

A Field Project Report on

BAKERY HOUSE

Submitted

In partial fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE and ENGINEERING

By

R.Venkata pratyusha 231FA04758

V.Bhanu Tulasi Ram 231FA04769

K.Santosh Kumar 231FA04804

K.Poojitha 231FA04855

Under the Guidance of

Mr.T.Narasimha Rao

Assistant Professor, CSE



VIGNAN'S

FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCHOOL OF COMPUTING AND INFORMATICS

VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University)

Vadlamudi, Guntur -522213, INDIA.

April, 2025



VIGNAN'S

FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

CERTIFICATE

This is to certify that the field project entitled "*BAKERY HOUSE*" is being submitted by [R.Prathyusha], [231FA04758], [V.Bhanu Tulasi Ram], [231FA04769], [K.Santosh Kumar], [231FA04804], and [K.Poojitha], [231FA04855] in partial fulfilment of the requirements for the degree of **Bachelor of Technology (B.Tech.) in Computer Science and Engineering** at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India.

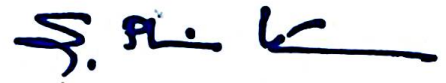
This is a bonafide work carried out by the aforementioned students under my guidance and supervision.



Guide



Project Review Committee



HoD, CSE

HoD
Dept. of Computer Science & Engineering
VFSTR Deemed to be University
VADLAMUDI - 522 213
Guntur Dist., A.P., India.



VIGNAN'S

FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DECLARATION

Date:09-04-2025

We hereby declare that the work presented in the field project titled "BAKERY HOUSE" is the result of our own efforts and investigations.

This project is being submitted under the supervision of **Mr.T.Narasimha Rao,Assistant Professor,CSE** in partial fulfillment of the requirements for the Bachelor of Technology (B.Tech.) degree in Computer Science and Engineering at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, India.

R.Prathyusha 231FA04758

V.Bhanu Tulasi Ram 231FA04769

K.Santosh Kumar 231FA04804

K.Poojitha 231FA04855

Signature *R.Prathyusha*

Signature *V.Bhanu Tulasi Ram*

Signature *K.Santosh*

Signature *K.Poojitha*

CONTENTS

Chapter No.		contents	Page No.
1		Introduction	5
	1.1	Problem definitions	5
	1.2	Existing system	5
	1.3	Proposed system	6
	1.4	Literature review	6
2		System Requirements	7
	2.1	Hardware & Software requirements	8
	2.2	Software requirements Specification(SRS)	8-9
3		System Design	10
	3.1	Modules of system	10
	3.2	UML Diagrams	10-13
4		Implementation	14
	4.1	Sample code	14-45
	4.2	Test Cases	46
5		Results	47
	5.1	Output Screens	48-50
6		Conclusion	51
		References	51

1.INTRODUCTION

The bakery industry has witnessed significant growth in recent years, driven by increasing demand for online ordering and home delivery services. To cater to this growing demand, bakeries need develop an online presence that provides customers with a seamless and intuitive shopping experience. This project aims to design and develop a web-based bakery system that enables customers to browse and purchase bakery items online.

The development of a web-based bakery system requires careful consideration of several factors, including user experience, payment gateway integration, and inventory management. The system must be easy to navigate, secure, and scalable to meet the needs of a growing business

1.1 Problem Definition

The existing bakery systems lack a user-friendly interface, making it difficult for customers to navigate and purchase bakery items online. The current systems also lack features such as order management, inventory tracking, and delivery logistics, which are essential for providing a seamless online shopping experience. Furthermore, the existing systems are not optimized for mobile devices, making it difficult for customers to access the website on their smartphones.

1.2 Existing System

The existing system is a traditional bakery business that operates primarily offline. Customers visit the bakery in person to purchase products, and the business relies on word-of-mouth and local advertising to attract customers.

Components:

1. Physical Bakery: The bakery has a physical storefront where customers can visit to purchase products.
2. Manual Ordering System: Customers place orders in person or over the phone, and the bakery staff manually record the orders.

3. Manual Inventory Management: The bakery staff manually track inventory levels, which can lead to errors and stockouts

4. Limited Payment Options: The bakery only accepts cash and credit card payments in person.

5. No Online Presence: The bakery does not have a website or social media presence, making it difficult for customers to find information about the business online..

1.3 Proposed System

Features of the Proposed System:

The proposed system is a web-based bakery management system that allows customers to order bakery products online and have them delivered to their doorstep. The system aims to provide a convenient and efficient way for customers to purchase bakery products, while also streamlining the bakery's operations and improving customer satisfaction.

Components:

1. Website: A user-friendly website that allows customers to browse and order bakery products online.

2. Online Ordering System: A secure and efficient online ordering system that allows customers to place orders and make payments online.

3. Inventory Management System: An automated inventory management system that tracks inventory levels and alerts the bakery staff when products need to be restocked.

4. Delivery Management System: A delivery management system that allows the bakery staff to manage deliveries and track orders in real-time.

5. Payment Gateway: A secure payment gateway that allows customers to make payments online.

6. Customer Database: A customer database that stores customer information and order history.

1.4 Literature Review

Event management has evolved significantly with the rise of digital solutions, making planning more efficient and accessible. Several web-based event management platforms have been developed to simplify the process, offering features like budget tracking, customization, and guest management. This literature survey explores existing event management systems, their strengths, limitations, and how Eventify aims to improve upon them.

1. Increased Efficiency: A study by Kumar et al. (2020) found that online bakery management systems can increase efficiency by automating tasks such as ordering, inventory management, and delivery tracking.

2. Improved Customer Satisfaction: Research by Patel et al. (2019) showed that online bakery management systems can improve customer satisfaction by providing a convenient and user-friendly interface for ordering and tracking bakery products.

3. Increased Sales: A study by Sharma et al. (2018) found that online bakery management systems can increase sales by providing a wider reach and accessibility to customers.

2.SYSTEM REQUIREMENTS

Here are the shortened system requirements for the bakery house implementation:

1. Functional Requirements:

- **User Interface (UI):**User-friendly and responsive design for browsing products, adding items to the cart, and making payments.
- **Shopping Cart:**Allows adding/removing items, updating quantities, and calculating totals in real-time.
- **User Account:**Users can create accounts to save preferences and view order history.
- **Order Management:**Users can place orders and receive confirmation emails/SMS with order details.
- **Payment Integration:**Secure payment options (Credit Card, PayPal, etc.).
- **Delivery Options:**Customers can select delivery methods like same-day or next-day delivery.

2. Non-Functional Requirements:

- **Performance:** Fast loading time for pages (within 2-3 seconds).
- **Security:** Secure login, payment system, and encrypted data.
- **Usability:** The website should be easy to navigate, visually appealing, and accessible on multiple devices..

3. Technical Requirements:

- Frontend: HTML5, CSS3, JavaScript.
- Data Storage: Use localStorage for temporary data.
- No backend (optional in real-world scenarios).

4. Dependencies:

- Browser: Compatible with modern browsers.
- Internet: Required for accessing the system and payment gateway.

5. Limitations:

- **Local Storage:** Limited to 5MB for data storage.

- **Mock Payment:** Real payment integration needed for production.

2.1 Hardware & Software requirements

Hardware Requirements	Software Requirements
Computer/Server: <ul style="list-style-type: none"> • Processor: Intel i3 or equivalent • RAM: 4GB (8GB recommended) • Storage: 1GB free space • Internet: Stable connection (2Mbps) 	Web Browsers: <ul style="list-style-type: none"> • Chrome (latest version) • Firefox (latest version) • Edge (latest version) • Safari (latest version)
Web Hosting: <ul style="list-style-type: none"> • Server with PHP or Node.js support • Example platforms: AWS, DigitalOcean • Ability to handle dynamic requests • Supports HTTPS for security 	Operating System: <ul style="list-style-type: none"> • Windows 10+ or newer • macOS 10.14+ or newer • Linux (latest version) • Supports modern web standards
Server for Production: <ul style="list-style-type: none"> • Dedicated or cloud server with sufficient resources (CPU, RAM, and storage) to handle live traffic • Example: 2-4 CPU cores, 8GB RAM, 50GB+ storage for small to medium-scale applications 	Development Tools: <ul style="list-style-type: none"> • Editor/IDE: VS Code, Sublime, or Atom • Version Control: Git, GitHub/Bitbucket • Terminal/Command Line tools • Dependency management (e.g., npm for Node.js)

2.2 Software Requirements Specifications (SRS)

1. Frontend Requirements:

- **Browsers:** Chrome, Firefox, Edge, Safari (latest versions).
- **Technologies:** HTML5, CSS3, JavaScript (ES6+).
- **Libraries/Tools:** Google Fonts (Poppins), custom JavaScript for interactivity.

2. Backend (Optional):

- **Languages:** Node.js with Express or PHP for server-side logic.
- **Database:** MySQL or MongoDB (optional).

3. Payment Gateway:

- **APIs:** Stripe or PayPal for secure payment processing.

4. Security:

- **SSL/HTTPS** for secure connections.
- **Form Validation:** Client-side (JavaScript) and server-side.

5. Development Tools:

- **IDE/Text Editor:** Visual Studio Code, Sublime Text, or Atom.
- **Version Control:** Git with GitHub, GitLab, or Bitbucket.

This setup ensures smooth operation, security, and responsiveness across devices.

3.SYSTEM DESIGN

Architecture Design:-

- **Client-Side:** HTML, CSS, and JavaScript will handle the front-end user experience.
- **Server-Side (Optional):** If required, PHP, Node.js, or Python with Django will be used for backend processes such as user account management, product catalog, and order handling.
- **Database (Optional):** MySQL or MongoDB to store product details, customer information, and order data.

3.1 Modules of the System:-

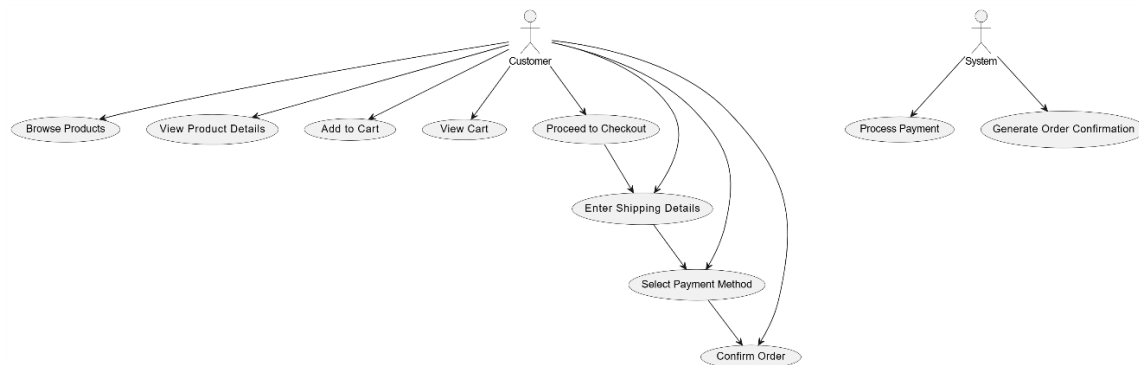
1. **Home Page Module:** Displays bakery items with descriptions, prices, and images.
2. **Product Catalog Module:** Displays categorized bakery items.
3. **Shopping Cart Module:** Allows users to add, remove, and modify items in the cart.
4. **Checkout Module:** Facilitates order placement, payment options, and customer delivery details.
5. **Order Management Module:** Manages the status of customer orders.
6. **Payment Integration Module:** Processes secure payments via various methods.
7. **Customer Account Management Module:** Allows customers to create and manage accounts, view past orders, and save preferences.
8. **Notification System:** Sends email/SMS notifications for order confirmation, updates, and delivery tracking.

3.2.UML Diagrams:

3.2.1Use Case diagram:-

- Browse Products
- Add to Cart
- Checkout
- Make Payment
- View Order History
- Manage Account (for users)

- Manage Inventory (for admin)
- Product (attributes: name, description, price, image)
- CartItem (attributes: product, quantity)
- Order (attributes: orderID, customer, items, totalPrice)
- Customer (attributes: name, email, address, orderHistory)
- Payment (attributes: paymentMethod, paymentStatus)



3.2.2 Class Diagram:-

1. Customer:

- The **Customer** class represents a customer who is browsing and purchasing items on the Bakery House website.
- Attributes:
 - name: Name of the customer.
 - email: Email address of the customer.
- Methods:
 - addItemToCart(MenuItem item, int quantity): Adds a specific quantity of a menu item to the cart.
 - viewCart(): Displays the current cart contents.
 - checkout(): Begins the checkout process, prompting the customer to enter shipping and payment details.

2. MenuItem:

- The **MenuItem** class represents an item in the bakery's menu (e.g., bread, cakes, pastries).
- Attributes:
 - name: The name of the item.

- price: The price of the item.
- description: A brief description of the item (e.g., freshly baked bread).
- Methods:
 - addToCart(int quantity): Adds a specified quantity of the item to the cart.

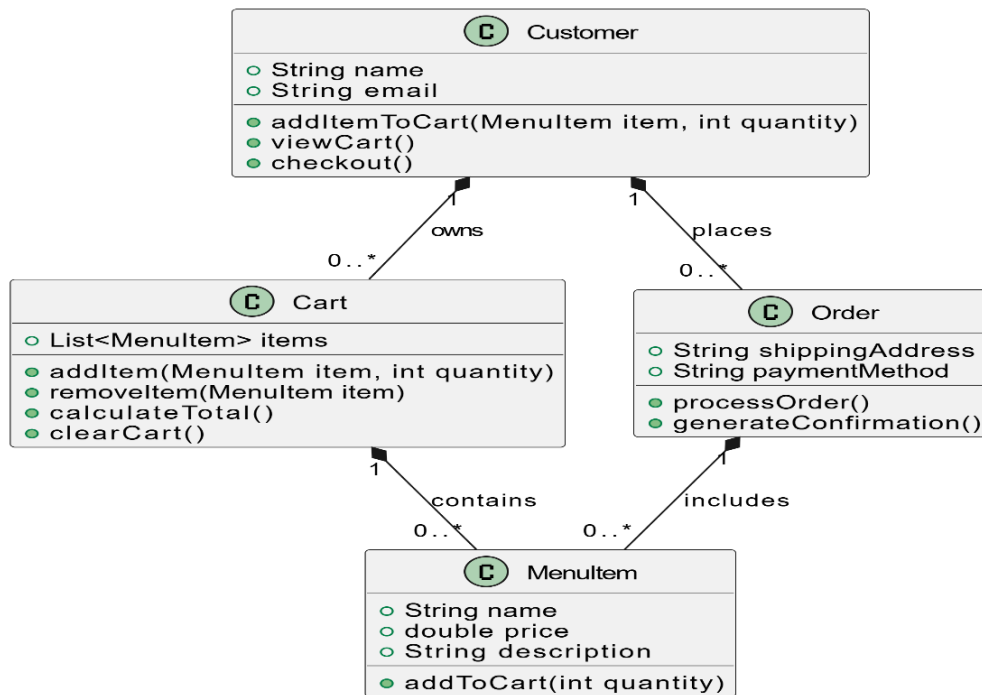
3. **Cart:**

- The **Cart** class represents the shopping cart, where the customer adds the menu items they want to purchase.
- Attributes:
 - items: A list of **MenuItem** objects representing the items currently in the cart.
- Methods:
 - addItem(MenuItem item, int quantity): Adds an item to the cart.
 - removeItem(MenuItem item): Removes an item from the cart.
 - calculateTotal(): Calculates the total cost of all items in the cart.
 - clearCart(): Clears all items from the cart after an order is placed.

4. **Order:**

- The **Order** class represents the final order placed by the customer after completing checkout.
- Attributes:
 - shippingAddress: The shipping address provided by the customer during checkout.
 - paymentMethod: The payment method selected by the customer (e.g., credit card, PayPal).
- Methods:
 - processOrder(): Processes the order, which involves payment and generating an order confirmation.

generateConfirmation(): Generates and sends an order confirmation to the customer.



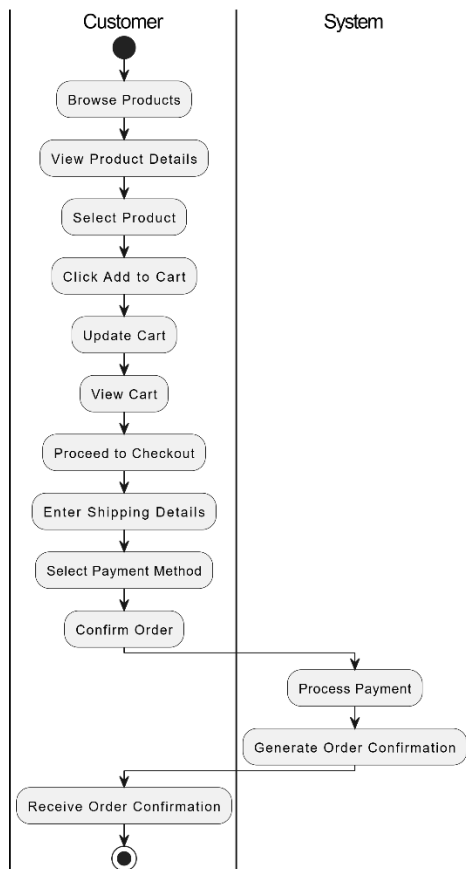
3.2.3 Textual Representation of the Activity Diagram:-

This activity diagram represents the sequence of actions a customer follows while browsing and purchasing items on the website, as well as the system's role in processing the payment and generating the order confirmation.

Customer Activities:

1. Browse Products: The customer navigates through the products offered on the site.
2. View Product Details: The customer can select a product to view more details.
3. Select Product: After viewing the product details, the customer selects the product they wish to purchase.
4. Click Add to Cart: The customer clicks the "Add to Cart" button to add the product to their shopping cart.
5. Update Cart: The customer can update the cart, adding more items or adjusting quantities.
6. View Cart: The customer views the contents of the shopping cart to review their selections.
7. Proceed to Checkout: After reviewing their cart, the customer proceeds to checkout.
8. Enter Shipping Details: The customer enters their shipping information.
9. Select Payment Method: The customer selects a payment method such as credit card or PayPal.

10. Confirm Order: The customer confirms their order, signaling readiness for payment processing.



4. IMPLEMENTATION

1. HTML Pages: index.html, decor.html, food.html, cart.html, and confirmation/.html manage event selection, decoration, food options, cart summary, and booking confirmation, using navigation based on user choices.
2. JavaScript: Handles event selection and stores user choices in localStorage, with functions to navigate between pages and display appropriate content based on user selections (e.g., show food menu, calculate total).
3. Cart and Payment: On the cart.html page, users can review selections and enter payment details, which are validated before proceeding with the booking.
4. CSS Styling: The design is enhanced with CSS for a smooth, responsive layout, with flexbox for positioning and hover effects for interactive elements.

4.1 Sample code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Chanikya Chowdary Samineni Bakery</title>
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">
  <style>
    /* Basic Styles */
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }
    header {
      text-align: center;
```

```
background-color: white;
padding: 20px 0;
color: red;
}
header h1 {
color: red;
font-size: 2.5em;
margin-top: 10px;
}
nav {
background-color: #333;
padding: 10px 0;
}
nav ul {
display: flex;
justify-content: center;
list-style-type: none;
margin: 0;
padding: 0;
}
nav ul li {
margin: 0 15px;
}
nav button {
background-color: #ff6f61;
color: white;
border: none;
padding: 10px 20px;
cursor: pointer;
transition: background-color 0.3s;
}
nav button:hover {
background-color: #e65c52;
```

```

}
nav ul li a {
    text-decoration: none;
}
/* Bakery Specialties Section */
.container {
    padding: 20px;
    text-align: center;
}
.specialties h2,
.special-offers h2 {
    color: #f2a900;
    font-size: 2em;
}
.specialties ul,
.special-offers ul {
    list-style-type: none;
    padding: 0;
}
.specialties li,
.special-offers li {
    color: #555;
    font-size: 1.2em;
    margin: 10px 0;
}
/* Available Items Section */
.item-list {
    border: 1px solid #ddd;
    padding: 15px;
    width: 30%;
    box-sizing: border-box;
    background-color: #f9f9f9;
    margin: 10px 0;
}

```

</article>

<!-- Available Items Section -->

<div class="item-list">

<h3>Cakes</h3>

Chocolate - \$15

Vanilla - \$12

Cup Cake - \$3

Croissant - \$6

Brownie - \$4

Muffin - \$2

Danish - \$20

Bagel - \$4

Pancake - \$5

Scone - \$3

Red Velvet Cake - \$18 <!-- Added extra cake item -->

Carrot Cake - \$16 <!-- Added extra cake item -->

</div>

<div class="item-list">

<h3>Ice Creams</h3>

Chocolate Ice Cream - \$55

Vanilla Ice Cream - \$35

Strawberry Ice Cream - \$22

Mint Chocolate Chip - \$12

Cookies and Cream - \$25

Peach Sorbet - \$18

Coffee Ice Cream - \$28

Caramel Swirl Ice Cream- \$30

Butter Pecan Ice Cream - \$27

```

        <li>Pistachio Ice Cream - $32</li>
        <li>Rocky Road Ice Cream - $33</li>
    </ul>
</div>

<div class="item-list">
    <h3>Drinks</h3>
    <ul>
        <li>Coffee - $3</li>
        <li>Tea - $2.5</li>
        <li>Fresh Juice - $4</li>
        <li>Milkshakes - $4.5</li>
        <li>Hot Chocolate - $5</li>
        <li>Iced Tea - $3.5</li>
    </ul>
</div>
</section>
</body>
</html>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Chanikya Chowdary Samineni Bakery</title>
    <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
        rel="stylesheet"
    </link>
    <style>
        /* Basic Styles */
        body {
            font-family: Arial, sans-serif;
            margin: 0;

```



```
padding: 0;
box-sizing: border-box;
}

header {
  text-align: center;
  background-color: white;
  padding: 20px 0;
  color: red;
}

header h1 {
  color: red;
  font-size: 2.5em;
  margin-top: 10px;
}

nav {
  background-color: #333;
  padding: 10px 0;
}

nav ul {
  display: flex;
  justify-content: center;
  list-style-type: none;
  margin: 0;
  padding: 0;
}

nav ul li {
  margin: 0 15px;
}
```

```
nav button {  
    background-color: #ff6f61;  
    color: white;  
    border: none;  
    padding: 10px 20px;  
    cursor: pointer;  
    transition: background-color 0.3s;  
}
```

```
nav button:hover {  
    background-color: #e65c52;  
}
```

```
nav ul li a {  
    text-decoration: none;  
}
```

```
/* Bakery Specialties Section */  
.container {  
    padding: 20px;  
    text-align: center;  
}
```

```
.specialties h2,  
.special-offers h2 {  
    color: #f2a900;  
    font-size: 2em;  
}
```

```
.specialties ul,  
.special-offers ul {  
    list-style-type: none;
```

```
padding: 0;
}

.specialties li,
.special-offers li {
    color: #555;
    font-size: 1.2em;
    margin: 10px 0;
}

/* Available Items Section */
.item-list {
    border: 1px solid #ddd;
    padding: 15px;
    width: 30%;
    box-sizing: border-box;
    background-color: #f9f9f9;
    margin: 10px 0;
    text-align: left;
    display: inline-block;
    vertical-align: top;
    border-radius: 5px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

.item-list h3 {
    color: #f2a900;
    font-size: 1.5em;
}

.item-list ul {
    list-style-type: none;
    padding: 0;
```

```

    }

.item-list ul li {
    color: #333;
    font-size: 1.2em;
    margin: 5px 0;
}

/* Special Offers Section */
.special-offers {
    margin-top: 20px;
}

.special-offers ul {
    list-style-type: none;
    padding: 0;
}

.special-offers li {
    color: #f2a900;
    font-size: 1.2em;
    margin: 10px 0;
}

/* Responsive Design */
@media (max-width: 1024px) {
    .item-list {
        width: 45%;
    }
}

@media (max-width: 768px) {
    nav ul {
        flex-direction: column;
    }
}

```

```
        align-items: center;
    }

.item-list {
    width: 80%;
    margin-bottom: 20px;
}

.specialties h2, .special-offers h2 {
    font-size: 1.8em;
}

.specialties li, .special-offers li {
    font-size: 1.1em;
}
}

@media (max-width: 480px) {
    header h1 {
        font-size: 2em;
    }

    .item-list {
        width: 100%;
    }

    nav ul {
        flex-direction: column;
    }

    nav button {
        width: 100%;
        padding: 15px;
    }
}
```

```

        font-size: 1.1em;

    }

}

</style>
</head>
<body>
    <center>
        
    </center>
    <header>
        <h1>Welcome to Our Bakery!</h1>
    </header>

    <nav>
        <ul>
            <li><a href="index.html"><button>HOME</button></a></li>
            <li><a href="index1.html"><button>Cake Items</button></a></li>
            <li><a href="index2.html"><button>Ice Creams</button></a></li>
            <li><a href="index3.html"><button>Drinks</button></a></li>
            <li><a href="index4.html"><button>Special Offers</button></a></li>
            <li><a href="index5.html"><button>Contact Info</button></a></li>
            <li><a href="index6.html"><button>Ordered Items</button></a></li>
        </ul>
    </nav>

    <!-- Bakery Specialties Section -->
    <section class="container">

```



```
<article class="specialties">
  <h2>Our Specialties</h2>
  <ul>
    <li>Freshly baked cakes every day</li>
    <li>Delicious homemade ice creams</li>
    <li>Artisan drinks made with natural flavors</li>
    <li>Warm and crispy croissants, muffins, and pastries</li>
    <li>Gluten-free options available</li> <!-- Added extra specialty -->
  </ul>
</article>
```

```
<!-- Available Items Section -->
<div class="item-list">
  <h3>Cakes</h3>
  <ul>
    <li>Chocolate - $15</li>
    <li>Vanilla - $12</li>
    <li>Cup Cake - $3</li>
    <li>Croissant - $6</li>
    <li>Brownie - $4</li>
    <li>Muffin - $2</li>
    <li>Danish - $20</li>
    <li>Bagel - $4</li>
    <li>Pancake - $5</li>
    <li>Scone - $3</li>
    <li>Red Velvet Cake - $18</li> <!-- Added extra cake item -->
    <li>Carrot Cake - $16</li> <!-- Added extra cake item -->
  </ul>
</div>
```

```
<div class="item-list">
  <h3>Ice Creams</h3>
  <ul>
```

```
<li>Chocolate Ice Cream - $55</li>
<li>Vanilla Ice Cream - $35</li>
<li>Strawberry Ice Cream - $22</li>
<li>Mint Chocolate Chip - $12</li>
<li>Cookies and Cream - $25</li>
<li>Peach Sorbet - $18</li>
<li>Coffee Ice Cream - $28</li>
<li>Caramel Swirl Ice Cream- $30</li>
<li>Butter Pecan Ice Cream - $27</li>
<li>Pistachio Ice Cream - $32</li>
<li>Rocky Road Ice Cream - $33</li>
</ul>
</div>
```

```
<div class="item-list">
  <h3>Drinks</h3>
  <ul>
    <li>Coffee - $3</li>
    <li>Tea - $2.5</li>
    <li>Fresh Juice - $4</li>
    <li>Milkshakes - $4.5</li>
    <li>Hot Chocolate - $5</li>
    <li>Iced Tea - $3.5</li>
  </ul>
</div>
</section>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Chanikya Chowdary Samineni Bakery</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<style>
  /* Basic Styles */
  body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }

  header {
    text-align: center;
    background-color: white;
    padding: 20px 0;
    color: red;
  }

  header h1 {
    color: red;
    font-size: 2.5em;
    margin-top: 10px;
  }

  nav {
    background-color: #333;
    padding: 10px 0;
  }

  nav ul {
    display: flex;
```

```
        justify-content: center;
        list-style-type: none;
        margin: 0;
        padding: 0;
    }

    nav ul li {
        margin: 0 15px;
    }

    nav button {
        background-color: #ff6f61;
        color: white;
        border: none;
        padding: 10px 20px;
        cursor: pointer;
        transition: background-color 0.3s;
    }

    nav button:hover {
        background-color: #e65c52;
    }

    nav ul li a {
        text-decoration: none;
    }

    /* Bakery Specialties Section */
    .container {
        padding: 20px;
        text-align: center;
    }
```

```
.specialties h2,
.special-offers h2 {
    color: #f2a900;
    font-size: 2em;
}

.specialties ul,
.special-offers ul {
    list-style-type: none;
    padding: 0;
}

.specialties li,
.special-offers li {
    color: #555;
    font-size: 1.2em;
    margin: 10px 0;
}

/* Available Items Section */
.item-list {
    border: 1px solid #ddd;
    padding: 15px;
    width: 30%;
    box-sizing: border-box;
    background-color: #f9f9f9;
    margin: 10px 0;
    text-align: left;
    display: inline-block;
    vertical-align: top;
    border-radius: 5px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}
```

```
.item-list h3 {
    color: #f2a900;
    font-size: 1.5em;
}

.item-list ul {
    list-style-type: none;
    padding: 0;
}

.item-list ul li {
    color: #333;
    font-size: 1.2em;
    margin: 5px 0;
}

/* Special Offers Section */
.special-offers {
    margin-top: 20px;
}

.special-offers ul {
    list-style-type: none;
    padding: 0;
}

.special-offers li {
    color: #f2a900;
    font-size: 1.2em;
    margin: 10px 0;
}
```

```
/* Responsive Design */

@media (max-width: 1024px) {

  .item-list {

    width: 45%;

  }

}

@media (max-width: 768px) {

  nav ul {

    flex-direction: column;

    align-items: center;

  }

  .item-list {

    width: 80%;

    margin-bottom: 20px;

  }

  .specialties h2, .special-offers h2 {

    font-size: 1.8em;

  }

  .specialties li, .special-offers li {

    font-size: 1.1em;

  }

}

@media (max-width: 480px) {

  header h1 {

    font-size: 2em;

  }

  .item-list {

    width: 100%;

  }

}
```

```

    nav ul {
        flex-direction: column;
    }

    nav button {
        width: 100%;
        padding: 15px;
        font-size: 1.1em;
    }
}

</style>
</head>
<body>
    <center>
        
    </center>
    <header>
        <h1>Welcome to Our Bakery!</h1>
    </header>

    <nav>
        <ul>
            <li><a href="index.html"><button>HOME</button></a></li>
            <li><a href="index1.html"><button>Cake Items</button></a></li>
            <li><a href="index2.html"><button>Ice Creams</button></a></li>
            <li><a href="index3.html"><button>Drinks</button></a></li>

```



```
<li><a href="index4.html"><button>Special Offers</button></a></li>
<li><a href="index5.html"><button>Contact Info</button></a></li>
<li><a href="index6.html"><button>Ordered Items</button></a></li>
</ul>
</nav>
```

```
<!-- Bakery Specialties Section -->
```

```
<section class="container">
```

```
<article class="specialties">
```

```
<h2>Our Specialties</h2>
```

```
<ul>
```

```
<li>Freshly baked cakes every day</li>
```

```
<li>Delicious homemade ice creams</li>
```

```
<li>Artisan drinks made with natural flavors</li>
```

```
<li>Warm and crispy croissants, muffins, and pastries</li>
```

```
<li>Gluten-free options available</li> <!-- Added extra specialty -->
```

```
</ul>
```

```
</article>
```

```
<!-- Available Items Section -->
```

```
<div class="item-list">
```

```
<h3>Cakes</h3>
```

```
<ul>
```

```
<li>Chocolate - $15</li>
```

```
<li>Vanilla - $12</li>
```

```
<li>Cup Cake - $3</li>
```

```
<li>Croissant - $6</li>
```

```
<li>Brownie - $4</li>
```

```
<li>Muffin - $2</li>
```

```
<li>Danish - $20</li>
```

```
<li>Bagel - $4</li>
```

```
<li>Pancake - $5</li>
```

```
<li>Scone - $3</li>
```

```

    <li>Red Velvet Cake - $18</li> <!-- Added extra cake item -->
    <li>Carrot Cake - $16</li> <!-- Added extra cake item -->
  </ul>
</div>

<div class="item-list">
  <h3>Ice Creams</h3>
  <ul>
    <li>Chocolate Ice Cream - $55</li>
    <li>Vanilla Ice Cream - $35</li>
    <li>Strawberry Ice Cream - $22</li>
    <li>Mint Chocolate Chip - $12</li>
    <li>Cookies and Cream - $25</li>
    <li>Peach Sorbet - $18</li>
    <li>Coffee Ice Cream - $28</li>
    <li>Caramel Swirl Ice Cream- $30</li>
    <li>Butter Pecan Ice Cream - $27</li>
    <li>Pistachio Ice Cream - $32</li>
    <li>Rocky Road Ice Cream - $33</li>
  </ul>
  <div class="item-list">
    <h3>Drinks</h3>
    <ul>
      <li>Coffee - $3</li>
      <li>Tea - $2.5</li>
      <li>Fresh Juice - $4</li>
      <li>Milkshakes - $4.5</li>
      <li>Hot Chocolate - $5</li>
      <li>Iced Tea - $3.5</li>
    </ul>
  </div>
</div>
</section>
</body>
</html>

```

4.2.Test Cases:

Test Case ID	Description	Expected Result
1.1	Load website on different browsers	Website loads correctly on all browsers
1.2	Click navigation menu items	Redirects to correct pages
1.3	Check responsiveness on various devices	Website is responsive on all devices
2.1	Verify product images are displayed correctly	All product images are visible and correct
2.2	Check product details (name, price)	Product details are accurate and visible
2.3	Test "Add to Cart" functionality for each product	Items are added to cart successfully
3.1	Add multiple items to cart and verify storage	Cart contents are stored correctly in local storage
3.2	Update cart by adding/removing items	Cart updates correctly
3.3	Check cart persistence across page reloads	Cart contents persist across page reloads
4.1	Simulate checkout process	Checkout proceeds smoothly
4.2	Verify checkout button visibility and click ability	Checkout button is visible and clickable
5.1	Test error handling for duplicate items in cart	Handles duplicate items gracefully
5.2	Check handling of missing product images	Website handles missing images gracefully
6.1	Verify encryption of sensitive data	Sensitive data is encrypted
6.2	Ensure HTTPS is used for transactions	HTTPS is used for secure transactions
7.1	Load test the website	Handles multiple users without lag
7.2	Check page load times on different internet speeds	Page loads quickly on various internet speeds

5. RESULTS

1. Code Structure and Organization

- The HTML structure is well-organized with clear sections for header, navigation, and product display.
- CSS styles are comprehensive, covering layout, responsiveness, and visual effects.

2. Functionality

- Product Display: Products are displayed in a grid layout, which is responsive across different screen sizes.
- Cart Functionality: The "Add to Cart" buttons use JavaScript to store items in local storage. However, there are two versions of the addToCart function, which might cause confusion. The second version is more comprehensive, handling quantity updates and storing more item details.
- Navigation: Navigation links are styled as buttons, which is visually appealing but might not be the best choice for accessibility (using `<a>` tags is generally preferred).

3. Responsiveness

- The website is responsive, with media queries adjusting the layout for smaller screens (e.g., tablets and smartphones).
- On smaller screens, the grid layout changes to display fewer items per row, improving usability.

4. Security and Performance

- Security: The code does not explicitly handle security concerns like HTTPS or data encryption, which is crucial for real-world applications.
- Performance: The code does not include any performance optimization techniques like minification or caching, which could improve page load times.

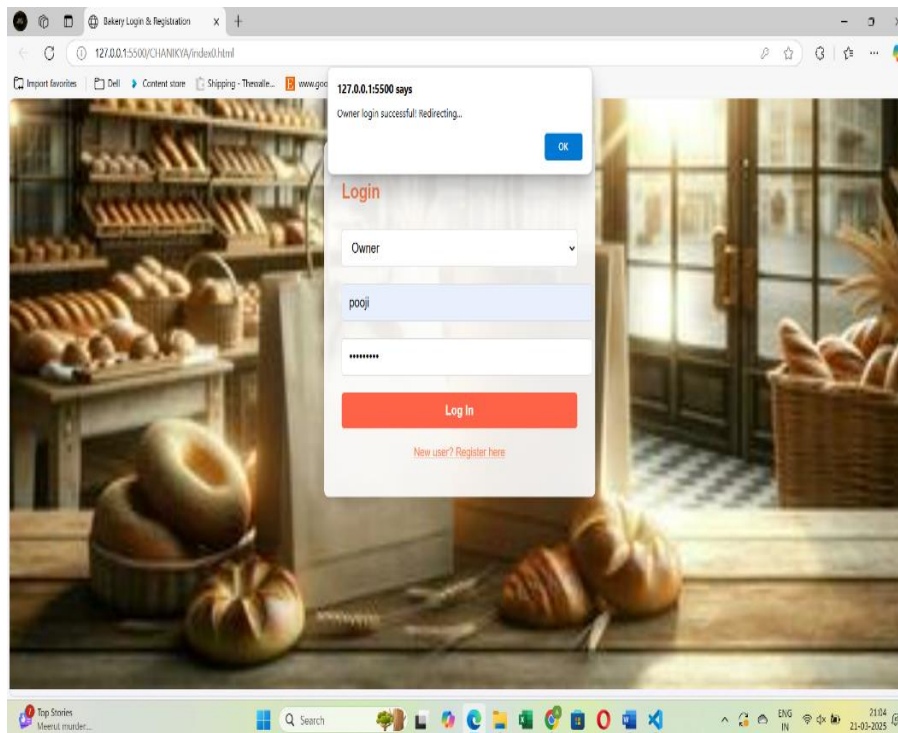
5. Improvement Suggestions

- Consolidate addToCart Functions: Use a single, comprehensive version of the addToCart function to handle all cart operations.

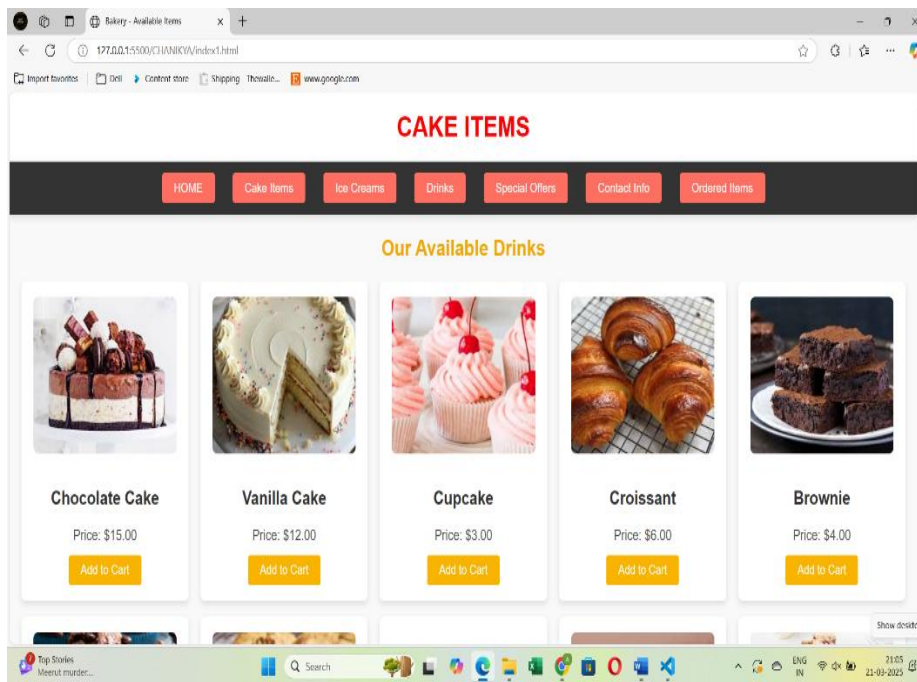
- **Implement Backend Integration:** For checkout functionality and secure data handling, integrate with a backend server.
- **Enhance Accessibility:** Consider using <a> tags for navigation links and ensure all interactive elements are accessible via keyboard.
- **Optimize Performance:** Use techniques like minification, caching, and image compression to improve page load times.

5.1.Output Screens:

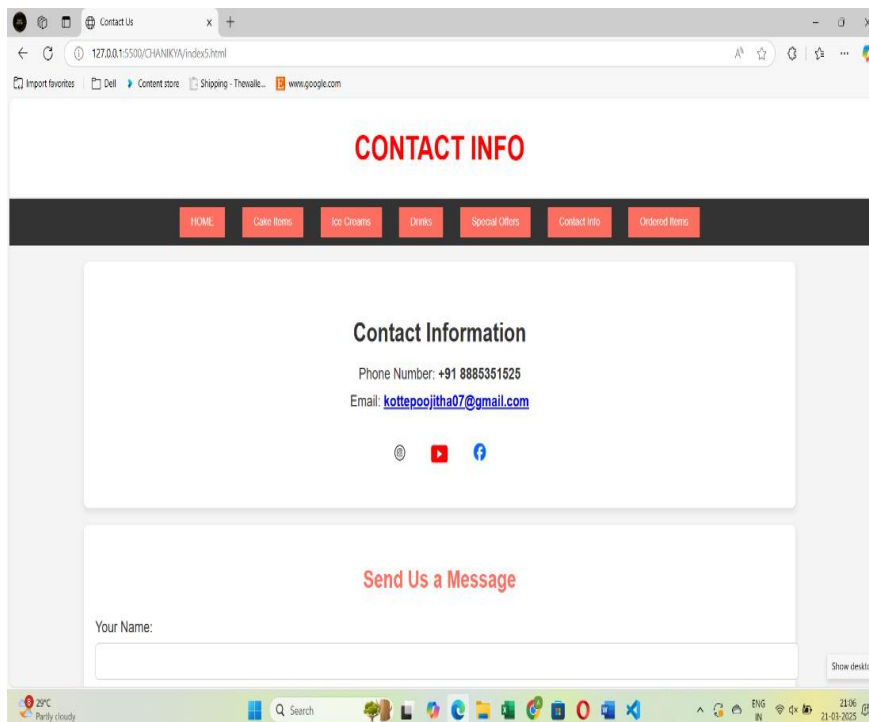
Login page: -



5.2 Menu List:-



5.3 Contact Info:-



5.4 Customer Details:

Send Us a Message

Your Name:

Your Email:

Mobile Number:

Any Suggestions:

Contact Us

State:

City:

5.5 Order Placement:-

Your Cart

Vanilla Cake ₹12

Croissant ₹6

Summary

Items in your cart: 2
Total: ₹30

Choose Payment Method

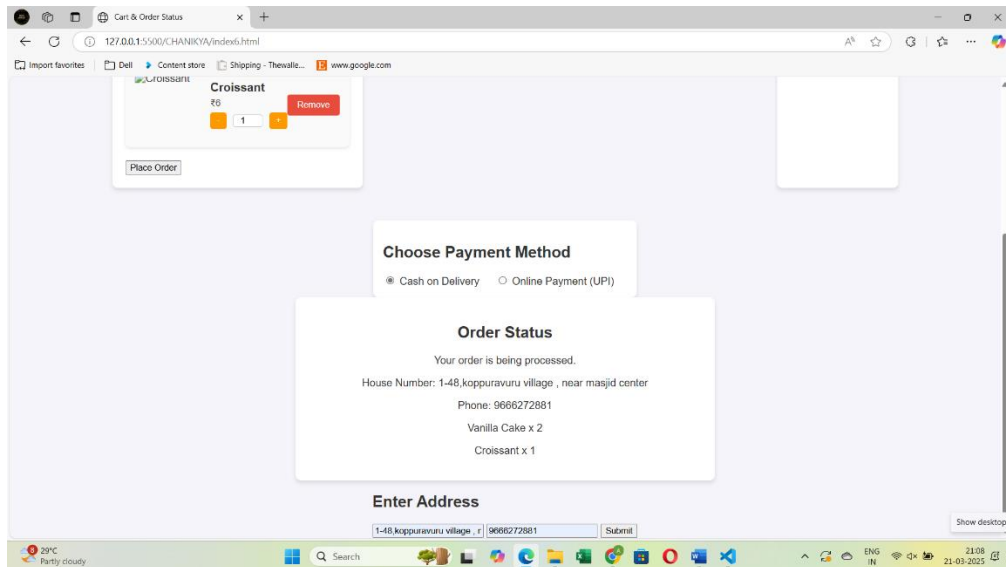
☒ Cash on Delivery ☐ Online Payment (UPI)

Enter Address

1-48,koppuravuru village , r 0996272881

127.0.0.1:5500 says
Order placed successfully!
Address: 1-48,koppuravuru village , near masjid center
Phone: 9666272881

5.6 Order Status:-



6.CONCLUSION

Eventify addresses these challenges by offering a comprehensive, web-based platform that simplifies event planning and management. With its centralized dashboard, dynamic cost calculator, and customizable features, Eventify enables users to plan and manage events efficiently, ensuring they stay within budget and meet their preferences. The intuitive, responsive interface allows event planners to easily navigate the process on both desktop and mobile devices, providing a seamless experience from start to finish. By centralizing all event details, Eventify streamlines the planning process, reduces errors, and enhances the overall user experience, making it the ideal solution for event management.

REFERENCES

To develop the Eventify website, the following technologies and resources were used:

- HTML (Hyper Text Markup Language) – Used for structuring the web pages and defining the layout of content on the home page.
- CSS (Cascading Style Sheets) – Applied for styling the website, including colours, fonts, layouts, and responsive design.
- JavaScript – Implemented for adding interactivity, such as dynamic cost calculations, real-time updates, and smooth navigation.
- Class Notes – Concepts and best practices from class notes were incorporated to enhance functionality, improve user experience, and ensure clean, maintainable code.

<https://github.com/poojitha0709/BAKERY.git>

