

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The Online Platform for Dessert Products is an innovative web-based solution designed to help a local bakery expand its customer base and streamline the ordering process. The project was built using the MEAN stack (MongoDB, Express, Angular, and Node.js), the platform provides customers with a convenient way to browse and order a wide variety of bakery products, both for personal and bulk event orders. The platform features an easy-to-use interface where customers can view detailed product descriptions, check product availability, and explore special promotions. Secure online payment options and real-time order tracking ensure a smooth and reliable customer experience. Additionally, the platform includes an inventory management system that enables the bakery to keep track of stock levels and fulfill orders efficiently. Incorporating a customer support system further enhances user experience, allowing customers to get quick assistance with any inquiries or issues.

The platform leverages modern web technologies to deliver a fast, secure, and scalable solution, providing a strong foundation for future growth. The system automates order processing, significantly reducing manual workload and improving overall operational efficiency. The integration of advanced data analytics enables the bakery to gain valuable insights into customer preferences and market trends, facilitating informed decision-making. The platform's scalability ensures it can handle increasing customer demand without compromising performance, supporting long-term business expansion. Additionally, features such as personalized product recommendations, secure payment processing, and real-time order tracking enhance customer satisfaction and build trust.

1.2 PROBLEM DEFINITION

The local bakery faces challenges in reaching a wider audience and efficiently managing customer orders. With the growing demand for convenience and online shopping, customers increasingly prefer to place orders digitally. The bakery's traditional in-store ordering system limits its ability to handle bulk orders, track inventory, and offer a seamless customer experience. Additionally, customers often face difficulties in accessing product availability information and tracking their orders. The bakery also struggles with maintaining accurate stock levels and providing timely responses to customer inquiries, leading to potential dissatisfaction and missed business opportunities. Without an online presence, the bakery risks losing customers to competitors offering more accessible and convenient platforms for placing orders.

This project aims to address these challenges by developing a robust and scalable online platform that allows customers to easily browse products, place orders, make secure payments, track orders in real-time, and communicate with customer support. The system will also streamline inventory management for the bakery, helping to improve operational efficiency and enhance customer satisfaction. By automating order processing and stock management, the platform will reduce manual errors and save time for bakery staff. The inclusion of a user-friendly interface will ensure a smooth and hassle-free shopping experience for customers. Moreover, the platform's ability to handle high traffic and support future business growth will provide the bakery with a competitive edge in the market. The platform will also gather customer data and feedback to offer personalized recommendations, further improving customer engagement and loyalty.

1.3 OBJECTIVES

The main objectives of the Online Platform for Dessert Products is to provide an easy and accessible way for customers to browse and order bakery products online. This includes allowing customers to place individual or bulk orders, view detailed product information, check availability, and explore special offers, all while ensuring secure online payments and real-time order tracking. The platform aims to improve the customer experience by offering a user-friendly interface that works seamlessly across devices and provides efficient support through an integrated customer service system. The goal is to create a convenient and reliable shopping experience that encourages customer retention and increases customer satisfaction.

The platform will also enable customers to customize their orders based on preferences, such as flavor and quantity, enhancing the overall shopping experience. Additionally, the platform will feature a feedback and rating system, allowing customers to share their opinions and helping the bakery improve its offerings. By simplifying the ordering process and improving customer interaction, the platform aims to build long-term customer loyalty and trust.

Another key objective is to enhance the bakery's operational efficiency by implementing real-time inventory management and automated order processing systems. This will allow the bakery to manage stock levels, fulfill orders promptly, and ensure smooth order fulfillment. The system will also reduce manual workload and human error, leading to improved accuracy and faster service. By integrating secure payment gateways, the platform will offer customers a trustworthy environment for transactions, increasing customer confidence and loyalty. The platform's scalability ensures that it can handle increasing demand and support future business growth without compromising performance. Additionally, the platform will provide valuable insights into customer behavior and purchasing patterns through data analysis, helping the bakery make informed business decisions and create targeted marketing strategies. By leveraging these features, the project aims to help the bakery reach a larger audience, drive sales, and ultimately contribute to long-term business growth and market competitiveness.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing system at the bakery relies on traditional in-store ordering, where customers visit the bakery to place orders, either for personal or bulk purchases. This manual process can lead to long waiting times and inefficiencies, particularly for bulk orders. There is no digital platform to facilitate online orders, check product availability, or track orders in real time. Inventory management is handled manually, leading to potential errors and delays in fulfilling orders. Customer support is limited to in-person or phone inquiries, which can be inefficient.

This system restricts the bakery's growth, limiting its ability to reach a wider audience and meet modern customer expectations. Additionally, customers face difficulties in accessing detailed product information and exploring available discounts or special offers. The absence of an automated system also makes it challenging to analyze customer behavior and preferences, reducing the bakery's ability to tailor its offerings. Without an online presence, the bakery struggles to compete with modern businesses that offer fast and convenient digital ordering solutions.

The existing system has following disadvantages.

- Limited accessibility as customers must visit the bakery in person to place orders.
- Inefficiency in order processing, leading to longer waiting times and potential errors.
- No real-time order tracking, causing uncertainty for customers.
- Manual inventory management increases the risk of stock discrepancies and delays.
- Limited customer support available only in person or via phone.
- Lack of an online presence, restricting the bakery's ability to reach a large audience.
- Scalability issues, unable to handle increased order volume or demand.

2.2 PROPOSED SYSTEM

The proposed system is a web-based Online Platform for Dessert Products, designed to streamline the bakery's ordering process and enhance customer experience. The platform will enable customers to browse and order bakery products online, either for personal or bulk orders, from the comfort of their homes. It will include features like detailed product information, availability checks, special offers, secure payment options, and real-time order tracking, providing a seamless shopping experience. The system will incorporate an automated inventory management feature to keep track of stock levels in real-time, ensuring accurate product availability and timely order fulfillment. Customer support will be integrated into the platform, offering prompt assistance through live chat or email, improving overall customer satisfaction. Built using the MEAN stack (MongoDB, Express, Angular, Node.js), the platform will be fast, secure, and scalable, supporting future growth. This digital solution will allow the bakery to reach a broader audience, improve order efficiency, and ultimately contribute to business growth by attracting more customers and offering a convenient online shopping experience.

The main advantages of using the proposed system are:

- The ordering process for bakery products is streamlined and user-friendly, making it easy for customers to place orders online.
- The user interface is intuitively designed, enhancing navigation and accessibility for users.
- Administrators can easily update product information and availability, ensuring customers have the latest details.
- The system includes dynamic features that allow for quick updates and changes in response to market demands.
- Information is presented clearly and dynamically, reducing confusion and improving the overall customer experience.

2.3 FEASIBILITY STUDY

Feasibility studies are essential for assessing the proposed system's functions, its potential impact on the bakery, the ability to meet customer requirements, and how effectively resources will be used. While any project can be considered feasible with unlimited resources and time, it is important to evaluate the practicality and associated risks of software development. The proposed Online Platform for Dessert Products will go

through a proof-of-concept phase to establish its viability before moving forward with full development. Understanding the relationship between feasibility and risk is crucial, especially when project risks are high and development feasibility is uncertain.

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

2.3.1 Technical Feasibility

Technical feasibility examines whether the necessary technology and resources are available for the successful development and implementation of the new system. This stage can be challenging, as the processes of analysis and definition must occur concurrently with the assessment of technical feasibility. A key factor to consider is whether the organization has sufficient resources for both development and implementation. Additionally, the availability of skilled personnel to operate and maintain the system is crucial to ensuring long-term success. Another important aspect is the system's ability to integrate with existing infrastructure without causing disruptions. Given that the proposed system requires minimal resources and leverages existing technologies, it is classified as technically feasible.

2.3.2 Operational Feasibility

Operational feasibility evaluates whether the system effectively meets the needs of both customers and administrators. The proposed system is designed to provide a user-friendly experience, ensuring ease of use for both customers and administrators while handling high traffic efficiently to support global accessibility. It integrates real-time features for smooth service delivery, automates order processing and inventory management to reduce manual errors, and enhances customer trust with secure payment options and real-time order tracking. Additionally, the system's scalability allows it to adapt to growing business needs, reinforcing its long-term operational success and creating better market opportunities for service providers. The system's ability to provide detailed sales and inventory reports enables better decision-making and operational planning. By centralizing customer and order data, the platform allows for quick resolution of customer issues, improving overall service quality.

2.3.3 Economical Feasibility

Economic feasibility assesses the project's development costs in relation to the potential revenue and benefits it will generate. The proposed project is advantageous because it does not require any additional hardware or software investments, ensuring its financial viability for the bakery. By carefully analyzing costs against expected returns, the project demonstrates strong economic rationale, ensuring that the investment will lead to positive financial outcomes for the bakery. This thorough assessment of economic feasibility underscores the project's capacity to deliver value and sustainability. The system's ability to automate processes reduces operational costs and increases overall efficiency. The projected increase in customer engagement and satisfaction is expected to enhance revenue streams. Furthermore, the scalability of the system allows for future growth without significant additional investment, reinforcing its long-term economic benefits.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE SPECIFICATION

- Processor : Intel® Core™ i5- 2.00GHz
- Ram : 8GB
- System type : 64-bit operating system
- Hard disk : 512GB
- Keyboard : Standard 102 keys

3.2 SOFTWARE SPECIFICATION

- Operating System : Windows 11
- Front End : HTML5.0, ANGULAR JS
- Back End : NODE JS, MONGO DB, EXPRESS JS
- Environment : Visual Studio, Chrome

3.2.1 Frontend

- **HTML:** HTML (Hypertext Markup Language) is the foundation of the web and a standard for creating webpages. It structures content by organizing text, images, and multimedia, defining elements like headings, paragraphs, links, tables, and forms to create an accessible and well-organized layout. HTML allows developers to create interactive and visually appealing interfaces by embedding multimedia content such as videos and audio files directly into webpages. For the Online Platform for Dessert Products, HTML ensures an easy-to-navigate interface for customers to browse product details, place orders, and access other essential features. The use of semantic HTML helps structure content logically, making it easier for search engines to index and improving search engine visibility. HTML's compatibility with modern browsers

ensures consistent performance across different platforms and devices, providing a seamless user experience. The ability to integrate with CSS and JavaScript enhances the look and functionality of the platform, improving interactivity and user engagement. Its straightforward structure makes it easier for developers to update and maintain content, ensuring that the platform remains up-to-date and functional over time.

- **JAVASCRIPT:** JavaScript is a client-side scripting language that enhances the interactivity and functionality of web applications. It allows developers to create dynamic content that can respond to user actions, such as form validation, animations, and real-time updates. JavaScript also enables asynchronous communication with the server through AJAX, allowing data to be fetched and updated without reloading the page. For the Online Platform for Dessert Products, JavaScript plays a crucial role in creating interactive features such as product search, order forms, real-time order tracking, and customer notifications. External JavaScript files provide better organization and separation of concerns, making the code easier to maintain and debug as the project grows in complexity.
- **ANGULAR JS:** AngularJS is a powerful, open-source front-end framework developed by Google, designed for building dynamic, single-page applications (SPAs). It extends HTML's capabilities with features like two-way data binding, which ensures that changes in the data model are automatically reflected in the user interface without manual intervention. AngularJS also offers dependency injection for easier service management, modular architecture for better code organization, and custom directives for creating reusable components. The framework's built-in tools for routing, testing, and data filtering make it ideal for developing complex, scalable, and maintainable web applications, such as the Online Platform for Dessert Products, providing a seamless user experience and efficient data management.
- **TYPE SCRIPT:** TypeScript is a superset of JavaScript that introduces static typing, helping developers catch errors at compile time rather than runtime. It extends JavaScript by adding features such as interfaces, classes, and modules, enabling the development of large-scale applications with better code organization and

maintainability. TypeScript improves developer productivity with advanced tooling, including auto completion, refactoring, and type inference, which makes code easier to navigate and debug. Its compatibility with JavaScript ensures that existing codebases can be gradually migrated, while its strict type-checking reduces bugs and enhances overall application stability. TypeScript also supports modern JavaScript features, ensuring compatibility with evolving web standards and frameworks like Angular, making it a preferred choice for building scalable and robust web applications.

3.2.2 Backend

- **MONGO DB:** MongoDB is a highly flexible and scalable NoSQL database system, designed to manage large and complex datasets with ease. Unlike traditional relational databases that rely on predefined schemas and tables, MongoDB stores data in flexible, JSON-like documents, which allow for dynamic and adaptable data structures. This flexibility is especially beneficial for applications like the Online Platform for Dessert Products, where various types of data need to be managed, including product details, customer profiles, order history, and payment information. MongoDB's document-based approach allows the system to handle unstructured or semi-structured data, making it easier to store and retrieve different kinds of information without the limitations of rigid schemas. This adaptability ensures that the system can evolve over time, accommodating new features and data types without requiring complex database migrations. Furthermore, MongoDB supports horizontal scaling, which enables the platform to grow in terms of both traffic and data storage. As the number of users and the amount of stored data increase, MongoDB can scale across multiple servers, ensuring that the platform remains responsive and efficient even under heavy loads, thus maintaining a smooth user experience.
- **EXPRESS JS:** Express.js is a lightweight and flexible web application framework for Node.js that simplifies the process of building web applications and APIs. It provides a robust set of features for handling HTTP requests, routing, and middleware integration, enabling developers to create scalable and maintainable server-side applications with minimal code. Express.js follows a minimalist design, allowing developers to extend its functionality using third-party middleware and

custom modules. Its event-driven and non-blocking architecture ensures high performance, making it suitable for real-time applications. Express.js offers a straightforward routing mechanism and middleware support for efficient request processing and error handling, improving the overall development experience and application performance. It also supports RESTful API development, making it easier to create and manage endpoints for data exchange between the client and server. The framework's flexibility allows seamless integration with databases like MongoDB, enhancing data handling and storage capabilities. Express.js's built-in security features, such as protection against cross-site scripting (XSS) and Cross-Site Request Forgery (CSRF), further improve the overall security of the application.

- **NODE JS:** Node.js serves as a powerful server-side runtime environment, enabling JavaScript to be executed outside the browser. It has gained widespread popularity for building scalable, high-performance web applications, particularly due to its non-blocking, event-driven architecture. For the Online Platform for Dessert Products, Node.js acts as the backbone of the server-side operations, handling incoming requests from customers, processing orders, and interfacing with the database to deliver responses. This real-time processing capability is essential for the platform, where multiple users might be browsing products, placing orders, or tracking deliveries simultaneously. Node.js excels at managing concurrent connections without becoming a bottleneck, thanks to its event-driven nature, which allows it to handle numerous requests simultaneously without blocking the execution of other tasks. This makes it particularly suitable for real-time applications, ensuring that customers experience fast and efficient service regardless of the platform's load. Additionally, its vast ecosystem of modules and packages allows developers to build feature-rich applications quickly and effectively, contributing to the overall scalability and performance of the platform.

CHAPTER 4

SYSTEM DESCRIPTION

4.1 MODULE DESCRIPTION

The project contains the following modules such as:

- User Management module
- Cart Module
- Product Module
- Review Module
- Payment Module
- Order Module
- Admin Module

4.1.1 User Management Module

The User Management Module allows users to manage their account details and settings efficiently. The login module enables users to log in by entering their username and password, ensuring security through encrypted storage and validation of credentials. The registration module requires users to provide a username, email, password, and confirmation password (cpassword) to create an account securely. Users can edit their profile details, such as updating their username, email, and password, and upload a profile picture. The module also includes an address management feature, allowing users to add, update, and delete delivery addresses. Additionally, it manages user sessions to maintain secure and consistent access, automatically logging out inactive users to enhance security.

4.1.2 Cart Module

This module allows users to update their personal details such as username, email, and password. Users can modify their profile information at any time, including uploading a profile picture. The module also provides an option to change security settings like the password, ensuring users can maintain the security of their accounts. Profile changes are subject to authentication checks for user safety. The module ensures that any sensitive data is encrypted, protecting user information from unauthorized access.

4.1.3 Product Module

This module focuses solely on displaying the products offered by the bakery. Each product is accompanied by detailed descriptions, including features, pricing, and any applicable promotions. Users can view images, product specifications, and related details, but there is no option for online purchase in this module. The system may provide guidance on how to obtain the products physically or via inquiry.

4.1.4 Review Module

The review module allows users to share their feedback and experiences regarding the bakery's products and services. Users can submit ratings and written reviews, which will be visible to other customers. This module helps the bakery gather valuable customer insights, improve product offerings, and enhance customer satisfaction. Reviews will be monitored to ensure the quality and relevance of feedback. Positive reviews can serve as social proof, encouraging more customers to engage with the platform.

4.1.5 Payment Module

This module handles payments for bakery products through UPI QR codes or cash. For UPI, users can scan a QR code using their mobile phone to complete the payment process, or deep linking can be provided for direct payments on a laptop. There is no support for other payment methods such as credit or debit cards. The system confirms payments and sends receipts to users after successful transactions.

4.1.6 Order Module

The Order Module allows customers to place and manage their orders efficiently. Customers can track order status in real-time and view their order history. The module supports secure payment processing and updates inventory automatically upon order completion. It also provides options for order cancellation and return requests.

4.1.7 Admin

The admin module enables administrators to manage and monitor all aspects of the platform. Admins can view and manage user accounts, update product details, and manage product listings. They also have the ability to track payment records, ensuring transactions through UPI or cash are logged. Additionally, admins can edit or remove content, ensuring that the system stays up to date with accurate information on products.

4.2 USE CASE DIAGRAM

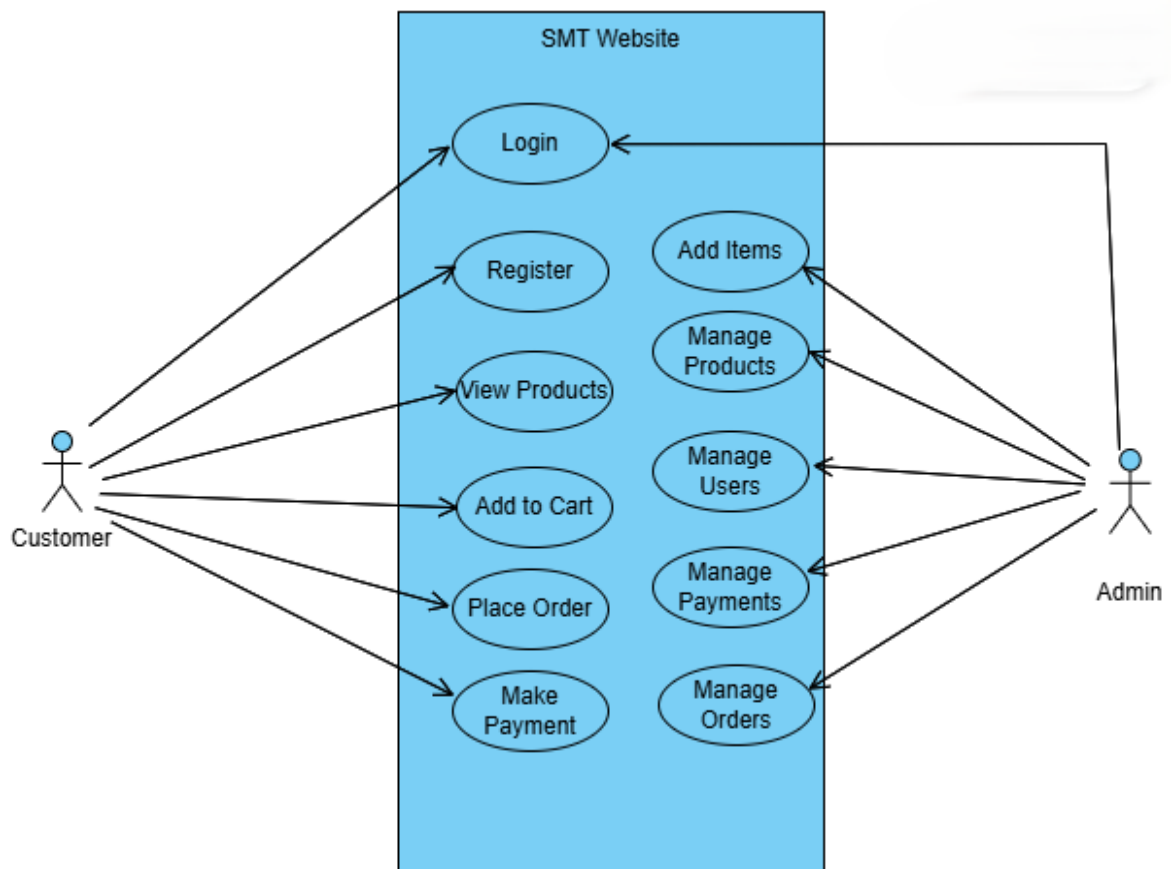


Figure 4.1 Use Case Diagram

4.3 SYSTEM FLOW DIAGRAM

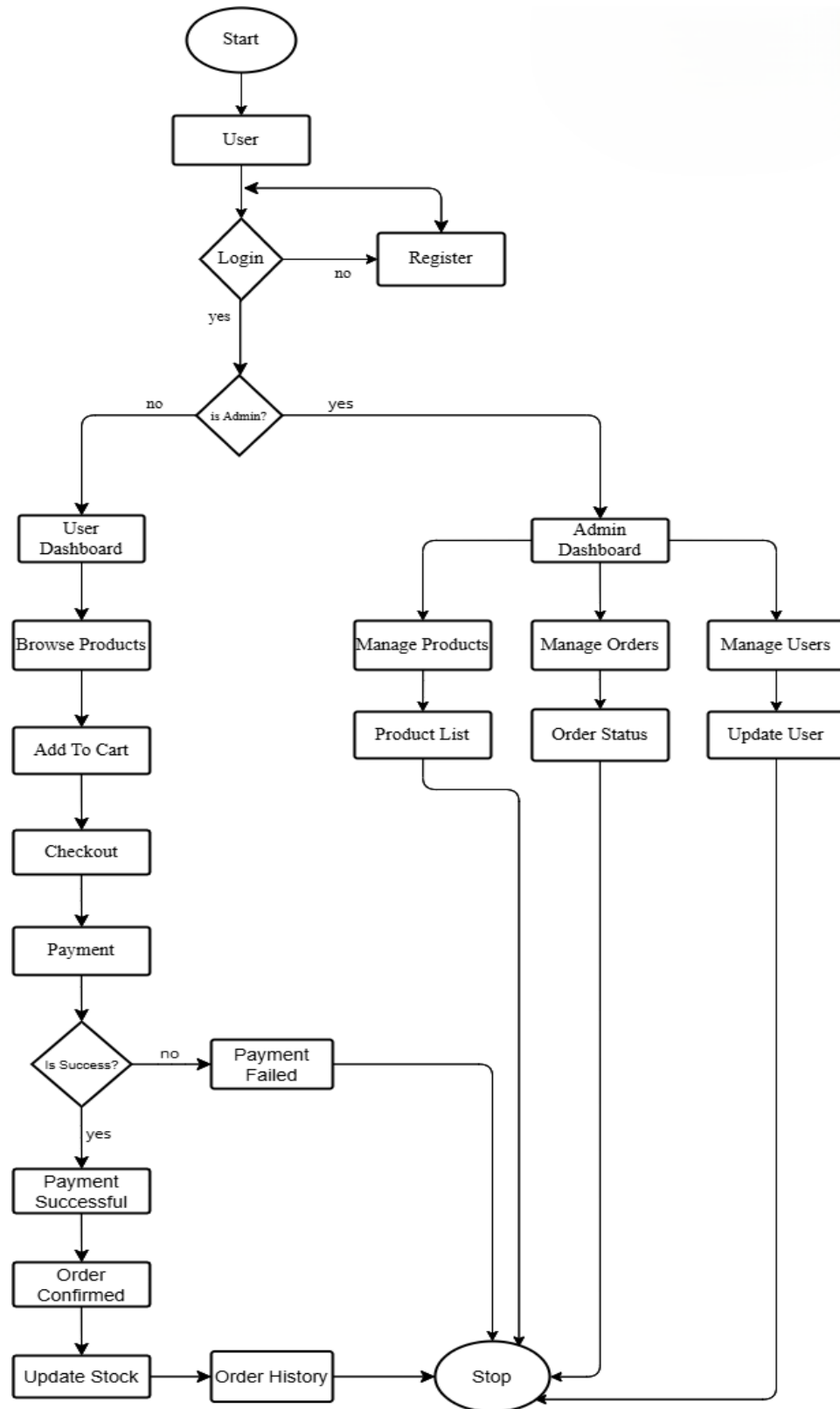


Figure 4.2 System Flow Diagram

4.4 DATA FLOW DIAGRAM

4.4.1 Data Flow Diagram Level 0

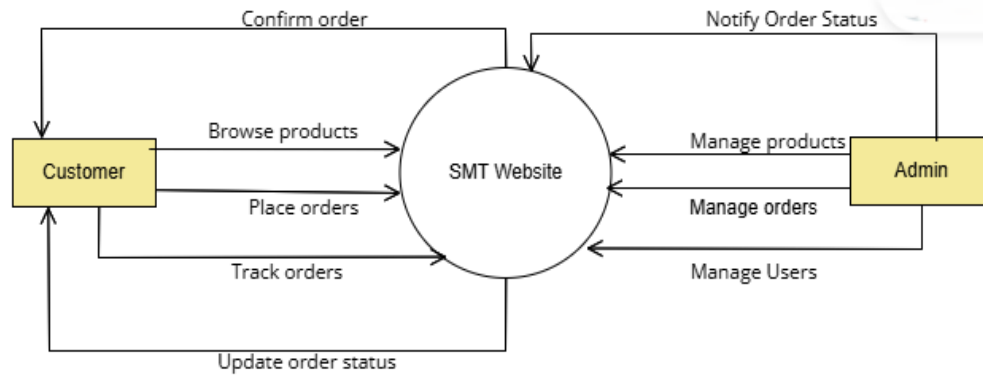


Figure 4.3 Data Flow Diagram Level 0

4.4.2 Data Flow Diagram Level 1

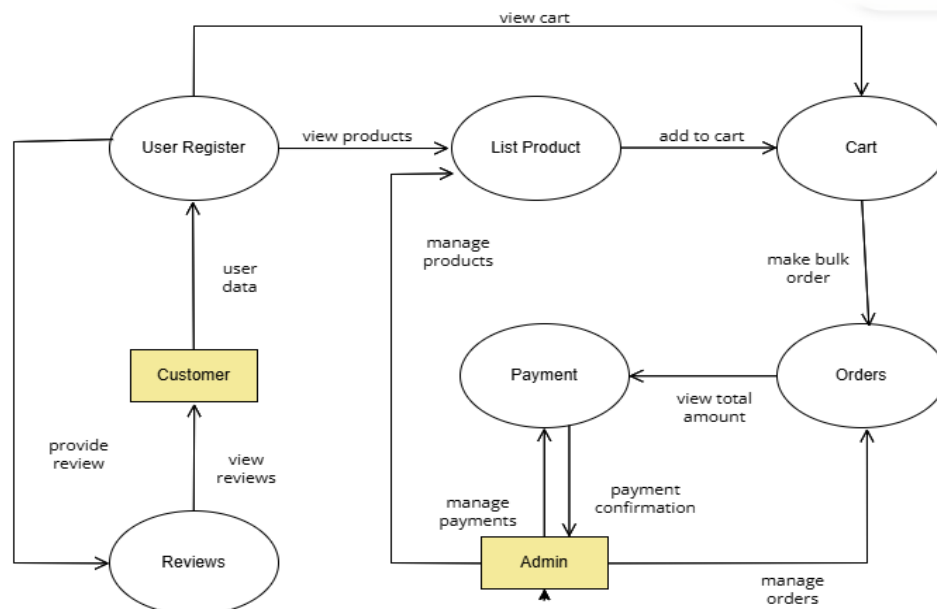


Figure 4.4 Data Flow Diagram Level 1

4.5 DATABASE DESIGN

COLLECTION NO : 4.6.1

COLLECTION NAME : User Details

DESCRIPTION : The table is used to store the login details of admin and the customers.

FIELD NAME	DATA TYPE	DESCRIPTION
_id	Object	ObjectId
name	String	User Name
email	String	User Email
Phno	Number	Phone number
fullname	String	User Full Name
role	String	User Role
password	String	User Password
pin_code	Number	PIN Code
country	String	User Country
createdAt	Date	Account Creation Date
updatedAt	Date	Account Updation Date

COLLECTION NO : 4.6.2

COLLECTION NAME : Product Details

DESCRIPTION : The table is used to store the products details.

FIELD NAME	DATA TYPE	DESCRIPTION
_id	Object	ObjectId
name	String	Product name
src	String	Product Image Path
type	String	Product Tyoe
desc	String	Product description
price	Number	Product Price
rating	Number	Product Rating
isVeg	Boolean	Specification

minquantity	Number	Product Minimum Quantity
pieces	Number	Product Pieces in Box
maxquantity	Number	Product Maximum Quantity
stock_quantity	Number	Product Quantity

COLLECTION NO : 4.6.3

COLLECTION NAME : User Addresses

DESCRIPTION : The table is used to store the user address details.

FIELD NAME	DATA TYPE	DESCRIPTION
_id	Object	ObjectId
userId	Object	User ObjectId
fullname	String	User Fullname
phoneNumber	Number	User Phone Number
pinCode	Number	User Current PIN Code
houseNo	String	User House Name/Number
roadNo	String	User Road Name/Number
state	String	User Current State
city	String	User City
address	String	User Current Address
createdAt	Date	Address Creation Time
updatedAt	Date	Address Updation Time

COLLECTION NO : 4.6.4

COLLECTION NAME : Reviews Details

DESCRIPTION : The table is used to store the user review details.

FIELD NAME	DATA TYPE	DESCRIPTION
_id	Object	ObjectId
userId	Object	User ObjectId
name	String	User Name
rating	Number	User Product Rating

comment	String	User Product Comments
image	String	User Product Image Path
date	Date	User Review Creation Date

COLLECTION NO : 4.6.5

COLLECTION NAME : Cart Details

DESCRIPTION : The table is used to store the user cart details.

FIELD NAME	DATA TYPE	DESCRIPTION
_id	Object	ObjectId
userId	Object	User ObjectId
productId	Object	Product Object Id
productName	String	Product Name
quantity	Number	Product Quantity
price	Number	Product Price
totalPrice	Number	Products Total Price
imgSrc	String	Product Image Path
stock_quantity	Number	Product Stock Quantity
product_is_veg	Boolean	Product Specification
product_type	String	Product Type
max_quantity	Number	Product Maximum Quantity
min_quantity	Number	Product Minimum Quantity
pack_quantity	Number	Product Pieces In Box
createdAt	Date	Cart Creation Time
updatedAt	Date	Cart Updation Time

COLLECTION NO : 4.6.6
 COLLECTION NAME : Delivery Details
 DESCRIPTION : The table is used to store the delivery details.

FIELD NAME	DATA TYPE	DESCRIPTION
_id	Object	ObjectId
city	String	City Name
state	String	State Name
charge	Number	Shipping Charge
isAvailable	Boolean	Shipping Available

COLLECTION NO : 4.6.7
 COLLECTION NAME : Coupon Codes Details
 DESCRIPTION : The table is used to store the coupon details.

FIELD NAME	DATA TYPE	DESCRIPTION
_id	Object	ObjectId
coupon_code	String	Coupon Code
exp_date	Date	Coupon Code Expiry Date
used_userIDs	Array	Used Code User ObjectId
max_use	Number	Maximum Usage

COLLECTION NO : 4.6.8
 COLLECTION NAME : Order Details
 DESCRIPTION : The table is used to store the order details.

FIELD NAME	DATA TYPE	DESCRIPTION
_id	Object	ObjectId
userId	Object	User ID
tems	Array	List of ordered items

totalAmount	Number	Total order amount
status	String	Order status
createdAt	Date	Order creation date
updatedAt	Date	Last update timestamp

4.6 INPUT DESIGN

In the proposed work, several key input forms have been designed to enhance the user experience and ensure smooth interaction with the bakery platform. These forms include the Customer Registration Form and the Product Details Form. The Customer Registration Form captures essential information such as First Name, Last Name, Email ID, and Password. It incorporates validation checks for critical fields like password strength and mobile number formatting, ensuring data accuracy and security. If the data entered is invalid, appropriate error messages are displayed to guide the user in correcting their input. The input design features the Login Form, Register Form, Edit Profile Form, and Order Form, each tailored to specific user needs.

The Login Form is the entry point for customers to access their accounts. It is designed with simplicity and security in mind, requiring a username (or email) and password for authentication. To enhance security, features like password recovery options and two-factor authentication are integrated, ensuring user safety. The form is responsive, adapting to different device sizes and platforms, ensuring a smooth experience for users on both mobile and desktop. Additionally, clear and concise error messages are presented when incorrect information is entered, helping users quickly resolve login issues.

The Register Form facilitates the creation of new customer accounts. It collects vital information, such as the customer's name, email address, and password, with a focus on maintaining high levels of security through measures like email verification and strong password requirements. The form's design is intuitive and user-friendly, guiding users step-by-step through the registration process. Options for social media logins can also be provided to simplify account creation. By clearly presenting terms of service and privacy policies, the form builds trust with users, increasing the likelihood of successful registrations and long-term engagement.

The Edit Profile Form allows customers to update their personal information securely and efficiently. It is designed to pre-fill existing customer data, enabling quick and easy modifications. Security is a priority, with password re-entry required for significant changes such as email or password updates, preventing unauthorized alterations. Validation messages ensure that all data entered is accurate, and clear instructions help users understand the steps involved. The design emphasizes ease of use, ensuring that even less tech-savvy users can comfortably update their profiles. The Order Form allows customers to place orders for bakery products quickly and efficiently. It captures essential details like the product name, quantity, delivery address, and payment method. Dropdown menus and checkboxes simplify the selection process, and a summary of the order, including total cost and delivery time, is provided before submission. Immediate confirmation upon submission reassures customers that their order is successful, and clear next steps, such as order tracking, are provided. The form is designed with a clean and organized layout, enhancing usability and making the ordering process straightforward and engaging for users, regardless of their technical expertise.

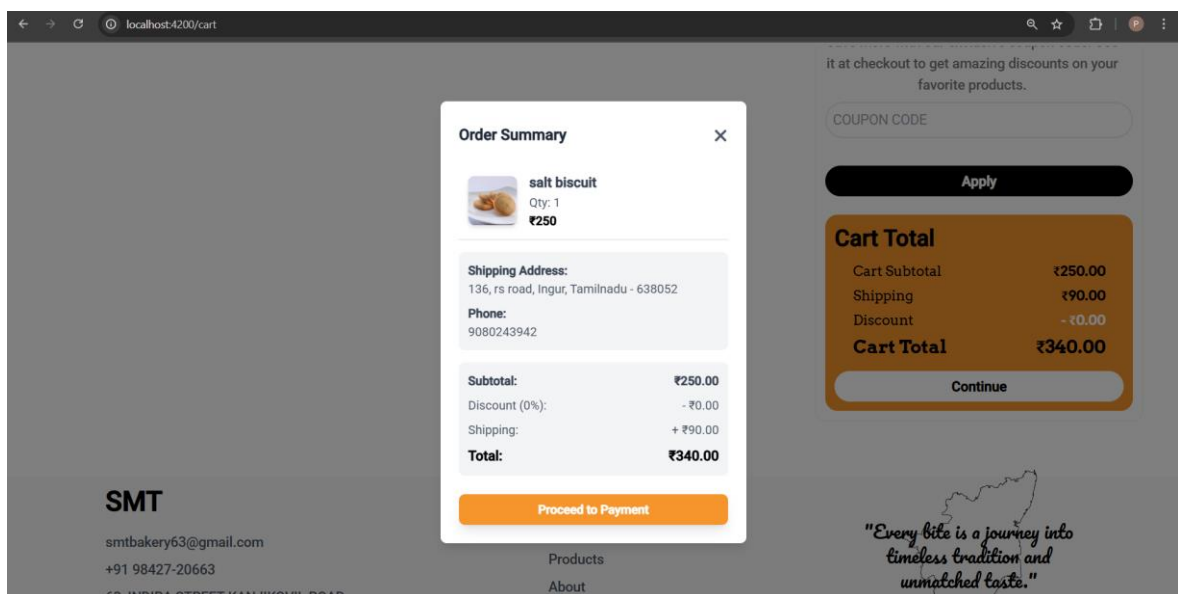


Figure 4.5 Checkout Page

4.7 OUTPUT DESIGN

The output in this project focuses on delivering critical information to customers in a clear and understandable manner, with the primary output being the Order Summary section, which provides detailed information about the products customers have ordered. This section enables customers to efficiently manage their orders and payments by displaying key details such as Product Name, Quantity, Payment Type, and Total Cost. The output design includes a flexible Payment Module supporting both online and offline methods. For online orders, customers are guided through a seamless payment process with clear instructions, while for offline orders, customers can make cash payments upon delivery or use a QR code for quick digital payments, ensuring convenience for all users. The information displayed in the Order Summary section helps customers keep track of their orders and payments, ensuring transparency and clarity in the financial aspect of their purchases. This organized and user-friendly design enhances the overall customer experience, allowing for easy reference and management of ongoing and completed orders. The goal is to deliver output that is both efficient and accessible, ensuring customers can interact with the system smoothly and confidently.

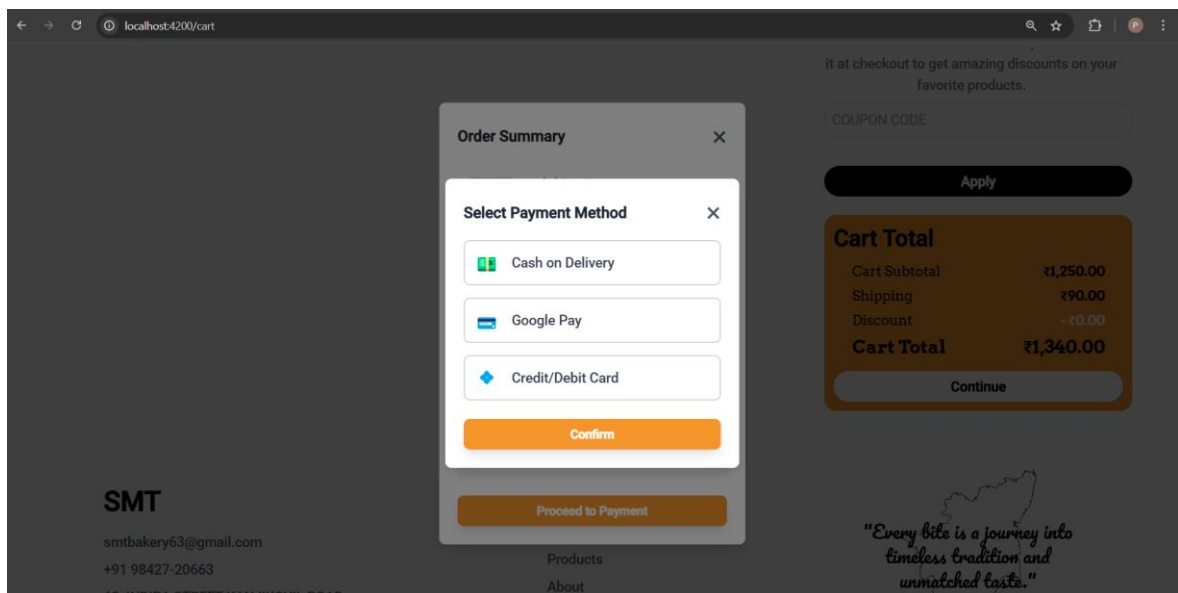


Figure 4.6 Payment Page

CHAPTER 5

SYSTEM TESTING

System testing is a type of testing that evaluates the overall functionality and performance of a complete and fully integrated software. System testing is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. It helps identify any defects in the system's interaction with integrated components and ensures that the system operates as expected under various conditions. This testing assesses both functional and non-functional aspects, verifying that the system meets all specified requirements and delivers a seamless user experience. Effective system testing helps improve software quality and ensures that the system is ready for deployment.

1. Unit Testing
2. Integration Testing
3. Validation Testing

5.1 UNIT TESTING

In unit testing, we must test the programs making up the system. By giving improper inputs, the errors occurred are noted and eliminated. This enables, to detection of errors in coding and logic that are contained within the module alone. The testing was carried out during the programming. In this system each form is considered as a separate unit and tested for errors. Every user input is unit tested for a valid accepted range.

Test Case 1

Module	: Admin Login
Login Type	: Loading of the appropriate form for the administrator
Input	: Username and Password
Expected Output	: Display admin menu

Test Case 2

Output	: Redirect to the main page and display the admin menus
Analysis	: The admin login form validates the username and password. If correct, the system redirects to the main page and displays the admin menu.

5.2 INTEGRATION TESTING

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions.

Test Case 1

Module	: Admin
Test Type	: Product Management and Order Management
Input	: Navigation between Admin options
Expected Output	: Navigation between modules is completed successfully

Test Case 2

Output	: On Clicking Login and other Admin modules the respective pages will open correctly.
Analysis	: Ensures that all admin functionalities navigate to the correct pages without errors.

5.3 VALIDATION TESTING

Verification and validation testing are two important tests, which are carried out before the product has been handed over to the customer. It determines whether the software functions as the user expected. Verification ensures that the software meets the specified requirements, while validation confirms that it satisfies the user's needs.

Test Case 1

Module	: Customer Registration
Test Type	: Register new user
Input	: Input to all required fields
Expected Output	: Required fields should not be empty

Test Case 2

Input : A required field is left empty
Output : Prompt user to provide all required fields.
Analysis : Navigation to the next page is blocked until all required fields are correctly.

CHAPTER 6

SYSTEM IMPLEMENTATION

System implementation is a critical phase in project development where the theoretical design transforms into a fully operational system. This stage requires meticulous planning and control, as any oversight can lead to significant errors and complications, affecting the system's success and user confidence in its functionality and effectiveness. The implementation process begins with the evaluation of the newly designed or revised system, created by the system analysis team, followed by a demonstration to end users to verify its logical flow and functionality through various combinations of test data. Once both end users and management approve the system, it proceeds to full implementation. Key activities in this stage include several major components essential for converting the design into an operational system.

CODING

Coding is the process in which the physical design specifications, created by the analysis team, are transformed into working instructions by the programming team. This stage involves selecting appropriate programming languages and tools that align with the system's requirements. The quality of the code is vital, as it lays the foundation for the system's functionality and performance.

TESTING

Testing begins once the coding process starts and continues in parallel. Each program module is subjected to rigorous testing to ensure it operates as intended. This process includes unit testing, integration testing, and system testing to validate that all components work together smoothly. Effective testing ensures that a high-quality product is delivered to the customers, minimizing the risk of defects and enhancing user satisfaction.

INSTALLATION

Installation is the process of replacing the current system with the new one. This includes converting existing data, updating software, and aligning documentation and work procedures with the new system. Careful planning is essential during installation to ensure a seamless transition, minimizing disruptions to daily operations and ensuring that users can adapt to the new system effectively.

DOCUMENTATION

Documentation emerges from the installation process and includes user guides that provide comprehensive information on how to operate the system and its flow. This documentation serves as a vital resource for users, helping them understand the system's functionalities, troubleshooting common issues, and facilitating smoother onboarding for new users.

TRAINING AND SUPPORT

A well-developed training plan is essential to equip users with the knowledge and skills needed to navigate the new system effectively. The training strategy likely begins in the earlier stages of the project, ensuring that users can quickly adjust to the system. Ongoing support is also critical, providing users with assistance and resources as they adapt to the changes, ultimately contributing to the system's overall success and user satisfaction.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

In conclusion, implementing the Online Platform for Dessert Products transforms design ideas into real solutions that effectively meet customer needs and align with the bakery's business goals. Careful planning and execution are essential to minimize errors and ensure a smooth transition to a fully operational platform. Thorough testing and performance validation will build customer confidence and enhance user trust. Providing comprehensive guidance and ongoing support will empower customers to navigate the platform effectively. A successful implementation enhances operational efficiency and customer satisfaction, positioning the bakery for long-term success and adaptability in a competitive market.

7.2 FUTURE ENHANCEMENTS

Enhancement to be made, modifying or redeveloping the code to support changes in the specification. It is necessary to keep up with the changing user requirements and operational environment.

Following are some of the features that can be considered as future enhancements:

1. **New Product Notification:** Customers will receive email notifications about new product launches and special offers to keep them updated and engaged.
2. **Discount and Offer Updates:** Regular updates on discounts, seasonal offers, and exclusive deals to encourage repeat purchases.
3. **Wishlist Feature:** Allow customers to create a wishlist of their favorite products for easy future access and quick ordering.

APPENDIX 1- SAMPLE CODING

HOME PAGE

```

<div class="ml-24 mr-24 mt-28 mobile_nav:ml-5 mobile_nav:mr-5 mobile_nav:mt-24">
  <div class="flex items-center justify-evenly mobile_nav:flex-col ">
    <div class="w-landing_img mobile_nav:w-auto ">
      <h1 class="text-7xl font-arvo font-semibold text-h-t-color mobile_nav:text-5xl">Sri
Murugan<br>Biscuits</h1>
      <p class="font-medium text-lg mt-6 break-words text-justify">At Sri Murugan Biscuit
Bakery, our biscuits are crafted to be enjoyed at any time of the day. Designed to suit all
climates, they retain their delightful taste and crunch wherever you are. Made with a blend
of natural and artificial ingredients, every bite delivers the perfect balance of flavor and
freshness!</p>
      <button class="bg-black h-14 w-52 text-white text-lg mt-6 rounded-lg text-center font-
medium transition ease-linear delay-100 hover:scale-105" routerLink="/about">View
More</button>
    </div>
    <div class="w-landing_img mobile_nav:w-auto mobile_nav:h-auto">
      
    </div>
  </div>
</div>

<div class="flex items-start justify-around mobile_nav:flex-col-reverse mobile_nav:gap-
y-2 rounded-xl shadow-lg border-2 border-gray-200 shadow-gray-200 w-auto">
  <div class="w-full rounded-xl p-3 mobile_nav:p-2 self-center mobile_nav:self-stretch">
    
  </div>
  <div class="flex items-start justify-center flex-col gap-y-4 p-3 mobile_nav:p-2">

```

```
<h1 class="text-4xl font-bold font-arvo text-">Fresh & Flavorful: Real and Carefully
Selected Ingredients</h1>
```

```
<p class="text-lg font-medium text-justify">
```

Welcome to Sri Murugan Biscuits Bakery, where every treat is crafted with real, carefully selected ingredients. We use the finest flours, rich chocolates, and fresh, locally-sourced produce to deliver pure, wholesome flavors. From cookies to cakes, enjoy delightful treats made with freshness you can taste. Indulge in the perfect blend of tradition and quality in every bite.

```
</p>
```

```
<button class="bg-black h-14 w-52 text-white text-lg mt-0 rounded-lg text-center font-
medium transition ease-linear delay-100 hover:scale-105" routerLink="/products">View
Products</button>
```

```
</div>
```

```
</div>
```

```
<div class="p-3 mt-20 flex items-center flex-row justify-between gap-x-5
mobile_nav:flex-col mobile_nav:gap-y-5 mobile_nav:mt-8">
```

```
<div class="flex items-start justify-start gap-y-5 flex-col ">
```

```
<h1 class="text-5xl font-medium mobile_nav:text-4xl">Why Choose Us?</h1>
```

<p class="text-lg text-justify font-medium">We deliver freshly baked, handcrafted desserts made with the finest ingredients and care. Our wide range of treats caters to every taste, all at affordable prices. With strict hygiene standards, exceptional customer service, and a convenient online shopping experience, we make every moment sweeter.</p>

```
<button class="bg-black h-14 w-52 text-white text-lg mt-0 rounded-lg text-center font-
medium transition ease-linear delay-100 hover:scale-105" routerLink="/reviews">See
Reviews</button>
```

```
</div>
```

```
<div class="w-auto">
```

```

```

```
</div>
```

```

<div class="flex items-center justify-center flex-col gap-y-5 w-full">
  <div class="flex items-center justify-center flex-col gap-y-3 border-4 rounded-2xl
border-gray-200 shadow-gray-100 shadow-xl ">
    
    <h1 class="text-xl font-semibold">Perfectly Paired</h1>
    <p class="p-2 text-center">Crafted to complement your coffee, delivering a delightful
experience with every bite.</p>
  </div>
  <div class="flex items-center justify-center flex-col gap-y-3 border-4 rounded-2xl
border-gray-200 shadow-gray-100 shadow-xl">
    
    <h1 class="text-xl font-semibold">Quality Assured</h1>
    <p class="p-2 text-center">Baked daily to provide consistent taste and satisfaction in
every bite.</p>
  </div>
</div>
</div>
</div>

<div class="flex items-center justify-center flex-col mt-20 gap-y-6 mobile_nav:mt-8 p-3">
  <h1 class="capitalize font-semibold text-5xl mobile_nav:text-4xl text-center">our best
selling product</h1>
  <p class="text-center text-lg">Discover our best-selling treats – baked fresh and crafted
to delight every bite. Your perfect indulgence awaits!</p>
  <div class="flex items-center justify-center flex-row gap-x-0 mobile_nav:flex-col
mobile_nav:gap-y-6 mobile_nav:w-auto">
    <div class="flex items-center justify-center flex-col gap-y-2 p-2">
      
      <h2 class="text-2xl font-bold ">Bread</h2>
      <p class="text-center text-lg font-normal">"Soft, fresh bread baked to perfection for
every meal."</p>
    </div>
    <div class="flex items-center justify-center flex-col gap-y-2 p-2 mobile_nav:p-0">

```



```

    
    <h2 class="text-2xl font-bold">Coconut Bun</h2>
    <p class="text-center text-lg font-normal">"Sweet, soft Coconut Buns with tropical
flavor in every bite."</p>
  </div>
  <div class="flex items-center justify-center flex-col gap-y-2 ">
    
    <h2 class="text-2xl font-bold">Special Rusk</h2>
    <p class="text-center text-lg font-normal">"Golden, crunchy Special Rusk – perfect
for tea time."</p>
  </div>
</div>
<button class="bg-black h-14 w-52 text-white text-lg mt-3 rounded-lg text-center font-
medium transition ease-linear delay-100 hover:scale-105 mobile_nav:mt-0"
routerLink="/products">View More</button>
</div>

<div #counterElement class="bg-black grid justify-around items-center grid-flow-col text-
white p-3 rounded-3xl mt-8 mobile_nav:grid-flow-row mobile_nav:gap-y-6 gap-1">
  <div class="text-center flex items-center justify-center gap-1 flex-col">
    <h1 class="text-4xl font-normal">{{ Counter.value1 }}%</h1>
    <hr class="w-full bg-h-t-color h-1">
    <p>Customer Satisfaction</p>
  </div>
  <div class="text-center flex items-center justify-center gap-1 flex-col ">
    <h1 class="text-4xl font-normal">{{ Counter.value2 }}+</h1>
    <hr class="w-full bg-h-t-color h-1">
    <p>Supply Over TamilNadu</p>
  </div>
  <div class="text-center flex items-center justify-center gap-1 flex-col">
    <h1 class="text-4xl font-normal">{{ Counter.value3 }}+</h1>

```

```

    <hr class="w-full bg-h-t-color h-1">
    <p>Supply to Other States</p>
  </div>
</div>

<div class="flex items-center justify-center flex-col gap-y-10 mt-20 mobile_nav:mt-8
mobile_nav:gap-y-3 p-1 w-full">
  <h1 class="capitalize font-semibold text-5xl mobile_nav:text-4xl text-center">contact
us</h1>
  <div class="flex items-center justify-center flex-row mobile_nav:mt-6">
    <form class="flex items-start justify-start flex-col gap-y-6" #formData="ngForm"
    (ngSubmit)="contact.handleSubmitBtn(formData)">
      <input type="text" placeholder="Name *" class="text-left w-full p-3 border-black
border-2 rounded-xl" name="name" ngModel required>
      <input type="email" placeholder="Email *" class="text-left w-full p-3 border-black
border-2 rounded-xl" name="email" ngModel required >
      <textarea placeholder="Message *" class="text-left w-full p-3 border-black border-2
rounded-xl" name="message" ngModel required></textarea>
      <button class="bg-black text-white font-medium text-lg text-center w-full p-2 rounded-
xl transition ease-linear delay-100 hover:scale-95">Submit</button>
    </form>
    <div class="">
      
    </div>
  </div>
</div>

<div>

</div>

<app-loading-spinner [isLoading]="contact.makeLoad"></app-loading-spinner>

```

PRODUCT PAGE

<!-- Popup State -->

```
<div *ngIf="popUpState" class="fixed inset-0 flex items-center justify-center bg-black bg-opacity-50 ">
```

```
  <div class="bg-white w-96 p-5 rounded-xl shadow-lg mobile_nav">
```

```
    <div class="flex justify-between items-center mb-4">
```

```
      <h1 class="text-2xl font-semibold">Currently Available States</h1>
```

```
      <button class="text-gray-500 hover:text-orange-500 text-2xl"
```

```
(click)="closeOrOpenStatePopup()">✕</button>
```

```
    </div>
```

```
  <div class="max-h-60 overflow-y-auto border rounded p-2">
```

```
    <div class="grid grid-cols-2 gap-2">
```

```
      <div class="p-2 bg-gray-100 rounded flex justify-between" *ngFor="let country of countryData">
```

```
        <span class="capitalize">{{ country.name }}</span> <span class="font-semibold">{{ { country.charge | currency:'INR' } }}</span>
```

```
      </div>
```

```
    </div>
```

```
  </div>
```

```
<div class="mt-4 text-right">
```

```
  <button class="px-4 py-2 bg-h-t-color text-white rounded hover:bg-orange-600"
```

```
(click)="closeOrOpenStatePopup()">Close</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

<!-- Description SMT -->

```
<div *ngIf="popUpDescription" class="fixed inset-0 flex items-center justify-center bg-
```

```

black bg-opacity-50" (click)="closeOrOpenDescriptionPopUp()">
<div id="goToProduct" class="bg-white w-96 p-5 rounded-xl shadow-lg mobile_nav">
  <div class="flex justify-between items-center mb-4">
    <h1 class="text-2xl font-semibold capitalize">description</h1>
    <button class="text-gray-500 hover:text-red-500 text-2xl">✕</button>
  </div>

```

```

<p class="text-gray-600 text-lg text-justify">

```

Sri Murugan Biscuit Bakery brings you the perfect blend of tradition and taste with our freshly baked delights. Crafted with the finest ingredients, our biscuits and cakes are a treat for those who appreciate quality and flavor in every bite.

```

</p>

```

```

</div>

```

```

</div>

```

```

<!-- Product Details -->

```

```

<div id="goToProduct" class="flex items-start justify-start gap-x-4 flex-row p-3
mobile_nav:flex-col mobile_nav:gap-y-3 mt-24 mobile_nav:mt-16">

```

```

  <div class="flex flex-col items-center justify-center">

```

```

    <div class="w-disp_img mobile_nav:w-auto">

```

```

      <img class="h-disp_image w-disp_img mobile_nav:w-auto mobile_nav:h-96 p-1
object-cover hover:scale-105 transition ease-linear delay-100 rounded-3xl flex-1"
[src]="http://localhost:8080' + productData.src" >

```

```

    </div>

```

```

    <div class=" hover:scale-95 transition ease-linear delay-100 flex items-center justify-
center flex-row w-full mt-4 gap-x-1 mobile_nav:bottom-0 mobile_nav:fixed
mobile_nav:p-1 ">

```

```

      <button class="uppercase h-10 w-full p-8 font-bold text-lg text-white bg-black border-
none rounded-md flex items-center justify-center gap-x-1"
(click)="onAddToCartBtnClicked()">

```

```

```

```

        Add to cart</button>

```

```

      </div>

```

```

</div>
<div class="flex items-start justify-start flex-col gap-y-2 mobile_nav:p-2 p-1">
  <div class="flex items-start justify-start flex-col gap-y-1">
    <h1 class="font-semibold text-4xl mobile_nav:text-3xl capitalize font-
arvo">{{productData.name}}</h1>
    <h2 class="text-xl font-normal text-justify">{{productData.desc}}</h2>
    <div class="flex items-center justify-start gap-x-2 mt-1">
      <p class="bg-h-t-color p-1 rounded-lg text-white font-normal text-xl font-
pacifico">{{productData.rating}}</p>
      <p [innerHTML]=productData.rating | starRating:5" class="text-3xl"></p>
    </div>
    <p class="font-arvo text-3xl mt-2 font-medium">{{productData.price |
currency:'INR'}}</p>
  </div>
  <p class="w-96 font-normal border-2 border-h-t-color bg-orange-50 p-2 rounded-lg mt-
1 text-justify flex items-start justify-start gap-x-1 mobile_nav:w-auto">
    
    This product has a shelf life of 6 months from the date of manufacture.</p>
    <!-- Other Details -->
    <div class="gap-x-20 mobile_nav:w-full w-disp_product flex flex-col gap-y-4">
      <div class="p-0 mt-5 flex items-center justify-between ">
        <h1 class=" font-medium text-xl">Delivery Available</h1>
        <p class=" text-blue-600 font-medium text-lg cursor-pointer"
(click)="closeOrOpenStatePopup()">View States</p>
      </div>
      <div class="p-0 mt-2 justify-between flex items-center ">
        <h1 class=" font-medium text-xl">Min Quandity</h1>
        <p class=" font-medium text-lg">{{productData.minquantity}} Boxes</p>
      </div>
      <div class="p-0 mt-2 justify-between flex items-start ">
        <h1 class=" font-medium text-xl">Highlights</h1>
        <ul class="font-medium text-lg flex flex-col gap-y-1 list-disc">
          <li>Type: {{productData.type}}</li>

```

```

    <li>Price per Slice: {{productData.price|currency:'INR'}}</li>
    <li>{{productData.isVeg=="true"? "Vegetarian": "Non-Vegetarian"}}</li>
  </ul>
</div>

<div class="p-0 mt-2 justify-between flex items-center ">
  <h1 class="font-medium text-xl">Seller</h1>
  <p class="text-blue-600 font-medium text-lg cursor-pointer"
(click)="closeOrOpenDescriptionPopUp()">Sri Murugan Biscuit Bakery</p>
</div>

<div class="p-0 mt-2 justify-between flex items-center ">
  <h1 class="font-medium text-xl">In The Box</h1>
  <p class="font-medium text-lg "><span>{{productData.pieces}}</span> Pieces </div>
</div>
</div>

<!-- <div class="p-0 mt-2 justify-between flex items-start gap-x-16">
  <h1 class="font-medium text-xl">Description</h1>
  <p class="font-medium text-lg w-disp_img mobile_nav:w-auto text-justify ">
    Sri Murugan Biscuit Bakery brings you the perfect blend of tradition and taste with our
    freshly baked delights. Crafted with the finest ingredients, our biscuits and cakes are a treat
    for those who appreciate quality and flavor in every bite.</p>
</div> -->
</div>

<hr class="border-2 border-gray-300 mt-4 m-3">

<div class="flex items-center justify-center gap-y-4 flex-col p-3 mobile_nav:mt-3">
  <div>
    <h1 class="text-3xl self-start font-medium mobile_nav:font-2xl">You May Also
Like</h1>
  </div>

```

```

<div class=" grid grid-cols-4 place-items-center mobile_nav:grid-cols-1 gap-y-5 gap-x-4">
  <div class="border-2 w-fit h-fit rounded-2xl border-items-color shadow-xl shadow-items-color" *ngFor="let product of randomProducts">
    <div class="flex items-start justify-start flex-col"
      (click)="onClickProduct(product); this.ScrollIntoElement();">
      <div class="p-2 bg-items-color rounded-2xl">
        <img [src]="http://localhost:8080' + product.src" class="w-72 h-64 object-cover rounded-2xl hover:scale-105 transition ease-linear delay-75">
      </div>
      <h1 class="text-2xl font-normal capitalize pl-2">{{ product.name }}</h1>
      <div class="flex items-center justify-between w-full p-1">
        <p class="float-start text-lg bg-items-color text-orange-700 p-2 rounded-xl"
          [ngStyle]="{'color':matchColor[product.type] || 'orange'}">{{ product.type }}</p>
        <p class="float-end text-3xl p-2 font-medium font-arvo">{{ product.price|currency:'INR' }}</p>
      </div>
    </div>
  </div>
</div>
<app-loading-spinner [isLoading]="makeLoading"></app-loading-spinner>

```

APPENDIX 2 - SCREENSHOTS

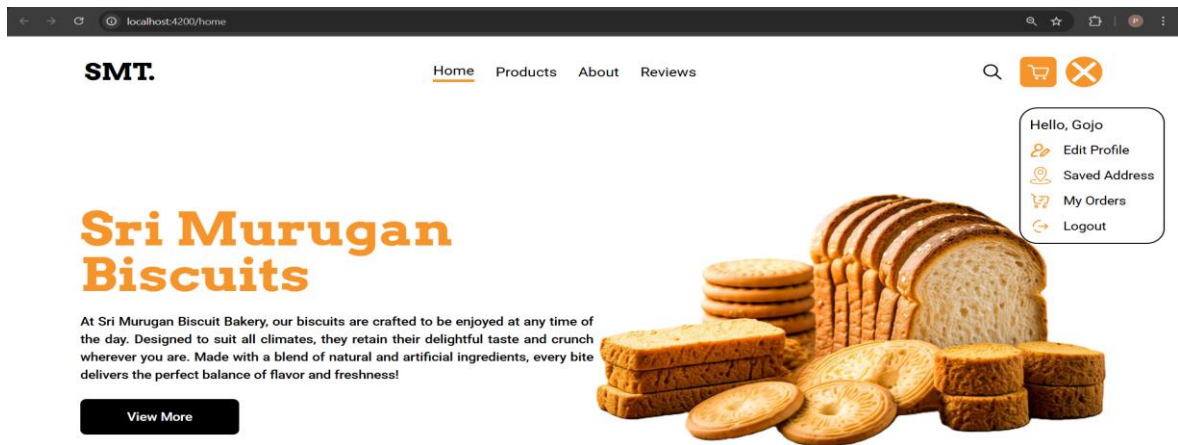
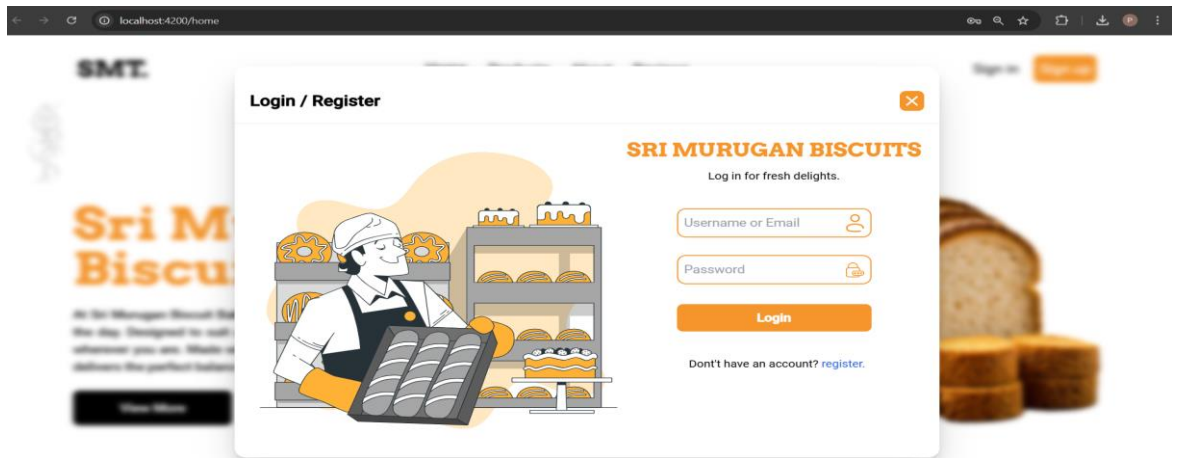


Figure. A 2.1 User Management Page

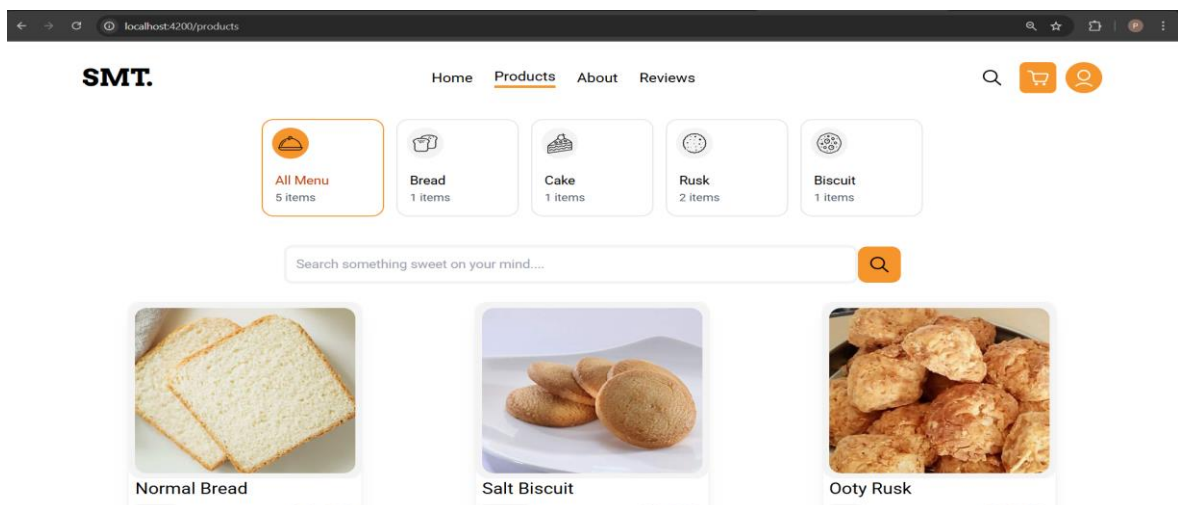


Figure. A 2.2 Product Page

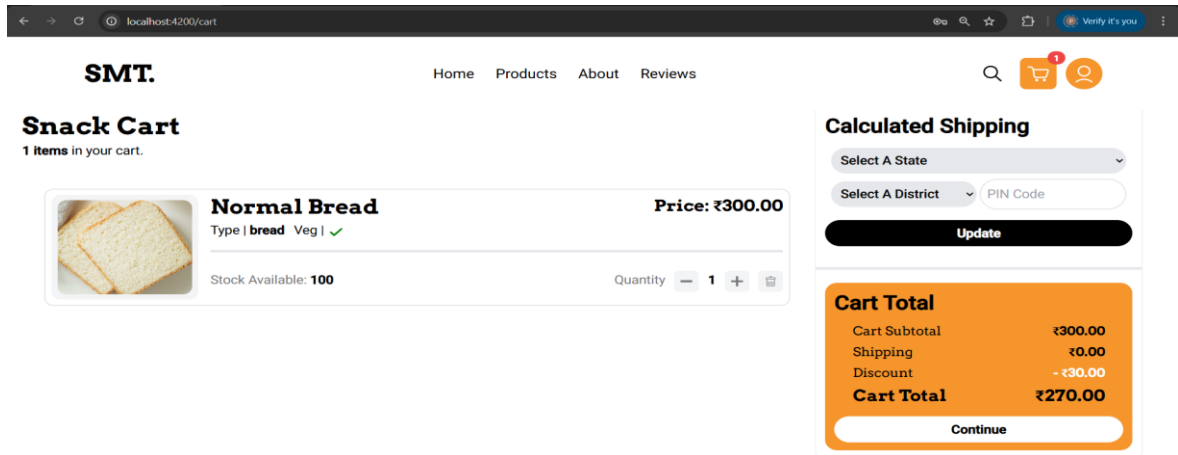


Figure. A 2.3 Cart Page

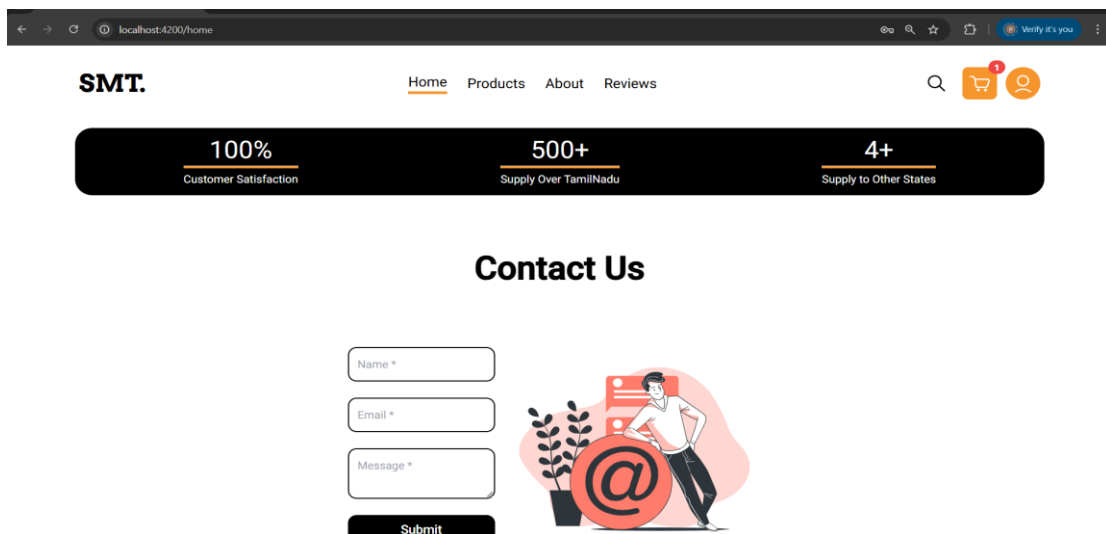
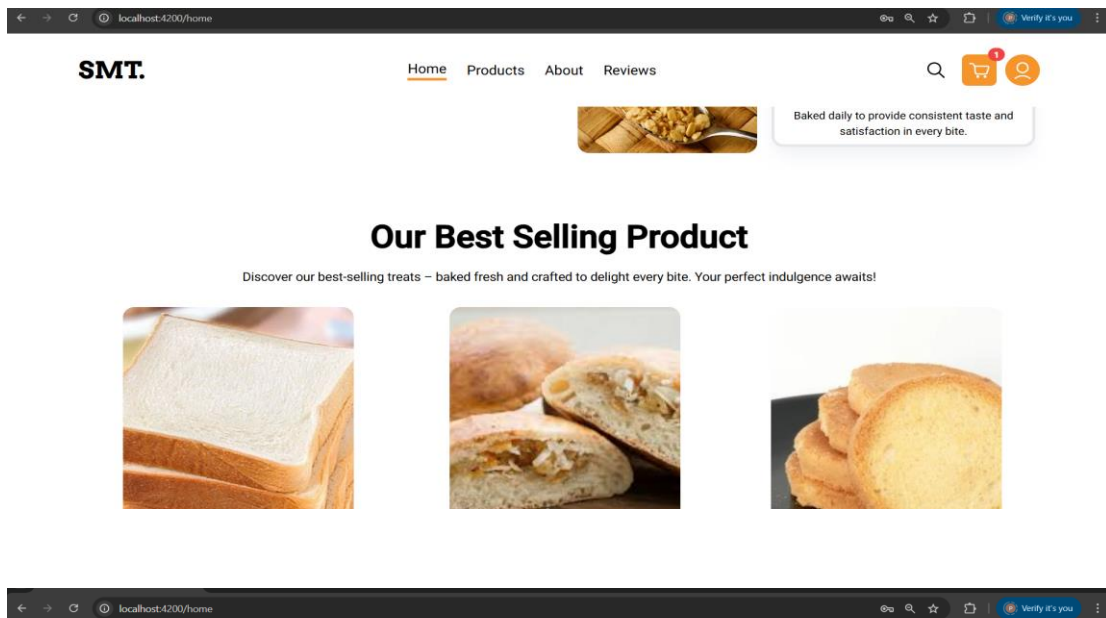
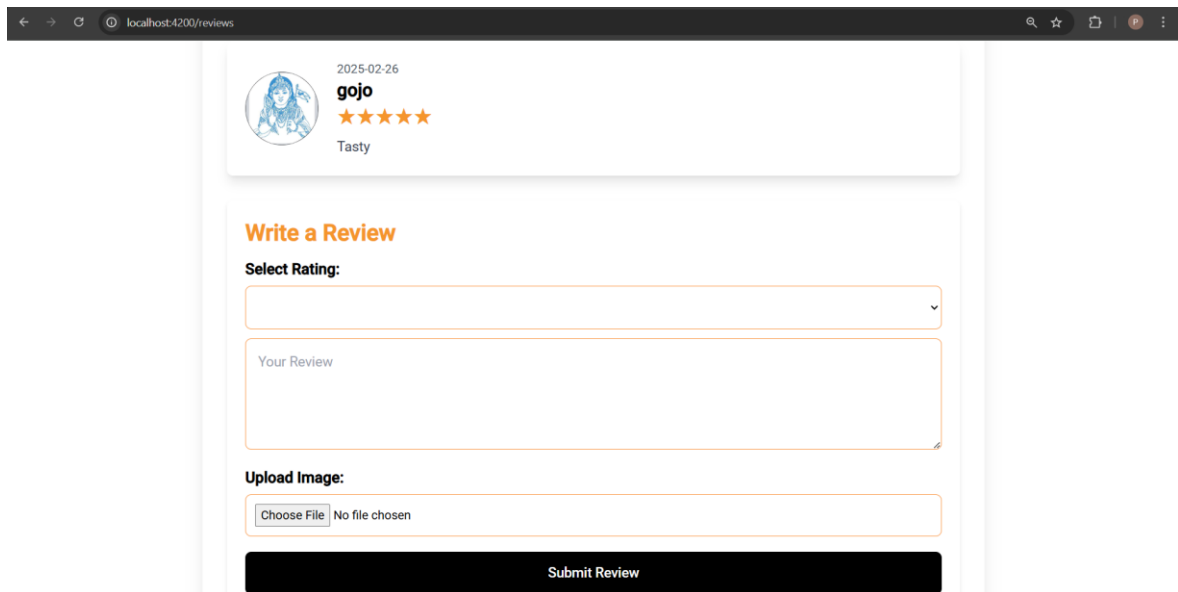
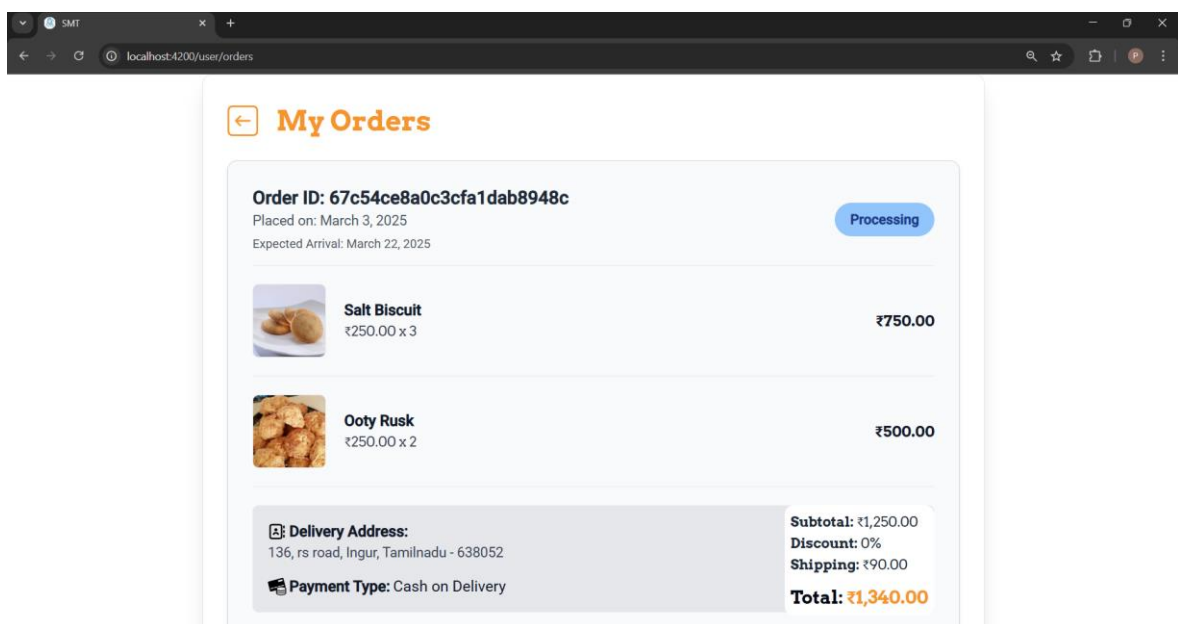


Figure. A 2.4 Home Page



The screenshot shows a web browser at localhost:4200/reviews. At the top, a user profile for 'gojo' is displayed with a date of 2025-02-26, a 5-star rating, and the word 'Tasty'. Below this is a 'Write a Review' section. It includes a 'Select Rating:' dropdown menu, a text area for 'Your Review', and an 'Upload Image:' section with a 'Choose File' button and the text 'No file chosen'. At the bottom is a black 'Submit Review' button.

Figure. A 2.5 Review Page



The screenshot shows a web browser at localhost:4200/user/orders. The page is titled 'My Orders' with a back arrow icon. It displays an order with ID '67c54ce8a0c3cfa1dab8948c'. The order was placed on March 3, 2025, and is currently in 'Processing' status. The expected arrival is March 22, 2025. The order contains two items: 'Salt Biscuit' (₹250.00 x 3) for ₹750.00 and 'Ooty Rusk' (₹250.00 x 2) for ₹500.00. The delivery address is 136, rs road, Ingur, Tamilnadu - 638052. The payment type is 'Cash on Delivery'. The subtotal is ₹1,250.00, the discount is 0%, the shipping is ₹90.00, and the total is ₹1,340.00.

Item	Price
Salt Biscuit (₹250.00 x 3)	₹750.00
Ooty Rusk (₹250.00 x 2)	₹500.00
Subtotal:	₹1,250.00
Discount:	0%
Shipping:	₹90.00
Total:	₹1,340.00

Figure. A 2.6 Order Page

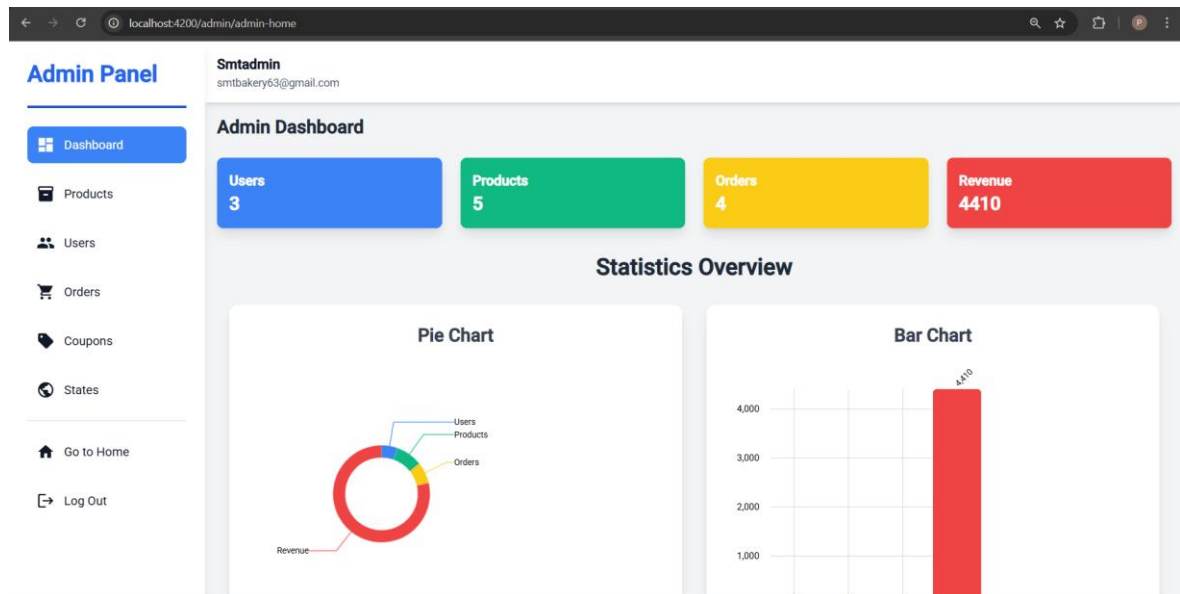


Figure. A 2.7 Admin Page

The screenshot shows the 'Manage Products' page. The sidebar is identical to the previous page, but the 'Products' menu item is highlighted. The main content area is titled 'Manage Products' and features an 'Add Product' form. The form has two columns of input fields: Product Name, Product Image (with a 'Choose File' button), Type (a dropdown menu), Description, Price (₹), Stock Quantity, Min Quantity, Max Quantity, Pieces, and Rating.

Field	Value
Product Name	Enter product name
Product Image	Choose File No file chosen
Type	Specify the product type
Description	Enter description
Price (₹)	Enter price
Stock Quantity	Enter stock quantity
Min Quantity	1
Max Quantity	5
Pieces	Enter number of pieces
Rating	0

Figure. A 2.8 Manage Product Page

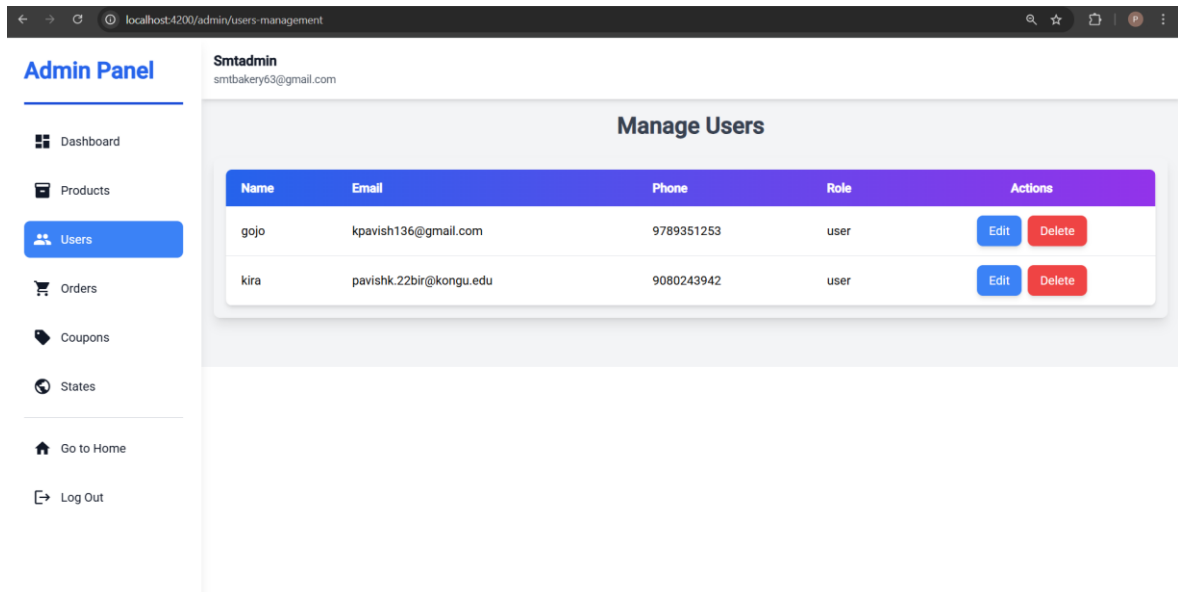


Figure. A 2.9 Manage Users Page

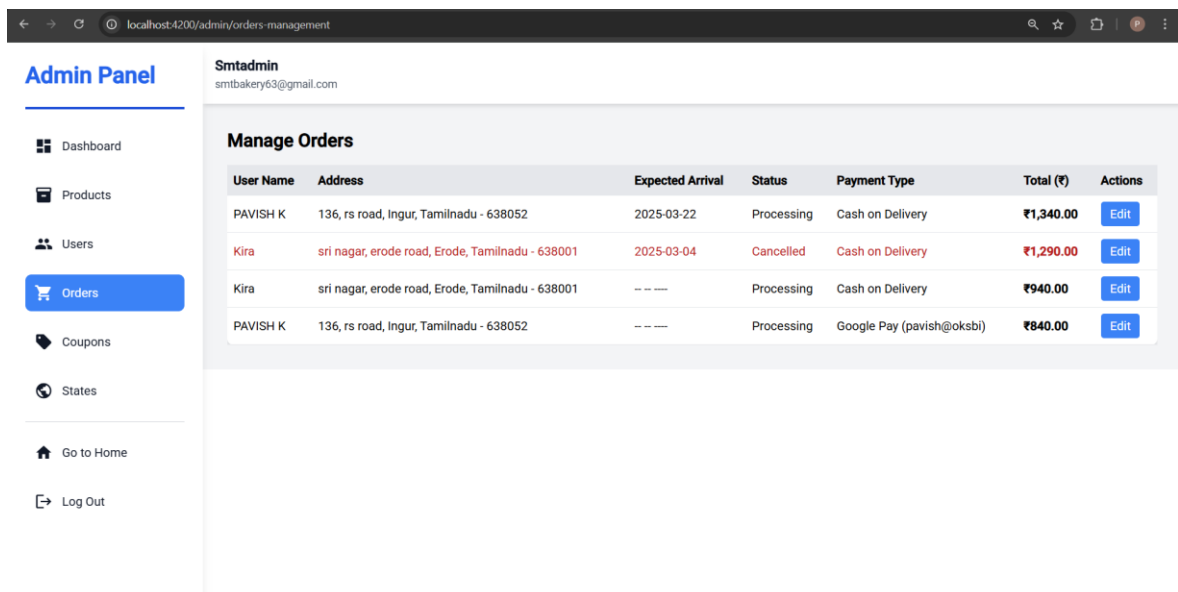


Figure. A 2.10 Manage Order Page

Admin Panel

Smtadmin
smtbakery63@gmail.com

Manage States

Add New State

District Name
Enter state name

State Name
Enter country/region

Charge
0

☒ Available

Add

States List

Name	State	Charge	Available	Actions
erode	tamilnadu	90	Yes	Edit Delete
chennai	tamilnadu	80	No	Edit Delete

Figure. A 2.11 Manage State Page

Admin Panel

Smtadmin
smtbakery63@gmail.com

Manage Coupons

Add New Coupon

Coupon Code
Enter coupon code

Expiration Date
mm/dd/yyyy

Discount
0

Max Usage
0

Add

Coupons List

Code	Expiration	Discount	Max Usage	Actions
				Edit Delete

Figure. A 2.12 Manage Coupon Page

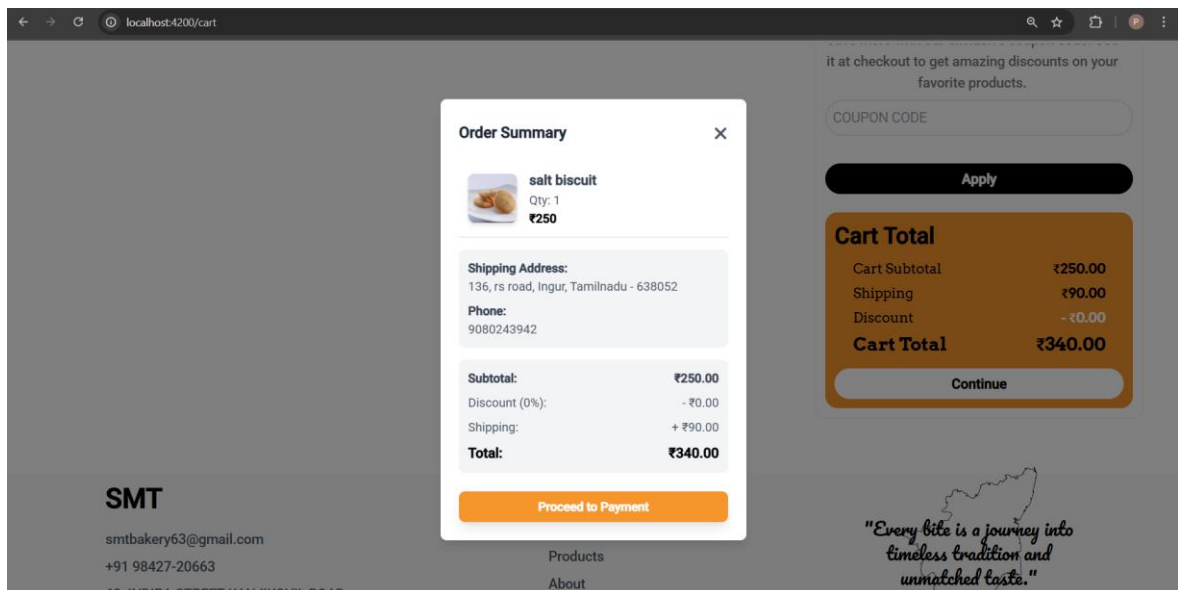
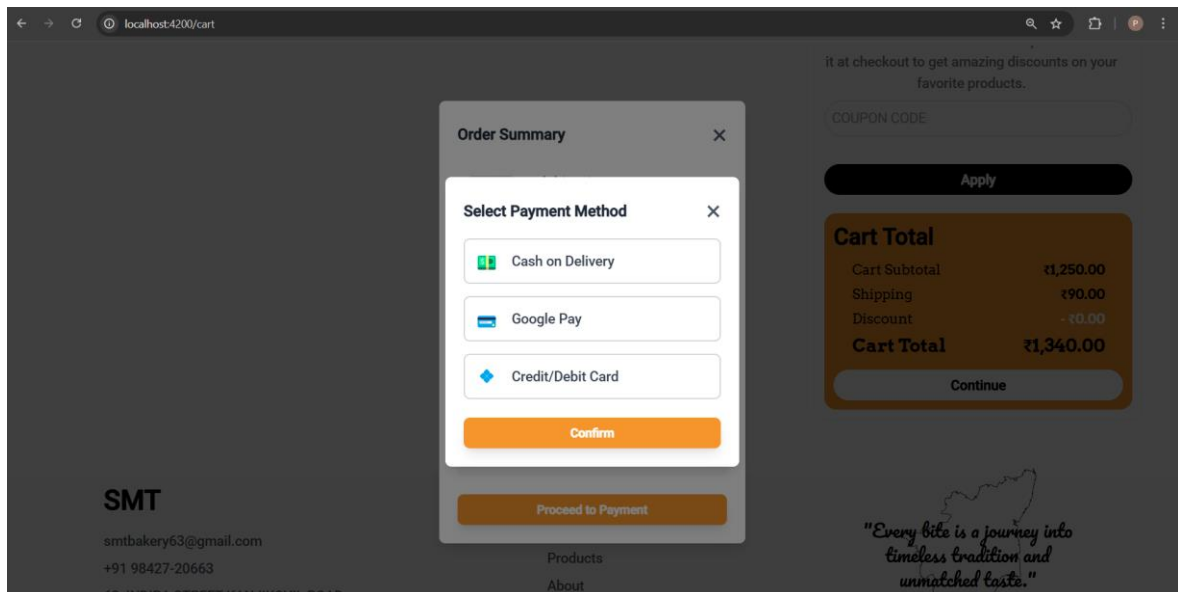


Figure. A 2.13 Payment Page

REFERENCES

- [1] **Shama Hoque**, "Learning MEAN Stack Development: Build web applications with MongoDB, Express.js, Angular, and Node.js," Packt Publishing, Second Edition, 2020.
- [2] **Simon Holmes & Clive Harber**, "Getting MEAN with Mongo, Express, Angular, and Node," Manning Publications, Second Edition, 2019.
- [3] **Amos Q. Haviv**, "MEAN Web Development," Packt Publishing, Second Edition, 2016.
- [4] **Vasan Subramanian**, "Pro MEAN Stack Development," Apress, First Edition, 2017.
- [5] **Valentin Bojinov**, "Angular for Enterprise-Ready Web Applications," Packt Publishing, Second Edition, 2020.

WEBSITES

- [1] MEAN Stack Official Documentation: <https://mean.io>
- [2] MongoDB Official Documentation: <https://www.mongodb.com/docs/manual/>
- [3] Express.js Guide: <https://expressjs.com/>
- [4] Angular Developer Guide: <https://angular.io/doc>
- [5] Node.js Official Website: <https://nodejs.org/>
- [6] Tutorials and Community Discussions: <https://stackoverflow.com/>