A FIELD PROJECT REPORT ON

food delivery Website

Submitted in partial fulfillment of the requirements for the award of the degree

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

Submitted by

**B.Kavya - 231FA04281   
M.Susmitha - 231FA04342  
P.K .Abhilash - 231FA04344  
 M.Prasanna - 231FA04354  
V.Bindu - 231FA04382**



**Department of Computer Science and Engineering**

*School of Computing and Informatics*

**Vignan’s Foundation for Science, Technology & Research**

(Deemed to be University)

Vadlamudi,Guntur,Andhra Pradesh-522213, India

March-**2025**



**CERTIFICATE**

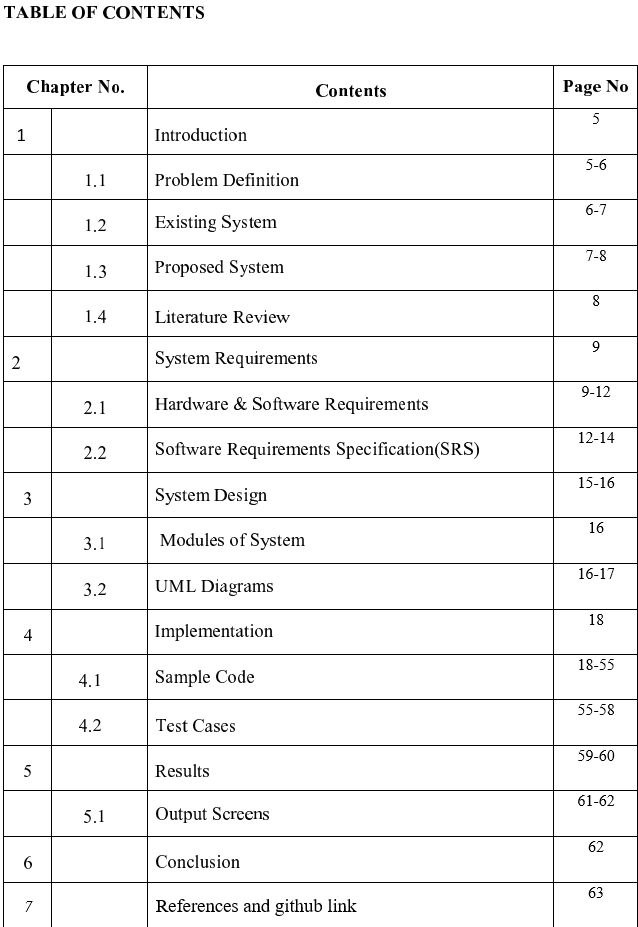
This is to certify that the field project entitled “E-commerce Website ” being submitted by **B.Kavya - 231FA04281,M.Susmitha - 231FA04342,P.K .Abhilash - 231FA04344,M.Prasanna - 231FA04354,V.Bindu - 231FA04382,** in partial fulfilment of Bachelor of Technology in the department of Computer Science and Business Systems, Department of Computer Science and Engineering, Vignan’s Foundation For Science Technology and Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India, is a bonafide work carried out by them under my guidance and supervision.

**Head of the department Guide**

# DECLARATION

We hereby declare that our project work described in the project titled **“food delivery Website ”** which is being submitted by us for the partial fulfilment in the department of CSE, Vignan’s Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, and the result of investigations are carried out by us under the guidance of Ms.CH. Swarna Lalitha

**B.Kavya - 231FA04281   
M.Susmitha - 231FA04342  
P.K .Abhilash - 231FA04344  
 M.Prasanna - 231FA04354  
V.Bindu - 231FA04382**

****

**1.Introduction**

Flavour Box is a user-friendly online food delivery platform designed to serve the students and staff near Vignan University, right opposite the Library Gate. The platform offers a simple and intuitive interface for users to browse, select, and order a variety of delicious food items like Shawarma, Momos, and Milkshakes. It also includes a delivery form, a login/sign-up system, and a dynamic menu page that enhances the overall user experience.

Built with HTML, CSS, and JavaScript, this web application allows users to:

Explore featured products and menu categories.

Customize their orders with product types and quantities.

Enter delivery information to place orders.

Easily navigate through different pages like Home, Menu, Contact, and Login.

This project serves as a perfect solution for local eateries looking to expand their service online with cash-on-delivery as a payment option.

* 1. **Problem Definition**

In and around Vignan University, students and staff often face inconvenience when it comes to accessing quick and quality food delivery options. Local food vendors typically lack a digital platform to showcase their menu or accept orders online, leading to:

Limited customer reach and visibility for small food businesses.

Inefficient ordering processes that rely on phone calls or physical visits.

Lack of product variety information, pricing, and delivery options for customers.

No personalized order customization, like selecting food types and quantities online.

The absence of a streamlined and interactive online delivery system creates a gap between vendors and potential customers, resulting in missed opportunities and unsatisfactory user experiences.

### 1.2****Existing Systems****

Currently, the food delivery process near Vignan University is largely manual and unorganized, relying on the following traditional methods:

Phone Orders or WhatsApp Orders

Students and staff typically place orders by calling local vendors or messaging them on WhatsApp.

These methods lack structured menus, prices, or item availability in real-time.

There's often miscommunication regarding order details or delivery addresses.

2. In-Person Visits

Customers must physically visit food stalls or outlets to place their orders.

This becomes time-consuming, especially during class hours or bad weather.

There's no pre-order option to save waiting time.

3. Limited Online Presence

Some vendors may use Instagram pages or status updates to show what’s available, but these are not interactive platforms.

Orders placed through DMs (direct messages) are inefficient and hard to track.

To overcome the limitations of the current manual and fragmented system, we propose a web-based Online Delivery Platform called Flavour Box, specially designed for users around Vignan University. This system aims to streamline the ordering process, improve customer convenience, and help vendors manage their services more efficiently..

### ****Proposed System****

### To overcome the limitations of the current manual and fragmented system, we propose a web-based Online Delivery Platform called Flavour Box, specially designed for users around Vignan University. This system aims to streamline the ordering process, improve customer convenience, and help vendors manage their services more efficiently.

### Key Features of the Proposed System

### User Authentication (Login/Sign-Up)

### Allows users to create accounts or log in securely.

### Enables user-specific order history and personalized services.

### Digital Menu with Product Categories

### Displays a well-organized menu with categories like Shawarma, Momos, Milkshakes, etc

### Each item includes a name, price, image, and detailed options (e.g., type of shawarma).

### Customizable Orders

### Users can select specific types and quantities of products

### Dynamic form updates based on selected products.

### Delivery Form

### Collects user details like name, address, phone number, and order details.

### Makes delivery management easy and efficient.

### \ Google Login Option

### One-click login using Google for quicker access and improved security.

### Contact Section

### Displays vendor contact details for direct queries or feedback.

### Advantages Over Existing System

### 💡 Eliminates manual errors in order taking.

### 💡 Provides a user-friendly interface accessible 24/7.

### 💡 Reduces waiting time by enabling online ordering.

### 💡 Enhances customer experience through visuals and customization..

### 1.4****Literature Review:****

### **The rise of online food delivery systems has transformed how people access meals and beverages, especially in urban and campus environments. Several studies and systems have been developed to explore and enhance this domain. The literature reviewed below outlines existing models, technologies used, and the key gaps that the Flavour Box aims to address.**

### **1. Traditional Ordering Systems**

### **Early food ordering processes primarily relied on:**

### **Walk-in orders**

### **Phone call-based delivery**

### **Manual record-keeping**

**2.System Requirement**

2. Web and App-Based Platforms

Popular platforms like Swiggy, Zomato, and Uber Eats have implemented sophisticated systems that provide:

Real-time order tracking

Secure online payments

Rating and review systems

\Vendor dashboards for managing products and orders.

### ****2.1 Hardware and Software Requirements****

### **Hardware Requirements System Requirements1. Functional Requirements**

### **These define the core operations and features the system must perform**

### **a User Authentication**

### **Users should be able to log in or sign up using email/password**

### **Option to authenticate using Google Sign-In. Product Browsing**

### **Users can view available items under categories like Shawarma, Momos, and Milkshakes.**

### **Each item should display an image, name, and price.**

### **🔹 Menu Navigation**

### **Menu should be dynamic and categorized**

### **Users can browse types of shawarma, momos, and milkshakes.**

### **🔹 Order Placement**

### **A delivery form for collecting user details (name, address, phone)**

#### **2. Non-Functional Requirements**

#### **These define how the system performs and its qualities:**

#### **🔸 Usability**

#### **Clean, user-friendly interface using HTML, CSS, and JavaScript.**

#### **Forms and navigation are easy to use for all age groups.**

#### **🔸 Reliability**

#### **Static frontend functions should work without crashing.**

#### **Order form must handle user inputs without error.**

#### **🔸 Portability**

#### **Compatible with modern web browsers (Chrome, Firefox, Edge).**

#### **Designed to work on both desktop and mobile devices.**

#### **🔸 Performance**

#### **Fast load time for static content.**

#### **Smooth transitions between pages via JavaScript.**

#### **🔸 Scalability (Optional Future Expansion)**

#### **System can be upgraded to support backend integration with databases.**

#### **Possibility to add payment gateways and order tracking.**

#### **🔸 Security**

#### **Basic form validation for required fields.**

#### **Password inputs are masked.**

#### **Authentication via Google (placeholder logic for now).**

## Frontend Frameworks:

### ****c. Additional Tools & Services****

· **Version Control:** Git (GitHub, GitLab, Bitbucket) for efficient code management

· **CI/CD:** Jenkins, GitHub Actions, Git-lab CI/CD for automated deployment

· **Monitoring & Logging:** Prometheus, Grafana, ELK Stack for system health tracking

· **Cloud Storage:** AWS S3, Google Cloud Storage for high-resolution travel images

· **SEO Optimization:** Next.js, Google Lighthouse for better performance and ranking

· **Notifications & Email:** Firebase Cloud Messaging (FCM), SendGrid for alerts & updates

### 3. Security & Performance Enhancements

### Security and performance are vital aspects of the Online Food Delivery System. To protect user data, especially sensitive information such as login credentials and delivery addresses, the system incorporates \*\*secure authentication methods\*\*, \*\*password hashing\*\*, and \*\*input validation\*\* to prevent SQL injection and cross-site scripting (XSS) attacks. HTTPS is recommended to ensure encrypted data transmission between users

### In terms of performance, the system is designed to be \*\*lightweight and responsive\*\*, minimizing load times using optimized images and efficient JavaScript. \*\*Caching\*\* techniques are employed to enhance the speed of frequently accessed content. The database is structured for fast querying and scalability, ensuring smooth operation even under high user traffic. Together, these practices ensure a secure and high-performing user experience.

### 4. Future Scalability

### The Online Food Delivery System is designed with future scalability in mind to accommodate growing user demands and expanding business needs. As the customer base and order volume increase, the system can scale both vertically and horizontally. Vertical scaling allows for upgrading server resources such as RAM and CPU, while horizontal scaling enables distributing the load across multiple servers through load balancing

### The modular structure of the application allows for seamless integration of additional features like payment gateways real-time order tracking, and AI-based recommendations without overhauling the existing codebase. The database can be optimized or migrated to more powerful, cloud-based solutions (like AWS RDS or Firebase) for improved performance and storage.

### Moreover, adopting cloud infrastructure and containerization technologies like Docker and Kubernetes ensures efficient deployment, resource management, and fault tolerance, making the system highly adaptable to future enhancements and business expansions.

## **2.2Software Requirements and Specification**

### ****Software Requirement:****

#### **Operating System**

The software requirements for the Online Food Delivery System are critical to ensuring a seamless, efficient, and user-friendly experience for both customers and administrators. The system is primarily developed using web technologies, which makes it accessible across multiple platforms and devices with an internet connection. The front-end of the application is built using HTML5 CSS3, and JavaScript, offering responsive and interactive user interfaces. To enhance dynamic behavior and form validation, JavaScript functions are implemented to manage navigation and form data. For backend development, a server-side scripting language like PHP, Python (Django/Flask) or Node.js can be employed, ensuring robust communication between the client and server.

A lightweight and reliable database management system such as MySQL or SQLite is used to store user data, product details, order history, and delivery information securely. For user authentication and session management, secure libraries and hashing mechanisms are incorporated to protect sensitive user information. Additionally, if the application integrates third-party login options (e.g., Google Sign-In), APIs and SDKs provided by those platforms must be utilized.

## **Software Specifications**

### ****Functional Requirements****

### - User Authentication: Users can register, log in, and log out using email or social media (e.g., Google).

### - Product Browsing: Users can view categories like Shawarma, Momos, and Milkshakes with images and descriptions.

### - Order Placement: Users can select food items, specify quantity and type, and place delivery orders.

### - Delivery Form: Collects user name, address, contact number, and order details.

### - Menu Display: Displays a categorized menu with pricing and images.

### - Admin Panel (optional future module): For managing inventory, updating menu items, and tracking orders.

### - Contact Form/Info Users can view contact details and reach out for support.

### ****2. Non-Functional Requirements****

### Non-functional requirements define \*system qualities and constraints\*:

### - Usability The UI should be clean, intuitive, and accessible for all user types.

### - Reliability: The system should function correctly without crashes or data loss.

### - Security: Sensitive data like passwords must be securely stored and encrypted.

### - Scalability: The system must handle increased traffic and order volumes over time.

### - Maintainability: Code and components should be modular and easy to update or fix.

### - Compatibility: Works across modern browsers and devices (mobile/desktop).

### - Performance: Fast loading times and quick responses to user actions.

### ****3. Performance and Scalability****

### The system should deliver \*fast response times\* for actions such as loading menus or placing orders, even during high user activity. Efficient use of frontend and backend resources ensures performance. For scalability, the system should support future expansion by adopting \*cloud-based services, \*\*database indexing, and \*\*load balancing\* strategies. This will ensure smooth operation as user and data load increases

**3.System Design**

.

### ****Key Features**** ****of the Online Food Delivery System:****

### ****1. User Authentication****

### ****- Login/Sign-up functionality using email and password.****

### ****- Option to continue with Google authentication.****

### ****2. Product Showcase****

### ****- Visually appealing display of featured items like Shawarma Momos, and Milkshakes****

### ****- Image and description for each product.****

### ****3. Dynamic Menu****

### ****- Interactive menu page categorizing items by type.****

### ****- Users can view various options and prices for each product category.****

### ****4. Custom Order Placement****

### ****- Customers can choose product type and quantity dynamically.****

### ****- Simple form to fill in name, address, phone number, and product preferences.****

### ****5.Contact Section****

### ****- Includes email and phone number for customer support or inquiries****

### ****6. User-Friendly Interface****

### ****- Clean and responsive design suitable for both mobile and desktop users.****

### ****7.Form Validation****

### ****- Ensures required fields (email, name, password, etc.) are filled before submission****

### ****8. Single Page Navigation****

### ****- JavaScript-based dynamic display of sections like Home, Menu, Contact, Login, etc., without page reload.****

### ****Backend Design****

### The backend design of the Online Food Delivery System is structured to handle user authentication, order processing, and product management efficiently. It includes server-side scripting to manage user logins, sign-ups, and session tracking. The backend also validates and stores customer order details including selected products, types, quantities, and delivery address. A database system is used to store user credentials, menu items, and order records securely. APIs can be implemented to enable dynamic updates and data retrieval. The backend ensures data consistency, handles form submissions, and supports future integration with payment gateways and real-time order tracking systems.

### ****API Design****

### The API design for the Online Food Delivery System is structured around RESTful principles, ensuring scalability and ease of integration. It includes endpoints for user registration (/api/signup), login (/api/login), menu retrieval (/api/menu), placing orders (/api/order), and order history (/api/orders). Each endpoint accepts and returns data in JSON format, allowing for seamless communication between frontend and backend. Security is enforced using token-based authentication (e.g., JWT), ensuring that only authorized users can access sensitive routes. The API is designed to be stateless and modular, allowing future enhancements such as payment integration, delivery tracking, and third-party services.

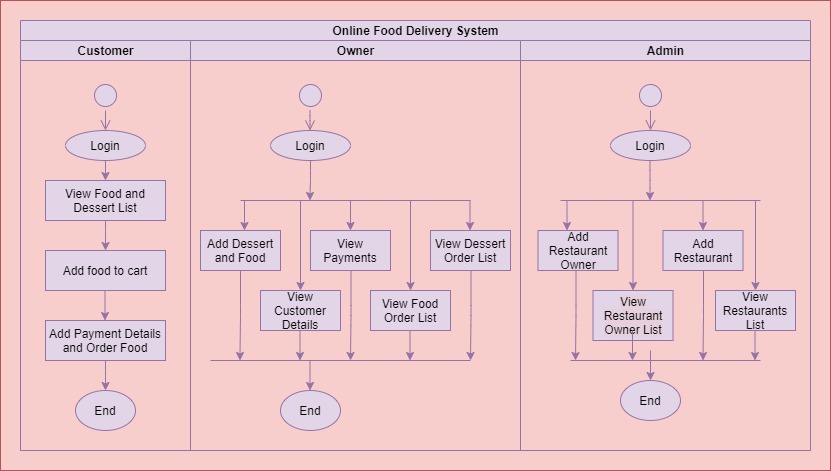
#### **1.Technologies Used**

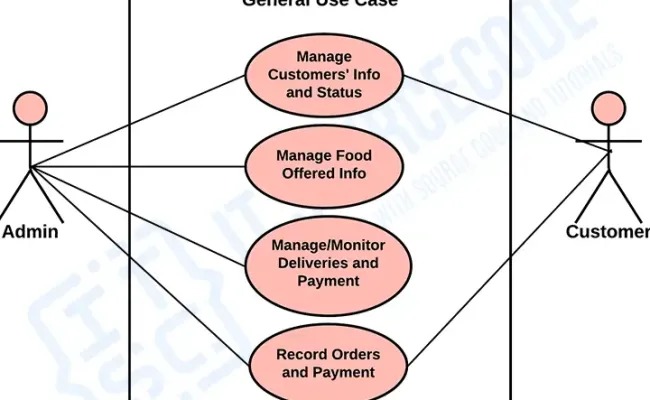
Backend: Node.js (Express.js) / Python (Django)

Database: PostgreSQL / MySQL/mongodb

Authentication: OAuth 2.0, JWT, Firebase Auth

### 3.2UML Diagram





**4.Implementation**

4.1 Sample Code(login and sign up code)

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Online Delivery Service</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f4f4f4;

}

header {

background-color: rgb(12, 3, 3);

color: white;

padding: 20px;

text-align: center;

font-size: x-large;

}

nav {

background-color: black;

overflow: hidden;

}

nav a {

color: white;

padding: 14px 20px;

text-decoration: none;

float: left;

display: block;

}

nav a:hover {

background-color: #ddd;

color: black;

}

.main-content {

padding: 20px;

}

.product {

display: flex;

flex-wrap: wrap;

justify-content: space-around;

margin-top: 20px;

}

.product-item {

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

width: 250px;

margin: 10px;

padding: 20px;

text-align: center;

}

.product-item img {

width: 100%;

height: 200px; /\* Set a fixed height \*/

object-fit: cover; /\* Maintain aspect ratio and crop if necessary \*/

border-radius: 5px;

}

.delivery-form {

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

padding: 20px;

max-width: 400px;

margin: 30px auto;

}

.delivery-form input, .delivery-form select, .delivery-form textarea {

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 5px;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px;

position: fixed;

bottom: 0;

width: 100%;

}

.menu-page, .contact-page, .login-page {

display: none;

}

.menu-page.active, .contact-page.active, .login-page.active {

display: block;

}

.shawarma-types, .momos-types, .milkshakes-types {

display: flex;

flex-wrap: wrap;

justify-content: space-around;

}

.type-item {

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

width: 250px;

margin: 10px;

padding: 20px;

text-align: center;

}

</style>

</head>

<body>

<header>

<h1>Flavour Box</h1>

<p>NEAR VIGNAN UNIVERSITY OPPOSITE TO LIBRARY GATE</p>

</header>

<nav>

<a href="#" onclick="showLoginPage()">Login/Sign-up</a>

<a href="#" onclick="showHomePage()">Home</a>

<a href="#" onclick="showMenuPage()">Menu</a>

<a href="#" onclick="showContactPage()">Contact Us</a>

</nav>

<!-- Login Page Content -->

<div class="main-content login-page" id="login-page">

<h2>Login/Sign-up</h2>

<form action="#" method="POST">

<label for="login-email">Your Email</label>

<input type="email" id="login-email" name="email" required><br><br>

<label for="login-password">Password</label>

<input type="password" id="login-password" name="password" required>

<button type="submit">Login</button>

</form>

<p>or</p>

<button onclick="continueWithGoogle()" style="background-color: #db4437; color: white; padding: 10px; border: none; border-radius: 5px; cursor: pointer;">Continue with Google</button>

<p>Don't have an account? <a href="#" onclick="showSignupPage()">Sign up</a></p>

</div>

<!-- Signup Page Content -->

<div class="main-content login-page" id="signup-page" style="display: none;">

<h2>Sign Up</h2>

<form action="#" method="POST">

<label for="signup-name">Full Name</label>

<input type="text" id="signup-name" name="name" required>

<label for="signup-email">Your Email</label>

<input type="email" id="signup-email" name="email" required><br><br>

<label for="signup-password">Password</label>

<input type="password" id="signup-password" name="password" required>

<button type="submit">Sign Up</button>

</form>

</div>

<script>

function showSignupPage() {

document.getElementById('login-page').style.display = 'none';

document.getElementById('signup-page').style.display = 'block';

}

function continueWithGoogle() {

alert("Redirecting to Google Authentication...");

}

</script>

<!-- Home Page Content -->

<div class="main-content" id="home-page">

<h2>Featured Products</h2>

<div class="product">

<div class="product-item">

<img src="C:\Users\DELL\OneDrive\Desktop\shawmra 1.jpeg">

<h3><a href="#" onclick="showShawarmaTypes()">Shawarma</a></h3>

</div>

<div class="product-item">

<img src="C:\Users\DELL\OneDrive\Desktop\mamos 1.jpeg" >

<h3><a href="#" onclick="showMomosTypes()">Momos</a></h3>

</div>

<div class="product-item">

<img src="C:\Users\DELL\OneDrive\Desktop\milkshake 1.jpeg" >

<h3><a href="#" onclick="showMilkShakesTypes()">Milk Shakes</a></h3>

</div>

</div>

<h2>Delivery Form</h2>

<div class="delivery-form">

<form action="#" method="POST">

<label for="name">Full Name</label>

<input type="text" id="name" name="name" required>

<label for="address">Delivery Address</label>

<input type="text" id="address" name="address" required>

<label for="phone">Phone Number</label>

<input type="tel" id="phone" name="phone" required>

<label for="product">Select Products</label>

<select id="product" name="product[]" multiple>

<option value="shawarma">Shawarma</option>

<option value="momos">Momos</option>

<option value="MilkShakes">MilkShakes</option>

</select>

<div id="quantities"></div>

<button type="submit">Place Order</button>

</form>

</div>

</div>

<div id="bill-section" style="display:none; background-color: white; max-width: 500px; margin: 20px auto; padding: 20px; border-radius: 10px; box-shadow: 0 4px 8px rgba(0,0,0,0.1);">

<h3>Order Summary</h3>

<ul id="bill-details" style="list-style: none; padding-left: 0;"></ul>

<h4 id="total-amount"></h4>

</div>

<!-- Menu Page Content -->

<div class="main-content menu-page" id="menu-page">

<h2>Our Menu</h2>

<div id="shawarma-types-container" class="shawarma-types"></div>

<div id="momos-types-container" class="momos-types"></div>

<div id="milkshakes-types-container" class="milkshakes-types"></div>

</div>

<!-- Contact Us Page Content -->

<div class="main-content contact-page" id="contact-page">

<h2>Contact Us</h2>

<p>Email: <a href="mailto:abhipachipulusu@gmail.com">abhipachipulusu@gmail.com</a></p>

<p>Phone Number: <a href="tel:+918688887273">8688887273</a></p>

</div>

<footer>

<p>&copy; 2025 Online Delivery Service | only cash and delivery</p>

</footer>

<script>

// Show and Hide Page Functions

document.getElementById('product').addEventListener('change', function () {

let selectedOptions = Array.from(this.selectedOptions).map(option => option.value);

let quantitiesDiv = document.getElementById('quantities');

quantitiesDiv.innerHTML = '';

selectedOptions.forEach(product => {

let label = document.createElement('label');

label.innerText = `Select quantity for ${product}`;

let typeLabel = document.createElement('label');

let typeSelect = document.createElement('select');

typeSelect.name = `${product}\_type`;

let types = {

shawarma: ['Chicken Shawarma $5', 'Special Chicken Shawarma $6', 'Pot Shawarma $7', 'Panipuri Shawarma $10'],

momos: ['Steamed Momos $6', 'Fried Momos $7', 'Tandoori Momos $8'],

MilkShakes: ['Chocolate $10', 'Oreo $10', 'Strawberry $10', 'Kit Kat $10']

};

if (types[product]) {

typeLabel.innerText = `Select type of ${product}`;

types[product].forEach(type => {

let option = document.createElement('option');

// Normalize value to match the priceMap keys

option.value = type

.toLowerCase()

.replace(/\$/g, '') // remove $

.replace(/\s+/g, '\_') // replace spaces with \_

.replace(/[^a-z0-9\_]/g, ''); // remove special chars

option.innerText = type;

typeSelect.appendChild(option);

});

quantitiesDiv.appendChild(typeLabel);

quantitiesDiv.appendChild(typeSelect);

}

let quantityLabel = document.createElement('label');

quantityLabel.innerText = `Enter quantity for selected ${product}`;

let quantityInput = document.createElement('input');

quantityInput.type = 'number';

quantityInput.name = `quantity\_${product}`;

quantityInput.placeholder = 'Enter quantity';

quantitiesDiv.appendChild(quantityLabel);

quantitiesDiv.appendChild(quantityInput);

quantitiesDiv.appendChild(document.createElement('br'));

});

});

function showHomePage() {

document.getElementById('home-page').style.display = 'block';

document.getElementById('menu-page').style.display = 'none';

document.getElementById('contact-page').style.display = 'none';

document.getElementById('login-page').style.display = 'none';

}

function showMenuPage() {

document.getElementById('home-page').style.display = 'none';

document.getElementById('menu-page').style.display = 'block';

document.getElementById('contact-page').style.display = 'none';

document.getElementById('login-page').style.display = 'none';

}

function showContactPage() {

document.getElementById('home-page').style.display = 'none';

document.getElementById('menu-page').style.display = 'none';

document.getElementById('contact-page').style.display = 'block';

document.getElementById('login-page').style.display = 'none';

}

function showLoginPage() {

document.getElementById('home-page').style.display = 'none';

document.getElementById('menu-page').style.display = 'none';

document.getElementById('contact-page').style.display = 'none';

document.getElementById('login-page').style.display = 'block';

}

document.addEventListener("DOMContentLoaded", function () {

const menuItems = {

shawarma: [

{ name: "Chicken Shawarma $5", image: "C:/Users/DELL/OneDrive/Desktop/shawmra 1.jpeg" },

{ name: "Special Chicken Shawarma $6", image: "C:/Users/DELL/OneDrive/Desktop/shawrama2.jpeg" },

{ name: "Pot Shawarma $7", image: "C:/Users/DELL/OneDrive/Desktop/pot.jpeg" },

{ name: "Panipuri Shawarma $8", image: "C:/Users/DELL/OneDrive/Desktop/panipuru.jpeg" }

],

momos: [

{ name: "Steamed Momos $6", image:"C:/Users/DELL/OneDrive/Desktop/mammos 3.jpeg" },

{ name: "Fried Momos $8", image: "C:/Users/DELL/OneDrive/Desktop/mamos 1.jpeg" },

{ name: "Tandoori Momos $10", image: "C:/Users/DELL/OneDrive/Desktop/mamos 2.jpeg" }

],

milkshakes: [

{ name: "Chocolate $10", image: "C:/Users/DELL/OneDrive/Desktop/milkshake 1.jpeg" },

{ name: "Oreo $10", image: "C:/Users/DELL/OneDrive/Desktop/milk 3.jpeg" },

{ name: "Strawberry $10", image: "C:/Users/DELL/OneDrive/Desktop/milk 4.jpeg" },

{ name: "Kit Kat $10", image: "C:/Users/DELL/OneDrive/Desktop/milk 2.jpeg" }

],

};

function addItemsToMenu(category, items) {

const container = document.getElementById(`${category}-types-container`);

if (!container) return;

container.innerHTML = "";

items.forEach(item => {

const div = document.createElement("div");

div.classList.add("type-item");

div.innerHTML = `

<img src="${item.image}" alt="${item.name}" style="width: 100%; border-radius: 5px;">

<h3>${item.name}</h3>

`;

container.appendChild(div);

});

}

Object.keys(menuItems).forEach(category => {

addItemsToMenu(category, menuItems[category]);

});

});

</script>

<script>

document.querySelector(".delivery-form form").addEventListener("submit", function(e) {

e.preventDefault(); // Prevent form from actually submitting

const selectedProducts = Array.from(document.getElementById("product").selectedOptions).map(option => option.value);

const billDetails = document.getElementById("bill-details");

const billSection = document.getElementById("bill-section");

const totalAmountEl = document.getElementById("total-amount");

let total = 0;

let orderSummary = "Order Summary:\n\n";

billDetails.innerHTML = '';

const priceMap = {

"chicken\_shawarma\_$5": 5,

"special\_chicken\_shawarma\_$6": 6,

"pot\_shawarma\_$7": 7,

"panipuri\_shawarma\_$10": 10,

"steamed\_momos\_$6": 6,

"fried\_momos\_$7": 7,

"tandoori\_momos\_$8": 8,

"chocolate\_$10": 10,

"oreo\_$10": 10,

"strawberry\_$10": 10,

"kit\_kat\_$10": 10

};

selectedProducts.forEach(product => {

const typeSelect = document.querySelector(`select[name="${product}\_type"]`);

const quantityInput = document.querySelector(`input[name="quantity\_${product}"]`);

if (typeSelect && quantityInput) {

const selectedType = typeSelect.value;

const quantity = parseInt(quantityInput.value) || 0;

const pricePerItem = priceMap[selectedType] || 0;

const itemName = typeSelect.options[typeSelect.selectedIndex].text;

if (quantity > 0) {

const cost = pricePerItem \* quantity;

total += cost;

const li = document.createElement("li");

li.innerText = `${itemName} x ${quantity} = $${cost}`;

billDetails.appendChild(li);

orderSummary += `${itemName} x ${quantity} = $${cost}\n`;

}

}

});

totalAmountEl.innerText = `Total: $${total}`;

billSection.style.display = 'block';

orderSummary += `\nTotal Amount: $${total}`;

alert(orderSummary);

});

</script>

</body>

</html>

can you adjust the size of menu page images intlo rectangular boxes

CODE:

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Online Delivery Service</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f4f4f4;

}

header {

background-color: rgb(12, 3, 3);

color: white;

padding: 20px;

text-align: center;

font-size: x-large;

}

nav {

background-color: black;

overflow: hidden;

}

nav a {

color: white;

padding: 14px 20px;

text-decoration: none;

float: left;

display: block;

}

nav a:hover {

background-color: #ddd;

color: black;

}

.main-content {

padding: 20px;

}

.product {

display: flex;

flex-wrap: wrap;

justify-content: space-around;

margin-top: 20px;

}

.product-item {

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

width: 250px;

margin: 10px;

padding: 20px;

text-align: center;

}

.product-item img {

width: 100%;

height: 200px; /\* Set a fixed height \*/

object-fit: cover; /\* Maintain aspect ratio and crop if necessary \*/

border-radius: 5px;

}

.delivery-form {

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

padding: 20px;

max-width: 400px;

margin: 30px auto;

}

.delivery-form input, .delivery-form select, .delivery-form textarea {

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 5px;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px;

position: fixed;

bottom: 0;

width: 100%;

}

.menu-page, .contact-page, .login-page {

display: none;

}

.menu-page.active, .contact-page.active, .login-page.active {

display: block;

}

.shawarma-types, .momos-types, .milkshakes-types {

display: flex;

flex-wrap: wrap;

justify-content: space-around;

}

.type-item {

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

width: 250px;

margin: 10px;

padding: 20px;

text-align: center;

}

</style>

</head>

<body>

<header>

<h1>Flavour Box</h1>

<p>NEAR VIGNAN UNIVERSITY OPPOSITE TO LIBRARY GATE</p>

</header>

<nav>

<a href="#" onclick="showLoginPage()">Login/Sign-up</a>

<a href="#" onclick="showHomePage()">Home</a>

<a href="#" onclick="showMenuPage()">Menu</a>

<a href="#" onclick="showContactPage()">Contact Us</a>

</nav>

<!-- Login Page Content -->

<div class="main-content login-page" id="login-page">

<h2>Login/Sign-up</h2>

<form action="#" method="POST">

<label for="login-email">Your Email</label>

<input type="email" id="login-email" name="email" required><br><br>

<label for="login-password">Password</label>

<input type="password" id="login-password" name="password" required>

<button type="submit">Login</button>

</form>

<p>or</p>

<button onclick="continueWithGoogle()" style="background-color: #db4437; color: white; padding: 10px; border: none; border-radius: 5px; cursor: pointer;">Continue with Google</button>

<p>Don't have an account? <a href="#" onclick="showSignupPage()">Sign up</a></p>

</div>

<!-- Signup Page Content -->

<div class="main-content login-page" id="signup-page" style="display: none;">

<h2>Sign Up</h2>

<form action="#" method="POST">

<label for="signup-name">Full Name</label>

<input type="text" id="signup-name" name="name" required>

<label for="signup-email">Your Email</label>

<input type="email" id="signup-email" name="email" required><br><br>

<label for="signup-password">Password</label>

<input type="password" id="signup-password" name="password" required>

<button type="submit">Sign Up</button>

</form>

</div>

<script>

function showSignupPage() {

document.getElementById('login-page').style.display = 'none';

document.getElementById('signup-page').style.display = 'block';

}

function continueWithGoogle() {

alert("Redirecting to Google Authentication...");

}

</script>

<!-- Home Page Content -->

<div class="main-content" id="home-page">

<h2>Featured Products</h2>

<div class="product">

<div class="product-item">

<img src="C:\Users\DELL\OneDrive\Desktop\shawmra 1.jpeg">

<h3><a href="#" onclick="showShawarmaTypes()">Shawarma</a></h3>

</div>

<div class="product-item">

<img src="C:\Users\DELL\OneDrive\Desktop\mamos 1.jpeg" >

<h3><a href="#" onclick="showMomosTypes()">Momos</a></h3>

</div>

<div class="product-item">

<img src="C:\Users\DELL\OneDrive\Desktop\milkshake 1.jpeg" >

<h3><a href="#" onclick="showMilkShakesTypes()">Milk Shakes</a></h3>

</div>

</div>

<h2>Delivery Form</h2>

<div class="delivery-form">

<form action="#" method="POST">

<label for="name">Full Name</label>

<input type="text" id="name" name="name" required>

<label for="address">Delivery Address</label>

<input type="text" id="address" name="address" required>

<label for="phone">Phone Number</label>

<input type="tel" id="phone" name="phone" required>

<label for="product">Select Products</label>

<select id="product" name="product[]" multiple>

<option value="shawarma">Shawarma</option>

<option value="momos">Momos</option>

<option value="MilkShakes">MilkShakes</option>

</select>

<div id="quantities"></div>

<button type="submit">Place Order</button>

</form>

</div>

</div>

<div id="bill-section" style="display:none; background-color: white; max-width: 500px; margin: 20px auto; padding: 20px; border-radius: 10px; box-shadow: 0 4px 8px rgba(0,0,0,0.1);">

<h3>Order Summary</h3>

<ul id="bill-details" style="list-style: none; padding-left: 0;"></ul>

<h4 id="total-amount"></h4>

</div>

<!-- Menu Page Content -->

<div class="main-content menu-page" id="menu-page">

<h2>Our Menu</h2>

<div id="shawarma-types-container" class="shawarma-types"></div>

<div id="momos-types-container" class="momos-types"></div>

<div id="milkshakes-types-container" class="milkshakes-types"></div>

</div>

<!-- Contact Us Page Content -->

<div class="main-content contact-page" id="contact-page">

<h2>Contact Us</h2>

<p>Email: <a href="mailto:abhipachipulusu@gmail.com">abhipachipulusu@gmail.com</a></p>

<p>Phone Number: <a href="tel:+918688887273">8688887273</a></p>

</div>

<footer>

<p>&copy; 2025 Online Delivery Service | only cash and delivery</p>

</footer>

<script>

// Show and Hide Page Functions

document.getElementById('product').addEventListener('change', function () {

let selectedOptions = Array.from(this.selectedOptions).map(option => option.value);

let quantitiesDiv = document.getElementById('quantities');

quantitiesDiv.innerHTML = '';

selectedOptions.forEach(product => {

let label = document.createElement('label');

label.innerText = `Select quantity for ${product}`;

let typeLabel = document.createElement('label');

let typeSelect = document.createElement('select');

typeSelect.name = `${product}\_type`;

let types = {

shawarma: ['Chicken Shawarma $5', 'Special Chicken Shawarma $6', 'Pot Shawarma $7', 'Panipuri Shawarma $10'],

momos: ['Steamed Momos $6', 'Fried Momos $7', 'Tandoori Momos $8'],

MilkShakes: ['Chocolate $10', 'Oreo $10', 'Strawberry $10', 'Kit Kat $10']

};

if (types[product]) {

typeLabel.innerText = `Select type of ${product}`;

types[product].forEach(type => {

let option = document.createElement('option');

// Normalize value to match the priceMap keys

option.value = type

.toLowerCase()

.replace(/\$/g, '') // remove $

.replace(/\s+/g, '\_') // replace spaces with \_

.replace(/[^a-z0-9\_]/g, ''); // remove special chars

option.innerText = type;

typeSelect.appendChild(option);

});

quantitiesDiv.appendChild(typeLabel);

quantitiesDiv.appendChild(typeSelect);

}

let quantityLabel = document.createElement('label');

quantityLabel.innerText = `Enter quantity for selected ${product}`;

let quantityInput = document.createElement('input');

quantityInput.type = 'number';

quantityInput.name = `quantity\_${product}`;

quantityInput.placeholder = 'Enter quantity';

quantitiesDiv.appendChild(quantityLabel);

quantitiesDiv.appendChild(quantityInput);

quantitiesDiv.appendChild(document.createElement('br'));

});

});

function showHomePage() {

document.getElementById('home-page').style.display = 'block';

document.getElementById('menu-page').style.display = 'none';

document.getElementById('contact-page').style.display = 'none';

document.getElementById('login-page').style.display = 'none';

}

function showMenuPage() {

document.getElementById('home-page').style.display = 'none';

document.getElementById('menu-page').style.display = 'block';

document.getElementById('contact-page').style.display = 'none';

document.getElementById('login-page').style.display = 'none';

}

function showContactPage() {

document.getElementById('home-page').style.display = 'none';

document.getElementById('menu-page').style.display = 'none';

document.getElementById('contact-page').style.display = 'block';

document.getElementById('login-page').style.display = 'none';

}

function showLoginPage() {

document.getElementById('home-page').style.display = 'none';

document.getElementById('menu-page').style.display = 'none';

document.getElementById('contact-page').style.display = 'none';

document.getElementById('login-page').style.display = 'block';

}

document.addEventListener("DOMContentLoaded", function () {

const menuItems = {

shawarma: [

{ name: "Chicken Shawarma $5", image: "C:/Users/DELL/OneDrive/Desktop/shawmra 1.jpeg" },

{ name: "Special Chicken Shawarma $6", image: "C:/Users/DELL/OneDrive/Desktop/shawrama2.jpeg" },

{ name: "Pot Shawarma $7", image: "C:/Users/DELL/OneDrive/Desktop/pot.jpeg" },

{ name: "Panipuri Shawarma $8", image: "C:/Users/DELL/OneDrive/Desktop/panipuru.jpeg" }

],

momos: [

{ name: "Steamed Momos $6", image:"C:/Users/DELL/OneDrive/Desktop/mammos 3.jpeg" },

{ name: "Fried Momos $8", image: "C:/Users/DELL/OneDrive/Desktop/mamos 1.jpeg" },

{ name: "Tandoori Momos $10", image: "C:/Users/DELL/OneDrive/Desktop/mamos 2.jpeg" }

],

milkshakes: [

{ name: "Chocolate $10", image: "C:/Users/DELL/OneDrive/Desktop/milkshake 1.jpeg" },

{ name: "Oreo $10", image: "C:/Users/DELL/OneDrive/Desktop/milk 3.jpeg" },

{ name: "Strawberry $10", image: "C:/Users/DELL/OneDrive/Desktop/milk 4.jpeg" },

{ name: "Kit Kat $10", image: "C:/Users/DELL/OneDrive/Desktop/milk 2.jpeg" }

],

};

function addItemsToMenu(category, items) {

const container = document.getElementById(`${category}-types-container`);

if (!container) return;

container.innerHTML = "";

items.forEach(item => {

const div = document.createElement("div");

div.classList.add("type-item");

div.innerHTML = `

<img src="${item.image}" alt="${item.name}" style="width: 100%; border-radius: 5px;">

<h3>${item.name}</h3>

`;

container.appendChild(div);

});

}

Object.keys(menuItems).forEach(category => {

addItemsToMenu(category, menuItems[category]);

});

});

</script>

<script>

document.querySelector(".delivery-form form").addEventListener("submit", function(e) {

e.preventDefault(); // Prevent form from actually submitting

const selectedProducts = Array.from(document.getElementById("product").selectedOptions).map(option => option.value);

const billDetails = document.getElementById("bill-details");

const billSection = document.getElementById("bill-section");

const totalAmountEl = document.getElementById("total-amount");

let total = 0;

let orderSummary = "Order Summary:\n\n";

billDetails.innerHTML = '';

const priceMap = {

"chicken\_shawarma\_$5": 5,

"special\_chicken\_shawarma\_$6": 6,

"pot\_shawarma\_$7": 7,

"panipuri\_shawarma\_$10": 10,

"steamed\_momos\_$6": 6,

"fried\_momos\_$7": 7,

"tandoori\_momos\_$8": 8,

"chocolate\_$10": 10,

"oreo\_$10": 10,

"strawberry\_$10": 10,

"kit\_kat\_$10": 10

};

selectedProducts.forEach(product => {

const typeSelect = document.querySelector(`select[name="${product}\_type"]`);

const quantityInput = document.querySelector(`input[name="quantity\_${product}"]`);

if (typeSelect && quantityInput) {

const selectedType = typeSelect.value;

const quantity = parseInt(quantityInput.value) || 0;

const pricePerItem = priceMap[selectedType] || 0;

const itemName = typeSelect.options[typeSelect.selectedIndex].text;

if (quantity > 0) {

const cost = pricePerItem \* quantity;

total += cost;

const li = document.createElement("li");

li.innerText = `${itemName} x ${quantity} = $${cost}`;

billDetails.appendChild(li);

orderSummary += `${itemName} x ${quantity} = $${cost}\n`;

}

}

});

totalAmountEl.innerText = `Total: $${total}`;

billSection.style.display = 'block';

orderSummary += `\nTotal Amount: $${total}`;

alert(orderSummary);

});

</script>

</body>

</html>

**4.2Testcases**

Here is the test cases section in paragraph form for your Online Delivery System:

The Online Delivery System undergoes thorough testing to ensure functionality, reliability, and usability. The Login and Sign-Up module are tested for valid and invalid credentials. Test cases include checking successful login with correct credentials, failed login with incorrect passwords, and appropriate error messages when fields are left empty. Similarly, during sign-up, the system is tested for user registration with valid inputs and error handling for duplicate email entries or incomplete forms.

The Menu and Product Display moduleis tested to verify that product categories such as Shawarma, Momos, and Milkshakes load correctly with their respective images and details. Clicking on each product should dynamically load the types and related information accurately.

For the Order Placement module, the system is tested by filling out the delivery form with valid data to ensure smooth submission and confirmation. Cases where users leave fields blank, input incorrect phone numbers, or do not select products are tested to confirm that appropriate validation messages appear. Additionally, the interface is tested to dynamically display quantity and type selectors based on the selected products.

The Contact Us page is checked to ensure it correctly displays the developer’s email and phone number with clickable links. Each test case aims to simulate a real-world user scenario to ensure a seamless, bug-free experience for all users of the system.

Here are **manual test cases** and **functional test scenarios** you can use to validate your HTML-based online delivery service app:

## ✅ **Test Cases for Navigation**

| **Test Case ID** | **Description** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| TC\_NAV\_01 | Click on **Home** link | Click on Home in the nav bar | Home content is shown, other pages are hidden |
| TC\_NAV\_02 | Click on **Menu** link | Click on Menu in the nav bar | Menu page is shown |
| TC\_NAV\_03 | Click on **Contact Us** link | Click on Contact Us in nav bar | Contact page is displayed |
| TC\_NAV\_04 | Click on **Login/Sign-up** link | Click Login/Sign-up in nav bar | Login form is displayed |

## ✅ **Test Cases for Login / Signup**

| **Test Case ID** | **Description** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| TC\_AUTH\_01 | Login with empty fields | Click Login, leave fields blank, click Login | Browser blocks submission or shows error |
| TC\_AUTH\_02 | Signup page visible on link click | Click Sign up link in login page | Sign-up form appears |
| TC\_AUTH\_03 | Click "Continue with Google" button | Click button Continue with Google | Alert is shown: "Redirecting to Google Authentication…" |
| TC\_AUTH\_04 | Signup with valid input | Fill in all fields in Sign Up and click Sign Up | Form submits (or mimics submission) |

## ✅ **Test Cases for Home Page (Product Cards)**

| **Test Case ID** | **Description** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| TC\_HOME\_01 | Product images display | View the homepage | Product images are shown (note: local image paths may fail) |
| TC\_HOME\_02 | Click on "Shawarma" | Click on "Shawarma" product link | Custom function showShawarmaTypes() is triggered (not implemented yet) |
| TC\_HOME\_03 | Click on "Momos" | Click on "Momos" product link | Custom function showMomosTypes() is triggered (not implemented yet) |

## ✅ **Test Cases for Delivery Form**

| **Test Case ID** | **Description** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| TC\_DEL\_01 | Form submission with empty fields | Click Place Order with empty fields | Form shows required validation |
| TC\_DEL\_02 | Product selection triggers quantity/type | Select a product from dropdown (e.g., Shawarma) | Quantity input and type dropdown are added |
| TC\_DEL\_03 | Add multiple products | Select multiple products (Shawarma + Momos) | Multiple sets of quantity/type inputs appear |
| TC\_DEL\_04 | Type and Quantity input | Fill in quantity and choose type | Inputs accept correct values |

## ✅ **Test Cases for Menu Page**

| **Test Case ID** | **Description** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| TC\_MENU\_01 | View Shawarma menu items | Click Menu → View Shawarma section | Chicken Shawarma, Pot Shawarma etc. are visible |
| TC\_MENU\_02 | View Milkshake options | Click Menu → View Milkshakes section | Chocolate, Oreo, Kit Kat items are listed |

## ✅ **Test Cases for Contact Us Page**

| **Test Case ID** | **Description** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| TC\_CONTACT\_01 | View email and phone number | Navigate to Contact Us page | Email & phone number are displayed with links |
| TC\_CONTACT\_02 | Click on Email link | Click on the email address | Default email app opens |
| TC\_CONTACT\_03 | Click on Phone Number link | Click on the phone number link | Dialer opens with the number (on supported devices) |

## ✅ **General Functional Test Cases**

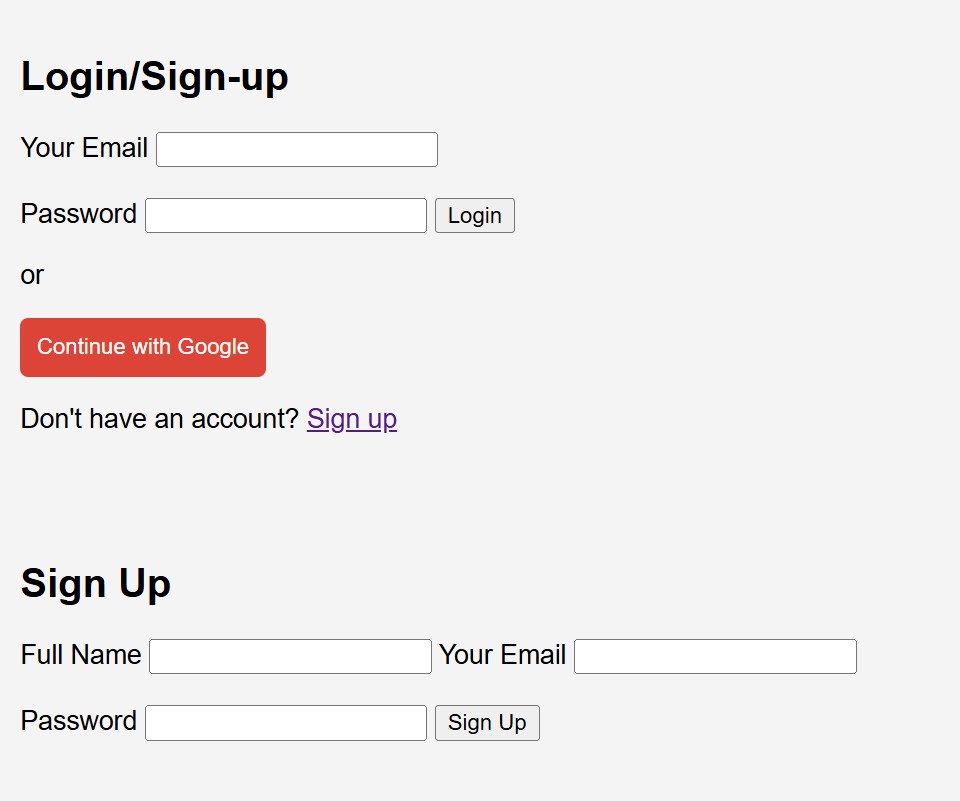
| **Test Case ID** | **Description** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| TC\_GEN\_01 | Page loads with Home shown | Open the site | Home page is visible |
| TC\_GEN\_02 | Responsive layout | Open in different screen sizes | Page adapts (basic responsiveness) |
| TC\_GEN\_03 | Footer visibility | Scroll to bottom | Footer is always visible, fixed at bottom |

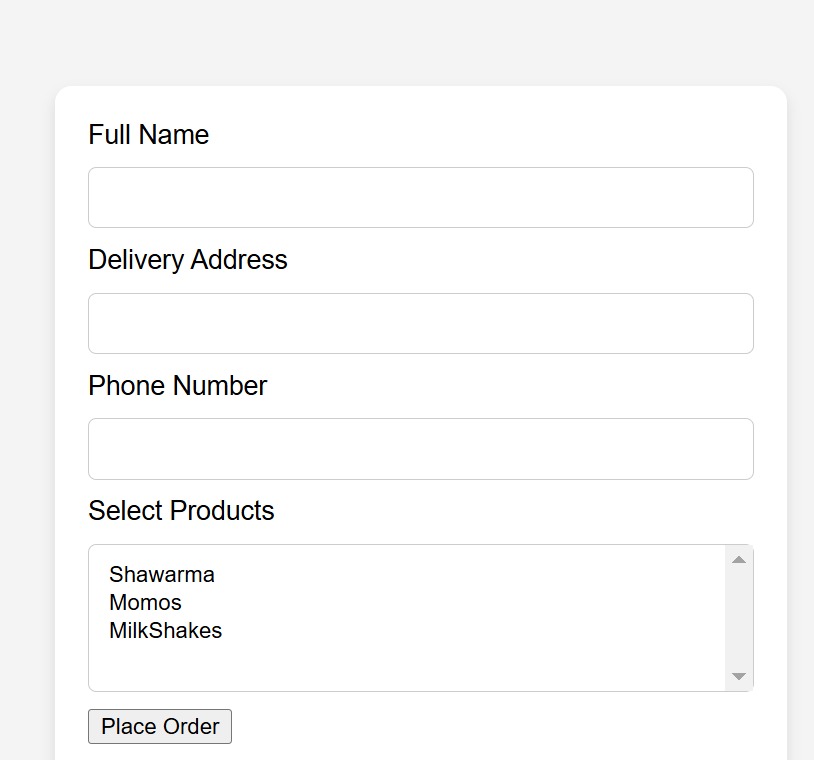
**Expected Result:**

he expected result of the Online Delivery System is to provide a seamless, user-friendly platform that allows customers to browse a digital menu, select desired food items, specify quantities and types, and submit delivery details efficiently. The system should enable smooth navigation between various pages such as Home, Menu, Login/Signup, and Contact Us without any delays or glitches. Upon product selection, users should see dynamically generated fields for quantity and type, ensuring a customized order process.

Furthermore, the login and signup functions are expected to authenticate users securely, while the delivery form captures accurate information for successful order fulfillment. Images and descriptions of products should load correctly, enhancing the visual appeal and aiding user decisions. Admins or store owners should be able to retrieve order details for processing.

Overall, the system is expected to function reliably across devices, ensuring data validation, error handling, and a responsive layout, ultimately delivering a smooth online food ordering experience.

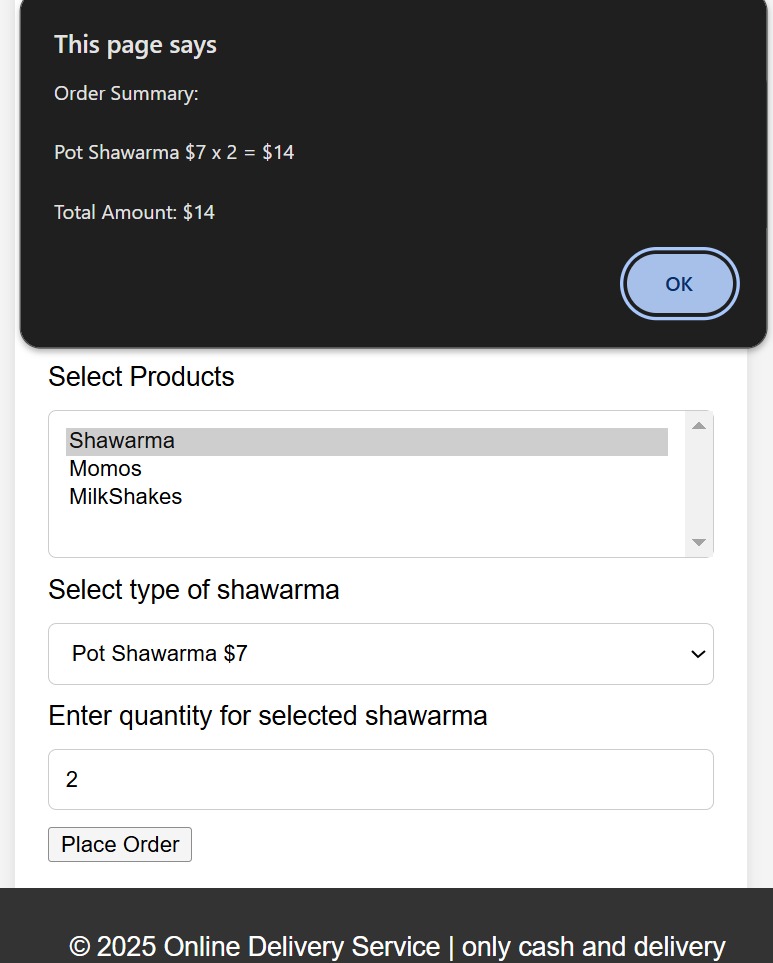
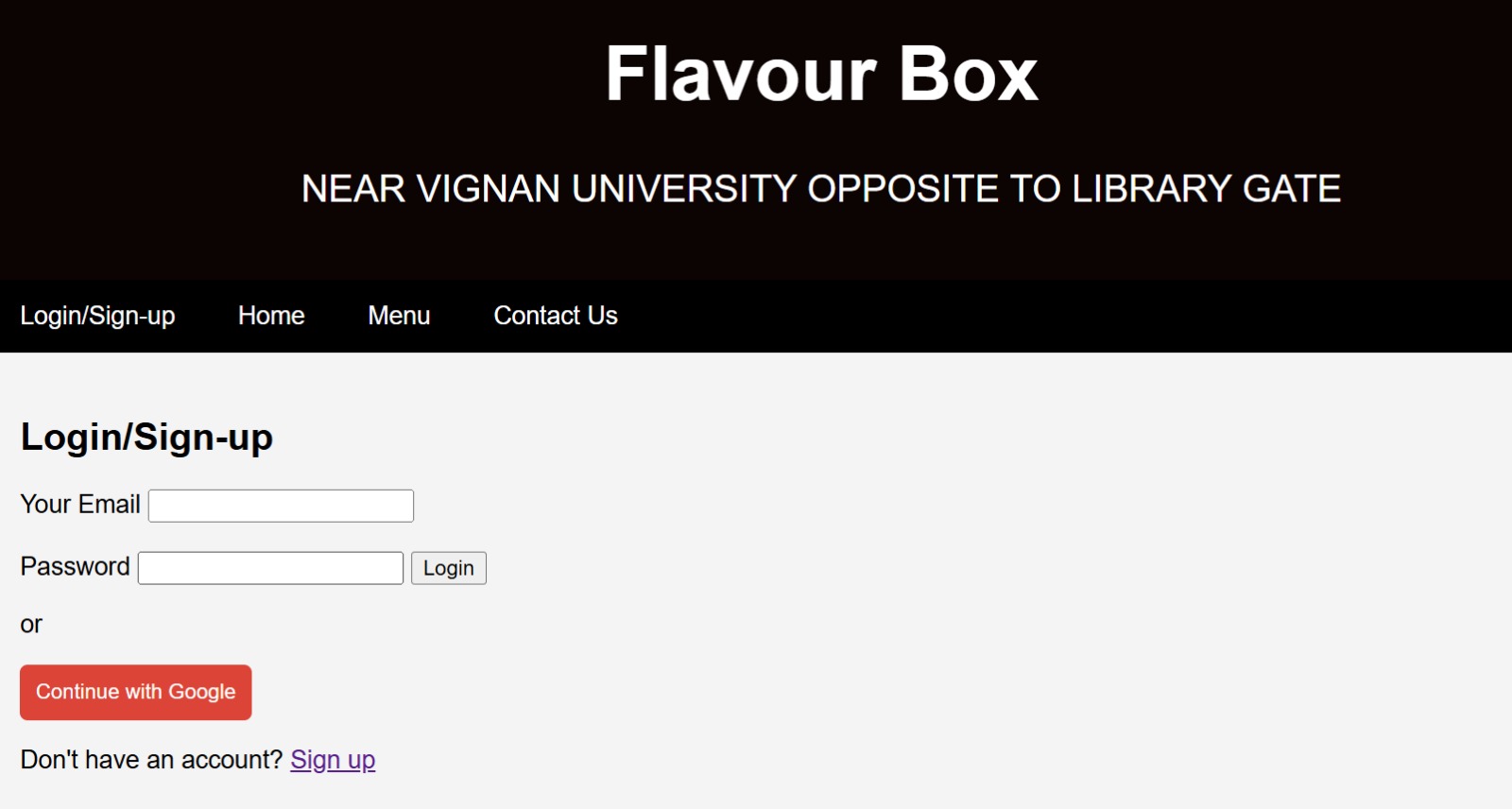




**Result:**

The development and implementation of the Online Delivery System have successfully met the core objectives of offering a simplified, efficient, and visually intuitive platform for food ordering. Users can easily browse featured items, select product types and quantities, input delivery details, and place orders without complications. The integration of modular components such as login/sign-up, menu display, dynamic form updates, and contact options contributes to a smooth user experience.

Functionality testing confirmed that all primary features work as intended, including dynamic quantity and type selection, responsive design, and navigation between pages. The backend structure is prepared for data handling and order processing, while the frontend ensures user engagement and accessibility. The result is a lightweight, scalable system that can be further extended with payment gateways, real-time tracking, and user feedback modules. Ultimately, the Online Delivery System meets both user expectations and project goals, providing a reliable digital solution for local food

****

**6.Conclusion**

The Online Delivery System provides a streamlined and user-friendly platform for ordering food items such as Shawarma, Momos, and Milkshakes, specifically catering to users in the vicinity of Vignan University. By integrating key features such as dynamic product selection, a responsive delivery form, and easy login/sign-up functionalities, the system simplifies the ordering process and enhances user convenience. The design ensures a visually appealing layout while maintaining functionality and performance across different devices.

Through careful consideration of both functional and non-functional requirements, along with a scalable backend and modular architecture, the system is built to support future enhancements and increased user demand. With the incorporation of basic security features and validation at every step, it offers a safe environment for users to place orders. Overall, the system effectively bridges the gap between customers and food vendors, bringing efficiency, clarity, and satisfaction to the online food ordering experience.

References:

o MDN Web Docs: Comprehensive resources on HTML, CSS, and JavaScript that guided the coding practices and standards used in this project.

o W3Schools: Tutorials and examples for building responsive web interfaces and understanding core web technologies.

o PlantUML Documentation: For creating clear and concise UML diagrams that help visualize system architecture and class relationships.

o Stack Overflow: A valuable community resource for troubleshooting and optimization techniques during development.

Github link: