages, and in-app messaging.
- **Wearable Devices:** Some delivery drivers may use wearable devices such as smartwatches for receiving notifications and updates related to orders and deliveries.

3.2. Functional Requirements

- **User Registration and Authentication:**
  Users should be able to create accounts and log in securely.
  User authentication is essential for verifying the identity of customers, drivers, and restaurant owners.
- **Menu Browsing and Ordering:**

Customers should be able to browse restaurant menus, view item details, prices, and add items to their cart.

Customers must have the ability to customize their orders, specify delivery preferences (e.g., delivery address, time), and place orders.

- **Order Management and Tracking:**

  Once an order is placed, customers should receive real-time updates on the status of their order.

  Restaurant owners should have a dashboard to accept or reject orders and update order preparation status.

  Drivers need access to order details, including pickup and delivery information, and should be able to update order status as they progress through the delivery process.

- **Payment Processing:**

  The application should securely handle payment transactions, supporting various payment methods such as credit/debit cards, digital wallets, and cash on delivery.

  The system must calculate and display the total order cost, including taxes, delivery fees, and any applicable discounts or promotions.

- **Rating and Review System:**

  Customers should be able to rate and leave reviews for restaurants and delivery Driver.

  This system helps maintain quality standards and provides valuable feedback to improve the service.

3.3. Use Cases

Restaurant Orders: Allow users to browse menus, place orders, and pay for food from their favorite local restaurants.

Scheduled Deliveries: Enable users to schedule food deliveries for later, making it convenient for meal planning.

Customizable Orders: Offer options for customizing orders, including special requests and dietary preferences.

Real-time Tracking: Provide live order tracking so customers can monitor their delivery's progress.

User Reviews and Ratings: Allow customers to leave reviews and ratings for restaurants and delivery drivers.

Promotions and Discounts: Offer promotions, discounts, and loyalty programs to attract and retain customers.

Multi-platform Access: Ensure compatibility with various devices, including smartphones, tablets, and desktops.

Driver Management: Streamline the onboarding and management of delivery drivers to ensure efficient order fulfillment.

3.4 Class/Objects

User: Represents registered users of the application. Contains information like name, contact details, and order history.

Restaurant: Represents the various restaurants available for food delivery. Contains information such as name, menu, location, and opening hours.

Order: Represents a user's order. Contains details about the items ordered, delivery address, payment information, and order status.

Menu Item: Represents individual items on a restaurant's menu. Contains details like name, description, price, and availability.

Cart: Represents a user's shopping cart, where they can add, remove, and modify items before placing an order.

Payment: Represents payment methods and transactions. Contains information about credit cards, digital wallets, and payment processing.

Delivery Driver: Represents the delivery personnel who transport orders from the restaurant to the user's location. Contains information like name, contact details, and current delivery status.

Review/Rating: Represents user reviews and ratings for both restaurants and delivery drivers. Contains feedback and rating scores to help improve the overall service.

3.5. Non-Functional Requirements

3.5.1. Performance

Response Time: The application should provide quick responses to user actions, such as searching for restaurants, placing orders, and tracking deliveries.

Load Balancing: Distribute incoming traffic evenly across multiple servers to prevent overloads on any single server.

3.5.2. Reliability

Availability: The application should be available 24/7 with minimal downtime for maintenance.

Fault Tolerance: The system should continue to function and recover gracefully from errors or failures.

Data Integrity: Ensure that user data, orders, and payments are accurate and secure.

3.5.3. Availability

The system should always be accessible, which means that a user can do so by using a web browser.

A substitute page will be displayed in the event of hardware failure or database corruption, and the administrator should download and store database backups from the server.

After that, the service will restart. It denotes availability every single day of the week.

3.5.4. Security

- The system uses SSL (secured socket layer) in all transactions that include any confidential customer information.
- After a certain amount of inactivity, the system needs to log off all users automatically.
- The system shouldn't save any cookies revealing the user's password on the client's computer.

3.5.5. Portability

Cross-Platform Compatibility:

The application should be compatible with multiple operating systems and devices, including iOS, Android, and web browsers, to ensure a broad user reach. It should adapt seamlessly to different screen sizes and resolutions.

Browser Compatibility:

For web-based versions of the application, it should be compatible with major web browsers (e.g., Chrome, Firefox, Safari, Edge) to provide a consistent user experience across different browsers and versions.

3.6 Inverse Requirements

Inverse requirements are also known as anti-requirements or non-requirements, describing what a system should explicitly avoid or not do. These requirements are typically formulated to prevent undesirable outcomes, mitigate risks, or ensure that certain negative scenarios do not occur. It is an important aspect of comprehensive requirements engineering as they help to clarify and manage project constraints and limitations.

- **No Downtime:** The website should avoid any unplanned or extended periods of downtime, ensuring continuous service availability.
- **No Unauthorized Access:** Prevent unauthorized access to user accounts, ensuring data security and privacy.
- **No Data Loss:** Avoid data loss by implementing robust data backup and recovery mechanisms.
- **No Misleading Information:** Avoid providing inaccurate or misleading information about menus, prices, and delivery times.
- **No Overcharging:** Prevent overcharging customers by accurately calculating prices, taxes, fees, and discounts.
- **No Excessive Complexity:** Keep the website's design and features user-friendly and straightforward.
- **No Discrimination:** Ensure the platform does not discriminate against any users.

3.7 Design Constraints

Design constraints are limitations on a design. These include imposed limitations that you don’t control, and limitations that are self-imposed to improve a design. They provide guidance for the

development process and help ensure a user-friendly, secure, and scalable platform that can adapt to changing needs and regulations.

- **Geographical Coverage:** Define the specific regions or cities where the platform will operate, which impacts restaurant partnerships and delivery logistics.
- **Payment Methods:** Determine accepted payment methods (credit cards, digital wallets) and integrate secure payment gateways.
- **Menu Complexity:** Set limits on the number of menu items and variations allowed per restaurant to maintain consistency and efficiency.
- **Data Security:** Implement robust security measures to protect user data, payment information, and restaurant details.
- **Delivery Logistics:** Design efficient delivery routing and tracking systems to optimize delivery times and costs.
- **Feedback and Ratings:** Implement mechanisms for user reviews and ratings to maintain quality control.
- **Regulatory Compliance:** Adhere to food safety, delivery, and online transaction regulations in different regions of operation.

3.8 Logical Database Requirements

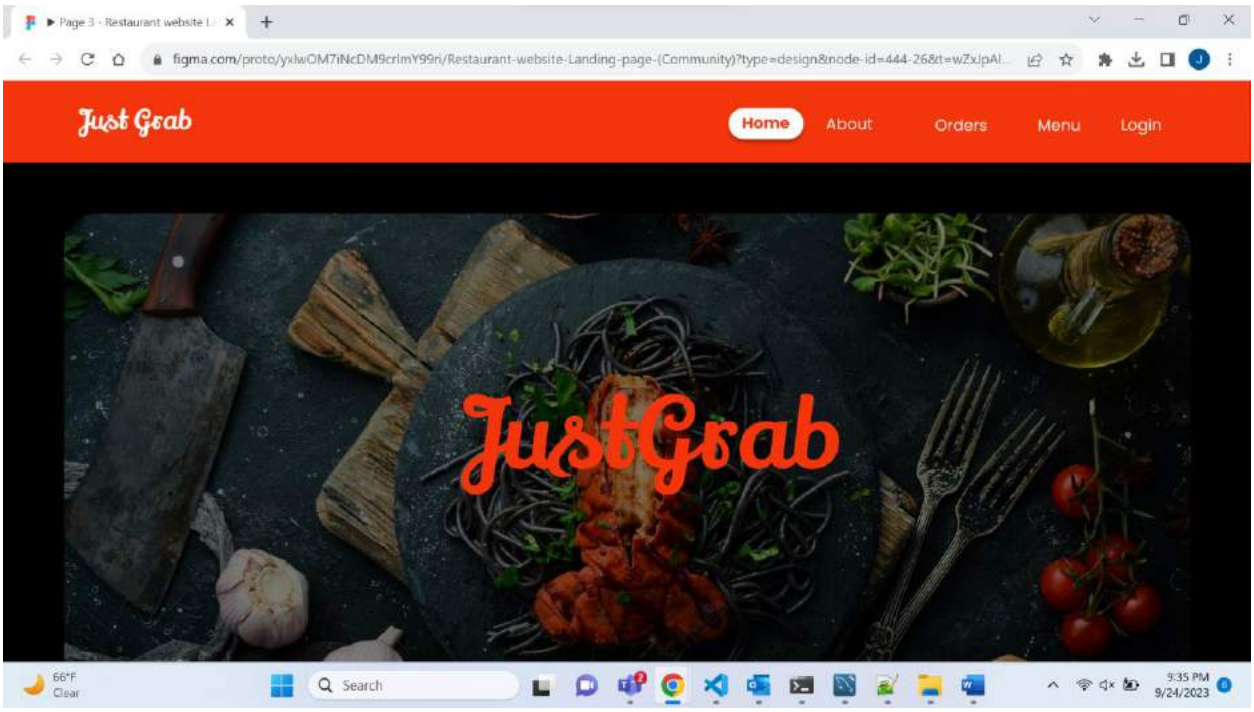
- **User Profiles:** Store user data, including personal information, delivery addresses, order history, and payment details.
- **Menu Items:** Maintain a database of available food items, including names, descriptions, prices, and dietary information.
- **Orders and Transactions:** Record order details, such as item selections, quantities, timestamps, and payment status.
- **Restaurants and Partners:** Store information about partnering restaurants, including menus, locations, and availability.
- **Ratings and Reviews:** Collect user reviews and ratings for restaurants and food items.
- **Delivery Tracking:** Track the status and location of food deliveries in real time.
- **Coupons and Promotions:** Manage promotional codes and discounts for users.
- **Notifications:** Store user preferences for receiving order updates and notifications.
- **Customer Support:** Maintain records of user inquiries, complaints, and support interactions.

3.9 Other Requirements: Certainly, in addition to design constraints, here are some other key requirements.

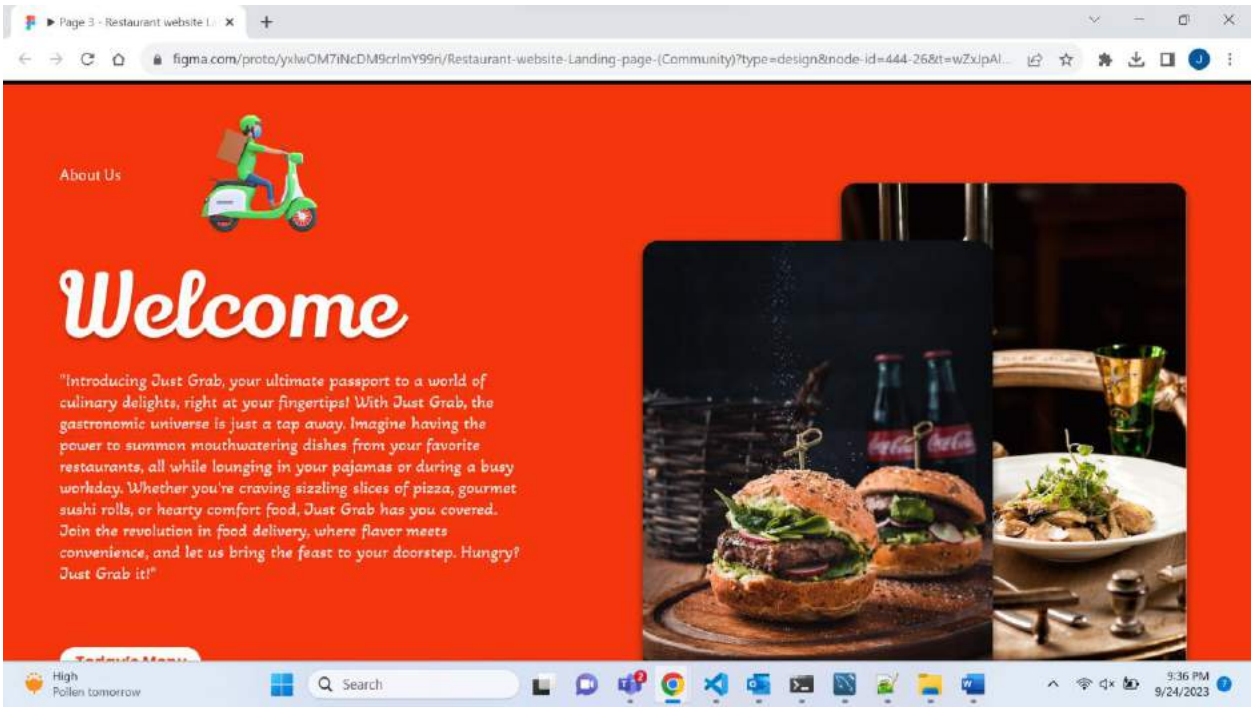
- **User Registration and Profiles:** Users should be able to register, create profiles, and manage their account information.
- **Restaurant Listings:** Display a comprehensive list of restaurants, including menus, prices, and ratings.
- **Ordering and Checkout:** Provide a user-friendly order placement process with secure payment options and order tracking.
- **Search and Filters:** Allow users to search for specific cuisines, dishes, or restaurants, and apply filters to narrow down choices.
- **Reviews and Ratings:** Enable users to leave reviews and ratings for restaurants and food items.
- **User Notifications:** Send order confirmations, updates, and delivery status notifications via email or SMS.
- **Feedback and Surveys:** Collect user feedback through surveys and ratings to improve service quality.
- **Data Backup and Recovery:** Implement data backup and recovery procedures to prevent data loss.

3.10 Prototype Initial Version:

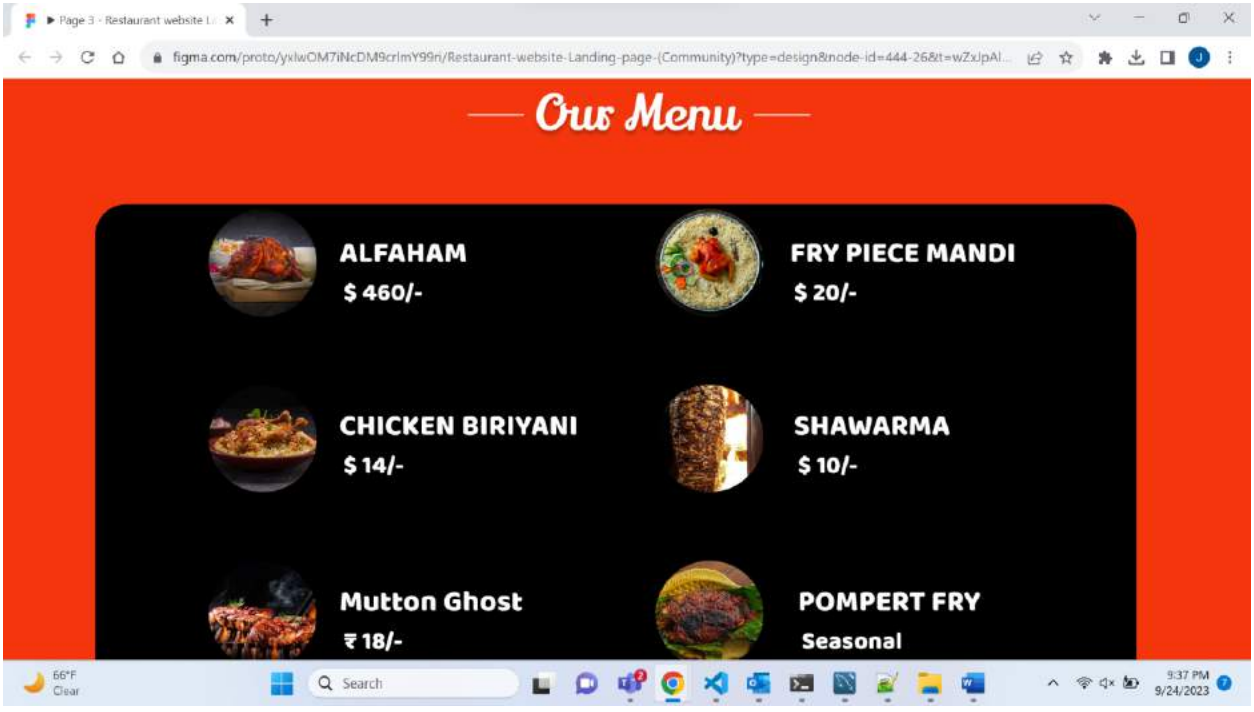
Landing Page:



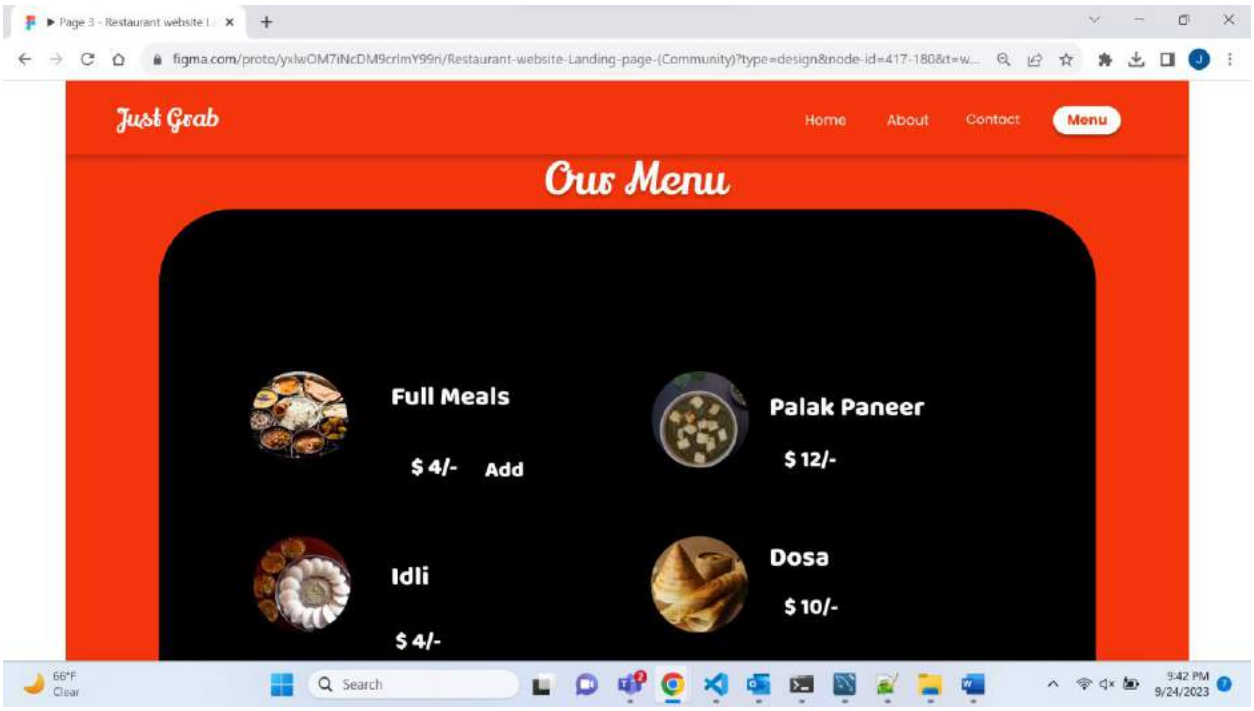
Food Ordering System



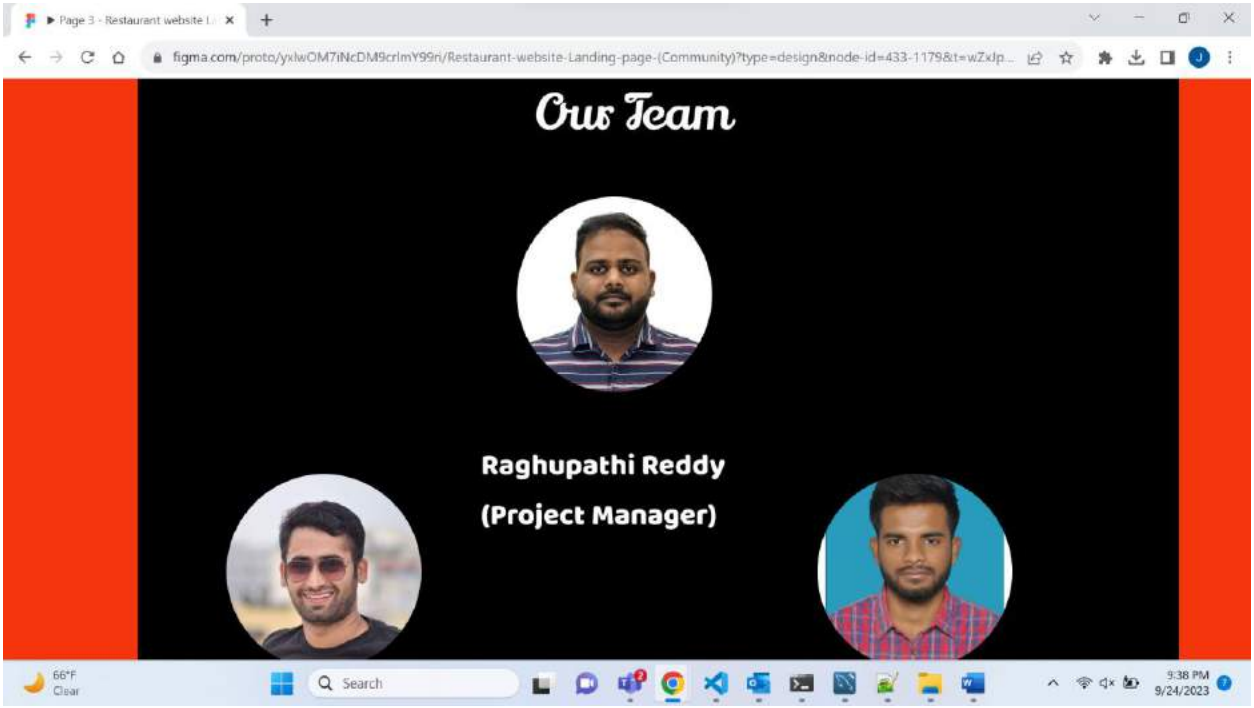
Menu Pages:



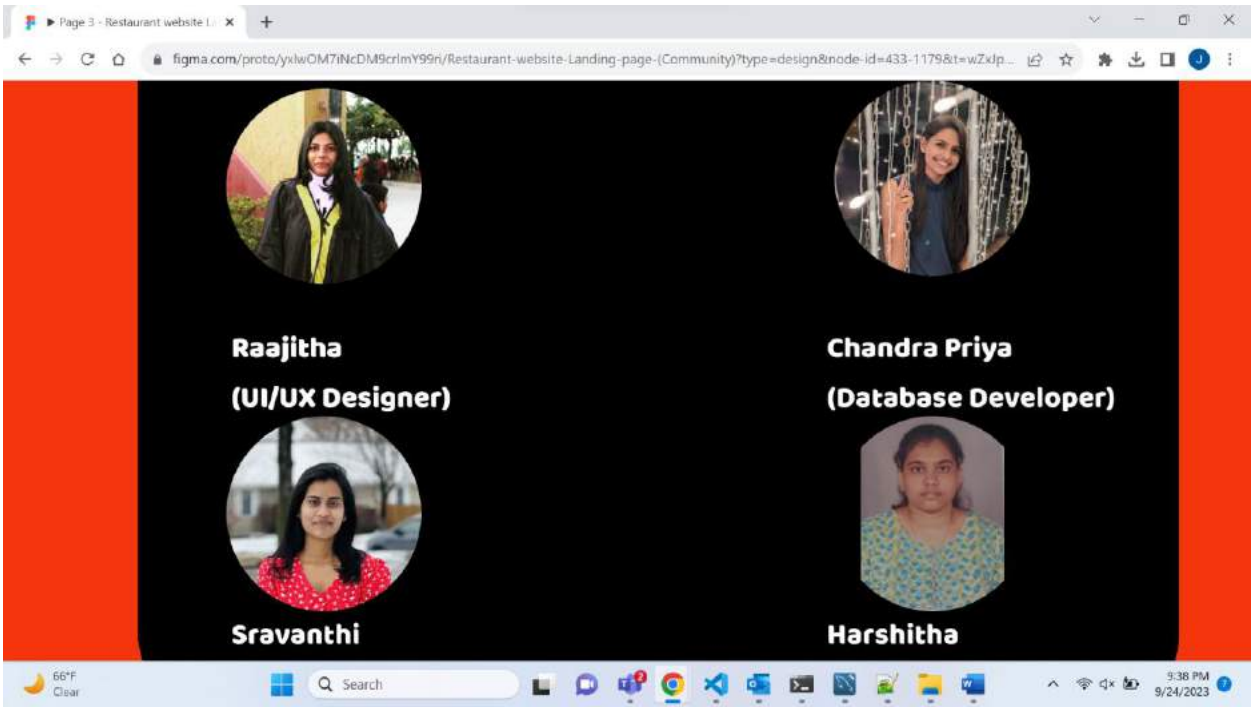
Food Ordering System



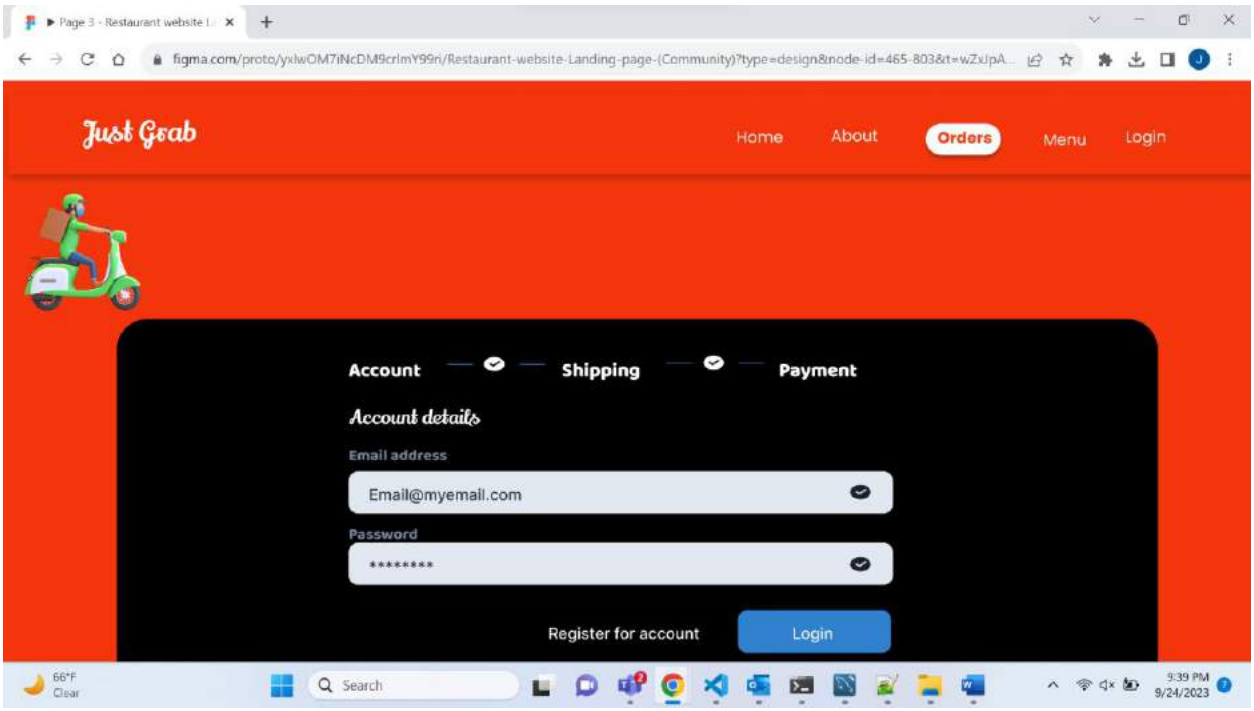
Our Team Pages:



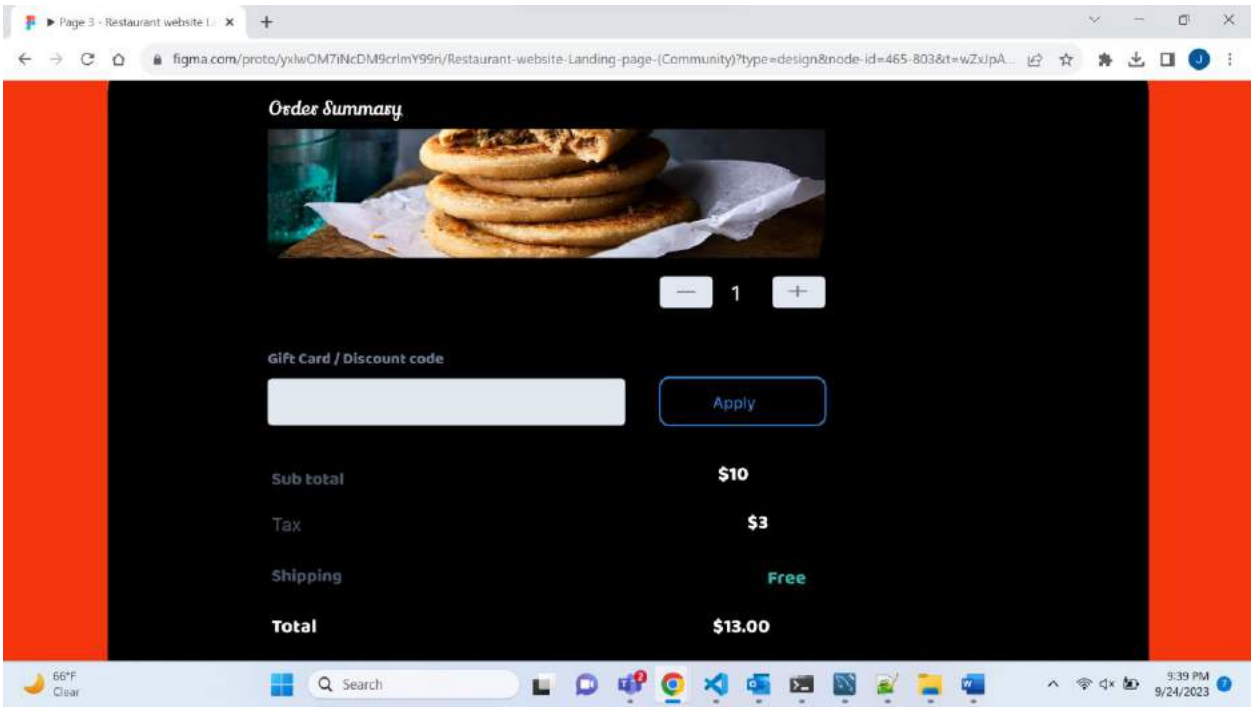
Food Ordering System



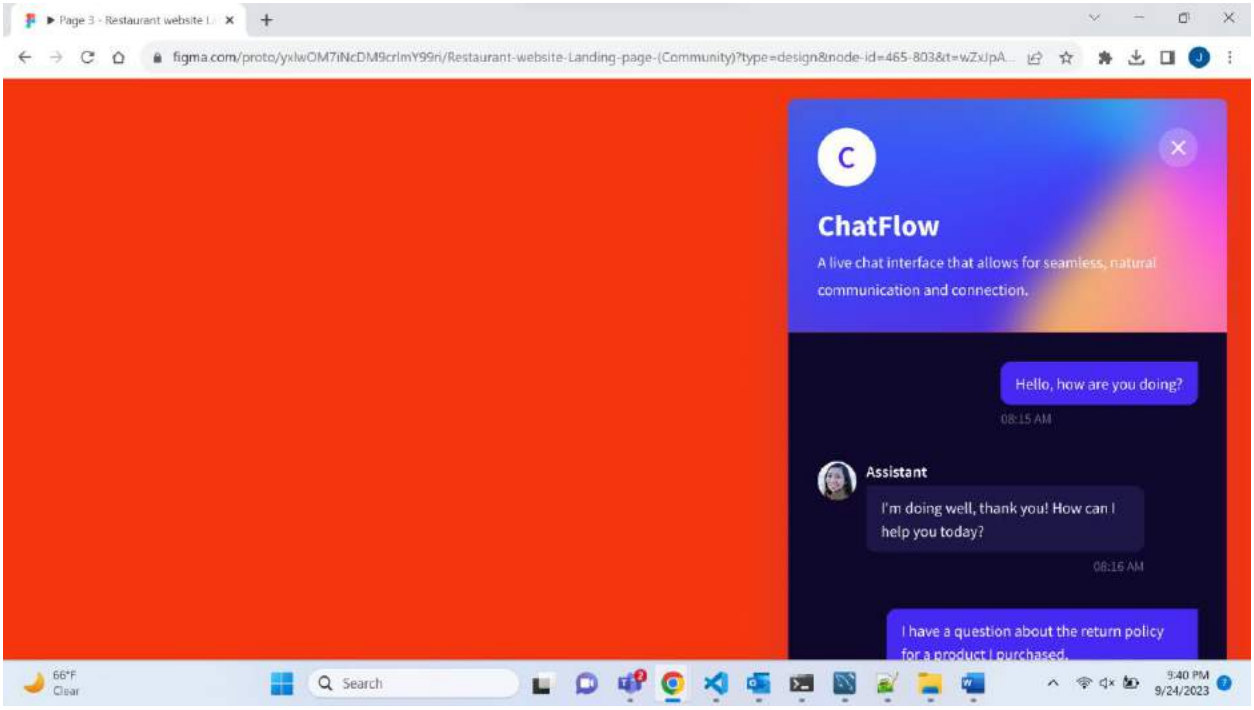
Order Pages:



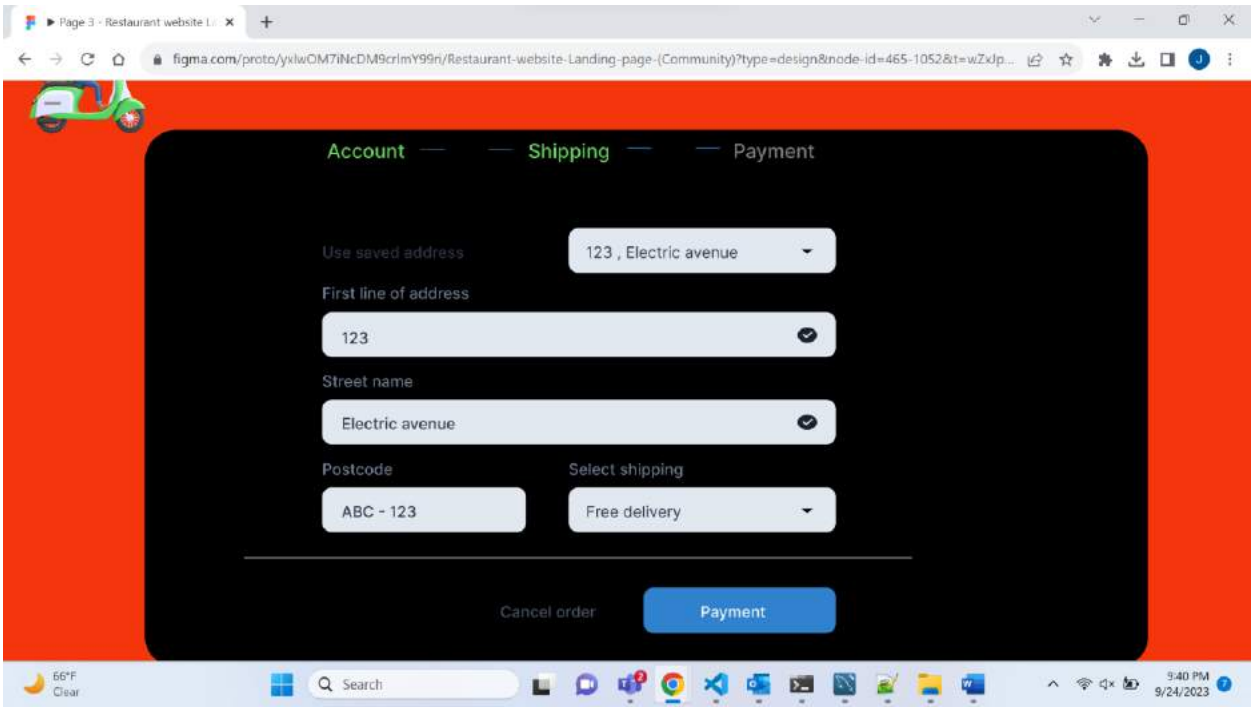
Food Ordering System



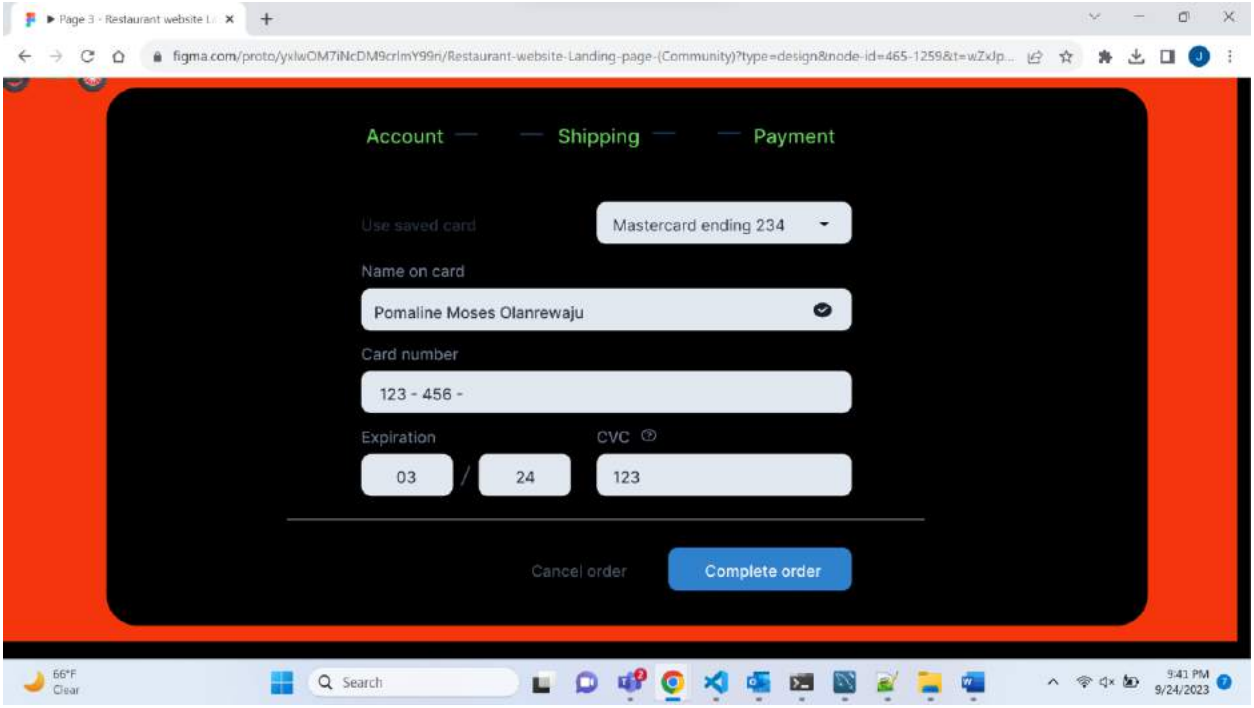
Chat Pages:



Food Ordering System

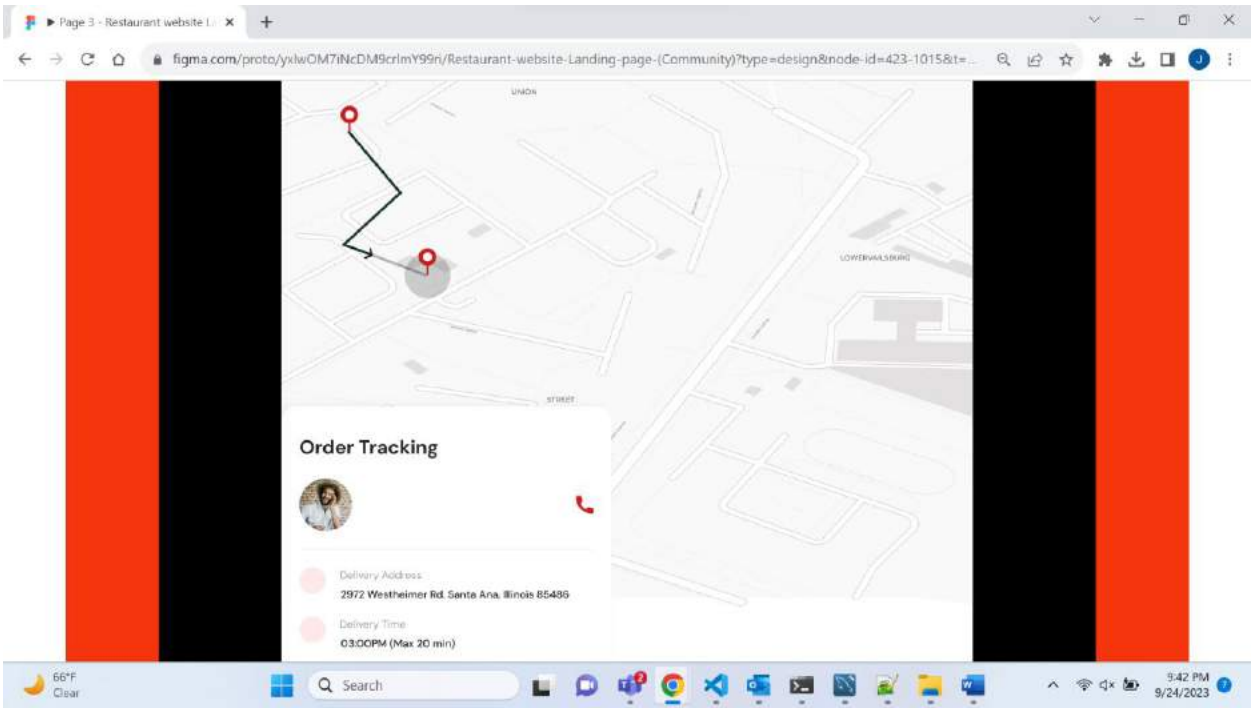


Payment Pages:

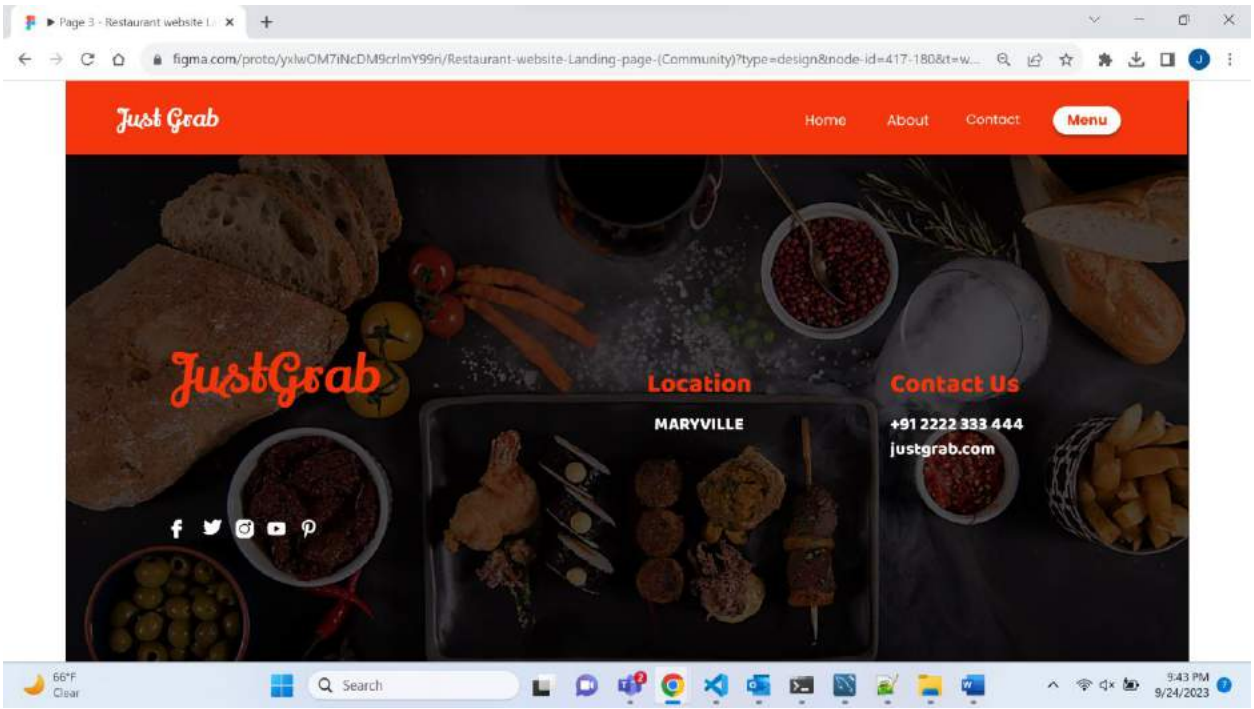


Food Ordering System

Order tracking Pages:



Footer Page of the Applicaion:



Food Ordering System

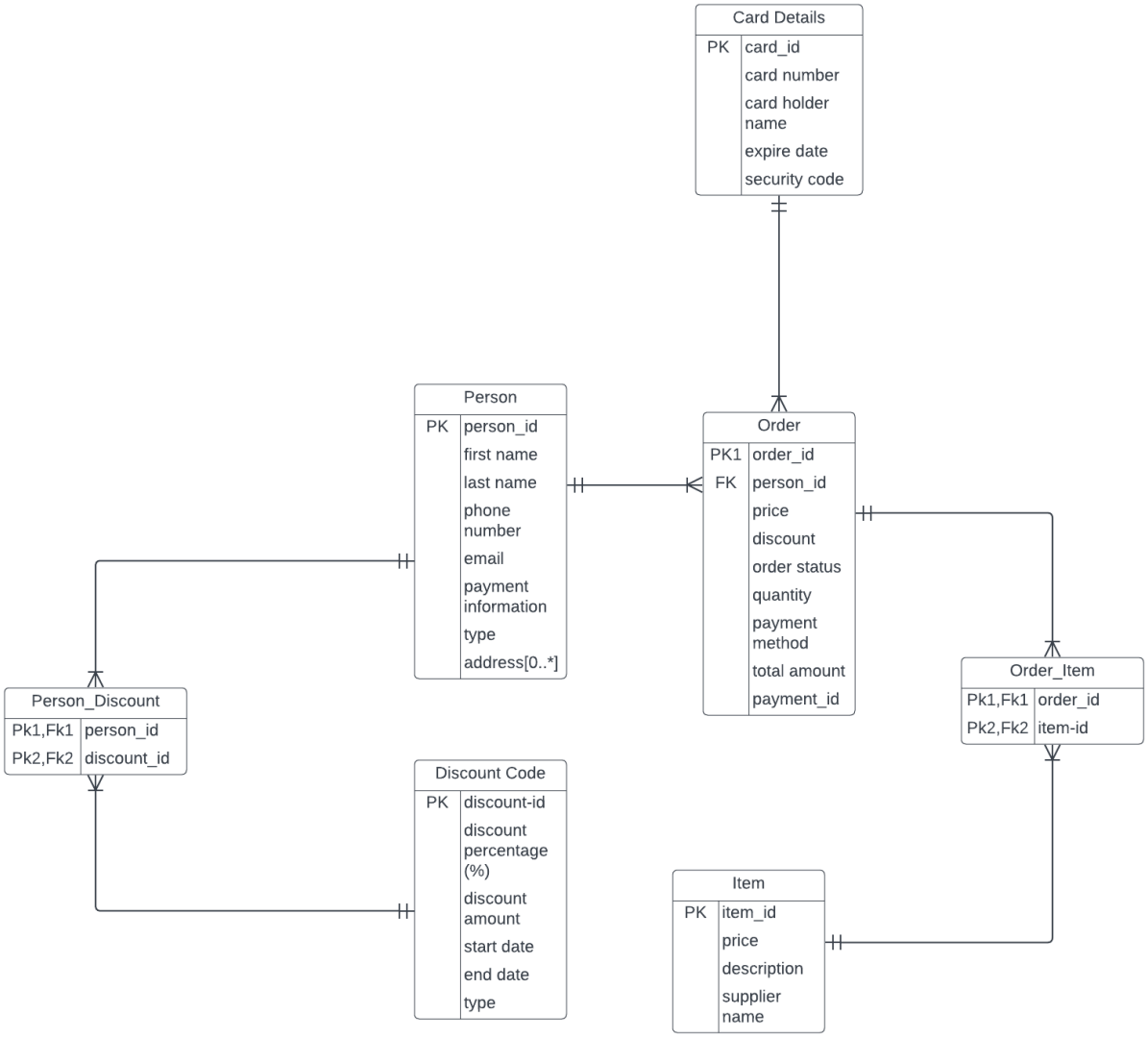
Prototype Figma File:

Figma Proto type Link : https://www.figma.com/proto/yxlwOM7iNcDM9crlmY99ri/Restaurant-website-Landing-page-(Community)?type=design&node-id=444-26&t=dj6Ym0lontQqHRsW-1&scaling=min-zoom&page-id=417%3A178&starting-point-node-id=444%3A26&mode=design

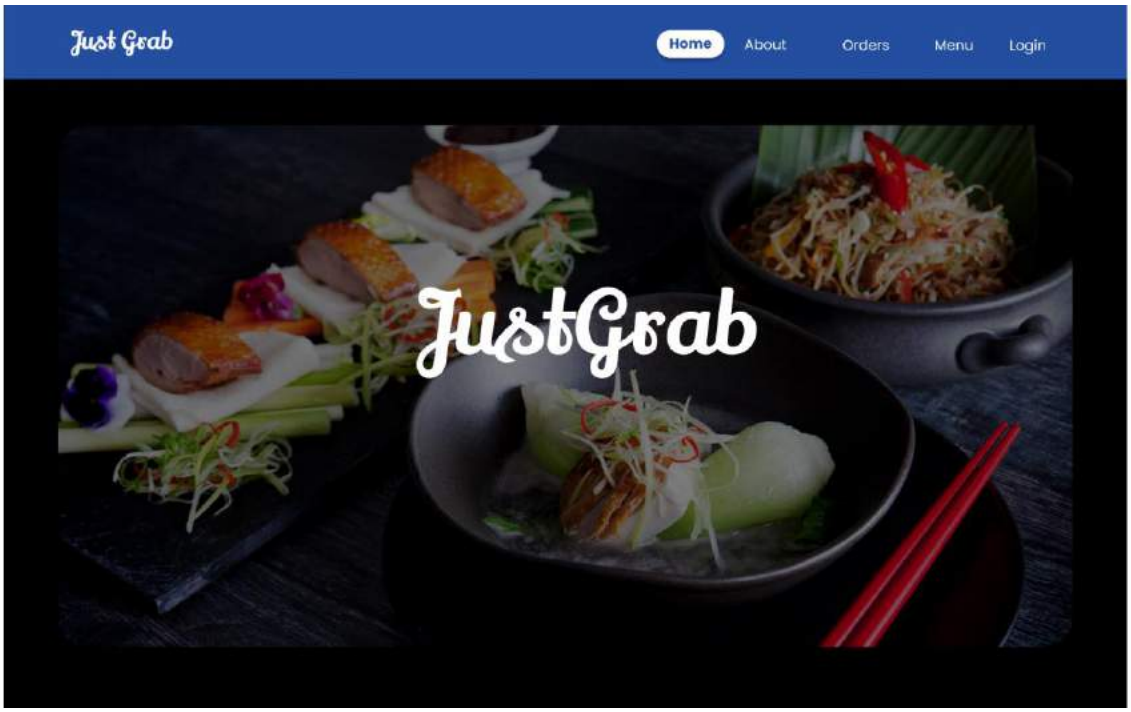
Group6GDPPrototype.zip

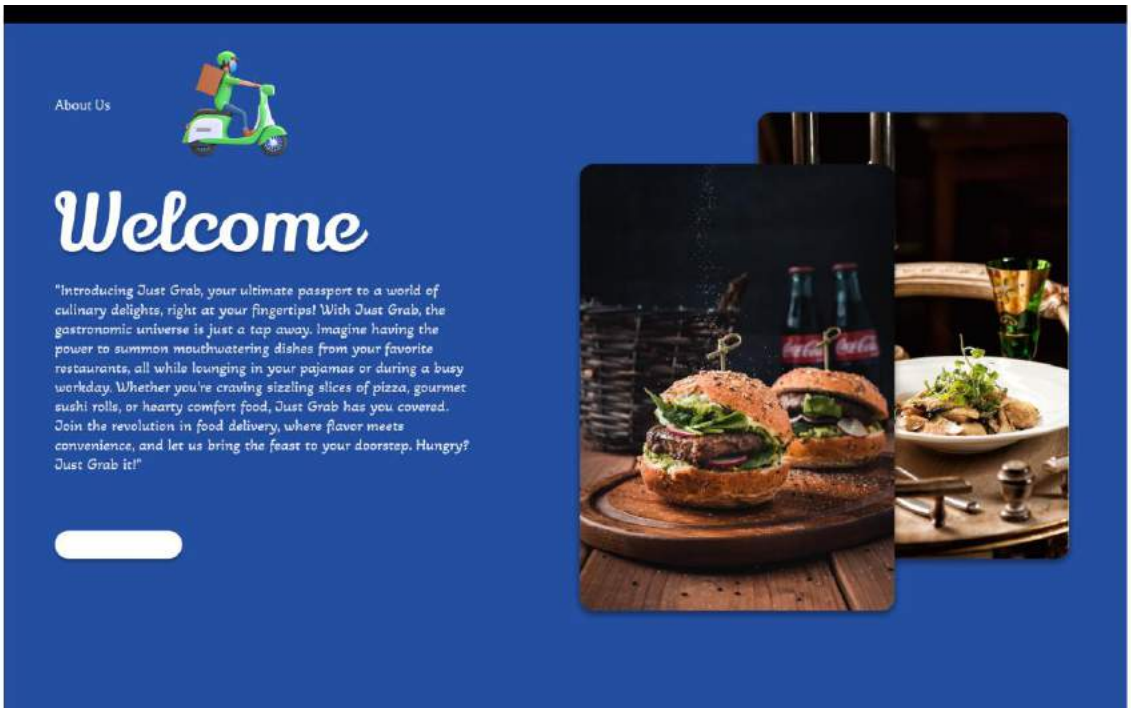
4. Introduction

4.1.ER Diagram

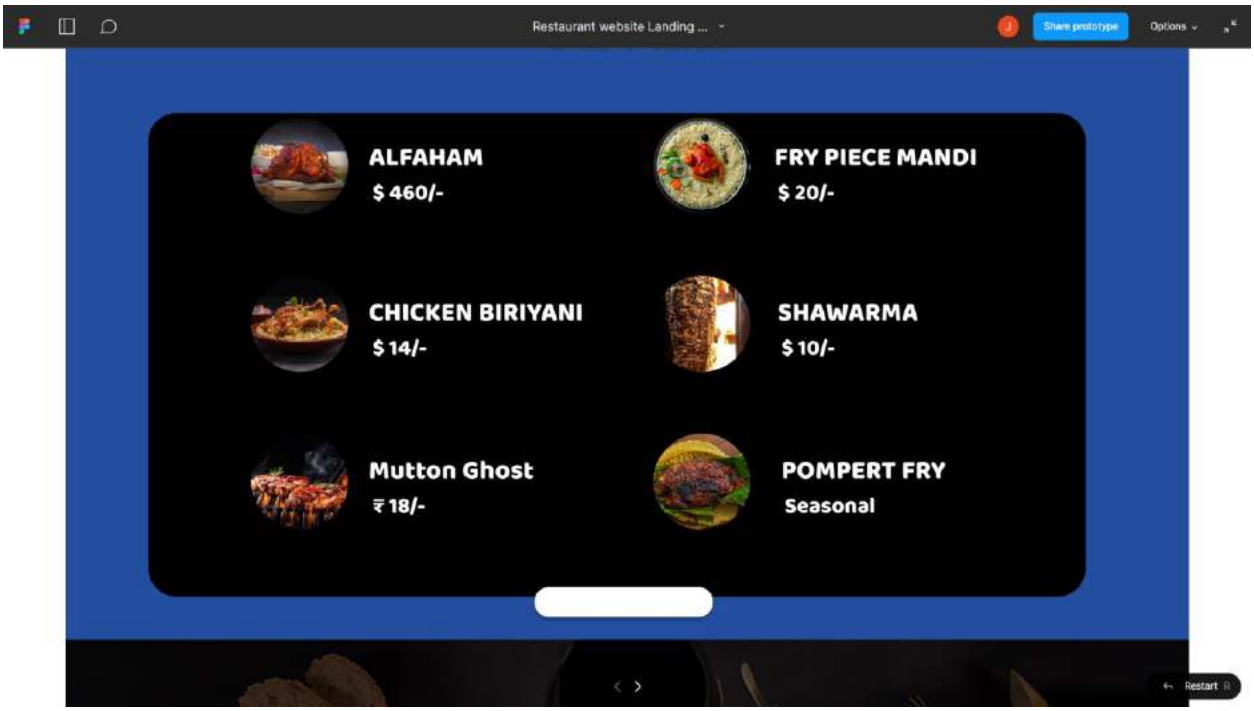


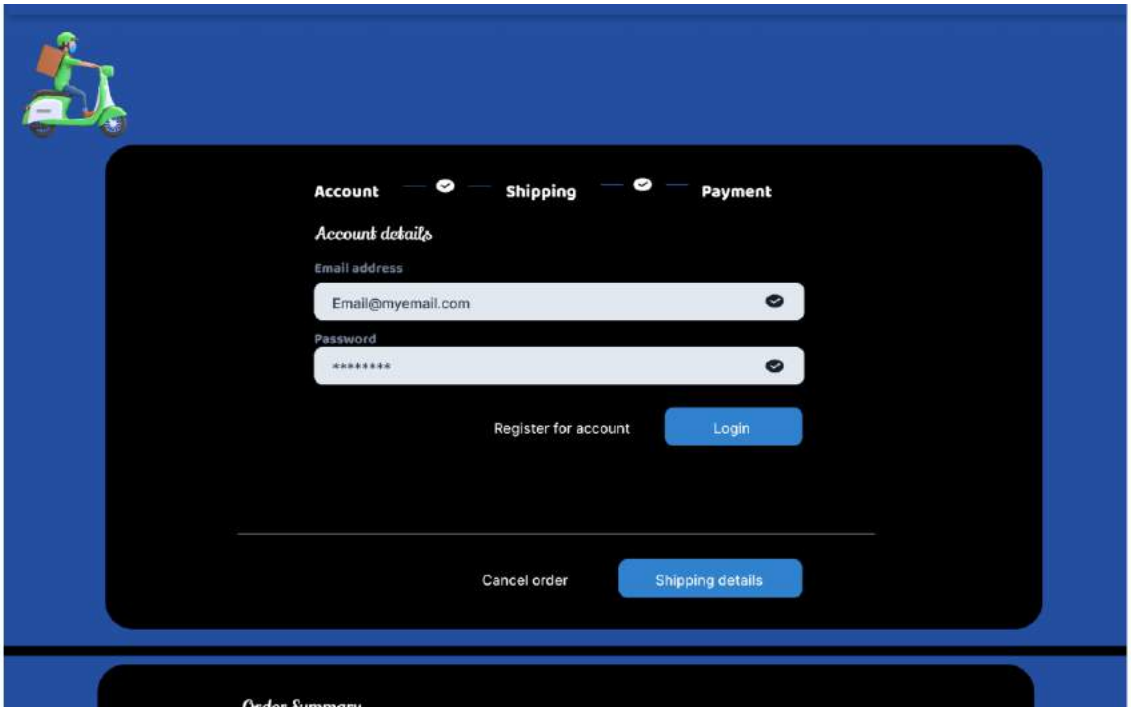
4.2.GUI


Just Grab
Home
About
Orders
Menu
Login
JustGrab

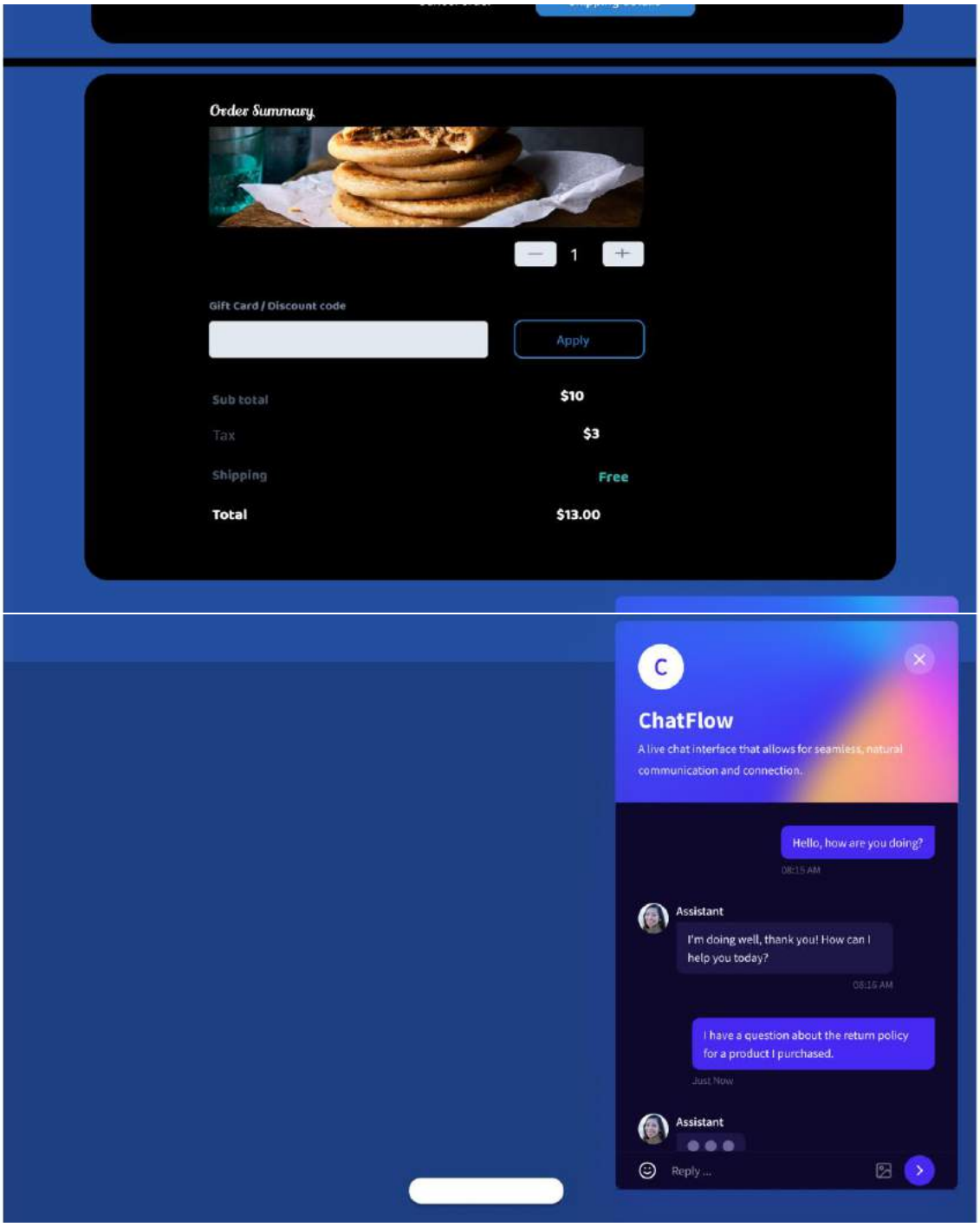

About Us
Welcome
"Introducing Just Grab, your ultimate passport to a world of culinary delights, right at your fingertips! With Just Grab, the gastronomic universe is just a tap away. Imagine having the power to summon mouthwatering dishes from your favorite restaurants, all while lounging in your pajamas or during a busy workday. Whether you're craving sizzling slices of pizza, gourmet sushi rolls, or hearty comfort food, Just Grab has you covered. Join the revolution in food delivery, where flavor meets convenience, and let us bring the feast to your doorstep. Hungry? Just Grab it!"

Food Ordering System

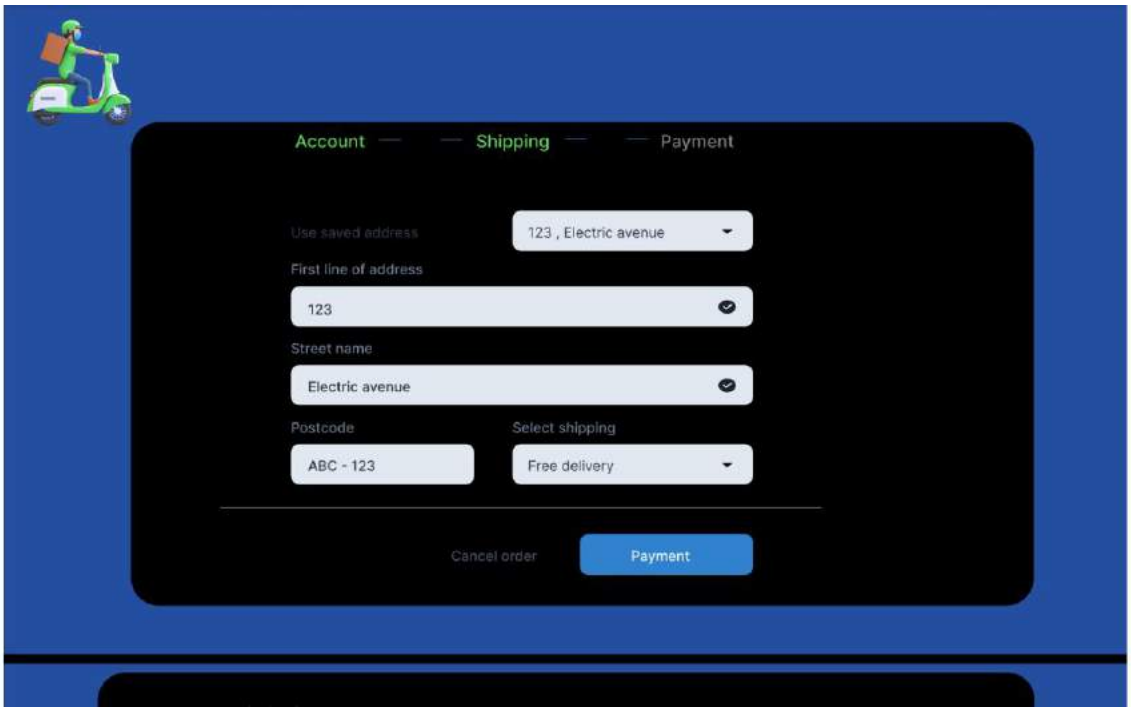




Food Ordering System



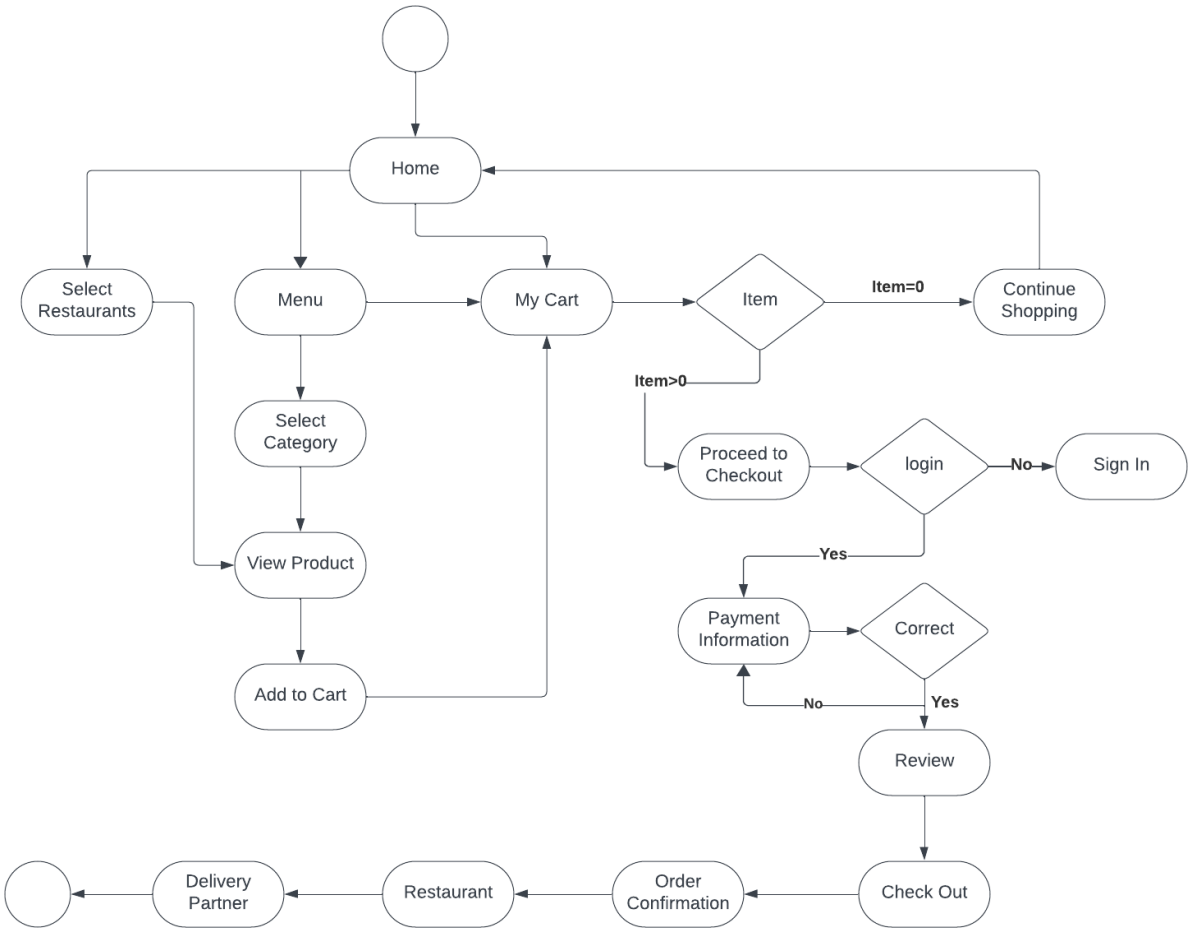
Food Ordering System



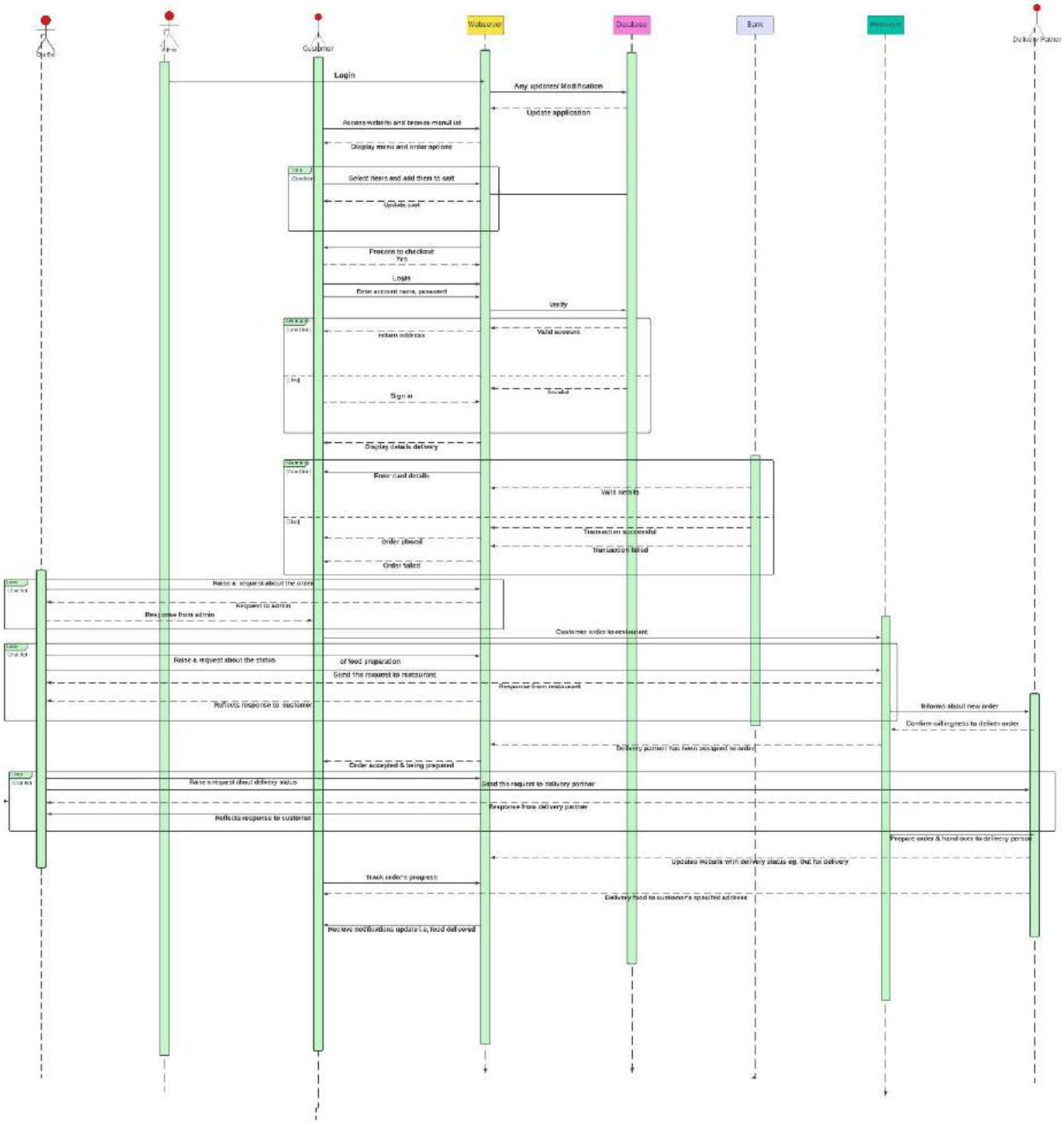
GITHUB Link : https://github.com/Jitender-Singh-NWM/FoodDeliveryApp-Web
GUI Link: https://www.figma.com/proto/yxlwOM7iNcDM9crlmY99ri/Restaurant-website-Landing-page-(Community)?type=design&node-id=444-26&t=XpUBcMXnEzpduXMh-1&scaling=min-zoom&page-id=417%3A178&starting-point-node-id=444%3A26&mode=design

5. Analysis Models

5.1 Data Flow Diagram



5.2 Data Sequence Diagram



Food Ordering System