

“ Sweet Bliss ”
Custom Cake Ordering System

Final Project Report



Sri Lanka Institute of Information Technology
IT2080 Information Technology Project

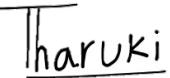
Group ITP_WD_B6_121

February 2025

Declaration

This project report is our original work and the content is not plagiarized from any other resource. References for all the content taken from external resources are correctly cited. To the best of our knowledge, this report does not contain any material published or written by third parties, except as acknowledged in the text.

Authors:

<u>Author SID</u>	<u>Author name</u>	<u>Signature</u>
IT23396272	Kariyawasam L L N	
IT23183018	Hirusha D G A D	
IT23222786	Budara V P R	
IT23222236	Eranthnage T N	
IT23179158	Budara H M K	

Date : 2025 – 05 – 15

Abstract

Sweet Bliss is web-based custom cake ordering system developed using MERN stack(MongoDB, Express.js, React.js and Node.js) to provide users with a seamless platform for personalizing and ordering cakes online. The platform allows users to design cakes according to their preferences, flavor, including color scheme, size, theme, toppings, custom texts and delivery date. It features user-friendly interfaces for customizing cakes, track delivery status and browse through the gift shop included.

An admin dashboard facilitates order management, inventory management, user management, review handling and the option to give discounts for customers. The system uses MongoDB for flexible data storage, Express.js and Node.js for more robust backend operations and React.js for more user-friendly and elegant interactive user interface experience. *Sweet Bliss* aims to streamline the custom cake ordering process, enhance the customer satisfaction and support the cake business reach a broader audience via internet.

Acknowledgement

We would like to extend our sincere gratitude to all those who supported and guided us throughout the development of our web-based solution, Sweet Bliss — a modern and customized cake ordering and management system. This project represents not just a technical implementation, but a meaningful collaboration that brought together creativity, innovation, and functionality to support a small business owner and university student striving to grow her custom cake-making enterprise.

First and foremost, we would like to express our heartfelt thanks to our supervisor, Mrs. Geethanjali, for their continuous encouragement, timely feedback, and valuable insights. Your guidance helped us stay focused and motivated, and your expertise contributed greatly to the successful design and implementation of the system.

We also wish to acknowledge our client, the founder of Sweet Bliss, for entrusting us with this real-world opportunity. Your detailed input, clear vision, and constant availability allowed us to truly understand the business needs, and tailor the platform accordingly. It was an honor to contribute to the digital transformation of your venture.

Our sincere thanks go to our university and the Department of Information Technology for providing the infrastructure, technical resources, and academic framework that enabled us to carry out this project successfully. The knowledge and skills acquired throughout our coursework proved invaluable in the practical implementation of this system.

To our team members, thank you for your dedication, commitment, and teamwork. Each of you brought unique strengths, and together we managed to overcome the challenges and achieve our project goals. Your enthusiasm, problem-solving mindset, and consistent communication played a pivotal role in delivering a high-quality product.

Finally, we are grateful to our families and friends for their unwavering support and encouragement throughout this journey. Their understanding and motivation helped us stay positive and focused during the demanding phases of this project.

Sweet Bliss is more than a cake-ordering website — it is a practical, innovative, and scalable solution that simplifies business processes, enhances customer experience, and opens doors for future growth. We are proud to have developed a platform that merges technology with passion and purpose.

TABLE OF CONTENTS

<u>DECLARATION</u>	2
<u>ABSTRACT</u>	3
<u>ACKNOWLEDGEMENT</u>	4
<u>CHAPTER 1</u>	8
<u>INTRODUCTION</u>	8
1.1. BACKGROUND	8
1.1.1. NATURE OF THE BUSINESS	8
1.1.2. NATURE OF THE CLIENT	8
1.2. PROBLEM AND MOTIVATION	9
1.2.1. PROBLEM	9
1.2.2. MOTIVATION	9
1.2.3. BENEFITS	10
1.3. LITERATURE REVIEW SUMMARY	10
1.4. AIMS AND OBJECTIVES	11
1.4.1. AIM	11
1.4.2. OBJECTIVES	11
1.5. SOLUTION OVERVIEW	12
1.5.1. CUSTOM ORDER MANAGEMENT	12
1.5.2. FINANCE MANAGEMENT	12
1.5.3. INVENTORY MANAGEMENT	12
1.5.4. FEEDBACK MANAGEMENT	13
1.5.5. USER MANAGEMENT	13
1.6. METHODOLOGY	14
1.6.1. AGILE METHODOLOGY	14
1.6.2. TOOLS AND TECHNOLOGIES	15
1.7. STRUCTURE OF THE REPORT	19
<u>CHAPTER 2</u>	20
<u>LITERATURE REVIEW AND REQUIREMENTS ANALYSIS</u>	20
2.1. LITERATURE REVIEW	20
2.1.1. STATE OF THE COMPETITIVE BACKGROUND	20
2.1.2. EXAMINE OF SIMILAR SOLUTIONS	21
2.1.3. MAIN LIMITATIONS IN EXISTING SOLUTIONS	23
2.1.4. PROPOSED SOLUTION FOR THE CAKE ORDERING PLATFORM	23
2.2. STAKEHOLDER ANALYSIS	24
2.3. REQUIREMENTS ANALYSIS	27

2.3.1. CUSTOM ORDERING MANAGEMENT	27
2.3.2. FINANCE MANAGEMENT	27
2.3.3. INVENTORY MANAGEMENT	28
2.3.4. FEEDBACK MANAGEMENT	29
2.3.5. USER MANAGEMENT	29
2.3.6. TECHNICAL REQUIREMENTS	30
2.4. REQUIREMENTS MODELLING	32
2.5. USE CASE SCENARIOS	33

CHAPTER 3

DESIGN AND DEVELOPMENT

3.1. SYSTEM ARCHITECTURE	36
3.2. DATABASE STRUCTURE	37
3.3. DATAFLOW DIAGRAM	41
3.4. ENTITY RELATIONSHIP DIAGRAM	42
3.5. NORMALIZED SCHEMA	43
3.6. FLOW CHART	44
3.7. WORKFLOW AND PLANS	45
3.8. KANBAN BOARD	45
3.9. PRODUCT BACKLOG	46
3.10. FIGMA UI DESIGNS	52
3.11. ACTUAL USER INTERFACES	59
3.11.1. COMMON USER INTERFACES	59
3.11.2. CUSTOMER MANAGEMENT USER INTERFACES	62
3.11.3. ORDER MANAGEMENT RELATED USER INTERFACES	65
3.11.4. INVENTORY MANAGEMENT RELATED USER INTERFACES	70
3.11.5. FINANCE MANAGEMENT RELATED USER INTERFACES	75
3.11.6. FEEDBACK MANAGEMENT RELATED USER INTERFACES	80

CHAPTER 4

TESTING

4.1. TEST CASE SCENARIOS	84
4.2. TEST CASE RESULTS	87

CHAPTER 5

EVALUATION

5.1. USER FEEDBACK	92
5.2. FUTURE IMPROVEMENTS	92
5.3. CONCLUSION	92

REFERENCES

Chapter 1

Introduction

1.1. Background

1.1.1. Nature of the business

Sweet Bliss is currently a small-scale custom cake business operated by Ms. Ranushi Warnakulasooriya. The business primarily takes orders through **WhatsApp and phone calls**, where customers communicate their cake preferences, share design inspirations, and finalize order details manually. Payments are handled through **bank transfers or cash on delivery**, with no integrated online payment system. Order tracking is managed manually, requiring constant back-and-forth communication between the business owner and customers. Since there is no dedicated platform, managing customer requests, tracking orders, and maintaining records is time-consuming and prone to errors. Additionally, marketing efforts rely on **social media promotions and word-of-mouth referrals**, with no structured system for customer retention or loyalty rewards.

1.1.2. Nature of the client

Our client is a university student who runs a custom cake business as a side hustle, managing orders manually through WhatsApp and calls. With a busy academic schedule, she faces challenges in tracking orders and needs a streamlined web solution to handle orders, inventory, and customer management efficiently.

1.2. Problem and Motivation

1.2.1. Problem

- The lack of a centralized system for managing custom cake orders, inventory, payments, and customer communication leads to inefficiencies, order mismanagement, and lost business opportunities, making it challenging for our client to scale and balance the business with academic commitments.

Currently, our client, a university student running a custom cake business, is facing significant challenges in managing orders efficiently. The entire order management process is handled manually via WhatsApp and phone calls, making it difficult to track and organize customer requests. This leads to delays, miscommunication, and potential order mix-ups, affecting customer satisfaction.

Since there is no centralized platform to manage orders, inventory tracking is also a challenge. The client manually keeps records of available ingredients and supplies, leading to last-minute shortages and rushed restocking. As the business grows, maintaining stock levels accurately becomes increasingly difficult.

Additionally, due to the client's busy academic schedule, there are times when orders cannot be fulfilled, but there is no system to inform customers in advance. This results in customer frustration and potential business loss. A lack of structured scheduling means the business struggles to balance production capacity with demand, leading to overbooking or missed opportunities.

Furthermore, the absence of automated payment processing forces the client to handle transactions manually, increasing the risk of payment errors, delays, and fraud. Customers also lack transparency regarding order statuses, as there is no system for real-time notifications.

Lastly, feedback collection is unstructured, making it difficult to evaluate customer satisfaction and improve the business based on user input. With no integrated promotion or loyalty system, there is no efficient way to engage repeat customers or attract new ones.

To overcome these challenges, we are developing a centralized web platform that automates and streamlines order management, inventory tracking, payments, customer communication, and business insights, ensuring a smooth and scalable cake ordering process.

1.2.2. Motivation

Imagine a world where ordering a custom cake is as seamless and enjoyable as eating one. Our platform leverages cutting-edge technology to automate the entire cake creation journey. By simplifying complex processes, we empower both our bakers and customers, allowing them to focus on the artistry and celebration. From intuitive design tools to automated order tracking, we're making custom cake ordering a delightful experience, while also ensuring cost-effectiveness.

1.2.3. Benefits

- Automating the custom cake ordering process increases efficiency and reduces manual workload.
- Implementing validation constraints enhances data accuracy and minimizes errors in orders and inventory.
- Provides effortless access to real-time order status, inventory levels, and customer interactions.
- Enables accurate business decisions at the right time through data-driven insights and reports.
- AI-powered cake design recommendations simplify the customization process and improve customer experience.
- Order scheduling and vacation mode allow better time management for the business owner.
- Secure payment and discount management streamline transactions while enhancing customer satisfaction.
- Strengthens customer trust by ensuring transparent order tracking and timely updates.
- The admin can easily track and manage customer feedback to improve services and product offerings.
- With a centralized system for managing financial transactions and promotions, the owner gains clear insights into business performance, leading to better decision-making and growth opportunities.

1.3. Literature Review Summary

Current solutions focus on general **e-commerce, food ordering, and custom product management**, but they lack a **specialized, integrated system** tailored to cake ordering.

- No Vacation Mode for Bakery Owners – Most systems do not allow creators to mark themselves as unavailable, causing mismanaged orders.
- Inflexible Order Approval Mechanisms – Platforms automatically process all orders without allowing the admin (baker) to accept or reject requests based on availability.

A detailed version of the review is included in **Chapter 2**.

1.4. Aims and Objectives

1.4.1. Aim

Our project aims to develop an efficient and user-friendly online custom cake ordering platform that streamlines order management, automates workflows, and enhances customer experience. By integrating AI-powered design recommendations, real-time order tracking, secure payment processing, and inventory management, the system will reduce manual workload, improve accuracy, and enable seamless business operations for the client, ensuring better time management and business growth.

1.4.2. Objectives

- Ensure an intuitive and accessible interface for customers and the admin.
- Enable easy creation, updating, and management of orders, products, and users.
- Assist customers in selecting designs based on trends and preferences.
- Ensure transparency with order statuses, ingredient usage, and stock updates.
- Send alerts for order updates, payments, promotions, and generate business insights.

1.5. Solution Overview

Sweet Bliss is a modern custom cake ordering platform designed to simplify the process of creating and managing personalized cake orders. It allows customers to customize cakes based on their preferences, upload design inspirations, or generate AI-powered cake recommendations. The platform ensures efficient order tracking, secure payments, and real-time notifications, making it easy for both customers and the business owner to manage orders seamlessly. With features like vacation mode, loyalty rewards, and customer cake showcases, Sweet Bliss enhances user experience while supporting the business's growth.

The system is divided into five different crucial sub-systems in order to be developed efficiently within the time frame that we were given.

- **Custom Order Management**
- **Finance Management**
- **Inventory Management**
- **Feedback Management**
- **User Management**

1.5.1. Custom Order Management

- Allows customers to personalize their cake orders by selecting flavors, sizes, and designs.
- Supports image uploads for custom designs and integrates AI-powered cake design recommendations.
- Includes an admin-controlled acceptance system and a vacation mode to manage order availability.

1.5.2. Finance Management

- Facilitates secure online transactions for customers through multiple payment gateways.
- Automates invoice generation and tracks financial records for business insights.
- Includes refund management for declined orders and logs payment histories for transparency.

1.5.3. Inventory Management

- Manages product listings, including cakes, cupcakes, and other bakery items.
- Tracks inventory levels, updates stock in real time, and prevents over-ordering of ingredients.
- Helps the admin organize promotions and discounts on selected items.

1.5.4. Feedback Management

- Enables customers to leave ratings, reviews, and feedback on their orders.
- Analyzes feedback trends to enhance product quality and customer satisfaction.
- Provides admins with sentiment analysis to make data-driven improvements.

1.5.5. User Management

- Handles customer registrations, user profiles, and role-based access for admins and customers.
- Supports password recovery, OTP verification, and account preferences for a personalized experience.
- Maintains user activity logs to improve security and customer support.

1.6. Methodology

1.6.1. Agile Methodology

The development of a dynamic and responsive online cake ordering system necessitates a methodology that prioritizes flexibility, adaptability, and continuous improvement. Agile methodologies, particularly Scrum, are ideally suited to meet these demands. Given the evolving nature of user preferences, technological advancements, and the inherent complexities of custom product creation, an iterative and collaborative approach is crucial.

How Agile Methodology Suits the Online Cake Ordering System:

- **Adaptability to Changing Requirements:**
 - The cake ordering platform will likely encounter fluctuating customer demands and evolving market trends. Agile's iterative nature allows for adjustments to features and functionalities throughout the development process, ensuring the platform remains relevant and competitive.
- **Rapid Prototyping and Feedback Loops:**
 - Agile's emphasis on short sprints enables the rapid development of prototypes and Minimum Viable Products (MVPs). This allows for early user feedback, which can be incorporated into subsequent iterations, ensuring the platform aligns with user needs.
- **Enhanced Collaboration and Communication:**
 - Agile promotes close collaboration between developers, designers, and stakeholders through daily stand-up meetings, sprint reviews, and retrospectives. This ensures everyone is aligned on project goals and facilitates effective communication, minimizing misunderstandings and delays.
- **Continuous Integration and Delivery:**
 - Agile practices like continuous integration and continuous delivery (CI/CD) allow for frequent releases of new features and updates. This ensures the platform is constantly evolving and improving, providing users with a consistently enhanced experience.
- **Prioritized Feature Development:**
 - Agile's prioritization of user stories and backlog items allow the development team to focus on the most critical features first. This ensures that the platform delivers core functionalities early on, providing immediate value to users.
- **Risk Mitigation:**
 - By working in short sprints, and constantly reviewing the work done, any potential issues are found and fixed quickly. This lowers the risk of large, costly mistakes.
- **Customer Centric Development:**
 - Agile methodology puts the customer first. By having constant reviews, and feedback loops, the final product is much more likely to meet the customer's needs.
- **Flexibility for Integration:**

- Because of the nature of the project, many API integrations will be needed. Agile methodology allows for the flexibility to integrate these APIs as needed.

Justification :

The selection of an Agile methodology for developing our online cake ordering platform is driven by the inherent need for flexibility and responsiveness. The dynamic nature of e-commerce, coupled with the unique demands of custom product creation, necessitates an iterative approach. Agile's emphasis on short sprints, continuous feedback, and adaptability ensures that we can rapidly respond to evolving user needs and market trends. This methodology fosters a collaborative environment, enabling us to deliver a high-quality, customer-centric platform that can readily adapt to the complexities of the digital landscape, ultimately maximizing user satisfaction and project success.

1.6.2. Tools and Technologies

Technologies

To develop the **Sweet Bliss Cake Ordering System**, we have chosen the **MERN (MongoDB, Express.js, React, Node.js) stack**, which provides a **modern, efficient, and scalable** solution for web application development.

What is MERN Stack?

MERN is a **JavaScript-based** technology stack that consists of the following components:

- **MongoDB** – A NoSQL database that stores data in a flexible, JSON-like format, making it ideal for dynamic applications.
- **Express.js** – A lightweight and powerful Node.js framework that simplifies backend development and API creation.
- **React.js** – A front-end JavaScript library used to build dynamic and interactive user interfaces.
- **Node.js** – A runtime environment that enables JavaScript to be executed on the server, allowing for a full-stack JavaScript application.



Figure 6 – MERN Stack

Alternative Solutions Considered

While the **MERN stack** is our primary choice, we explored several other technology stacks:

- **Angular + Node.js (MEAN Stack)**
 - Pros: Angular provides a structured and maintainable frontend framework with built-in features like dependency injection.
 - Cons: Angular has a **steeper learning curve** than React, leading to longer development time for dynamic UI updates.
- **Java Spring Boot + Angular**
 - Pros: Spring Boot is highly scalable and widely used in **enterprise-level applications** with strong security features.
 - Cons: **Development speed is slower** as Java requires more boilerplate code compared to JavaScript frameworks like Node.js.
- **PHP Laravel + MySQL**
 - Pros: Laravel is a **powerful backend framework** with built-in authentication and MVC architecture, making it reliable for web applications.
 - Cons: PHP is not optimized for real-time applications, and Laravel requires **additional configurations** for full-stack JavaScript support.

Justification for Choosing MERN Stack

We have chosen the **MERN stack** due to the following advantages:

- **Full-Stack JavaScript** – MERN allows developers to work with a single language (**JavaScript**) across the entire application, simplifying development and reducing the need for multiple skill sets.
- **Scalability & Performance** – Node.js and MongoDB provide high-speed, scalable solutions, making the system capable of handling growing user demands.
- **Flexibility & Efficiency** – React.js offers fast rendering, reusable components, and a **rich ecosystem**, making it ideal for dynamic user interfaces.
- **RESTful APIs & Microservices** – Express.js allows for easy API development, enabling **seamless integration** with third-party services such as **payment gateways** and **AI-powered recommendations**.

- **Cloud Compatibility** – MongoDB works well with cloud-based solutions like **AWS, Firebase, and DigitalOcean**, ensuring **easy deployment and data accessibility**.
- **Rapid Development & Maintenance** – The MERN stack is widely used, has **strong community support**, and enables **faster prototyping**, making it an ideal choice for startups and small businesses like **Sweet Bliss**.

By leveraging the MERN stack, we can build a robust, feature-rich, and scalable web platform for Sweet Bliss, ensuring smooth order management, enhanced customer experience, and future expansion opportunities.

Tools

Our development process for the custom cake ordering platform is built on a foundation of industry-standard tools and practices, ensuring efficiency, collaboration, and high-quality code.

We leverage **Microsoft Visual Studio Code (VS Code)** as our primary code editor. VS Code's versatility, extensive extensions, and powerful debugging capabilities provide our developers with an optimal environment for writing and maintaining clean, efficient code. Its seamless integration with version control systems further streamlines our workflow.

For version control and collaborative development, we utilize **GitHub**. This platform allows us to manage our codebase effectively, track changes, and facilitate seamless method integration. Through GitHub's branching and pull request system, we ensure that code changes are thoroughly reviewed and tested before being merged into the main codebase, minimizing errors and promoting code quality. This method integration is also very useful when collaborating on the APIs.

To manage our project workflow and track progress, we rely on **JetBrains YouTrack**. This project management tool provides a centralized hub for task management, bug tracking, and issue resolution. YouTrack's customizable workflows and agile boards enable us to effectively prioritize tasks, track progress, and maintain transparency throughout the development lifecycle. This helps keep the project on track, and allows for effective communication between all team members.

By combining the strengths of VS Code, GitHub, and YouTrack, we have created a robust and efficient development environment that enables us to deliver a high-quality custom cake ordering platform that meets the needs of our customers and bakers.

Justification :

- **Visual Studio Code (VS Code):**
 - Lightweight and highly customizable code editor.
 - Extensive extension ecosystem for enhanced productivity.
 - Robust debugging capabilities.
- **GitHub:**
 - Centralized version control and collaboration platform.
 - Seamless code sharing, review, and integration.
 - Branching and pull request system for code quality assurance.
- **Postman:**
 - Comprehensive API testing, debugging, and documentation platform.
 - Ensures robust communication between front-end and back-end services.

- Facilitates seamless integration with external APIs.
- **JetBrains YouTrack:**
 - Comprehensive project management capabilities.
 - Centralized task tracking, issue resolution, and agile workflow management.
 - Ensures project organization, transparency, and on-schedule delivery.

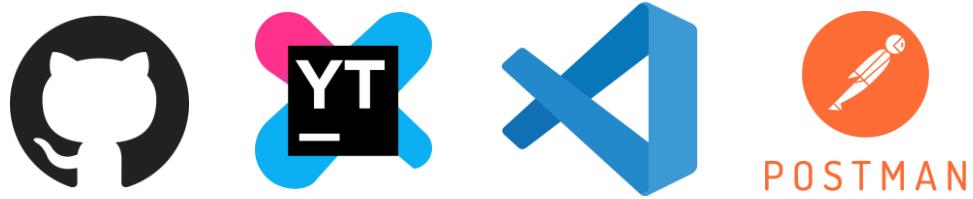


Figure 1 - Github , You Track , VS Code , Postman

1.7. Structure of the report

The report is structured into several chapters as included in below to give detailed analysis of relevant topics of *Sweet Bliss*.

Chapter 2 - Literature review and requirements analysis. Detailed version of the research done around functional and non-functional requirements of the project and the background research of the industry.

Chapter 3 – Design and development. Includes the system models, architecture and the implementation.

Chapter 4 – Testing. Includes the testing strategies, test cases and the relevant results.

Chapter 5 – Evaluation and conclusion. Analysis of final results, challenges resolved and future improvements.

Chapter 2

Literature Review and Requirements analysis

2.1. Literature Review

2.1.1. State of the competitive background

The confectionary art of cake creation and distribution, a practice with deep historical roots extending back to the medieval period, has undergone continuous refinement throughout subsequent ages. This evolution, however, has not been mirrored by a parallel advancement in digital platforms catering to the modern consumer's demand for bespoke cake experiences. Even within the hyper-connected and technologically driven environment of the 2020s, a notable absence persists: a comprehensive, user-centric web platform that effectively facilitates the creation and procurement of fully customized cakes.

Although numerous online vendors offer pre-designed cakes for various celebratory occasions, these platforms consistently exhibit critical shortcomings. A primary deficiency lies in the limited scope of customization options, often restricting consumers to pre-defined templates and neglecting the nuanced personalization demanded by special events. This lack of flexibility is compounded by logistical challenges, including unreliable delivery schedules, inadequate temperature control during transport, and a general lack of transparency in the delivery process.

Furthermore, issues pertaining to platform reliability and service quality are frequently reported. Concerns range from inaccurate order fulfillment and inconsistent product quality to unresponsive customer support and a lack of secure payment processing. These deficiencies collectively contribute to a fragmented and often frustrating online cake ordering experience, fostering a climate of distrust and hindering the realization of a seamless, personalized transaction.

The cumulative impact of these limitations reveals a significant disparity between the potential of digital platforms and the actual experience offered to consumers seeking custom cakes. This disparity underscores the urgent need for a novel platform that not only addresses the logistical and technical challenges inherent in online cake ordering but also prioritizes the customer's desire for creative control, reliable service, and a truly personalized celebratory experience. Such a platform would need to incorporate advanced customization tools, robust delivery management systems, stringent quality control measures, and a commitment to transparent and responsive customer service. The absence of such a comprehensive solution represents a significant gap in the current digital marketplace and highlights a compelling opportunity for innovation.

2.1.2. Examine of Similar Solutions

Instance 01 : A&M Cupcakes Colpetty, CakeFactory, Divine LK

"A&M Cupcakes, CakeFactory and Divine LK provides a seamless online ordering experience for custom cakes and baked goods, integrating payment and delivery options for better customer engagement."

Pros:

- Provides order tracking and estimated delivery times.
- Supports digital payments through multiple payment gateways.

Cons:

- Lacks AI-based cake design generation for personalized recommendations.
- Does not allow user to place online cake orders with customization options.
- No vacation mode to handle temporary unavailability of the baker.
- Limited admin flexibility in accepting or declining orders based on workload.

E-Commerce Platforms for Food & Beverages

Instance 02 : Shopify (Food & Beverage Storefronts) and WooCommerce

"WooCommerce allows small bakery businesses to set up an online storefront with an integrated ordering system and inventory management features."

Pros:

- Provides a structured product listing and shopping cart system.
- Supports various payment gateways and coupon integrations.
- Offers inventory management for tracking available products.

Cons:

- Does not focus on custom cake ordering workflows, requiring third-party plugins.
- Lacks built-in admin-controlled availability features like vacation mode.
- No AI-powered cake design assistance for users without a pre-made design.

Custom Order Management Systems

Instance 03 : CustomMade & Zazzle

"CustomMade specializes in allowing users to place highly personalized orders for unique products, connecting them with artisans."

Pros:

- Supports highly detailed custom order requests.
- Provides a messaging system for direct communication with the creator.
- Order tracking with production status updates.

Cons:

- Lacks industry-specific features such as AI-assisted cake design suggestions.
- Does not allow real-time availability control for the cake creator.
- No built-in feature to provide refunds for declined orders instantly.

Online Food Delivery Platforms

Instance 04: Uber Eats, DoorDash

"Uber Eats and DoorDash allow bakeries to list products and accept online orders for local delivery, connecting customers with nearby options."

Pros:

- Provides a platform to increase customer reach for small bakeries.
- Handles delivery logistics with third-party services.
- Offers real-time order tracking.

Cons:

- Charges high commission fees, reducing profit margins for small bakery owners.
- Does not provide an integrated cake customization workflow.
- No option for direct admin approval or decline of custom cake orders.

Feedback Management Systems

For instance: Trustpilot & Google Reviews

"Trustpilot provides a review aggregation platform that allows customers to leave feedback on businesses, helping improve credibility and trust."

Pros:

- Collects customer reviews and ratings.
- Help build trust and credibility for the business.
- Allow public and private responses from business owners.

Cons:

- No integration with the cake ordering system to directly collect feedback after order completion.
- Lacks structured feedback options for custom cake orders, such as taste, design accuracy, and delivery experience.
- Does not provide **sentiment analysis** to automatically categorize reviews (positive, neutral, negative).

2.1.3. Main Limitations in Existing Solutions

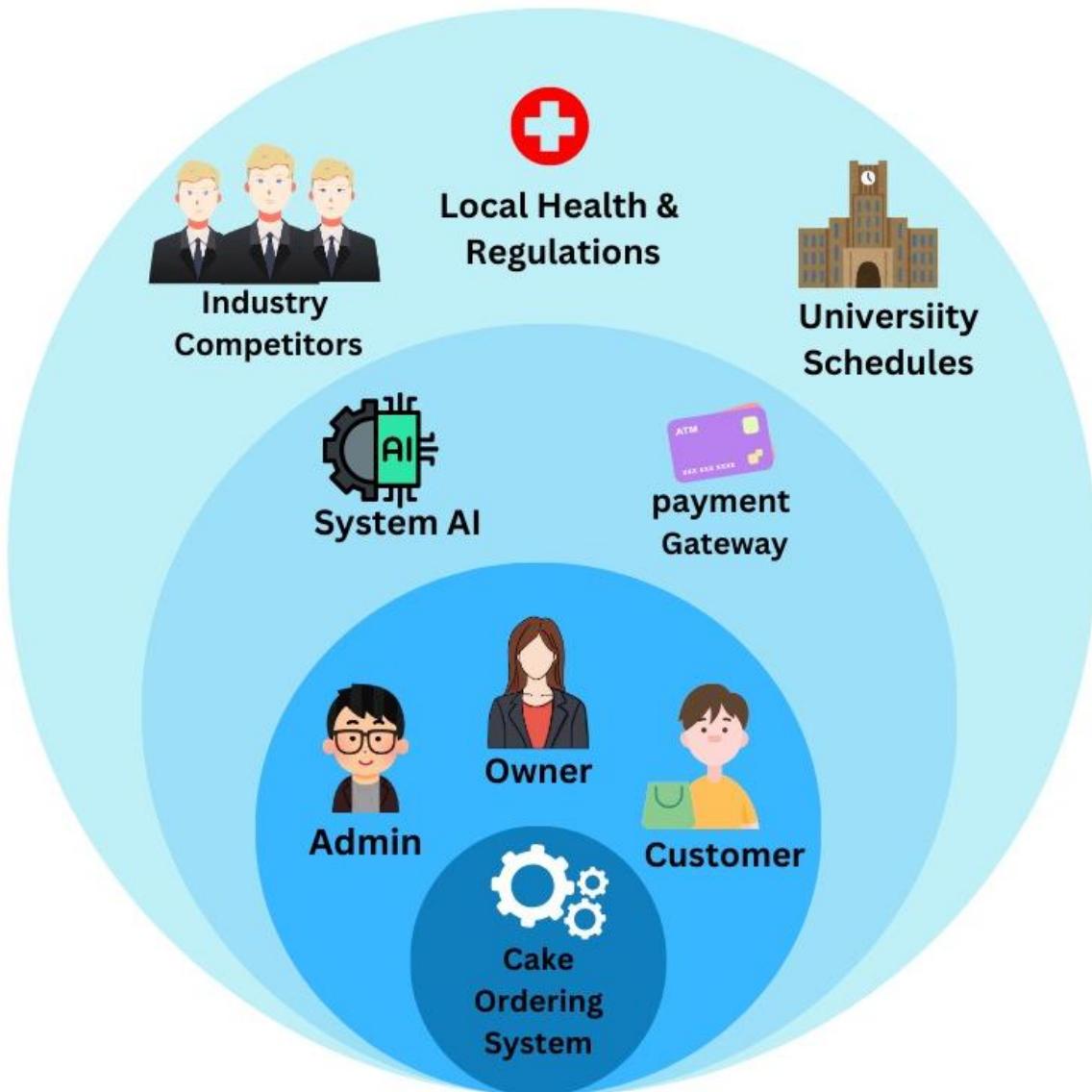
- **Lack of AI-Powered Design Assistance** – No existing solution suggests cake designs based on customer preferences dynamically.
- **No Vacation Mode for Bakery Owners** – Most systems do not allow creators to mark themselves as unavailable, causing mismanaged orders.
- **Inflexible Order Approval Mechanisms** – Platforms automatically process all orders without allowing the admin (baker) to accept or reject requests based on availability.
- **Limited Support for Refund Automation** – If an order is declined, manual processing is required instead of an immediate refund system.
- **Inadequate Support for Small Bakeries** – Many existing systems are built for large-scale operations, making them inefficient for independent bakers managing fluctuating workloads.

2.1.4. Proposed Solution for the Cake Ordering Platform

A **custom-built platform** tailored specifically for cake ordering is needed to address these gaps. This system would:

- Seamlessly handle custom cake orders with AI-powered design suggestions.
- Allow bakers to activate vacation mode to prevent unnecessary order processing.
- Provide a structured order approval process, enabling admin control over workload.
- Automate refunds for declined orders to enhance customer satisfaction.
- Improve efficiency and profitability for small, independent bakers with specialized tools.
- This system would bridge the limitations of existing platforms by offering a **fully integrated, flexible, and bakery-centric** online ordering experience.

2.2. Stakeholder Analysis



Core Layer: Cake ordering system

Role – Central software platform

Responsibilities:

- Enable browsing and ordering cakes.
- Handle order management, inventory and customer interactions.
- Store and manage customer data securely.
- Provide reporting dashboards for business insights.

Second Layer: Primary Users

Owner:

Role – Business operator and strategic decision-maker.

Responsibilities:

- Monitor overall business performance.
- Set product pricing and manage cake listings.
- Approve offers, promotions, and seasonal deals.
- Analyze sales reports and customer feedback for business improvement.
- Ensure legal and health regulation compliance.

Admin:

Role – System manager and support personnel.

Responsibilities:

- Maintain and update the backend system (e.g.: product listings, pricing etc.).
- Resolve technical issues and bugs.
- Moderate customer reviews or content if applicable.
- Support customer inquiries and escalate critical issues to the owner.
- Ensure data accuracy and system uptime.

Customer:

Role – End-user and buyer.

Responsibilities:

- Browse cake catalog and place orders.
- Customizing cake options (e.g.: flavor, message, size).
- Make payments through integrated gateways.
- Track orders and request support if needed.
- Leave feedback or ratings for services improvements.

Third Layer: Supporting Tech and services

Payment Gateway:

Role – Financial transaction handler.

Responsibilities:

- Process secure online payments.
- Manage refunds, cancellations, and chargebacks.
- Ensure PCI-DSS compliance.
- Notify system of successful/failed transactions in real-time.

AI system:

Role – Intelligence engine for automated cake design generation.

Responsibilities:

- Analyze customer purchase patterns to recommend more suited designs.
- Generate designs as relevant as possible according to customer request.

Fourth Layer: External Influences

Local Health and Regulations:

Role – Regulatory authority.

Responsibilities:

- Define and enforce hygiene and food safety standards.
- Conduct inspections and issue certifications/licenses.
- Penalize or warn for non-compliance.
- Regulate packaging, labeling and food handling practices.

Industry Competitors:

Role – Market influencers and benchmarks.

Responsibilities:

- Indirectly influence pricing, innovation, and feature sets.
- Drive the need for differentiation and better customer service.
- Affect promotional strategies and seasonal offerings.

University Schedules:

Role – External event influencer.

Responsibilities:

- Influence peak demand times.
- Help with planning targeted promotions or special offers for students.
- Align delivery schedules or staffing during known low or high activity periods.
- Enable batch pre-orders for campus clubs, parties or academic events.

2.3. Requirements Analysis

In the following section includes the functional and non-functional requirements component wise.

2.3.1. Custom ordering management

Functional Requirements

- Order Placement & Customization
- Order Scheduling
- Creator Availability (Vacation Mode)
- Image Recommendation
- Admin Order Management
- Customer Order History

Non-Functional Requirements

- Performance
- Scalability
- Availability
- Reliability

2.3.2. Finance management

Functional Requirements

- ❖ Transaction history
 - Customers and admins should be able to view a detailed history of payments.
 - Invoice & Receipt Generation
 - Generate invoices for successful transactions and provide downloadable receipts.
 - Refund Processing
- ❖ Promo Code system
 - Customers should be able to enter promo codes for discounts during checkouts.
- ❖ Sales and Revenue reports

Non-Functional Requirements

- ❖ Performance Requirements
 - Payment processing should take less than 5 seconds
 - Reports should be generated within 30 seconds, even with large accounting tools.
- ❖ Security
 - Transactions should use end-to-end encryption (SSL/TLS)
 - Protection against cyber threats.
 - Data encryption.

2.3.3. Inventory management

Functional Requirements

- ❖ Product Management
 - Add, edit and delete products.
 - Set product categories.
 - Manage stock levels.
 - Receive low-stock alerts.
 - Track inventory supplies.
- ❖ Set automated restocking reminders.
- ❖ Shopping cart
 - Add and remove items from the cart.
 - Adjust item quantity before checkout.
 - Apply discount codes.

Non-Functional Requirements

- ❖ Performance requirements
 - Fast page load times.
 - High availability.
- ❖ Usability & UX requirements.
 - Responsive design.
 - Error handling and feedback.
- ❖ Security

- Role-based access control.
- Protection against cyber threats.
- Data encryption.

2.3.4. Feedback management

Functional Requirements

- ❖ Feedback Submission
 - Customers can submit feedback after an order is completed.
 - Allow customers to rate orders (1-5 stars).
 - Admin can approve, reject, or delete feedback.
- ❖ The ability to hide certain feedback from public view.
- ❖ Flagging & Reporting Inappropriate Feedback.

Non-Functional Requirements

- ❖ Maintainability and Extensibility.
- ❖ Data Integrity and Accuracy.
- ❖ Availability and Reliability.

2.3.5. User management

Functional Requirements

- ❖ User Registration and Authentication.
 - Users can log in using their credentials.
 - Users must verify their account via email or OTP.
 - Users can reset passwords via email if forgotten.
 - Role-Based Access Control.
 - Define user roles: Admin, Customer.
 - Profile & Account Management.
- ❖ Admins can manage all accounts, including suspending/deleting users.
- ❖ Report Generation.
- ❖ Role Based Access Control for the admins.
- ❖ Profile Management.
- ❖ Personalized pre-scheduled notification system.
- ❖ Badge & Achievements system.

Non-Functional Requirements

- ❖ Security
 - User data should be encrypted, and the system should be protected against common vulnerabilities like SQL injections and XSS.
- ❖ Performance
 - Login, registration, and user management actions should be processed within 2 seconds.
- ❖ Scalability
 - The system should handle a large number of users without performance degradation.

2.3.6. Technical Requirements

At the heart of our custom cake ordering platform lies **MongoDB**, a powerful and flexible **NoSQL database** that acts as our central data repository. We've chosen MongoDB to ensure a truly seamless and efficient experience for both our bakers and customers. Imagine a system where every detail, from intricate cake designs and customer preferences to real-time inventory levels and financial transactions, is instantly accessible and consistently updated.

MongoDB's strength lies in its ability to handle diverse data structures without the rigid schema of traditional relational databases. This flexibility is crucial for our platform, which manages a wide array of information, including user profiles, complex custom order specifications, dynamic ingredient inventories, detailed financial records, and valuable customer feedback. By consolidating all application data into a single, unified MongoDB database, we achieve several key advantages:

- ❖ **Real-Time Data Access:** Bakers can instantly view and update order details, inventory levels, and customer preferences, ensuring smooth and efficient workflows. Customers can track their orders in real-time and make changes as needed.
- ❖ **Scalability and High Availability:** As our platform grows and handles increasing volumes of data and user traffic, MongoDB's inherent scalability ensures that performance remains consistently high. Its high availability architecture minimizes downtime, guaranteeing uninterrupted access to our services.
- ❖ **Seamless Integration:** Our various modules—User Management, Custom Order Management, Inventory Management, Finance Management, and Feedback Management—interact effortlessly with the centralized MongoDB database. This eliminates data silos and ensures that all modules are working with the most up-to-date information.
- ❖ **Data Consistency and Reduced Redundancy:** By storing all data in a single location, we minimize the risk of data inconsistencies and eliminate unnecessary data duplication. This leads to more accurate reporting, streamlined operations, and a more reliable platform.
- ❖ **Enhanced Flexibility for Customization:** Due to the nature of custom cake orders, the data associated with each order can be very different. MongoDB is perfect for this, as it allows us to store the data in a very flexible way.

When building web applications with Node.js and Express.js, MongoDB is frequently used as the database. Here's a breakdown of how these technologies work together to handle data:

1. Node.js and Express.js as the Backend:

❖ Node.js:

- Node.js provides the runtime environment for executing JavaScript on the server-side.
- It's known for its event-driven, non-blocking architecture, which makes it highly efficient for handling concurrent requests.

❖ Express.js:

- Express.js is a web application framework built on top of Node.js.
- It simplifies the process of building web applications and APIs by providing features like routing, middleware, and¹
- request/response handling.
- Express.js handles the incoming HTTP requests from clients (e.g., web browsers or mobile apps).

2. MongoDB as the Database:

❖ MongoDB:

- MongoDB is a NoSQL document database that stores data in flexible, JSON-like documents.
- Its flexible schema allows for easy adaptation to changing data requirements.
- It's designed for scalability and high performance.

3. How They Interact:

❖ Connecting to MongoDB:

- Node.js applications use the official MongoDB Node.js driver or a library like Mongoose to connect to a MongoDB database.
- Mongoose provides an Object Data Modeling (ODM) layer, which simplifies the interaction with MongoDB by providing a schema-based approach.

❖ Data Flow:

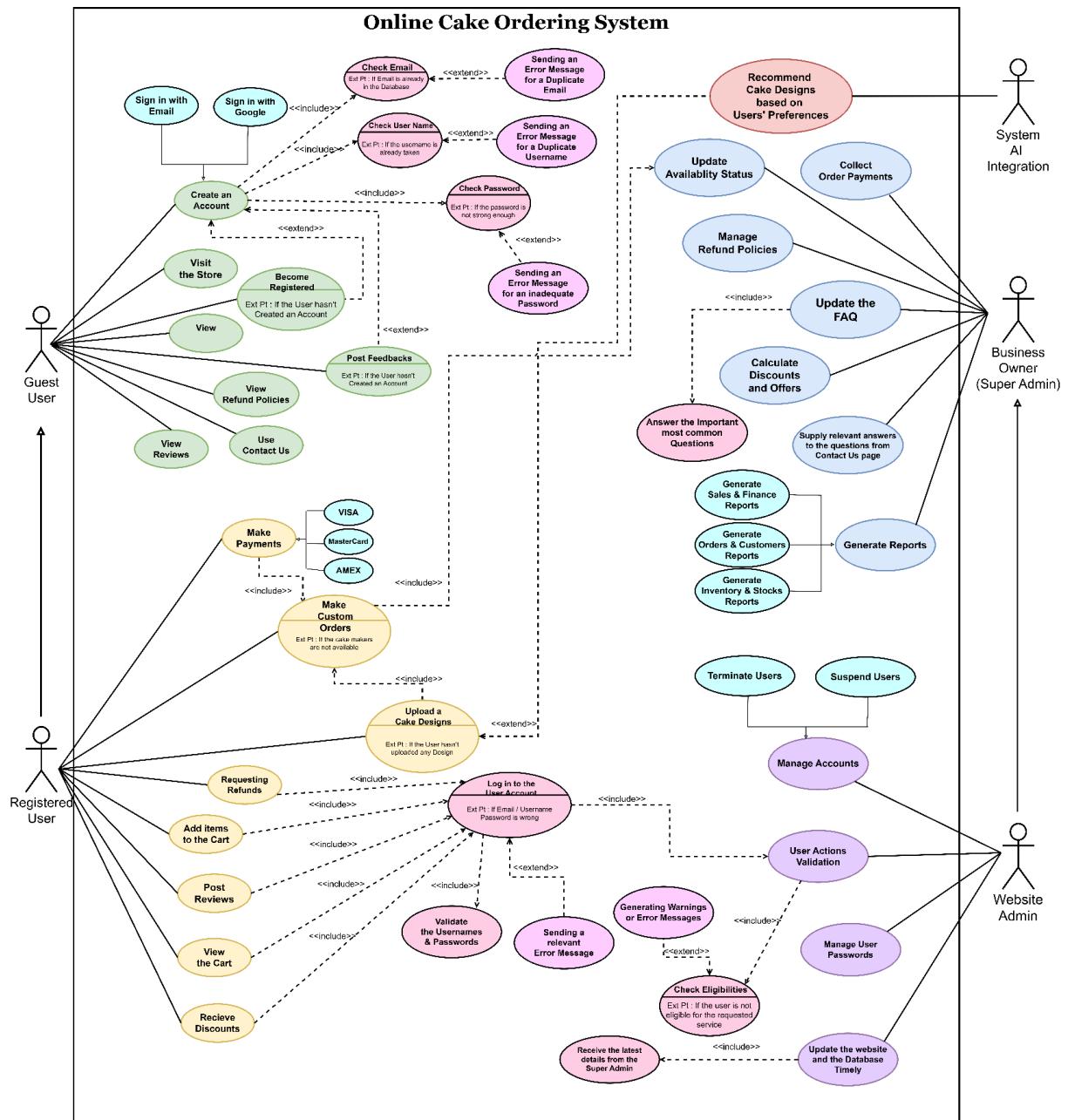
- When a client sends a request to the Express.js server, the server processes the request and may need to access or modify data in the database.
- Express.js uses the MongoDB driver or Mongoose to perform database operations, such as:
 - **Creating (Inserting) Data:** When a user submits a new order, for example, Express.js creates a new document in the MongoDB collection.
 - **Reading (Querying) Data:** When a user requests a list of their past orders, Express.js queries the MongoDB database and retrieves the relevant documents.
 - **Updating Data:** When a user modifies their order details, Express.js updates the corresponding document in the database.
 - **Deleting Data:** When a user cancels an order, Express.js deletes the corresponding document from the database.

❖ Data Exchange:

- Data is exchanged between Express.js and MongoDB in JSON format, which makes it easy to work with JavaScript objects.

Express.js then sends the processed data back to the client as a response.

2.4. Requirements Modelling



2.5. Use case scenarios

Number	UCS001	
Use Case ID	IT23183018	
Use Case Name	Make a Custom Order	
Summary	The user is directed to the cake order page in order to fill the relevant form of action	
Priority	01	
Pre-Condition	The user is prompted to log in to the system or create a new account	
Post-Condition	The user is directed to the payment gateway after successfully submitting the form	
Primary Actor(s)	The Registered User	
Trigger	The user wants to place an order for a cake	
Main Scenario	Step	
	01	The user logs into the system using his/her user credentials
	02	The user clicks on the “Order Now” button
	03	The user is directed to the Ordering Page
	04	A form with filtering options is displayed by the system
	05	The user fills all the necessary details
	06	The user uploads an image of his preferred cake design to the system
	07	The user successfully submits the form
	08	The system redirects the user to the payment gateway
Extensions	Step	
	01.a	The system generates an error message if the email is invalid
	01.b	The system generates an error message if the password is invalid
	02.a	If the user isn't registered, the system renders the login page
	04.a	If the vacation mode is turned on, the system will not render the form. Instead of it displays a message saying the cake maker(s) are not available at that moment
	06.a	If the user hasn't come up with any design to be uploaded, the system's AI integration will recommend sample images of cake designs based on the values that he/she filled in the form
Open Issues	#	
	1	The system has to be more optimized for the better results of image generations

Number	UCS002	
Use Case ID	IT23222786	
Use Case Name	Post Feedbacks	
Summary	A user gives feedback on a cake order by rating, commenting, and leaving a review.	
Priority	05	
Pre-Condition	The user should either register for a new account or log in to the system.	
Post-Condition	The feedback is successfully recorded and stored in the system.	
Primary Actor(s)	The User	
Trigger	After receiving the cake they bought, a customer wishes to give the service a rating	
Main Scenario	Step	
	01	The user enters their login credentials to access the system.
	02	The user Navigates to Feedback Section
	03	The user clicks on add new feedback button.
	04	The system presents a feedback form with choices to rate the cake and leave remarks.
	05	The user enters a rating, writes a review regarding the System
	06	User uploads desired images of cake that he/she received
	07	User submits the Feedback from
	08	The system collects and stores input and a confirmation message appears: "Thank you for your feedback!"
Extensions	Step	
	01.a	The user enters wrong login information, and the system returns an error notice.
	01.b	Login is not possible because the user's account cannot be found or has been deactivated.
	01.c	If a user is not registered, the system displays the login page.
	02.a	The user has no completed orders, so the feedback section is empty.
	07.a	When the system discovers that the same feedback has been submitted several times, it blocks the duplicate.
Open Issues	#	
	1	User can't give a feedback without login to the account.

Number	UCS003	
Use Case ID	IT23179158	
Use Case Name	Generate Reports	
Summary	The admin generates necessary reports for the due time.	
Priority	05	
Pre-Condition	The admin accesses the reports section.	
Post-Condition	The admin observes the platform's analytics and makes reports.	
Primary Actor(s)	The Administrator	
Trigger	The admin decides to generate reports to make the future arrangements of the website.	
Main Scenario	Step	
	01	The admin must log in to the platform using valid credentials to gain access.
	02	After authentication, the admin must navigate to the admin panel.
	03	Upon accessing the admin panel, the system must direct the user to the Ordering Page.
	04	The user must select the desired report type from the available options.
	05	The admin must define and apply parameters to filter the relevant data.
	06	The admin must initiate the report generation process by clicking the "Generate Report" button.
	07	Once the report is generated, the admin must click the "Print" button to obtain a hardcopy.
Extensions	Step	
	01.a	The system generates an error message if the username is wrong.
	01.b	The system generates an error message if the password is wrong
	04.a	The user can generate a report regarding Sales & Finance.
	04.b	The admin can generate a report regarding User Activity & Recipe Performance.
	05.a	The admin can apply a monthly-based filter for the selected report type.
	05.b	The admin can apply a monthly-based filter for the selected report type.
Open Issues	#	
	1	Ensuring all the relevant data is included before the report is generated.

Chapter 3

Design and Development

3.1. System Architecture

Sweet Bliss employs client-server architecture based on **MERN** stack, which enables modularity, scalability and responsiveness.

- ❖ **Frontend** – React.js for more dynamic, responsive and elegant UI.
- ❖ **Backend** – Express.js with Node.js for RESTful APIs.
- ❖ **Database** – MongoDB for flexibility and scalability in data storage.
- ❖ **3rd party APIs** – For automated image generation.

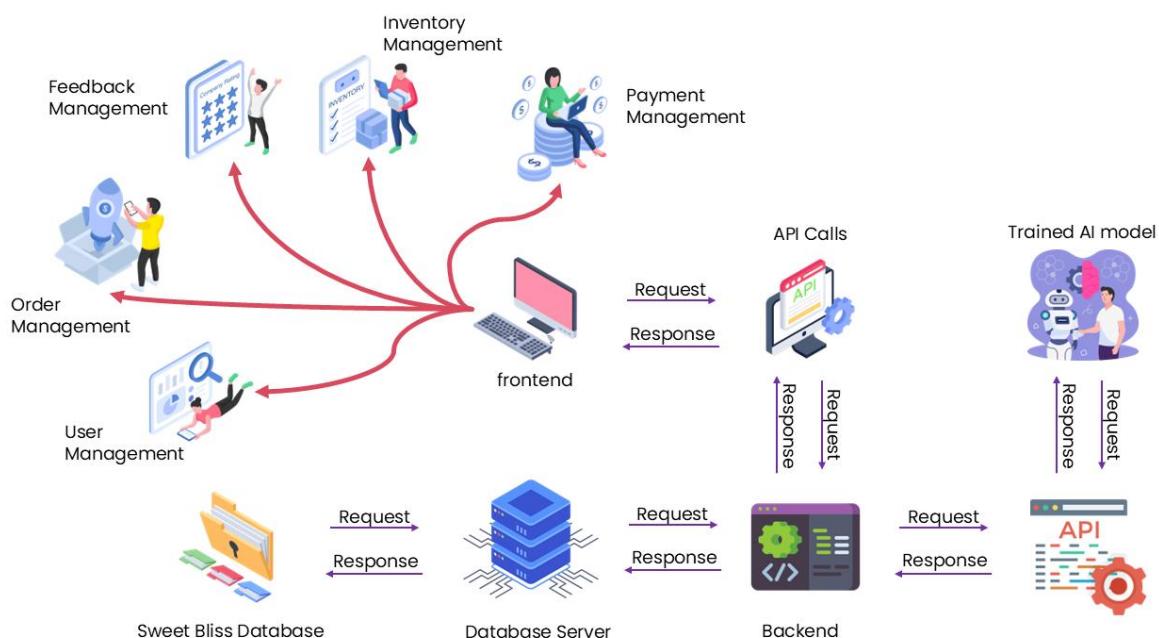


Figure 2 system diagram for system overview

3.2. Database Structure

Since the database is powered by MongoDB platform, which is a NoSQL platform, the data is represented as Collections rather than Tables.

Users Collection

- **_id** (ObjectId) - Primary Key
- **name** (String)
- **email** (String, Unique)
- **password** (String, Hashed)
- **phone** (String)
- **address** (String)
- **role** (String) - Enum: ['admin', 'customer']
- **created_at** (ISODate)

Sample JSON output:

```
{  
    "_id": ObjectId,  
    "name": "John Doe",  
    "email": "johndoe@example.com",  
    "password": "hashed_password",  
    "phone": "+123456789",  
    "address": "123 Street, City",  
    "role": "customer"  
    "created_at": ISODate("2025-02-14T12:00:00Z")  
}
```

Orders Collection

- **_id (ObjectId)** - Primary Key
- **user_id (ObjectId)** - References Users Collection
- **items (Array of Objects)**
 - **product_id (ObjectId)** - References Products Collection
 - **name (String)**
 - **weight (Number)** - Measured in kg/lb
 - **color (String)** - Example: 'Red', 'Blue', 'White'
 - **image_url (String)** - Custom design image uploaded by user
 - **toppings (Array of Strings)** - Example: ['Choco Chips', 'Fruits', 'Nuts']
 - **additional_description (String)** - User's custom message
 - **quantity (Number)**
 - **price (Number)**
- **total_price (Number)**
- **status (String)** - Enum: ['pending', 'processing', 'completed', 'cancelled']
- **order_date (ISODate)** - Selected order date from calendar
- **created_at (ISODate)**

Sample JSON output:

```
{  
    "_id": "65a3f1b7e4b09f24a3b8e123",
```

```

"user_id": "65a3e4c6f1a2b34789d4f789",
"items": [
  {
    "product_id": "65a3f6d5c9a0b123f9e4d567",
    "name": "Chocolate Fudge Cake",
    "weight": 1.5,
    "color": "Brown",
    "image_url":
      "https://example.com/uploads/custom_cake.jpg",
    "toppings": ["Choco Chips", "Caramel Drizzle"],
    "additional_description": "Please write 'Happy
    Birthday, Anna!' on top.",
    "quantity": 1,
    "price": 25.99
  },
  {
    "product_id": "65a3f9e2d7b2a567f8c3d890",
    "name": "Vanilla Cream Cake",
    "weight": 2,
    "color": "White",
    "image_url":
      "https://example.com/uploads/vanilla_cake.jpg",
    "toppings": ["Fruits", "Whipped Cream"],
    "additional_description": "No nuts, please!",
    "quantity": 1,
    "price": 30.99
  }
],
"total_price": 56.98,
"status": "pending",
"order_date": "2025-02-20T10:00:00.000Z",
"created_at": "2025-02-14T14:30:00.000Z"
}

```

Products Collection

- **_id** (ObjectId) - Primary Key
- **name** (String)
- **description** (String)
- **price** (Number)
- **stock** (Number)
- **image_url** (String)
- **created_at** (ISODate)

Sample JSON output:

```
{
  "_id": ObjectId,
  "name": "Vanilla Cake",
  "description": "A delicious vanilla cake with cream
frosting.",
  "price": 12.99,
  "stock": 10,
  "image_url": "https://example.com/cake.jpg",
  "created_at": ISODate("2025-02-14T12:00:00Z")
}
```

Payments Collection

- **_id** (ObjectId) - Primary Key
- **order_id** (ObjectId) - References Orders Collection
- **amount** (Number)
- **method** (String) - Enum: ['cash', 'credit_card', 'bank_transfer']
- **status** (String) - Enum: ['pending', 'completed', 'failed']
- **payment_date** (ISODate)

Sample JSON output:

```
{  
  "_id": ObjectId,  
  "order_id": ObjectId("order_id"),  
  "amount": 31.98,  
  "method": "credit_card",  
  "status": "completed",  
  "payment_date": ISODate("2025-02-14T12:00:00Z")  
}
```

Inventory Collection

- **_id** (ObjectId) - Primary Key
- **name** (String)
- **quantity** (Number)
- **unit** (String) - Example: 'kg', 'liters'
- **threshold** (Number) - Minimum stock alert level
- **last_updated** (ISODate)

Sample JSON output:

```
{  
  "_id": ObjectId,  
  "name": "Flour",  
  "quantity": 50,  
  "unit": "kg",  
  "threshold": 5,  
  "last_updated": ISODate("2025-02-14T12:00:00Z")  
}
```

Reviews Collection

- **_id** (ObjectId) - Primary Key
- **user_id** (ObjectId) - References Users Collection
- **product_id** (ObjectId) - References Products Collection
- **rating** (Number) - 1 to 5
- **likes** (Number)
- **comment** (String)
- **created_at** (ISODate)

Sample JOSN output:

```
{
```

```
        "_id": ObjectId,
        "user_id": ObjectId("user_id"),
        "product_id": ObjectId("product_id"),
        "rating": 5,
        "comment": "Absolutely delicious!",
        "created_at": ISODate("2025-02-14T12:00:00Z")
    }
```

Promotions Collection

- **_id** (ObjectId) - Primary Key
- **title** (String)
- **description** (String)
- **discount** (Number) - Percentage discount
- **start_date** (ISODate)
- **end_date** (ISODate)

Sample JSON output:

```
{
    "_id": ObjectId,
    "title": "Valentine's Day Discount",
    "description": "Get 20% off on all cakes!",
    "discount": 20,
    "start_date": ISODate("2025-02-10T00:00:00Z"),
    "end_date": ISODate("2025-02-14T23:59:59Z")
}
```

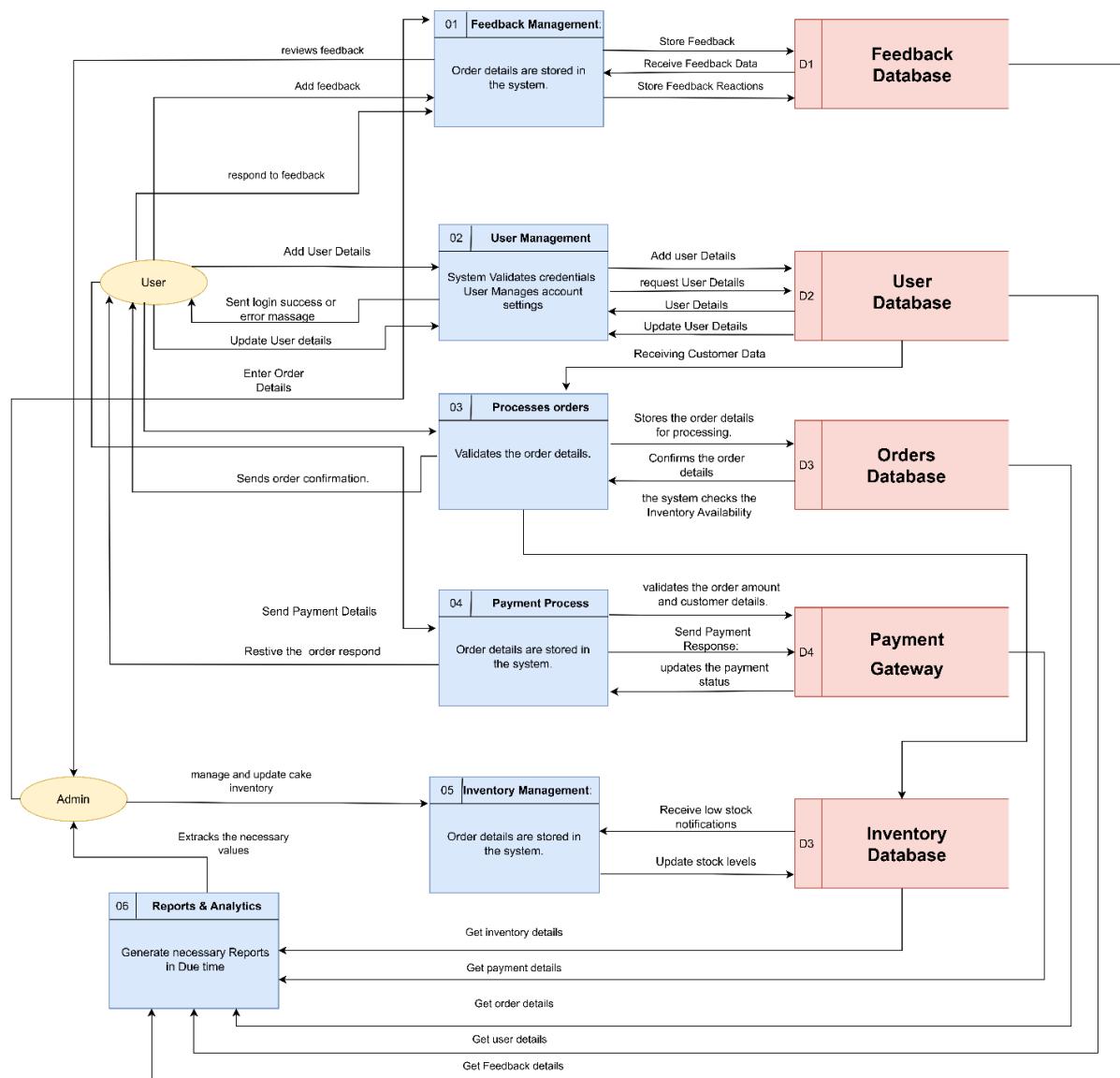
Reports Collection

- **_id** (ObjectId) - Primary Key
- **title** (String)
- **report_type** (String) - Enum: ['sales', 'inventory', 'customer']
- **generated_at** (ISODate)
- **file_url** (String)

Sample JSON output:

```
{
    "_id": ObjectId,
    "title": "Monthly Sales Report - Jan 2025",
    "report_type": "sales",
    "generated_at": ISODate("2025-02-14T12:00:00Z"),
    "file_url": "https://example.com/repo/feb-2025.pdf"
}
```

3.3. Dataflow diagram



3.4. Entity Relationship diagram

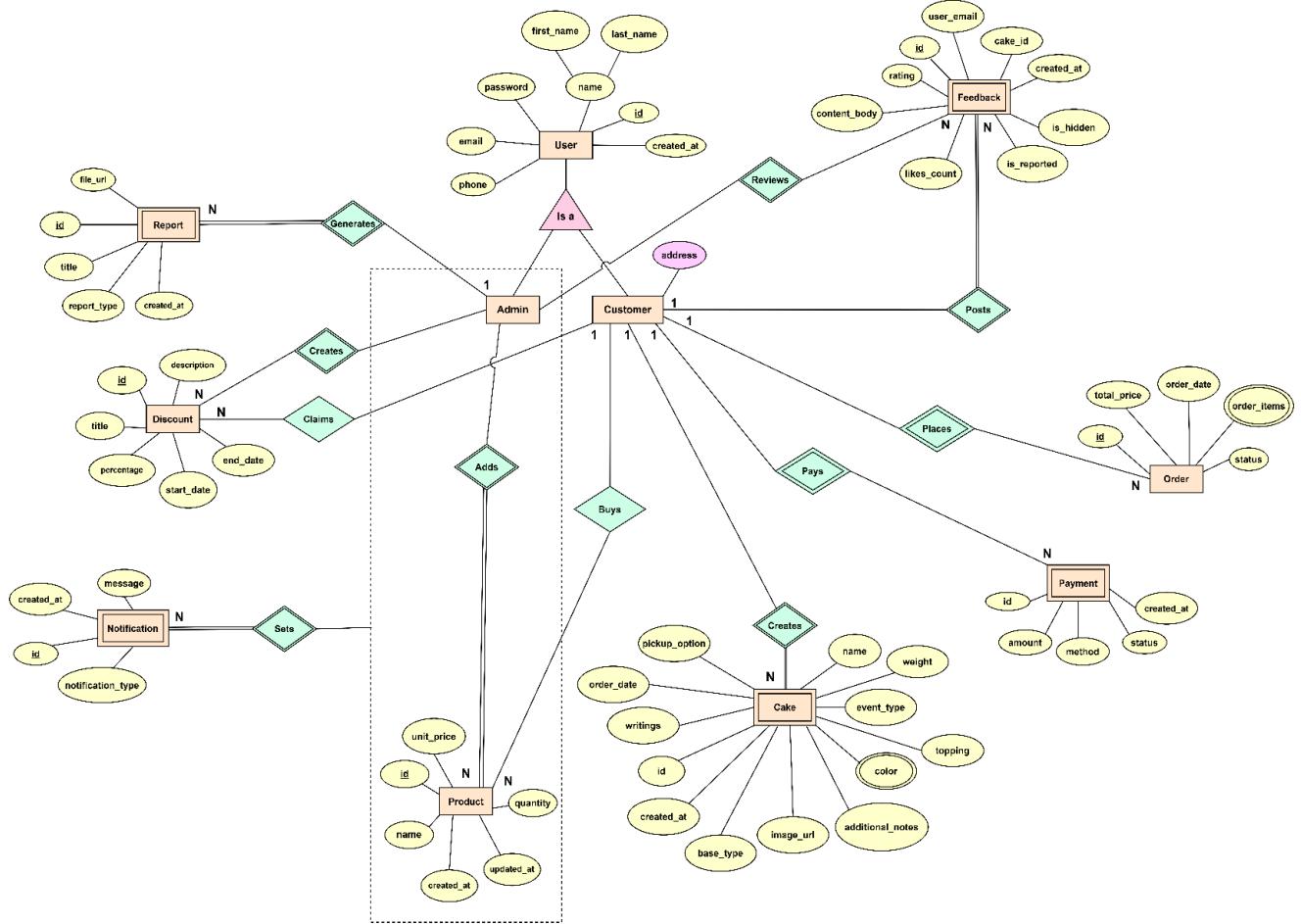


Figure 3 ER diagram

3.5. Normalized Schema

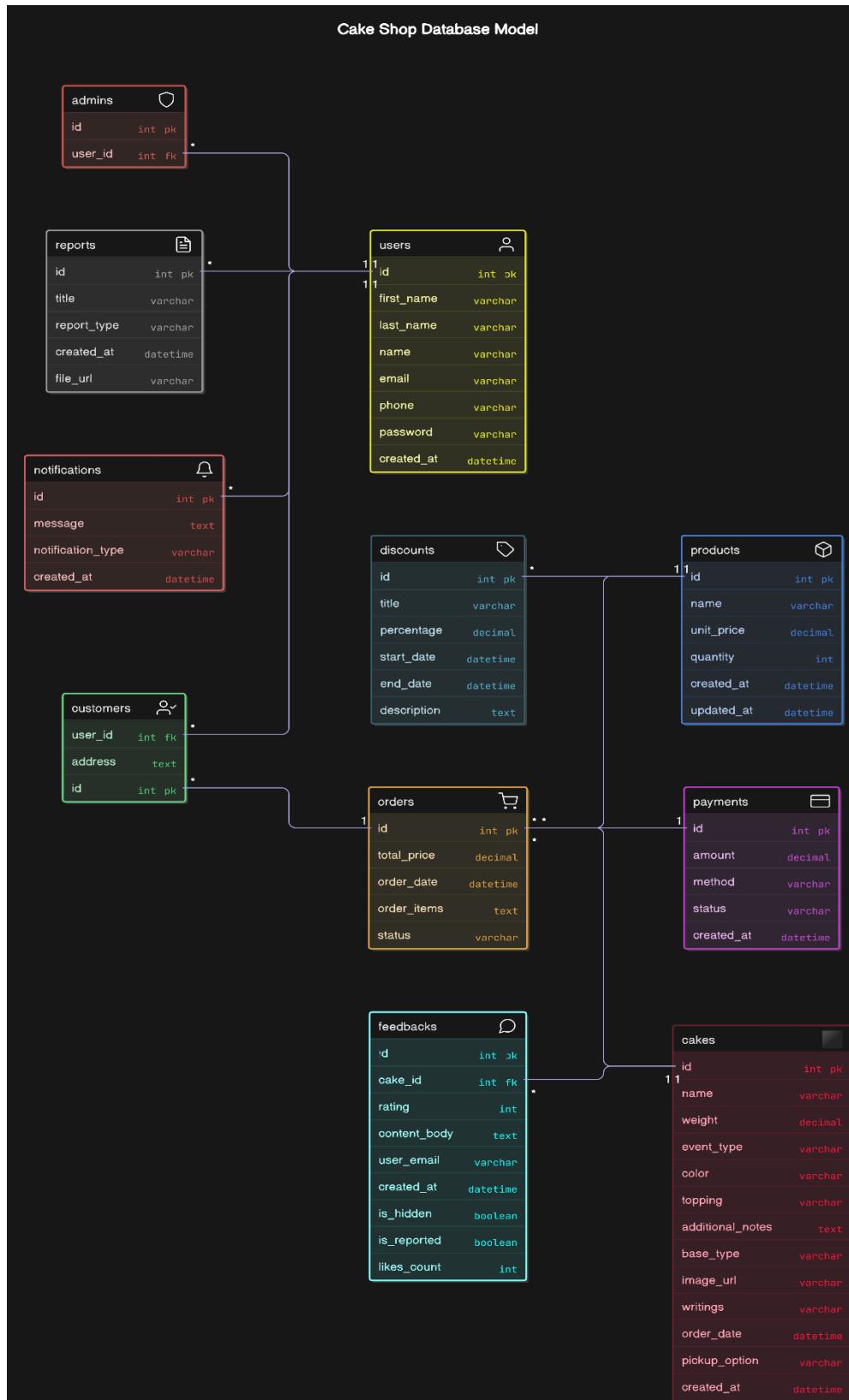


Figure 4 normalized schema of the database

3.6. Flow Chart

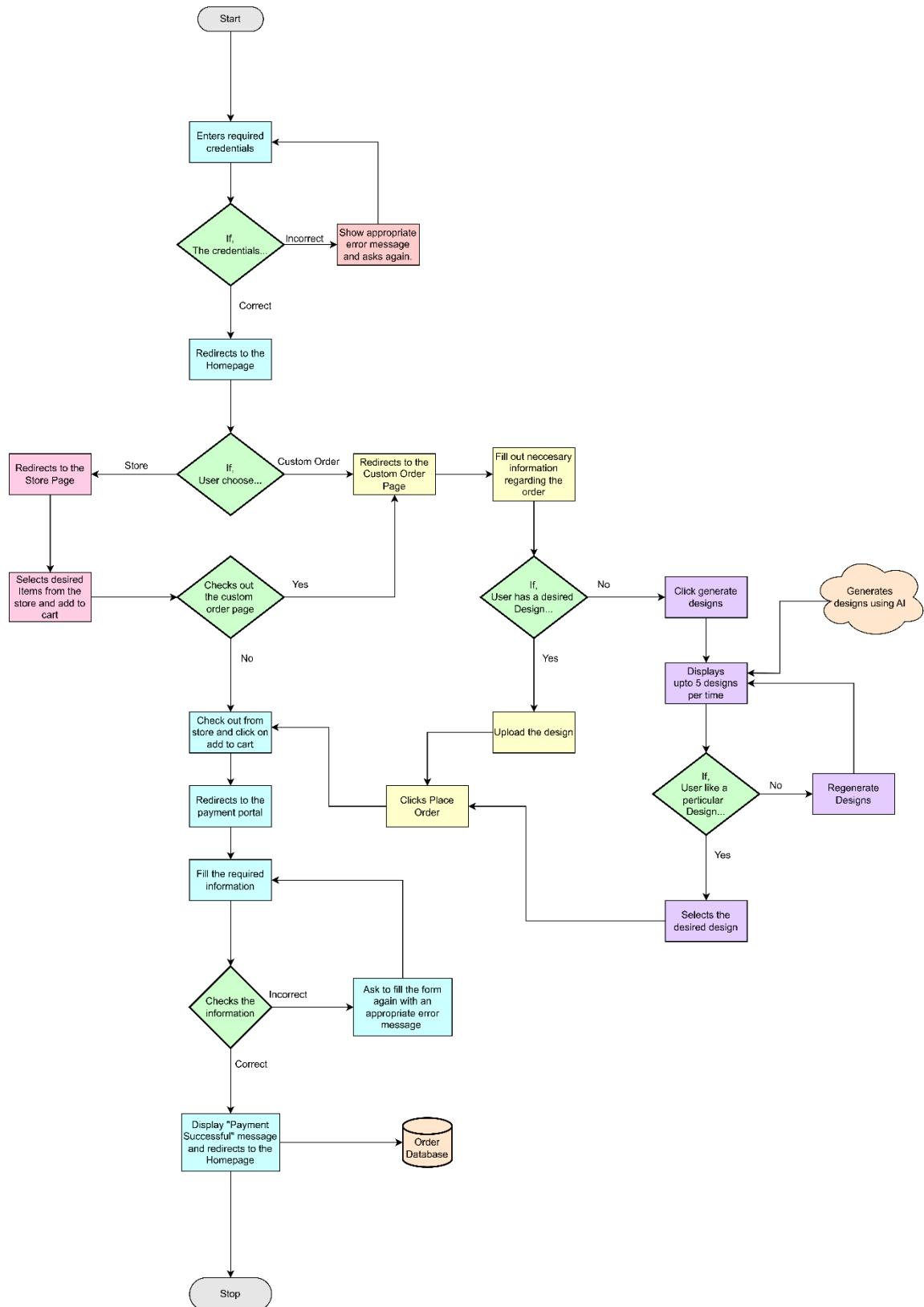


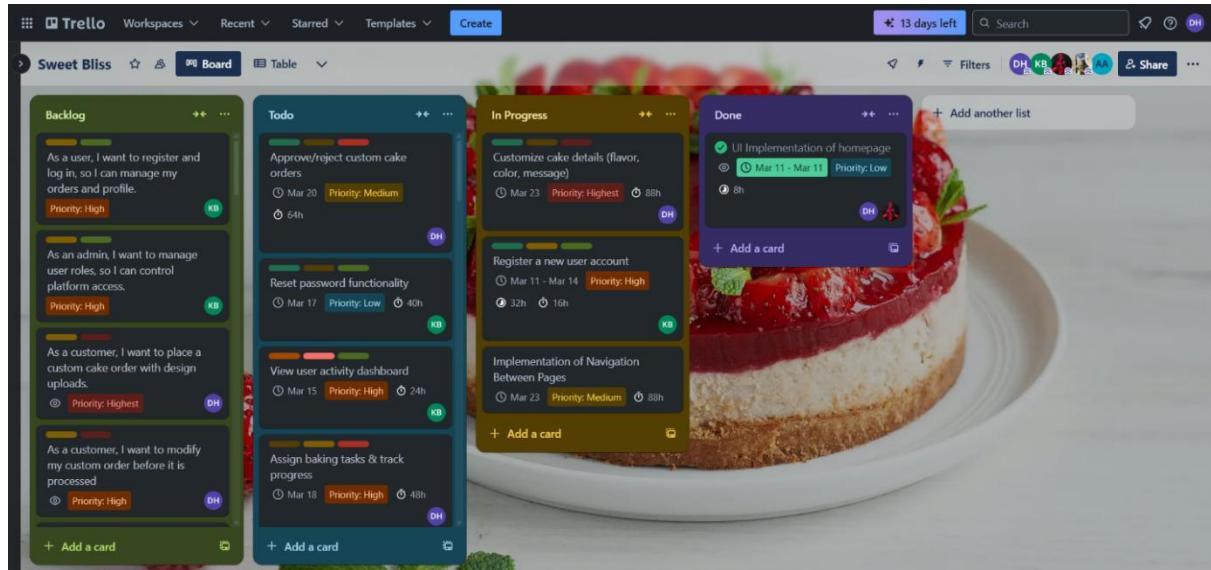
Figure 5 flow chart of a user placing an order

3.7. Workflow and plans



Figure 6 Gantt chart

3.8. Kanban Board



3.9. Product Backlog

Order ID	User Story ID	User Story	Issue Type	Est. Hours	Priority	Status	Assignee	Sprint	Tags
01	001	"As a user, I want to register and log in, so I can manage my orders and profile."	Feature	12	Critical	To Be Started	Ms. K. Budara	Sprint 1	user-management
02	002	"As an admin, I want to manage user roles, so I can control platform access."	Improvement	10	Major	To Be Started	Ms. K. Budara	Sprint 1	user-management
03	003	"As a customer, I want to place a custom cake order with design uploads."	Feature	18	Show-stopper	To Be Started	Mr. D. Hirusha	Sprint 1	custom-orders
04	004	"As a customer, I want to modify my custom order before it is processed."	Feature	16	Critical	To Be Started	Mr. D. Hirusha	Sprint 1	custom-orders

05	005	"As a business owner, I want to accept or reject orders based on availability."	Feature	14	Critical	To Be Started	Mr. D. Hirusha	Sprint 2	custom-orders
06	006	"As a customer, I want to track my order status in real-time."	Feature	16	Critical	To Be Started	Mr. D. Hirusha	Sprint 2	order-management
07	007	"As an admin, I want to update inventory levels when stock is low."	Feature	14	Major	To Be Started	Mr. L. Navanga	Sprint 2	inventory
08	008	"As an admin, I want to receive low-stock alerts to restock ingredients."	Feature	12	Critical	To Be Started	Mr. L. Navanga	Sprint 2	inventory
09	009	"As a customer, I want to view available cakes and stock levels before ordering."	Feature	10	Normal	To Be Started	Mr. L. Navanga	Sprint 3	inventory
10	010	"As an admin, I want to manage promotions and discount offers."	Feature	14	Major	To Be Started	Mr. L. Navanga	Sprint 3	inventory

11	011	"As a customer, I want multiple payment options, including credit card and PayPal."	Feature	18	Showstopper	To Be Started	Ms. R. Budara	Sprint 1	finance
12	012	"As an admin, I want to generate invoices and send payment confirmations."	Feature	12	Critical	To Be Started	Ms. R. Budara	Sprint 2	finance
13	013	"As an admin, I want to issue refunds for rejected orders automatically."	Feature	16	Critical	To Be Started	Ms. R. Budara	Sprint 2	finance
14	014	"As a business owner, I want to view financial reports on sales and revenue."	Feature	18	Major	To Be Started	Ms. R. Budara	Sprint 3	finance
15	015	"As a customer, I want to leave feedback on my order experience."	Feature	10	Normal	To Be Started	Ms. T. Nethmini	Sprint 3	feedback
16	016	"As an admin, I want to moderate and respond to customer feedback."	Feature	12	Critical	To Be Started	Ms. T. Nethmini	Sprint 3	feedback

17	017	"As a customer, I want to see top-rated cakes and reviews on the homepage."	Feature	14	Major	To Be Started	Ms. T. Nethmini	Sprint 3	feedback
18	018	"As a customer, I want to receive notifications on order status updates."	Feature	10	Normal	To Be Started	Mr. D. Hirusha	Sprint 2	custom-orders
19	019	"As a business owner, I want a vacation mode to temporarily disable orders."	Feature	16	Critical	To Be Started	Mr. D. Hirusha	Sprint 3	custom-orders
20	020	"As a returning customer, I want loyalty rewards for repeat purchases."	Feature	20	Major	To Be Started	Ms. T. Nethmini	Sprint 3	feedback
21	021	"As a customer, I want to cancel my order before it is accepted."	Feature	14	Critical	To Be Started	Mr. D. Hirusha	Sprint 2	custom-orders
22	022	"As an admin, I want to export sales reports to Excel."	Feature	12	Normal	To Be Started	Ms. R. Budara	Sprint 3	finance

23	023	"As an admin, I want to assign customer support tickets to the right team members."	Feature	12	Major	To Be Started	Ms. K. Budara	Sprint 3	user-management
24	024	"As a customer, I want to change my delivery address before the order is shipped."	Feature	14	Major	To Be Started	Mr. D. Hirusha	Sprint 2	custom-orders
25	025	"As an admin, I want to send personalized discount codes to loyal customers."	Feature	16	Major	To Be Started	Ms. T. Nethmini	Sprint 3	feedback
26	026	"As a business owner, I want an analytics dashboard for order trends."	Feature	18	Critical	To Be Started	Ms. R. Budara	Sprint 3	finance
27	027	"As a customer, I want to pre-order seasonal cakes ahead of time."	Feature	16	Major	To Be Started	Mr. D. Hirusha	Sprint 3	custom-orders
28	028	"As an admin, I want to verify customer reviews before they are published."	Feature	12	Critical	To Be Started	Ms. T. Nethmini	Sprint 3	feedback

29	029	"As a business owner, I want an emergency order option for last-minute cakes."	Feature	18	Show-stopper	To Be Started	Mr. D. Hirusha	Sprint 3	custom-orders
30	030	"As an admin, I want to generate tax reports for financial compliance."	Feature	14	Critical	To Be Started	Ms. R. Budara	Sprint 3	finance

3.10. Figma UI designs



Login

Stay updated on your most desired recipes

Email or Username

Password

[Forgot Password ?](#)

Log in

or

[Sign in with Google](#)

Have no account ? [Join Now](#)

[Terms and Conditions](#) [Privacy Policies](#) [FAQ](#) [Store](#) [Contact Us](#) [Blog](#)

@Developed by WD_I2I Team



Register

Stay updated on your most desired recipes

Email

Username

Password (8 characters minimum)

Repeat Password

By clicking accept and register, you agree to the [Terms of Use](#),
the [Privacy Policy](#) and [Cookies Policy](#) of this website.

Accept and Register

or

[Sign in with Google](#)

Already registered ? [Log in](#)

[Terms and Conditions](#) [Privacy Policies](#) [FAQ](#) [Store](#) [Contact Us](#) [Blog](#)

@Developed by WD_I2I Team

=

Email

New Password

Confirm New Password

OTP

Submit

[Privacy Policy](#) | [Refund/Return Policy](#) | [Copyright@www.sweetbliss.lk](#)



Unveil your Desires

Choose your Order Date:

Month 2000 >



Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7



[view more...](#)

Nature of the Event

Base Colors

Do you require a Design ?

I have one already Yes I do

↑

Base type of the cake

Pickup Option

*Note: delivery will apply additional charges.

Date of Requirement

Toppings

*Note: delivery will apply additional charges.

Cake Size

Writings on Top

*Note: delivery will apply additional charges.

Additional Notes

Submit

[Terms and Conditions](#) [Privacy Policies](#) [FAQ](#) [Store](#) [Contact Us](#) [Blog](#)

@Developed by WD_I2I Team



Unveil your Desires

Choose your Order Date

Month 2000 > < <						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7



view more...

Nature of the Event

Base Colors

Do you require a Design ?

Base type of the cake

Pickup Option

I have one already Yes I do

Date of Requirement

Toppings

Cake Size

Writings on Top

*Note: delivery will apply additional charges.

*Note: delivery will apply additional charges.

*Note: delivery will apply additional charges.

Submit

Terms and Conditions Privacy Policies FAQ Store Contact Us Blog

@Developed by WD_I2I Team



Select your card type

VISA MasterCard PayPal

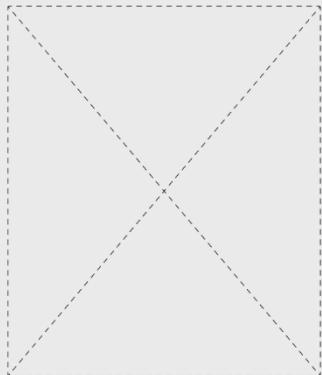
Card Number

Card Holder's Name

Expiry Date

CVC

Pay Now



Terms and Conditions Privacy Policies FAQ Store Contact Us Blog

@Developed by WD_I2I Team



Name : John Doe
Email : john.doe@gmail.com
Address : 121/Baker st,England

Edit

My Orders

Order No.	Items	Qty	Price	Order Status	Confirmation
001	Item Name	2	3600	Baking...	✓
002	Item Name	3	5400	Waiting..	✓ X
003	Item Name	1	1800	Waiting...	✓ X
004	Item Name	4	7200	Done...	✓

[Terms and Conditions](#) [Privacy Policies](#) [FAQ](#) [Store](#) [Contact Us](#) [Blog](#)

@Developed by WD_J2I Team



- [Orders](#)
- [Revenue](#)
- [Deliveries](#)
- [Feedback](#)
- [Profile](#)
- [Discounts](#)
- [Categories](#)
- [Price List](#)

Orders

Accepted Orders

Order No.	Items	Qty	Price	Order Status
001	Item Name	2	3600	Baking...
002	Item Name	4	7200	Paid...

Pending Orders

Order No.	Items	Qty	Price	Order Status	Confirmation
001	Item Name	3	5400	Waiting..	✓ X
002	Item Name	1	1800	Waiting...	✓ X



Revenue

Revenues

[Monthly](#) [Annually](#)

[Orders](#)

[Revenue](#)

[Deliveries](#)

[Feedback](#)

[Profile](#)

[Discounts](#)

[Categories](#)

[Price List](#)

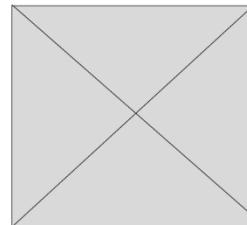
Item	Expences	Income
Cost of Ingredients	48000	—
Cost of Delivery	10000	—
Discounts / Offers	5000	—
Revenue from Pre-designed Orders		
Butter Cakes	—	18000
Chocolate Cakes	—	20000
Brownies	—	30000
Cupcakes	—	15000
Revenue from Custom orders		
Bento Cake(butter)	—	10800
Bento Cake(chocolate)	—	6000
Mini Cake(butter)	—	15400
Mini Cake(chocolate)	—	5000
Icing Cake(butter)	—	20800
Icing Cake(chocolate)	—	32000
Total	63000	173000
Net Revenue		110000



About Us

Norem ipsum dolor sit amet, consectetur adipiscing elit. Etiam eu turpis molestie, dictum est a, mattis tellus. Sed dignissim, metus nec fringilla accumsan, risus sem sollicitudin lacus, ut interdum tellus elit sed risus. Maecenas eget condimentum velit, sit amet feugiat lectus. Class aptent taciti sociosqu ad littera torquent per conubia nostra, per inceptos himenaeos. Proesent auctor purus luctus enim egestas, ac scelerisque ante pulvinar. Donec ut rhoncus ex. Suspendisse ac rhoncus nisl, eu tempor urna. Curabitur vel bibendum lorem. Morbi convallis convallis diam sit amet lacinia. Aliquam in elementum tellus.

Curabitur tempor quis eros tempus lacinia. Nam bibendum pellentesque quam a convallis. Sed ut vulputate nisl. Integer in felis sed leo vestibulum venenatis. Suspendisse quis arcu sem. Aenean feugiat ex eu vestibulum vestibulum. Morbi a eleifend magna. Nam metus lacus, porttitor eu mauris a, blandit ultrices nibh. Mauris sit amet magna non ligula vestibulum eleifend. Nulla varius volutpat turpis sed lacinia. Nam eget mi in purus lobortis eleifend. Sed nec ante dictum sem condimentum ullamcorper quis venenatis nisl. Proin vitae facilisis nisi, ac posuere leo.





Terms & Conditions

1. **Introduction:** The introduction outlines the agreement between the website owner and the users who access and use the site.
2. **User Agreement:** Users must agree to abide by the terms and conditions in order to use the website, product, or mobile application.
3. **Intellectual Property Rights:** The website owner or its licensors own all rights to the intellectual property and material contained in the website.
4. **User Restrictions:** The terms and conditions define what actions are prohibited on the website.
5. **User Content:** Any content that users choose to display on the website is considered their content.
6. **License to User Content:** By displaying their content, users grant the website owner a non-exclusive, worldwide, irrevocable, royalty-free, sublicensable license to use, reproduce, adapt, publish, translate and distribute it in any and all media.
7. **Confidentiality:** Any user ID and password for the website are confidential and users must maintain the confidentiality of such information.

[Terms and Conditions](#)

[Privacy Policies](#)

[FAQ](#)

[Store](#)

[Contact Us](#)

[Blog](#)

@Developed by WD_121 Team



Privacy Policies

At our platform, we prioritize your privacy. We are committed to protecting and respecting your personal data. Our policies are designed to ensure transparency, promote user trust, and comply with legal obligations. We collect, use, and store your personal data securely and responsibly. We encourage you to read our policies to understand your rights and our responsibilities.

1. **Data Collection:** We collect personal data necessary for service delivery and improvement.
2. **Data Use:** We use your data to personalize and improve your experience.
3. **Data Protection:** We implement robust security measures to protect your data.
4. **Data Sharing:** We do not sell your data to third parties.
5. **User Rights:** You have the right to access, correct, or delete your personal data.
6. **Cookies:** We use cookies to enhance user experience and site functionality.
7. **Policy Updates:** We periodically update our policies and notify users of significant changes.
8. **Contact Us:** For any privacy concerns or queries, users can contact us directly.

[Terms and Conditions](#)

[Privacy Policies](#)

[FAQ](#)

[Store](#)

[Contact Us](#)

[Blog](#)

@Developed by WD_121 Team



Have a Sweet Idea or a Custom Cake Request? Let's Talk

Name*

Email*

Phone*

Company / Organization

Tell us your about Desires

Colombo , Sri Lanka

info.reciprealm@outlook.com

(+94) 11 777 8880 , (+94) 11 777 8881

Send Message

[Terms and Conditions](#)

[Privacy Policies](#)

[FAQ](#)

[Store](#)

[Contact Us](#)

[Blog](#)

@Developed by WD_I2I Team

3.11. Actual User Interfaces

3.11.1. Common User Interfaces

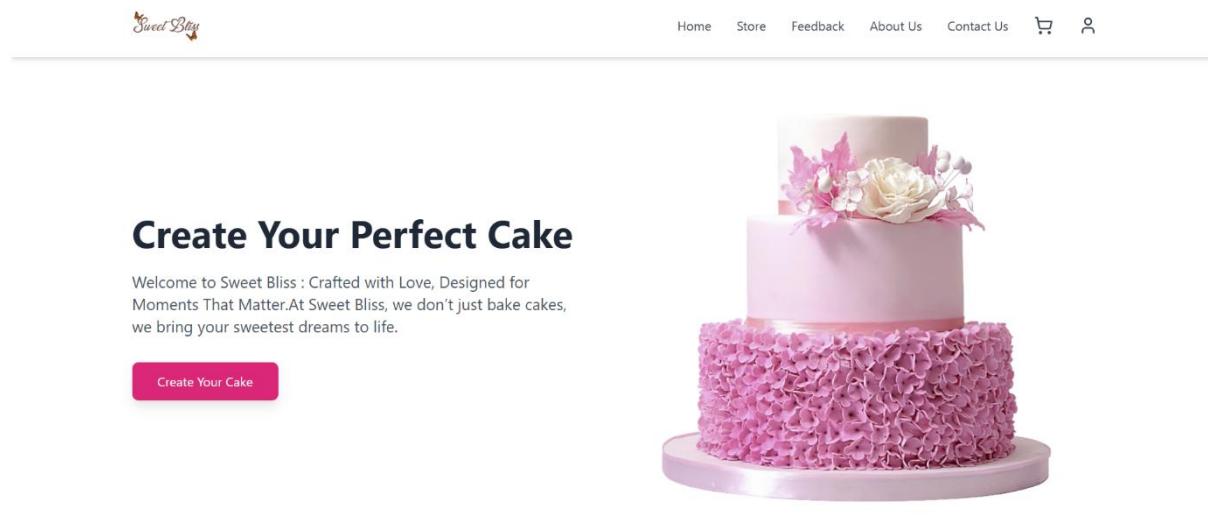


Figure 7 - Home Page (Hero Section)

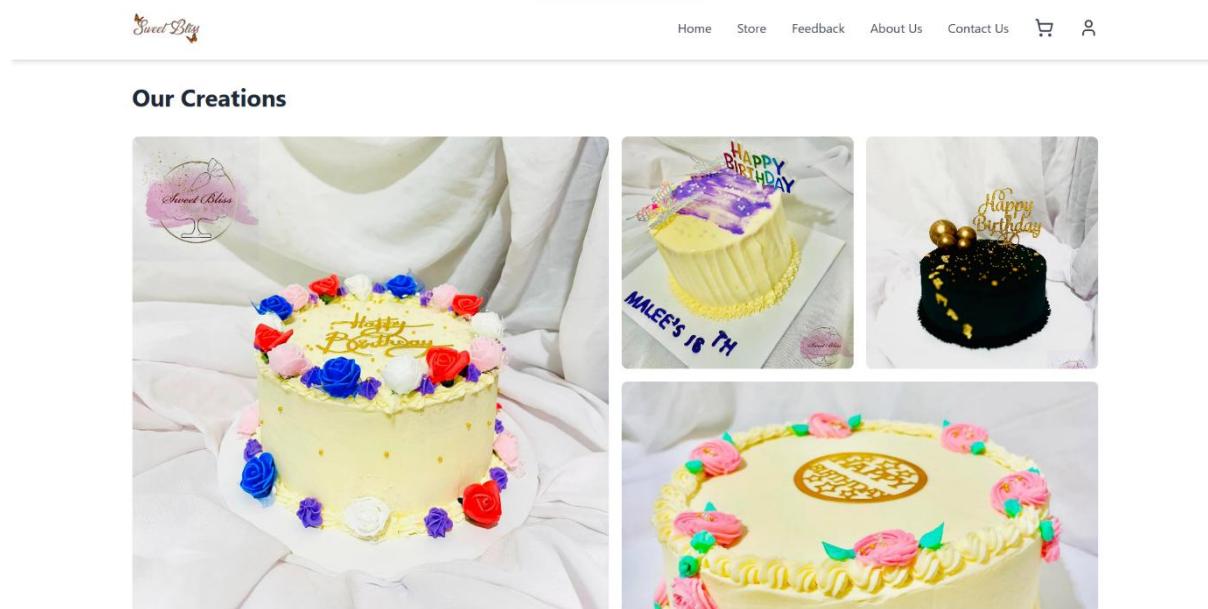


Figure 8 - Home Page (Our Creation Section)



Home Store Feedback About Us Contact Us

Contact Us

We'd love to hear from you! Let us know how we can help.

Get in Touch

Phone
+1 (555) 123-4567

Email
info@sweetbliss.com

Address
123 Sweet Street, Bakery Lane
New York, NY 10001

Hours
Mon-Sat: 9:00 AM - 8:00 PM
Sunday: 10:00 AM - 6:00 PM

Send us a Message

Your Name

Email Address

Subject

Message

Send Message

localhost:5173/feedback

Figure 9 - Contact Us Page



Home Store Feedback About Us Contact Us

About Us

Creating Sweet Moments Since 2015

Our Story

Sweet Bliss began its journey with a simple passion for creating delightful moments through exquisite cakes and pastries. Founded in 2015, we've grown from a small home bakery to one of the most trusted names in custom cake creation. Our team of skilled artisans combines traditional baking methods with modern techniques to create stunning masterpieces that not only look beautiful but taste amazing.

We pride ourselves on using only the finest ingredients and creating each cake with meticulous attention to detail. Whether it's a wedding, birthday, or special celebration, we're dedicated to making your moments sweeter and more memorable.

Banking Information

Commercial Bank

Account Name:
Sweet Bliss Cakes Ltd
Account Number:
1234-5678-9012-3456
Branch:
Main Street Branch

Bank of Ceylon

Account Name:
Sweet Bliss Cakes Ltd
Account Number:
9876-5432-1098-7654
Branch:
Central Branch

People's Bank

Account Name:
Sweet Bliss Cakes Ltd
Account Number:
4567-8901-2345-6789
Branch:
City Center Branch

Figure 10 - About Us Page

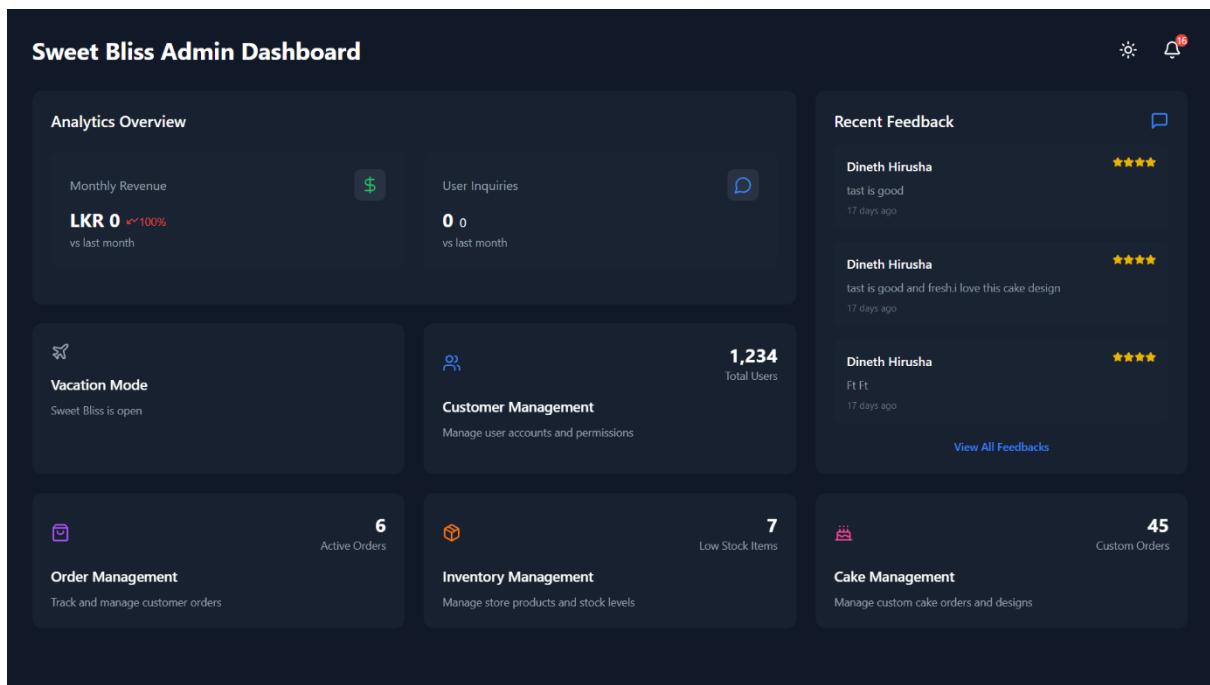


Figure 11 - Admin Dashboard

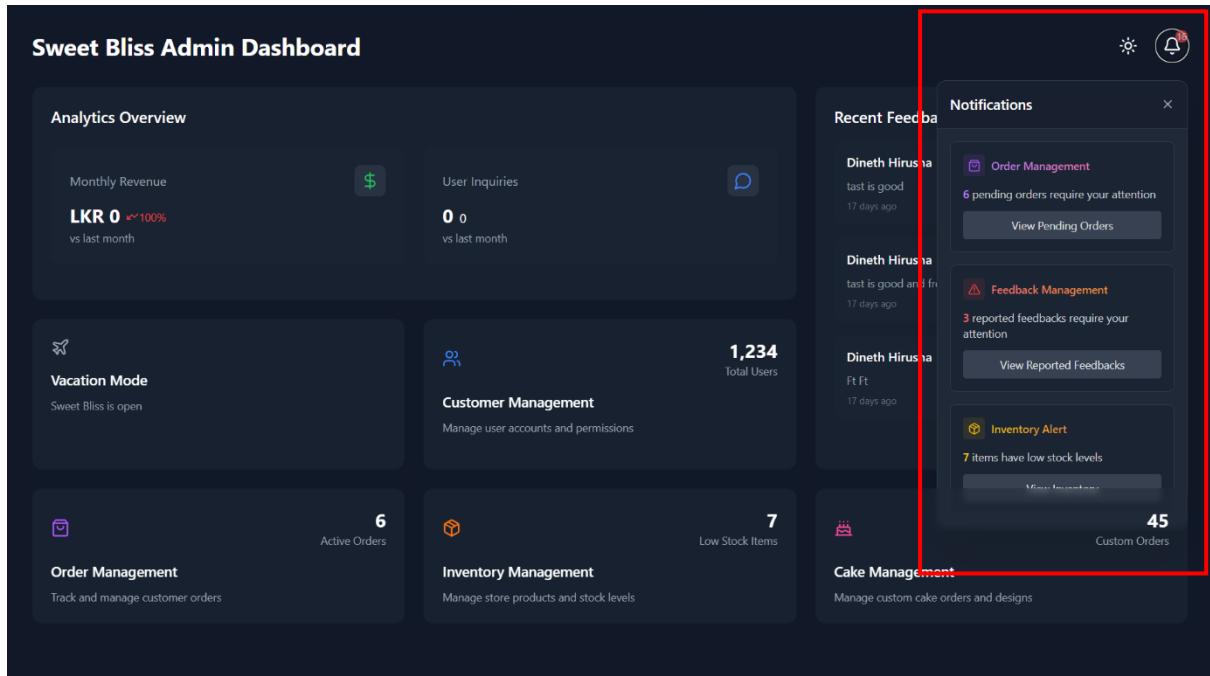


Figure 12 - Admin dashboard (Notification System)

3.11.2. Customer Management User Interfaces

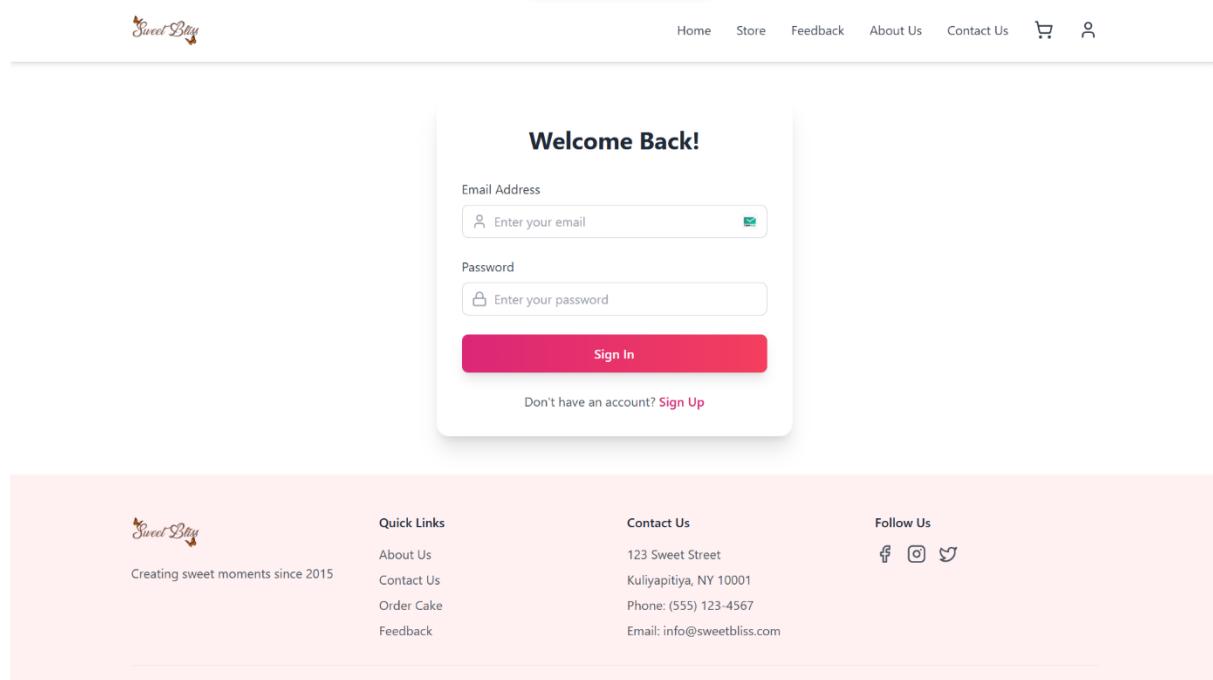


Figure 13 - Login Page

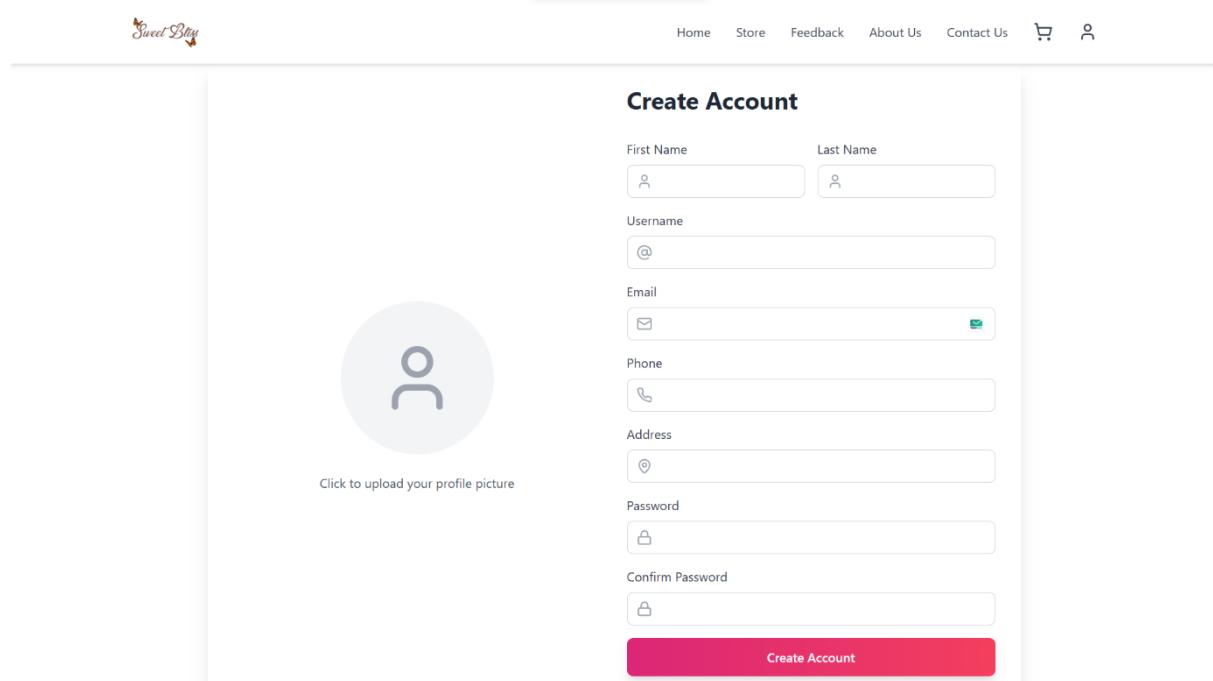


Figure 14 - Register Page

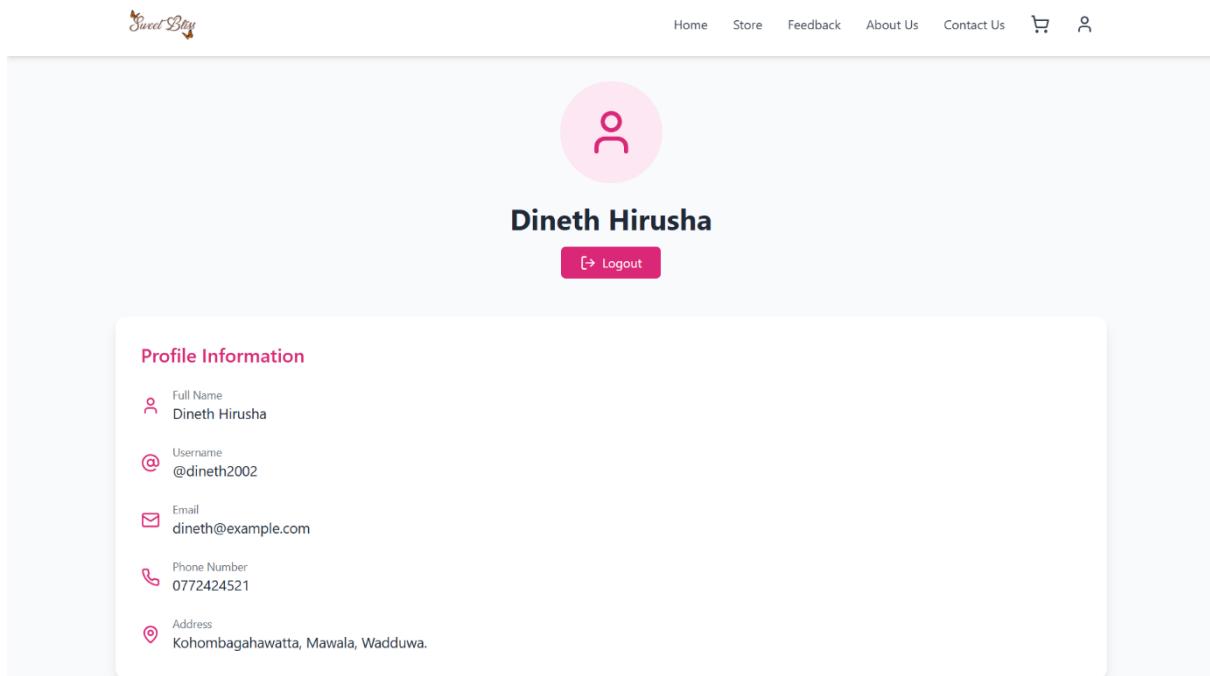


Figure 15 - User Profile Page

The screenshot shows the 'My Orders' page. At the top, there is a close button 'X'. The title 'My Orders' is centered above a table. The table has columns: Order Details, Items, Total, Status, and Actions. There are two rows of data:

Order Details	Items	Total	Status	Actions
Order #6b6beb Apr 28, 2025, 08:16 AM	Custom Ribbon Cake Qty: 1 x Rs. 1470 Capybara Plushie Qty: 2 x Rs. 900	Rs. 2370	Pending	
Order #33f25d Apr 28, 2025, 04:15 AM	Custom Butter Cake Qty: 1 x Rs. 1870	Rs. 1870	Completed	

Figure 16 - Customer Orders Page

Delivery Tracking

Real-time monitoring of your scrap collection deliveries

Map View List View Detail View

All Deliveries					
CUSTOMER	ADDRESS	ITEMS	DATE	STATUS	ACTIONS

Customer Information

Customer Name*

Contact Number*

Address*

Items for Collection*

Add

Scheduled Date*

...

Initial Status

▼

Schedule Delivery

Delivery Tracking

Real-time monitoring of your scrap collection deliveries

Map View List View Detail View

Delivery Map



Delivery Details

Select a delivery on the map to view details

Figure 17 - Delivery Tracking Page

3.11.3. Order Management related User Interfaces

The screenshot shows the 'Custom Cake Order' page from a website. At the top, there's a navigation bar with links for Home, Store, Feedback, About Us, Contact Us, a shopping cart icon, and a user profile icon. Below the navigation is the main title 'Custom Cake Order' in a large pink font, followed by the subtitle 'Unveil your Desires' in a smaller pink font. On the left, there's a pink sidebar containing a calendar for 'May 2025'. The calendar shows dates from 1 to 31, with May 15th highlighted. A note at the bottom of the sidebar says: '* Orders require at least 2 days advance notice'. To the right of the sidebar are several input fields: 'Nature of the Event' (empty), 'Base Type' (set to 'Butter Cake'), 'Date of Requirement' (empty), 'Cake Weight' (set to '500g'), 'Pickup Option' (set to 'Self Pickup'), 'Base Colors' (three small circles), 'Toppings' (set to 'Sprinkles'), 'Writings on Top' (empty), 'Do you require a Design?' (radio buttons for 'I have one already' and 'Yes I do'), and 'Additional Notes' (a text area with placeholder '0/100 words').

Figure 18 - Custom Cake Order Page (Unfilled)

This screenshot shows the same 'Custom Cake Order' page but with filled-in information. The calendar now displays 'February 2026' with February 14th highlighted. The 'Nature of the Event' field contains 'Valentine Event', 'Base Type' is 'Butter Cake', 'Date of Requirement' is 'February 14, 2026', 'Cake Weight' is '1kg', 'Pickup Option' is 'Self Pickup', and 'Base Colors' are three specific colors (red, yellow, and pink). The 'Toppings' field is 'Cherries on top', 'Writings on Top' is 'Happy Valentines Day', 'Do you require a Design?' has 'I have one already' selected, and the file 'cake_image(14).jpg' is attached. The 'Additional Notes' field contains 'None'.

Figure 19 - Custom Cake Order Page (Filled)

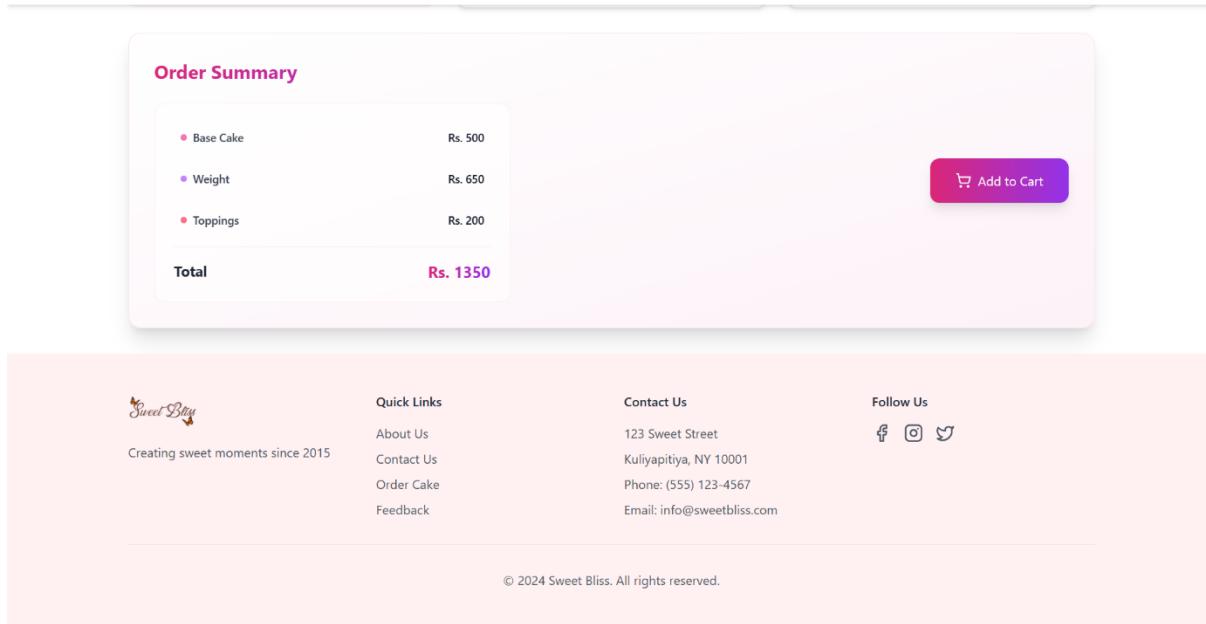


Figure 20 - Order Summary Section

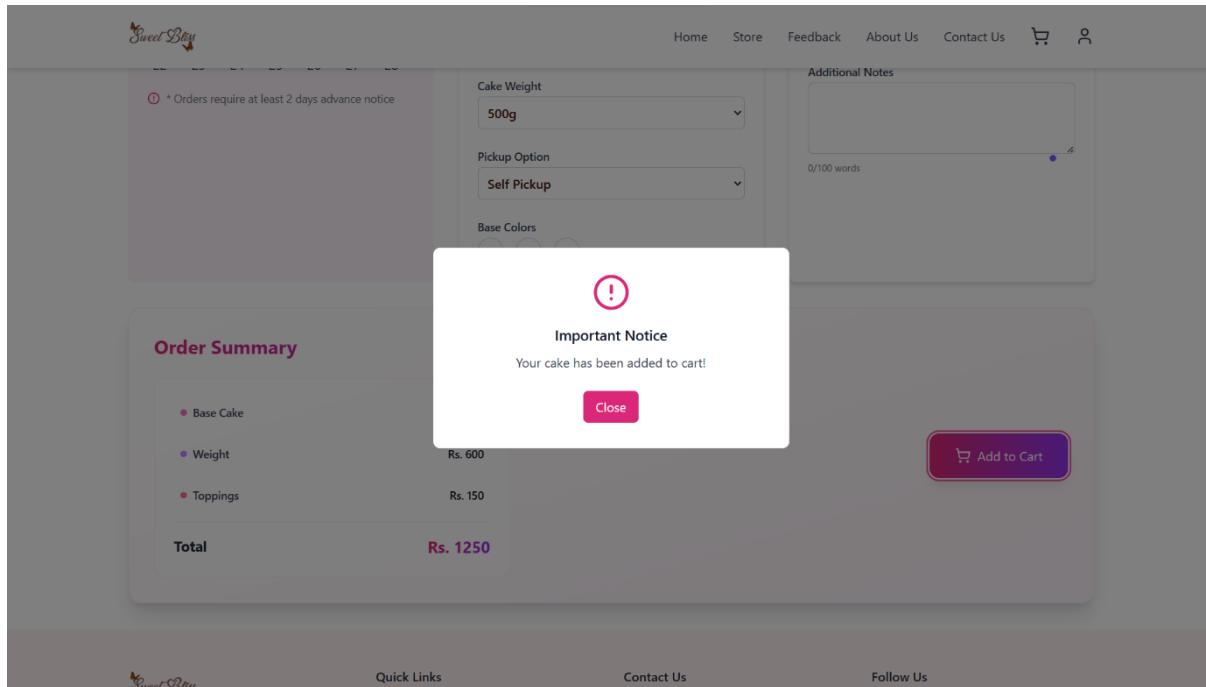


Figure 21 - Order Confirmation Notification

Your Shopping Cart

Custom Cakes

Valentine Event Cake
Size: 1kg
Delivery: Self Pickup

Rs. 1350

Store Items

Balloons
- 6 +

Rs. 450

Order Summary

Enter promo code

Subtotal **Rs. 1860.00**

Total **Rs. 1860.00**

Proceed to Checkout →

Figure 22 - Cart Page (Containing Custom Cake)

My Orders

Order Details	Items	Total	Status	Actions
Order #6e72c2 May 15, 2025, 04:10 PM	Custom Butter Cake Qty: 1 x Rs. 1350 Balloons Qty: 6 x Rs. 450 Party Hats Qty: 4 x Rs. 60	Rs. 1860	Pending	
Order #6b6beb Apr 28, 2025, 08:16 AM	Custom Ribbon Cake Qty: 1 x Rs. 1470 Capybara Plushie Qty: 2 x Rs. 900	Rs. 2370	Pending	
Order #33f25d Apr 28, 2025, 04:15 AM	Custom Butter Cake Qty: 1 x Rs. 1870	Rs. 1870	Completed	

Figure 23 - Customer Orders Page (Updated with Custom Cakes)

Customer Orders					
Start Date	End Date			Export to Excel	Export to PDF
CUSTOMER NAME	ORDER ITEMS	ORDER PRICE	STATUS	CREATED AT	ACTIONS
dineth@example.com	 Custom Ribbon Cake Qty: 1 × \$1470  Capybara Plushie Qty: 4 × \$900	Rs. 2370	🕒 ● Pending	Apr 28, 2025, 08:16 AM	 
kumuthu@example.com	 Capybara Plushie Qty: 11 × \$4950  Balloons Qty: 7 × \$525  Party Hats Qty: 4 × \$60  Confetti Cannon Qty: 2 × \$500	Rs. 6035	🕒 ● Pending	Apr 28, 2025, 09:05 AM	 
risini@example.com	 Capybara Plushie small Qty: 2 × \$9000	Rs. 9000	🕒 ● Pending	Apr 28, 2025, 09:43 AM	 
kumuthu@example.com	 Capybara Plushie small Qty: 2 × \$9000	Rs. 9000	🕒 ● Pending	Apr 28, 2025, 10:04 AM	 
kumuthu@example.com	 Capybara Plushie small Qty: 2 × \$9000	Rs. 9000	🕒 ● Pending	Apr 28, 2025, 10:05 AM	 
dineth@example.com	 Custom Butter Cake Qty: 1 × \$1870	Rs. 1870	🕒 ● Completed	Apr 28, 2025, 04:15 AM	 
kumuthu@example.com	 Confetti Cannon Qty: 3 × \$750	Rs. 750	🕒 ● Completed	Apr 28, 2025, 07:52 AM	 
kumuthu@example.com	 Balloons Qty: 5 × \$375	Rs. 375	🕒 ● Completed	Apr 28, 2025, 07:58 AM	 
kumuthu@example.com	 Party Hats Qty: 4 × \$60	Rs. 60	🕒 ● Completed	Apr 28, 2025, 09:06 AM	 
kumuthu@example.com	 Balloons Qty: 3 × \$325	Rs. 225	🕒 ● Completed	Apr 28, 2025, 09:40 AM	 
kumuthu@example.com	 Capybara Plushie small Qty: 2 × \$9000	Rs. 9000	🕒 ● Completed	Apr 28, 2025, 09:51 AM	 
kumuthu@example.com	 Capybara Plushie small Qty: 2 × \$9000	Rs. 9000	🕒 ● Completed	Apr 28, 2025, 09:55 AM	 

Figure 25 - Order Management Section (Admin)

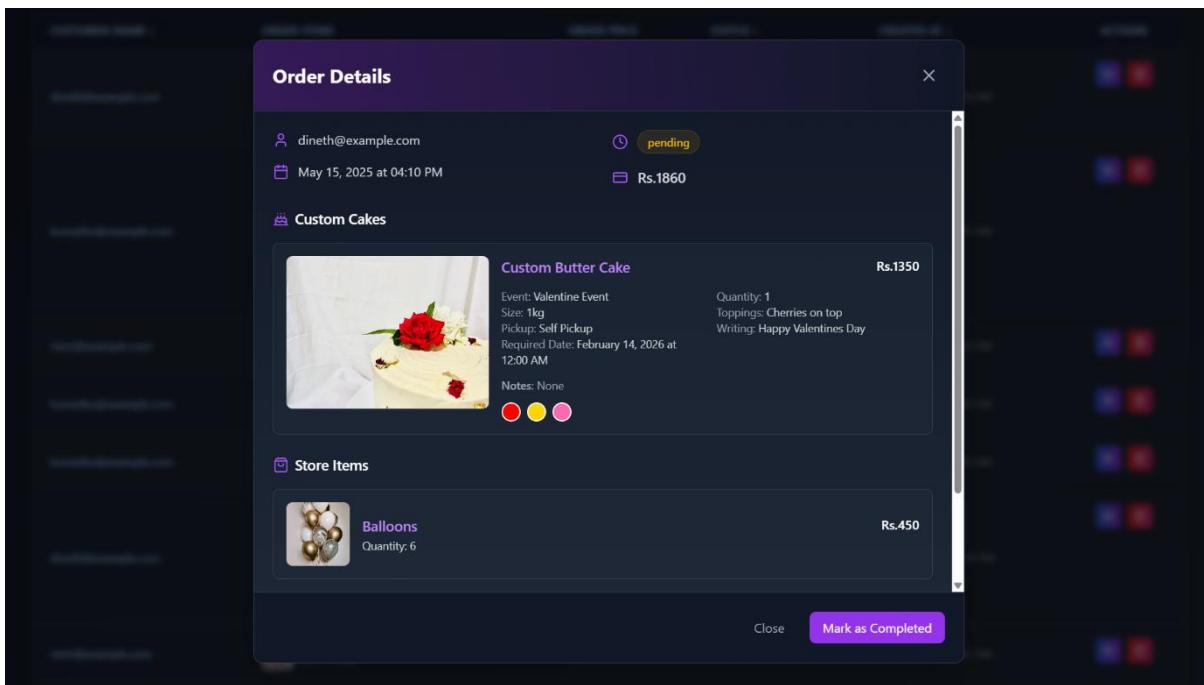


Figure 24 - Order Details View

sweet_bliss_orders (1).pdf

Sweet Bliss - Order Report

Generated on: 5/15/2025, 4:18:15 PM

Customer Email	Order Items	Total Price	Status	Created At
dineth@example.com	Custom Butter Cake, Balloons, Party Hats	Rs.1860	completed	May 15, 2025, 04:10 PM
kumuthu@example.com	Capybara Plushie small	Rs.9000	completed	Apr 28, 2025, 10:05 AM
kumuthu@example.com	Capybara Plushie small	Rs.9000	completed	Apr 28, 2025, 10:04 AM
kumuthu@example.com	Capybara Plushie small	Rs.9000	completed	Apr 28, 2025, 09:55 AM
kumuthu@example.com	Capybara Plushie small	Rs.9000	completed	Apr 28, 2025, 09:51 AM
risiniq@example.com	Capybara Plushie small	Rs.9000	completed	Apr 28, 2025, 09:43 AM
kumuthu@example.com	Balloons	Rs.225	completed	Apr 28, 2025, 09:40 AM
kumuthu@example.com	Party Hats	Rs.60	completed	Apr 28, 2025, 09:06 AM
kumuthu@example.com	Capybara Plushie, Balloons, Party Hats, Confetti Cannon	Rs.6035	completed	Apr 28, 2025, 09:05 AM
dineth@example.com	Custom Ribbon Cake, Capybara Plushie	Rs.2370	completed	Apr 28, 2025, 08:16 AM
kumuthu@example.com	Balloons	Rs.375	completed	Apr 28, 2025, 07:58 AM
kumuthu@example.com	Confetti Cannon	Rs.750	completed	Apr 28, 2025, 07:52 AM
dineth@example.com	Custom Butter Cake	Rs.1870	completed	Apr 28, 2025, 04:15 AM
risiniq@example.com	Confetti Cannon	Rs.500	completed	Apr 28, 2025, 03:15 AM
risiniq@example.com	Happy Birthday Hanging Decor	Rs.200	completed	Apr 28, 2025, 03:09 AM

Figure 26 - Order Management Related Report Generation[1] (Admin)

sweet_bliss_orders (1).pdf

Sweet Bliss - Order Report

Generated on: 5/15/2025, 4:18:15 PM

Customer Email	Order Items	Total Price	Status	Created At
kumuthu@example.com	Capybara Plushie small	Rs.9000	completed	Apr 28, 2025, 09:55 AM
risiniq@example.com	Capybara Plushie small	Rs.9000	completed	Apr 28, 2025, 09:43 AM
kumuthu@example.com	Balloons	Rs.225	completed	Apr 28, 2025, 09:40 AM
kumuthu@example.com	Party Hats	Rs.60	completed	Apr 28, 2025, 09:06 AM
kumuthu@example.com	Capybara Plushie, Balloons, Party Hats, Confetti Cannon	Rs.6035	completed	Apr 28, 2025, 09:05 AM
dineth@example.com	Custom Ribbon Cake, Capybara Plushie	Rs.2370	completed	Apr 28, 2025, 08:16 AM
kumuthu@example.com	Balloons	Rs.375	completed	Apr 28, 2025, 07:58 AM
kumuthu@example.com	Confetti Cannon	Rs.750	completed	Apr 28, 2025, 07:52 AM
dineth@example.com	Custom Butter Cake	Rs.1870	completed	Apr 28, 2025, 04:15 AM
risiniq@example.com	Confetti Cannon	Rs.500	completed	Apr 28, 2025, 03:15 AM
risiniq@example.com	Happy Birthday Hanging Decor	Rs.200	completed	Apr 28, 2025, 03:09 AM
risiniq@example.com	Birthday Cards	Rs.200	completed	Apr 28, 2025, 02:54 AM
risiniq@example.com	Confetti Cannon	Rs.500	completed	Apr 28, 2025, 02:41 AM

Order Summary

Total Orders: 17	Completed Orders: 17
Total Revenue: Rs.59945.00	Pending Orders: 0

Page 1 Generated by Sweet Bliss Admin System

Figure 27 - Order Management related Report Generation[2] (Admin)

3.11.4. Inventory Management related User Interfaces

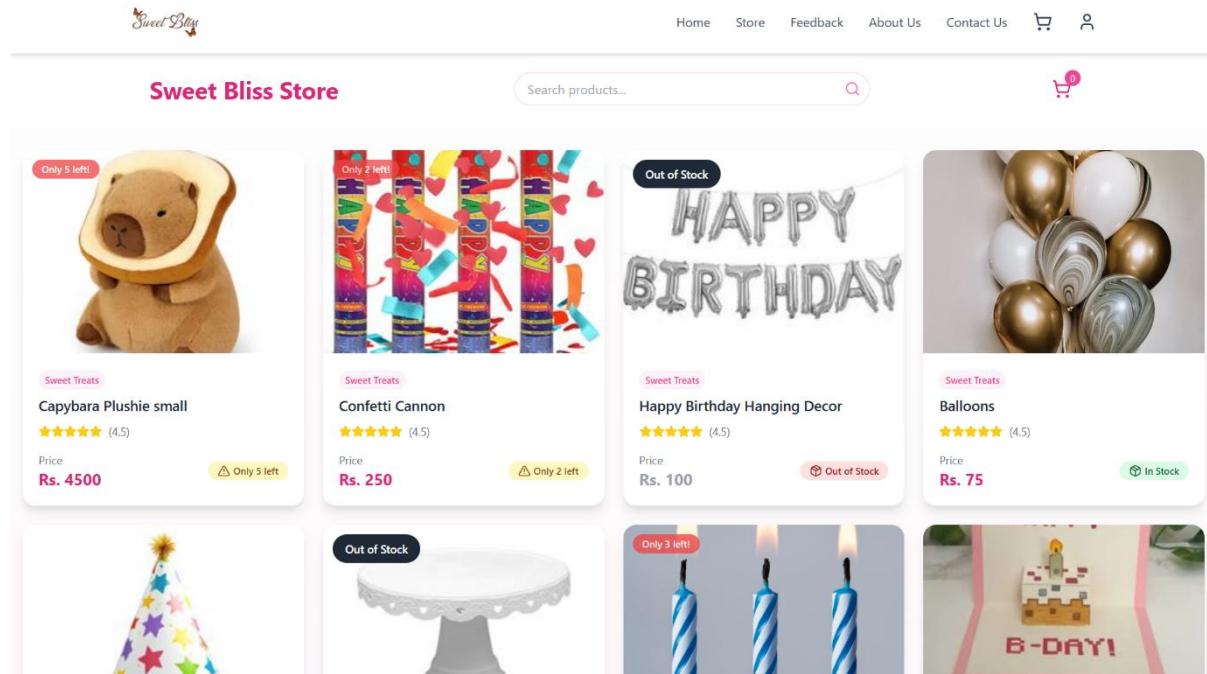


Figure 28 - Sweet Bliss Store Page

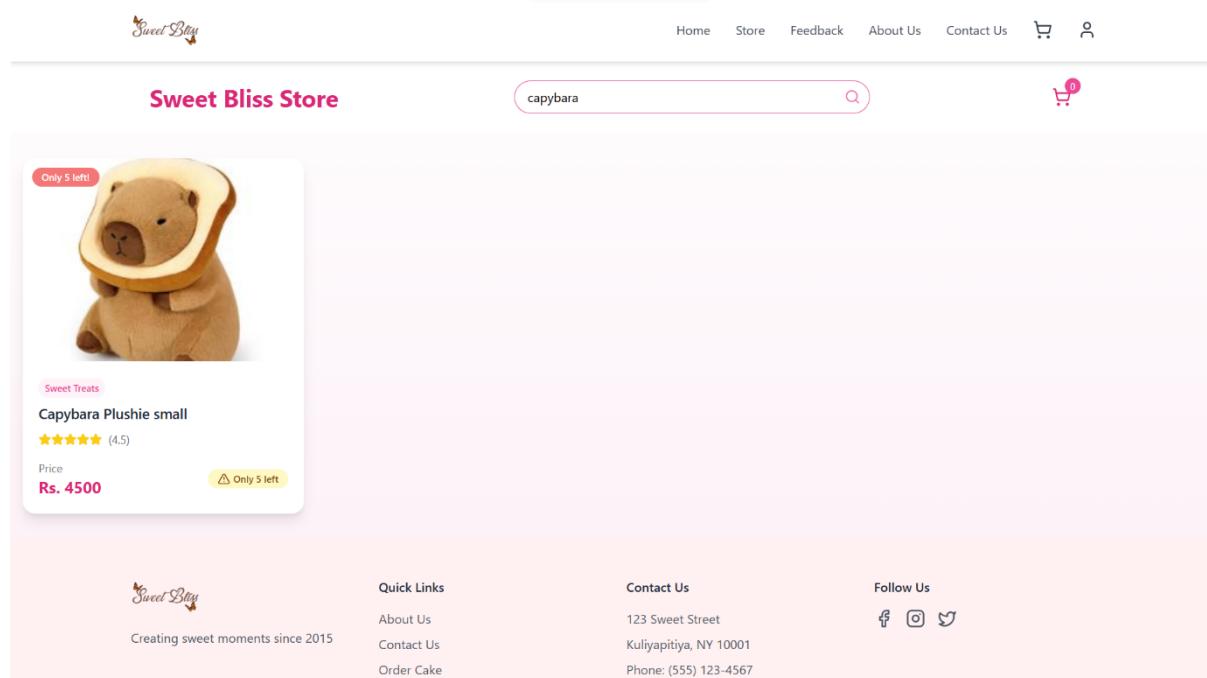


Figure 29 - Search on Specific Item in the Store Page

Your Shopping Cart

Store Items

No store items in cart

Browse Store →

Order Summary

Enter promo code **Apply**

Subtotal **Rs. 0.00**

Total **Rs. 0.00**

Proceed to Checkout →

SweetBlossom
Creating sweet moments since 2015

Quick Links
About Us
Contact Us
Order Cake

Contact Us
123 Sweet Street
Kuliyapitiya, NY 10001
Phone: (555) 123-4567

Follow Us
[Facebook](#) [Instagram](#) [Twitter](#)

Figure 30 - Cart Page

Store Items

Balloons
- 6 + **Rs. 450** **Remove**

Party Hats
- 4 + **Rs. 60** **Remove**

Order Summary

Enter promo code **Apply**

Subtotal **Rs. 1860.00**

Total **Rs. 1860.00**

Proceed to Checkout →

SweetBlossom
Creating sweet moments since 2015

Quick Links
About Us
Contact Us
Order Cake

Contact Us
123 Sweet Street
Kuliyapitiya, NY 10001
Phone: (555) 123-4567

Follow Us
[Facebook](#) [Instagram](#) [Twitter](#)

Figure 31 - Cart Page (Containing Store Items)

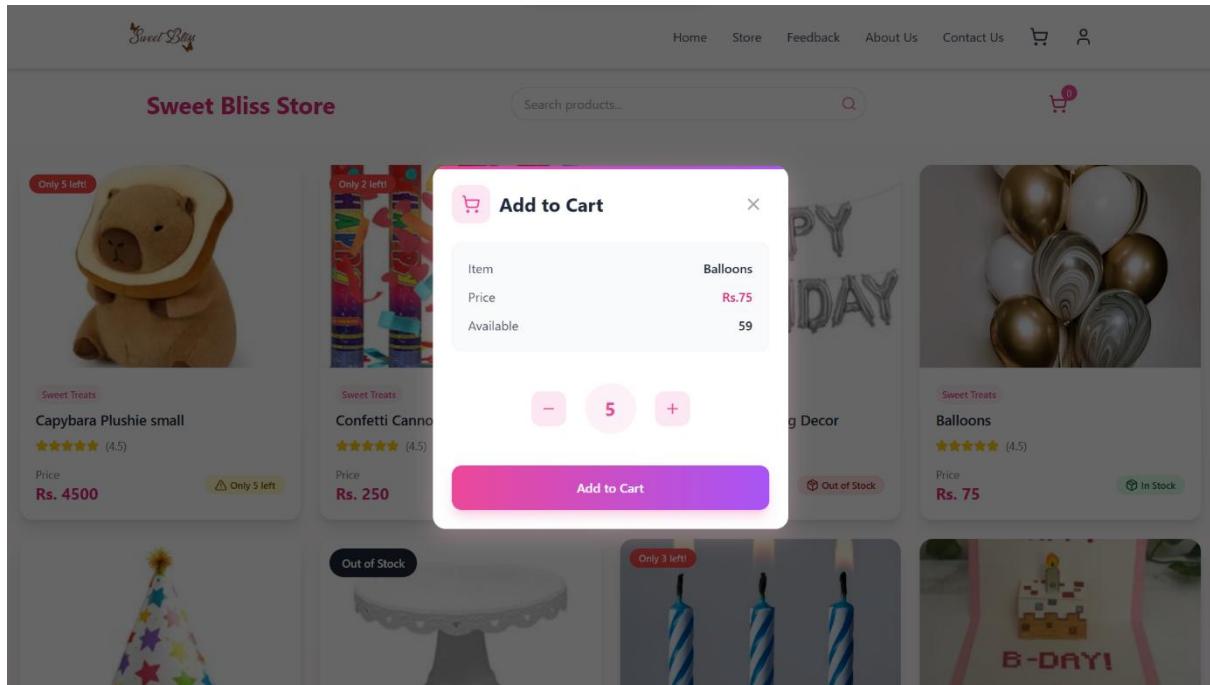


Figure 32 - Add to Cart Dialog View

Inventory Management					+ Add New Item
Product Name	Image	Price	Stock Level	Actions	
Capybara Plushie small		Rs.4500	5 ▲		
Confetti Cannon		Rs.250	2 ▲		
Happy Birthday Hanging Decor		Rs.100	0 ▲		
Balloons		Rs.75	54		
Party Hats		Rs.15	41		
Cake Trays		Rs.400	0 ▲		
Candles		Rs.150	3 ▲		
Birthday Cards		Rs.100	191		
Paper Plates		Rs.250	18		

Figure 33 - Inventory Management Section (Admin)

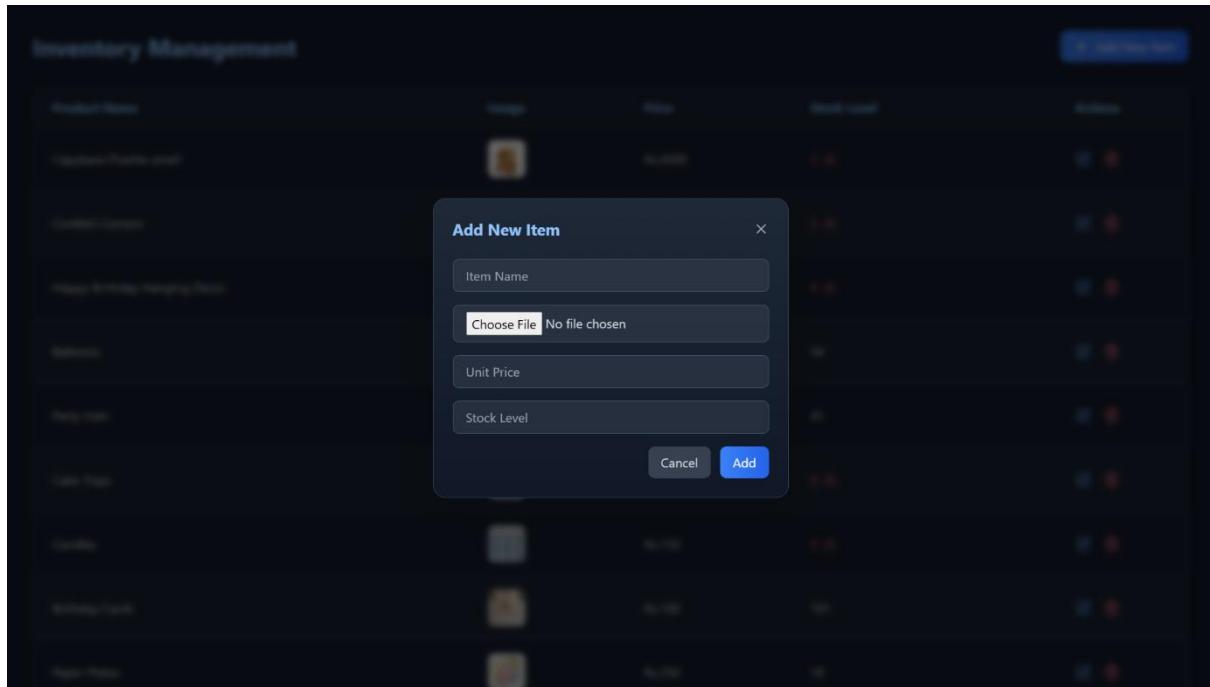


Figure 34 - Adding New Item to the Inventory

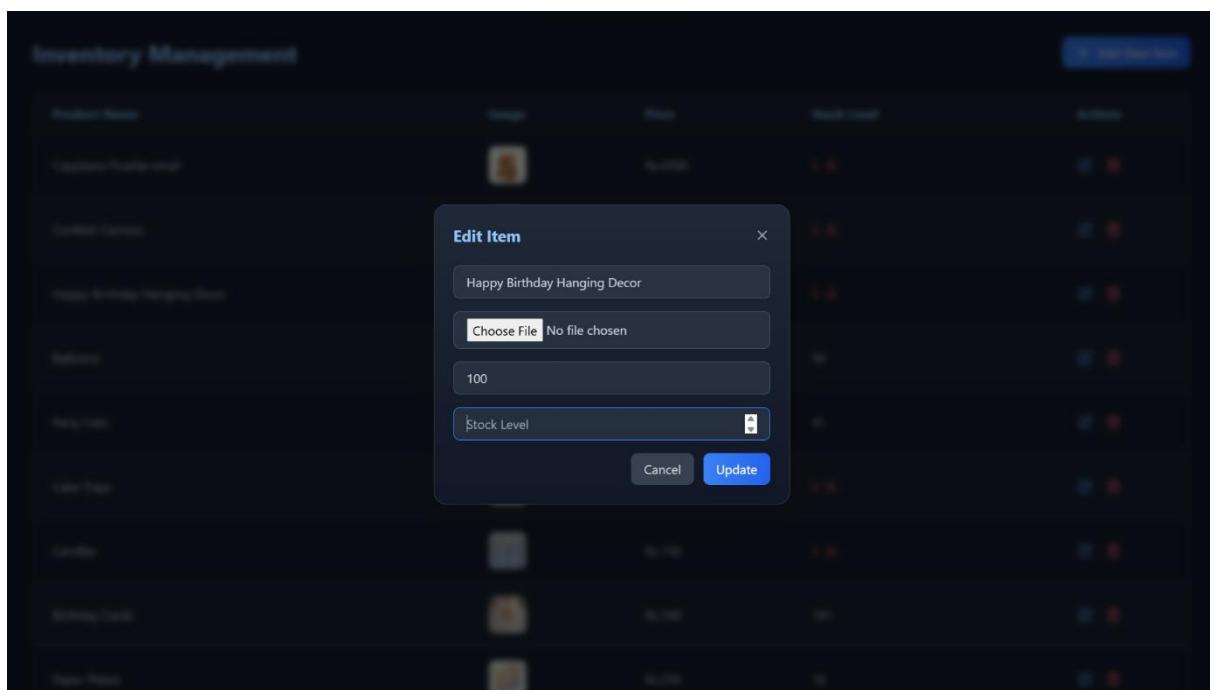


Figure 35 - Managing the Stock Levels and Other Inventory Information

Inventory Management					+ Add New Item
Product Name	Image	Price	Stock Level	Actions	
Capybara Plushie small		Rs.4500	5 ▲		
Confetti Cannon		Rs.250	2 ▲		
Happy Birthday Hanging Decor		Rs.100	75		
Balloons		Rs.75	54		
Party Hats		Rs.15	41		
Cake Trays		Rs.400	0 ▲		
Candles		Rs.150	3 ▲		
Birthday Cards		Rs.100	191		
Paper Plates		Rs.250	18		

Figure 36 - Inventory Items with Updated Stock Levels

3.11.5. Finance Management related User Interfaces

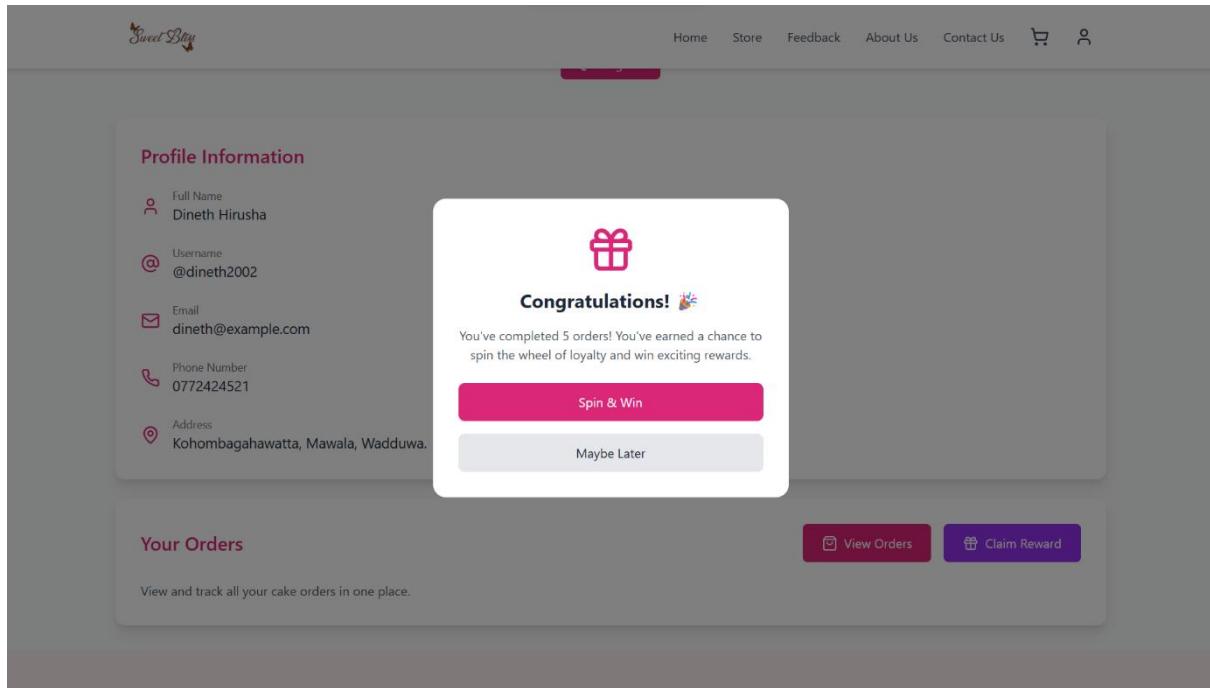


Figure 37 - Offer Dialog View after 5 Completed Orders

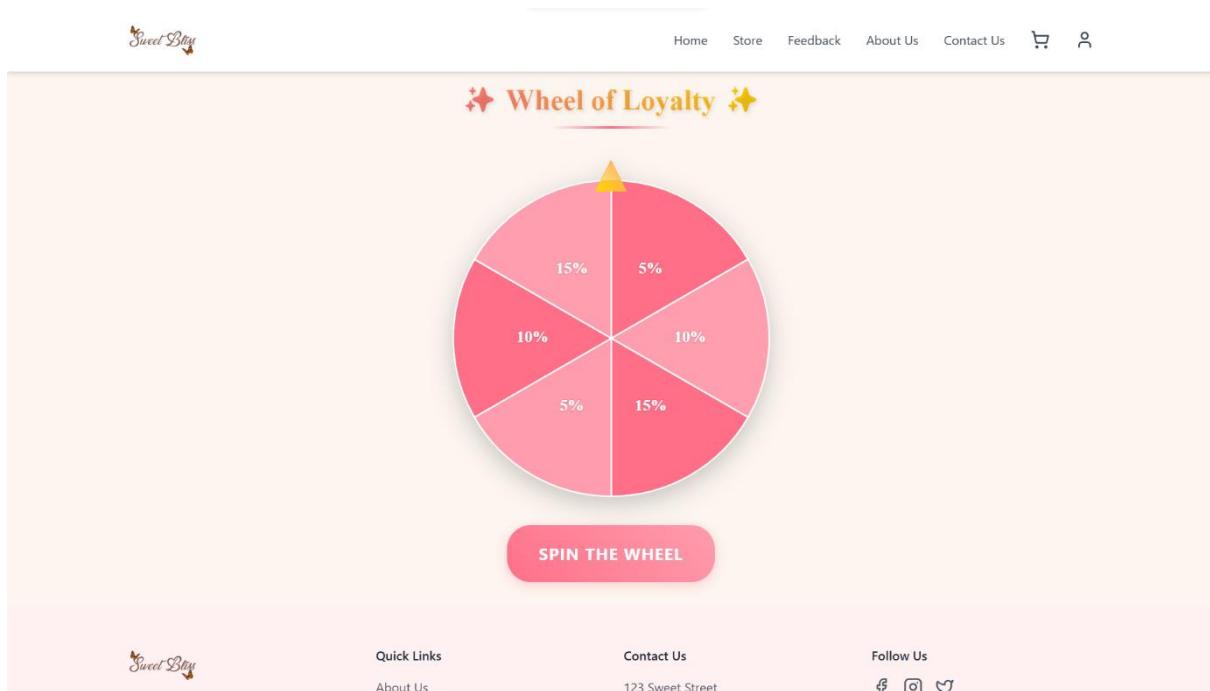


Figure 38 - The Wheel of Loyalty Page

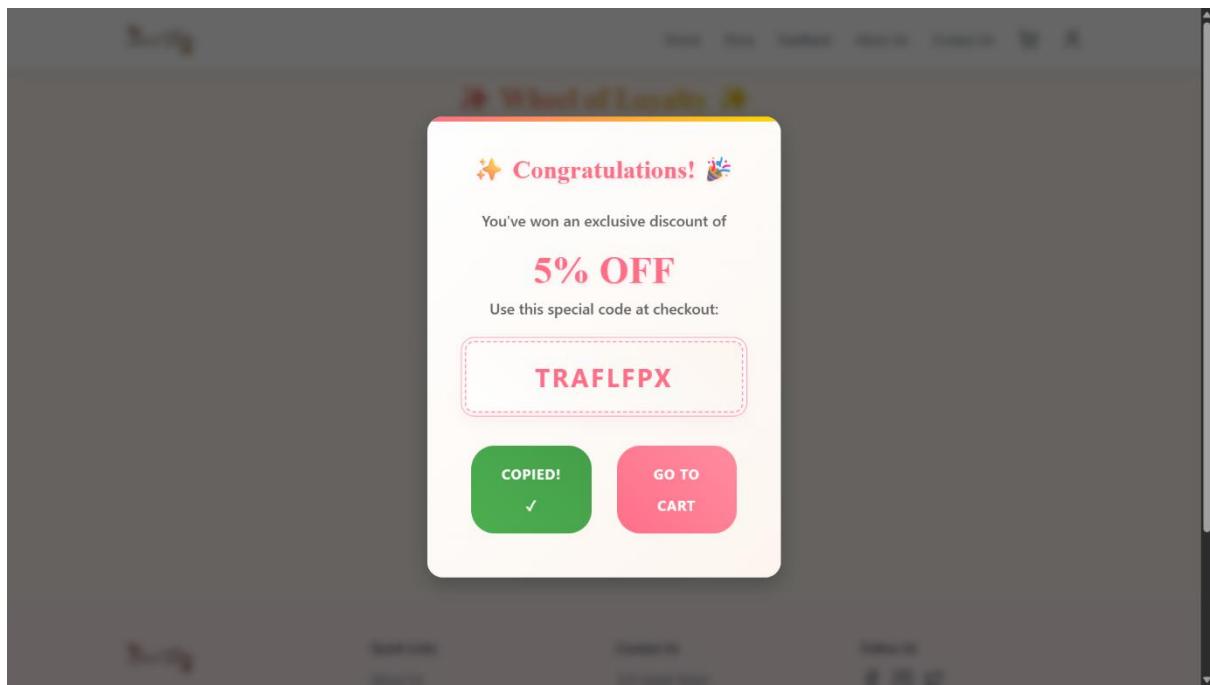


Figure 39 - Discount Code Generated Dialog View

Figure 40 - Application of Discount Code

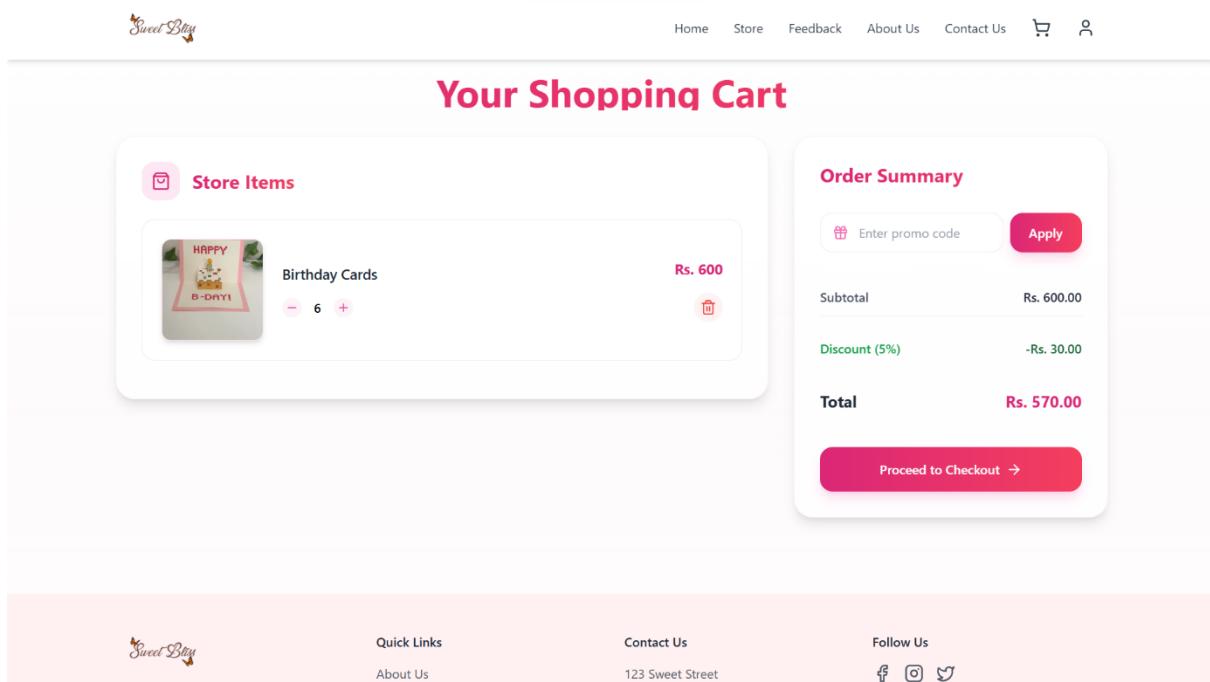


Figure 41 - Granted Discount Application

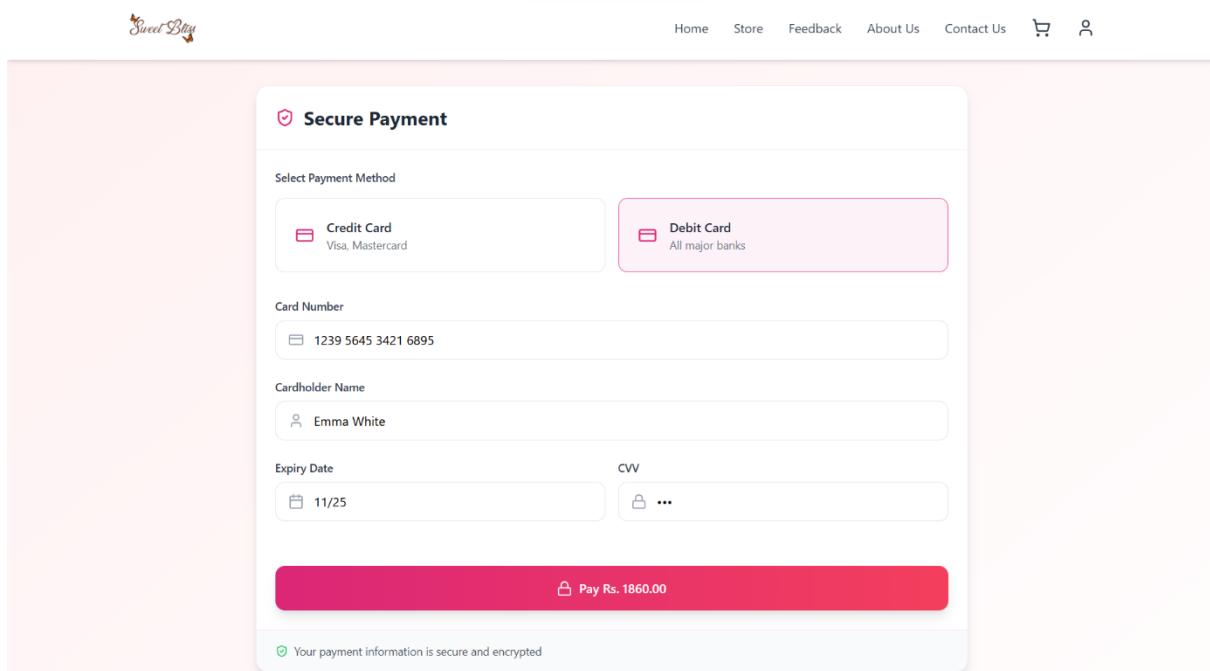


Figure 42 - Payment Gateway

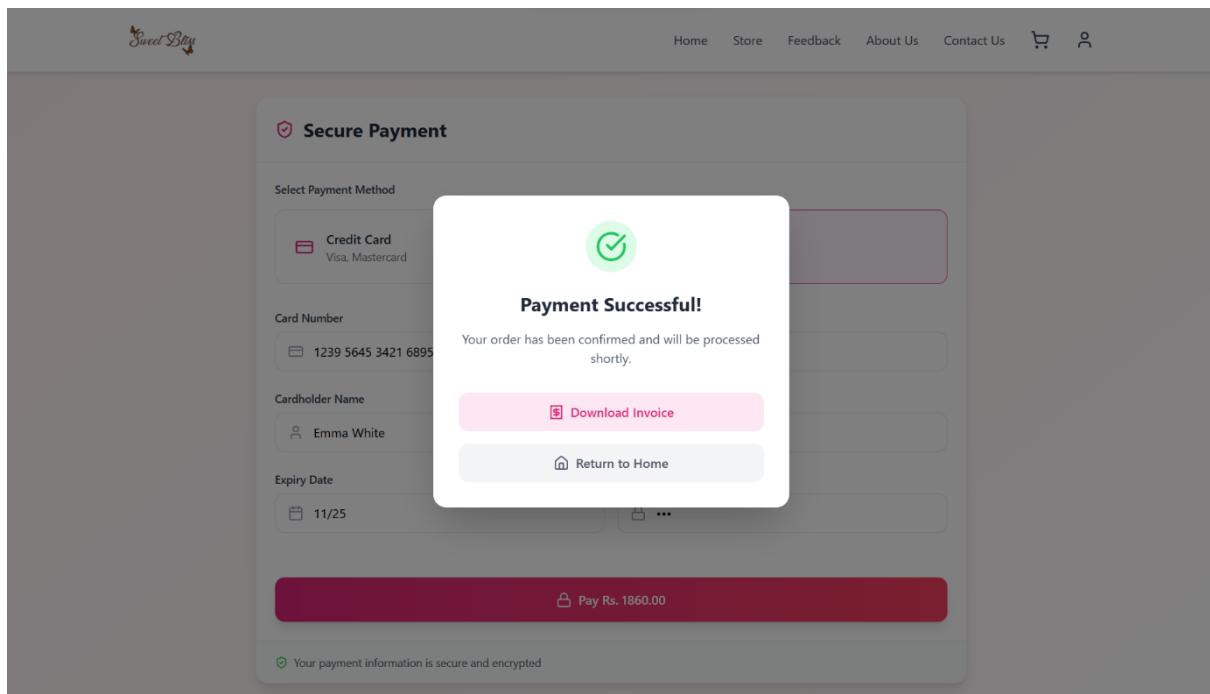


Figure 43 - Payment Successful Dialog View

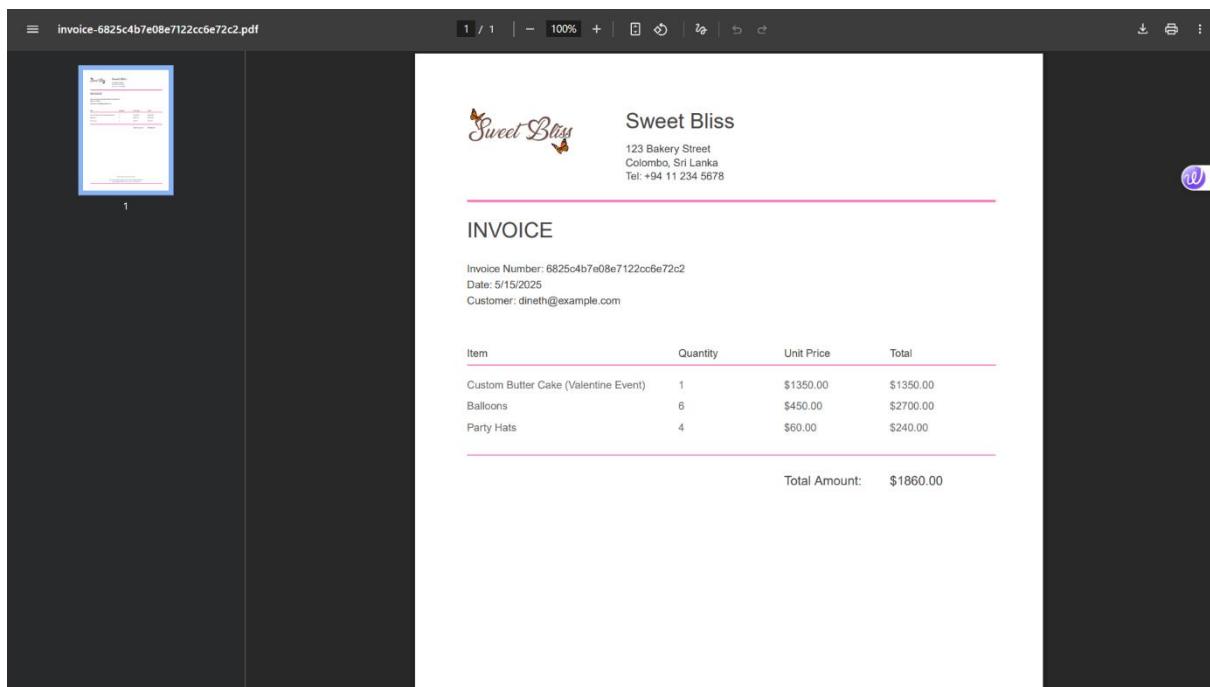


Figure 44 - Payment Invoice

Cake Orders Management							
Manage and track custom cake orders							
Customer	Event Type	Cake Type	Size	Price	Required Date	Pickup Option	Actions
dineth@example.com	Valentine Event	Butter Cake	1kg	LKR 1350	Feb 14, 2026, 12:00 AM	Self Pickup	 
dineth@example.com	Valentine Event	Butter Cake	1kg	LKR 1350	Feb 14, 2026, 12:00 AM	Self Pickup	 
dineth@example.com	Valentine Event	Butter Cake	1kg	LKR 1350	Feb 14, 2026, 12:00 AM	Self Pickup	 
dineth@example.com	Birthday Event	Butter Cake	1kg	LKR 1350	May 2, 2025, 12:00 AM	Self Pickup	 
dineth@example.com	Wedding	Ribbon Cake	1kg	LKR 1470	May 20, 2025, 12:00 AM	Self Pickup	 
dineth@example.com	Birthday Event	Butter Cake	1kg	LKR 1870	May 3, 2025, 12:00 AM	Delivery	 
risini@example.com	Birthday Party	Chocolate Cake	1kg	LKR 1350	May 19, 2025, 12:00 AM	Self Pickup	 
dineth@example.com	New Year Event	Butter Cake	500g	LKR 1750	May 6, 2025, 12:00 AM	Delivery	 
dineth@example.com	Hajji Event	Butter Cake	2kg	LKR 1400	Jun 3, 2025, 12:00 AM	Self Pickup	 
dineth@example.com	Birthday Event	Ribbon Cake	1kg	LKR 1470	May 15, 2025, 12:00 AM	Self Pickup	 

Figure 45 - Cake Orders View (Admin)

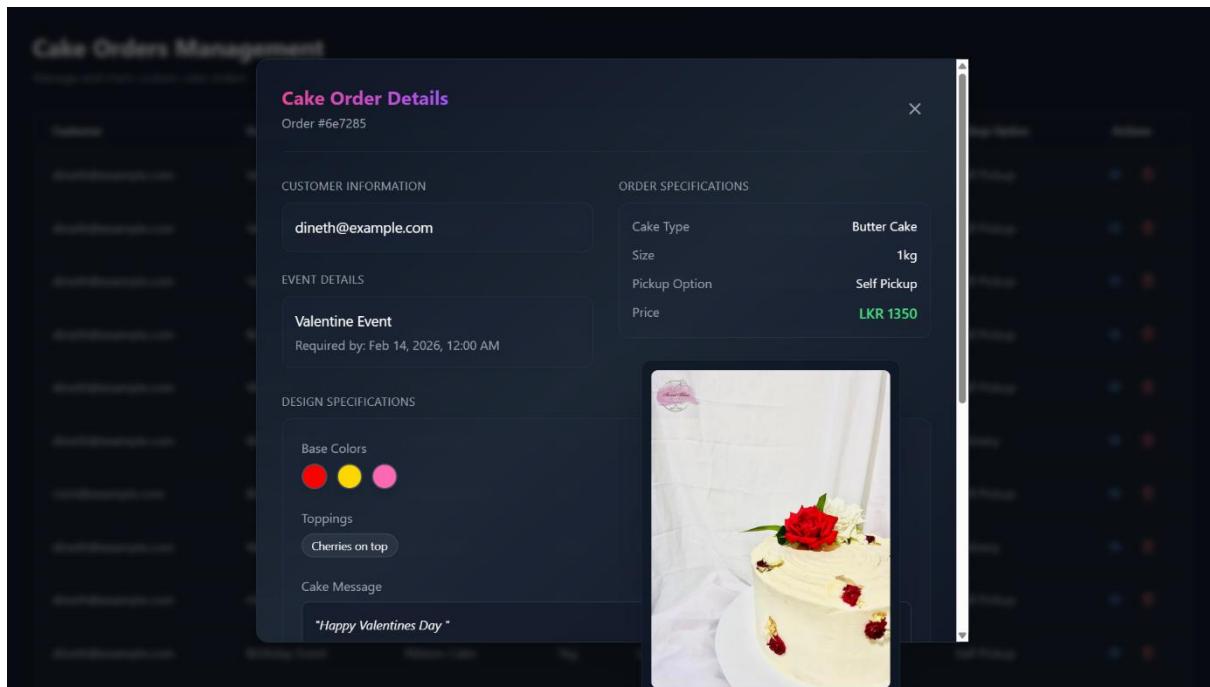


Figure 46 - Detailed View of Cake Orders Payment (Admin)

3.11.6. Feedback Management related User Interfaces

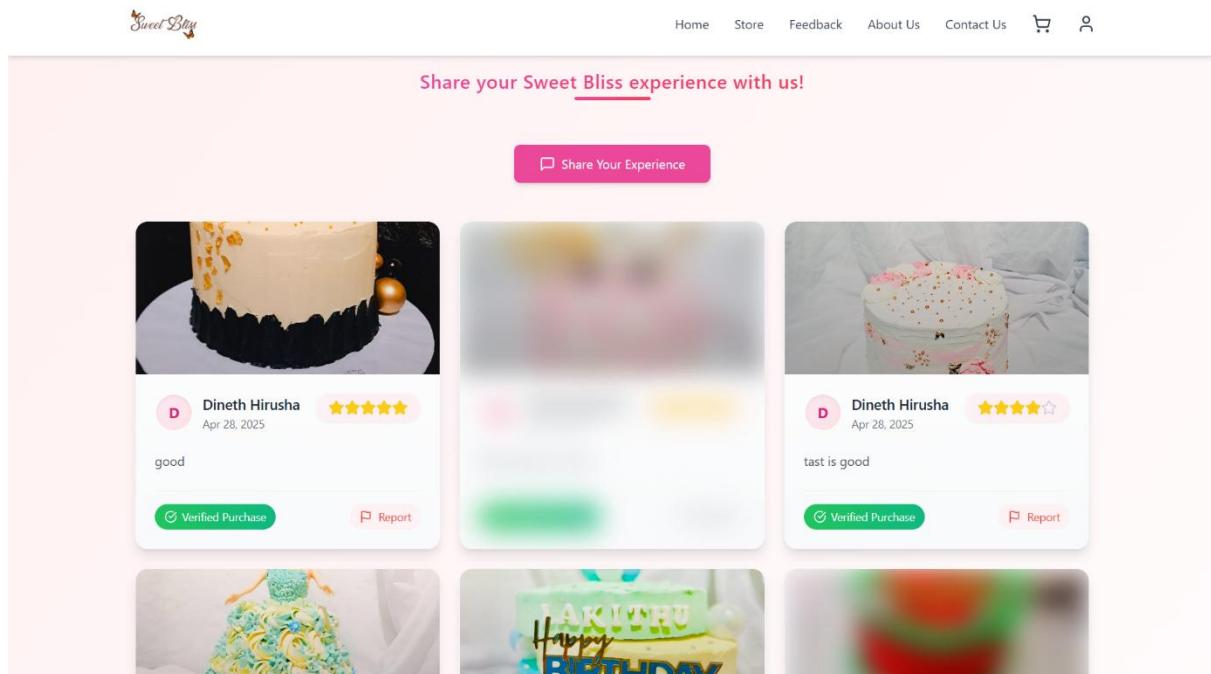


Figure 47 - Feedback Page

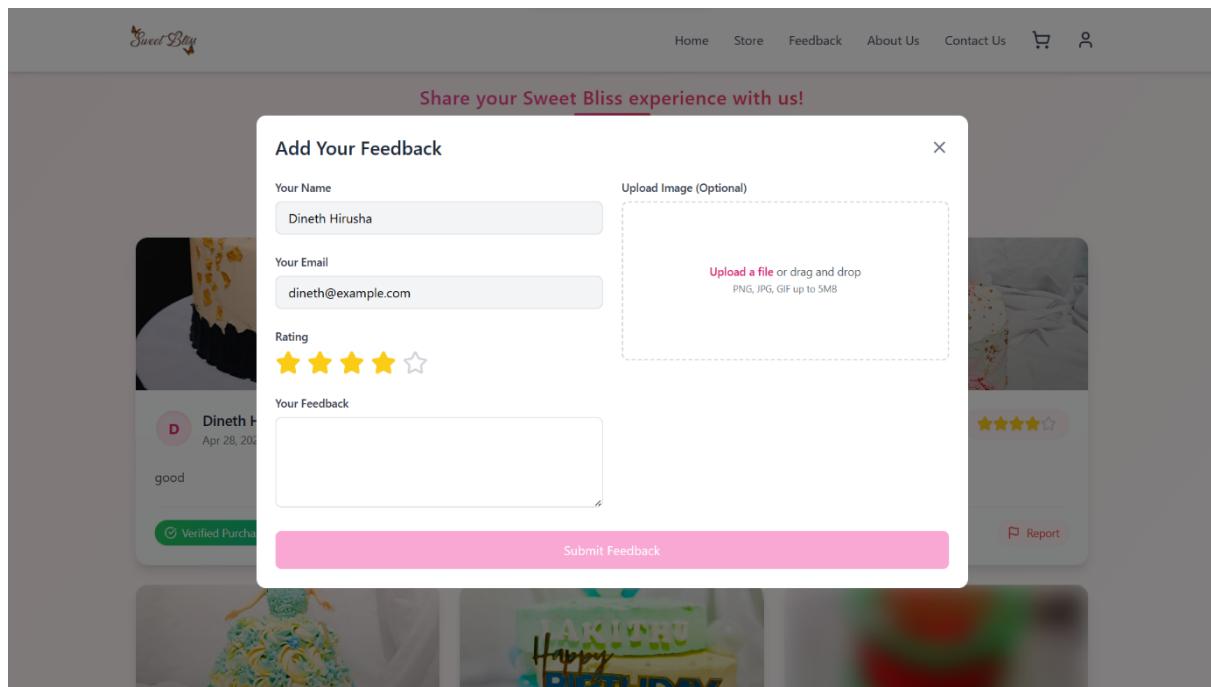


Figure 48 - Add New Feedback Dialog

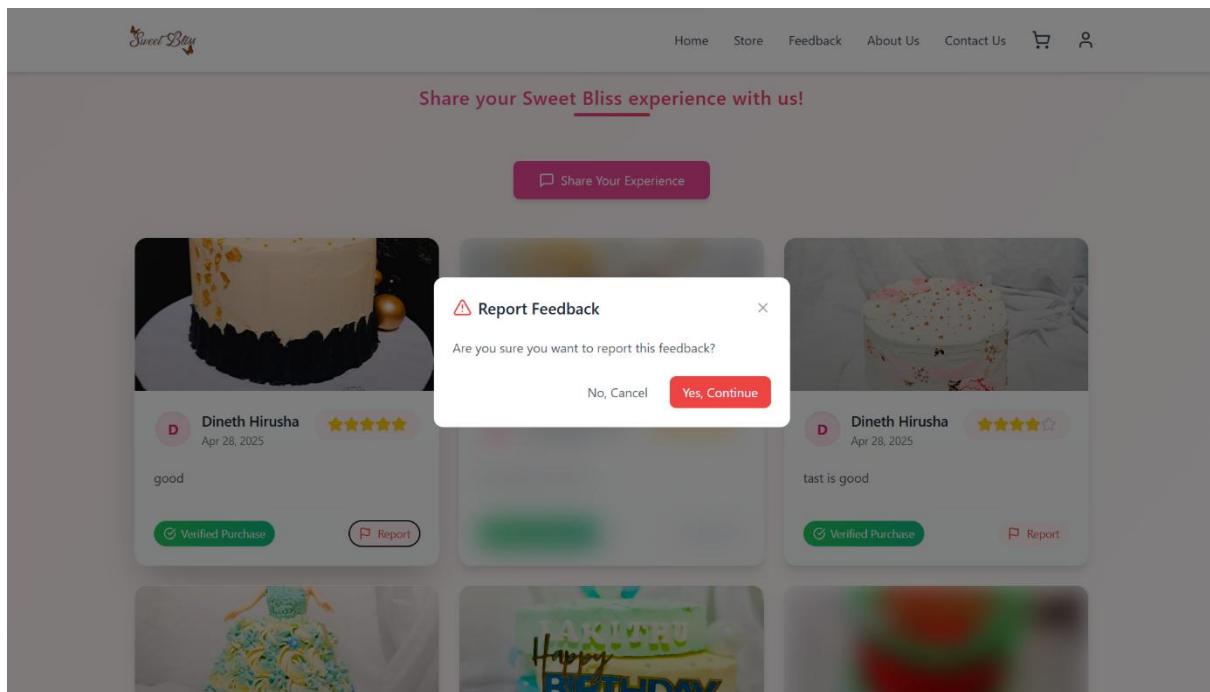


Figure 49 - Report Feedback Dialog View

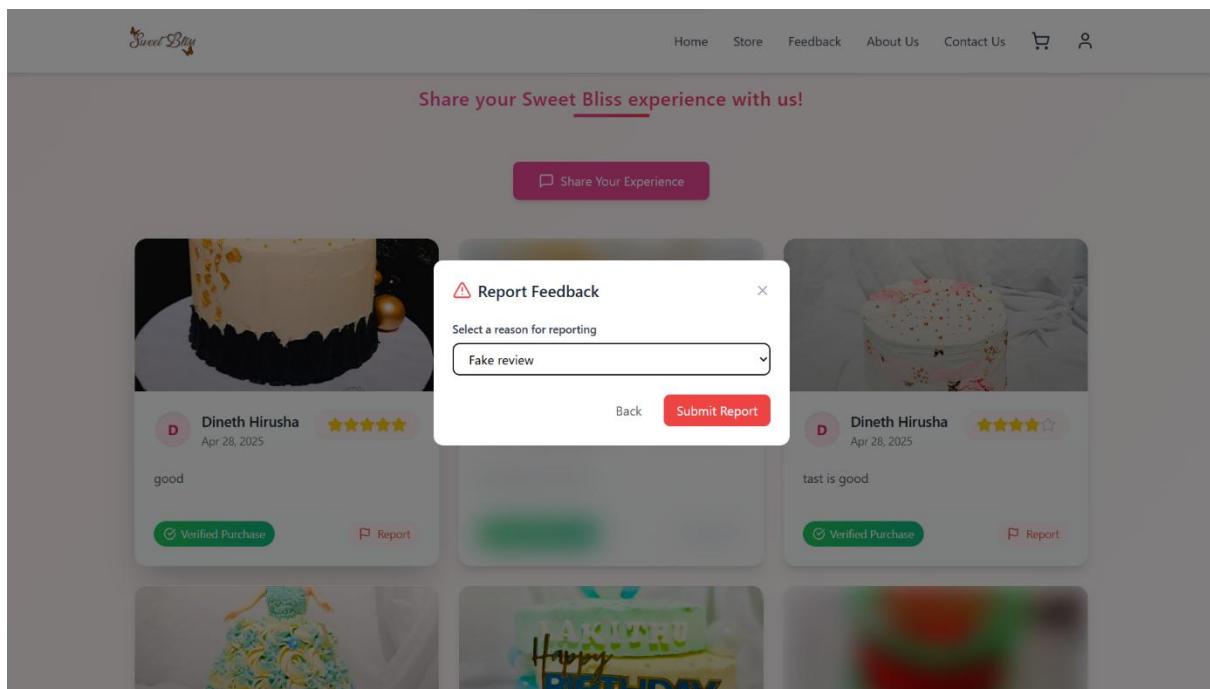


Figure 50 - Adding Reason of Reporting

User Email	Feedback Image	Reported	Description	Created At	Actions
dineth@example.com		● Reported	good	Apr 28, 2025	
dineth@example.com		● Reported	best product.i love it	Apr 28, 2025	
dineth@example.com		● Safe	tast is good	Apr 28, 2025	
dineth@example.com		● Safe	tast is good and fresh.i love this cake design	Apr 28, 2025	
dineth@example.com		● Safe	Ft Ft	Apr 27, 2025	
locha@mail.com		● Safe	Cupiri	Apr 27, 2025	

Figure 51 - Feedback Management Section (Admin)

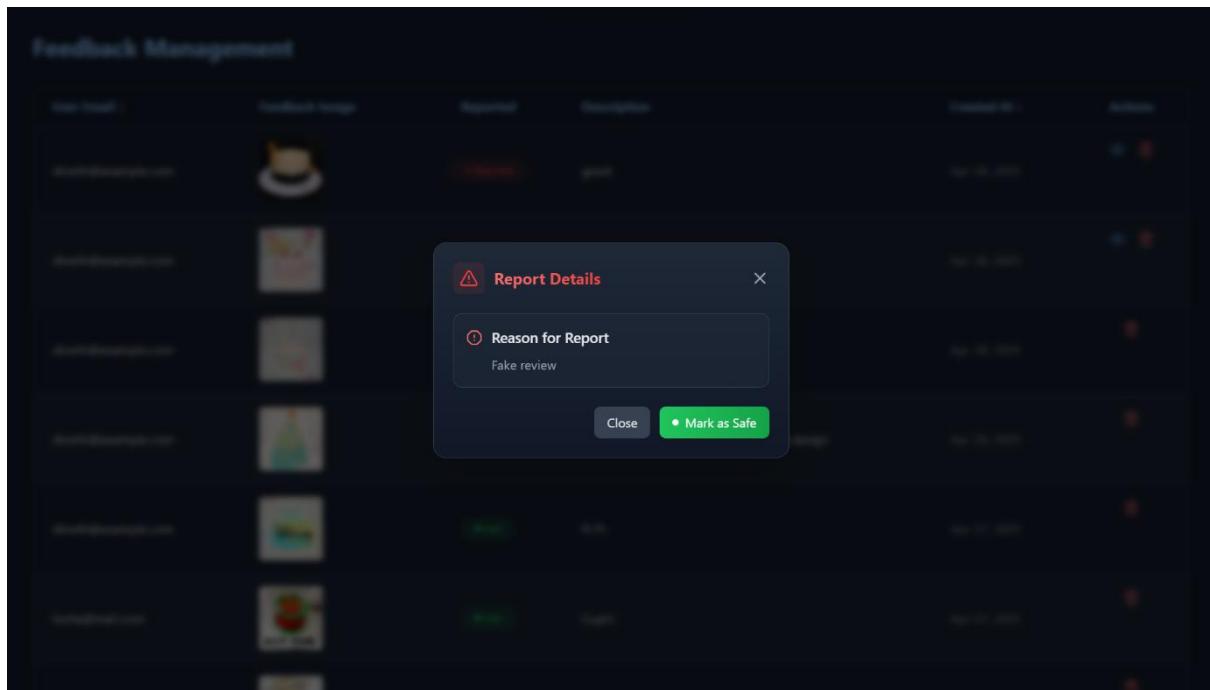


Figure 52 - Action Dialog View (Admin)

User Email	Feedback Image	Reported	Description	Created At	Actions
dineth@example.com		Reported	good	Apr 28, 2025	
dineth@example.com		Safe	best product.i love it	Apr 28, 2025	
dineth@example.com		Safe	tast is good	Apr 28, 2025	
dineth@example.com		Safe	tast is good and fresh.i love this cake design	Apr 28, 2025	
dineth@example.com		Safe	Ft Ft	Apr 27, 2025	
locha@mail.com		Safe	Cupiri	Apr 27, 2025	

Figure 53 - Resolved Feedbacks after Taking Action

Chapter 4

Testing

These test case scenarios and test cases are designed to test and validate both functional and non-functional requirements of each component with automated and manual testing. For the automated testing Postman was used.

4.1. Test case scenarios

Test Case ID	ITP25_B6_121_TC01
Project ID	ITP25_B6_121
Project Name	Sweet Bliss – Custom Cake Ordering Management System
Testing Function	Custom Cake Order Form Submission
Test case designed and executed by	Reg No.: IT23183018 Name: HIRUSHA D G A D
Test Priority	High

Test Case ID	ITP25_B6_121_TC03
Project ID	ITP25_B6_121
Project Name	Sweet Bliss – Store, Cart and Inventory Management System
Testing Function	Adding, Updating, Deleting and notification regarding low stocks.
Test case designed and executed by	Reg No.: IT23396272 Name: Kariyawasam L L N
Test Priority	High

Test Case ID	ITP25_B6_121_TC02
Project ID	ITP25_B6_121
Project Name	Sweet Bliss – Store, Cart and Inventory Management System
Testing Function	Adding Items to the cart and updating quantity
Test case designed and executed by	Reg No.: IT23396272 Name: Kariyawasam L L N
Test Priority	High

Test Case ID	ITP25_B6_121_TC04
Project ID	ITP25_B6_121
Project Name	Sweet Bliss – User Management System
Testing Function	User Management
Test case designed and executed by	Reg No.: IT23179158 Name: Budra H M K
Test Priority	High

Test Case ID	ITP25_B6_121_TC05
Project ID	ITP25_B6_121
Project Name	Sweet Bliss – Order,Payment Management System
Testing Function	Order Payment Submission

Test case designed and executed by	Reg No.: IT23222786 Name: Budara V.P.R.
Test Priority	High

Test Case ID	ITP25_B6_121_TC06
Project ID	ITP25_B6_121
Project Name	Sweet Bliss – Feedback Management System
Test case designed and executed by	Reg No.: IT23222236 Name: Erathnage TN
Test Priority	Medium

4.2. Test case Results

Custom Ordering

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comment
Test_001a	Valid order details including cake type, size, colors, pickup option, toppings, writing on top, date, and additional notes.	The order is successfully created and added to the cart.	The order is created without errors.	Pass	The system correctly processes valid inputs and adds the order to the cart.
Test_001b	Missing required fields (e.g., cake type or size not selected).	The system should prompt the user to complete all required fields.	The system displays an error message indicating missing information.	Pass	Ensures validation prevents submission of incomplete orders.
Test_001c	Selecting a date within 2 days of the current date.	The system should display a pop-up: " The date should be at least 2 days after this date. "	The system correctly prevents selection and shows the message.	Pass	Ensures that users cannot place urgent last-minute orders.
Test_001d	Selecting a past date.	The system should display a pop-up: " Invalid user input. "	The system correctly prevents selection and shows the message.	Pass	Verifies that users cannot select a date that has already passed.
Test_001e	Selecting "I have one already" for design requirement.	AI-generated images should be displayed in a carousel.	AI images section appears but shows placeholder images as AI generation is unavailable.	Pass	Ensures that the UI dynamically updates based on user input.
Test_001f	Selecting "Yes I do" for design requirement.	File upload option should be enabled.	File upload option appears correctly.	Pass	Ensures users can upload custom cake designs.

Test_001g	Selecting different options for cake type, size, toppings, and pickup method.	The final price should update dynamically.	The final price updates based on selections.	Pass	Ensures correct price calculation logic.
Test_001h	Clicking the "Add to Cart" button after filling all fields correctly.	The order should be added to the cart successfully.	The order appears in the cart with correct details.	Pass	Confirms successful order placement.

Store and Inventory

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comment
Test_002a	Selects available items.	The item has added successfully	The item was added without errors.	Pass	The system correctly processes valid inputs and adds the item to the cart.
Test_002b	Selects an unavailable item/s.	The system should prompt an error message.	The system displays the correct error message.	Pass	Ensures validation prevents adding items which are unavailable.
Test_002c	Updating quantity of an item.	The system should let the user to update the quantity.	The system correctly updates the quantity one by one.	Pass	Ensures that users can update the quantity.

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comment
Test_003a	Add a new item to inventory	The item should be added successfully	The item was added successfully	Pass	The system correctly adds items to inventory
Test_003b	Update item details	The item information should be updated successfully	The system updates item details correctly	Pass	Users can modify item details as required
Test_003c	Check for low stock notification	The system should notify when stock is below threshold	The system flagged the item as low stock	Pass	Ensures admins are alerted for low stock
Test_003d	Delete an item from inventory	The system should prompt a confirmation before deleting	The system prompts confirmation and deletes the item	Pass	Ensures secure deletion with confirmation prompt

User Management

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comment
Test_004a	Valid user details (username, password, email, phone number, profile picture)	The user account is successfully created.	The account is created without errors.	Pass	The system correctly processes valid inputs and creates the user account.
Test_004b	Missing required fields (e.g., username or password not provided).	The system should prompt the user to complete all required fields.	The system displays an error message indicating missing information.	Pass	Ensures validation prevents account creation with incomplete information.
Test_004c	Username already exists in the system.	The system should display an error message: "Username already taken."	The system correctly prevents account creation and shows the message.	Pass	Verifies that the system prevents duplicate usernames.
Test_004d	Invalid email format (e.g., missing "@" or domain).	The system should display an error message: "Invalid email address."	The system correctly prevents account creation and shows the message.	Pass	Ensures proper email validation.
Test_004e	Selecting "Change Profile Picture" option.	Profile picture upload option should be enabled.	The upload option appears and works as expected.	Pass	Confirms users can upload new profile pictures.
Test_004f	Clicking "Save Changes" after updating user details.	The updated details should be saved successfully.	The system updates the user profile without errors.	Pass	Confirms successful update of user profile details
Test_004g	Changing password with old password verification.	The password should update successfully after old password verification.	Password is updated and the old password is correctly verified.	Pass	Verifies password change functionality with old password check.
Test_004h	Clicking the "Delete	The user account should	The account is deleted and the	Pass	Ensures successful account deletion

	Account" button.	be deleted after confirmation.	user is logged out		with proper confirmation.
--	------------------	--------------------------------	--------------------	--	---------------------------

Finance Management

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comment
Test_005a	Valid credit card details and payment amount	Payment should be processed and transaction recorded	Payment processed successfully.	Pass	Confirms successful payment processing
Test_005b	Missing required payment details (e.g., no card number)	System should prompt for missing information.	Error message prompts for missing detail	Pass	Ensures form validation works properly
Test_005c	Invalid dummy card number (not in allowed list)	System should reject the payment and show error	"Invalid card number"	Pass	Ensures only dummy card numbers are accepted
Test_005d	Invalid CVV (incorrect format or not in dummy list)	System should reject the payment and show error	Error message: "Invalid CVV"	Pass	Ensures CVV validation works properly
Test_005e	Selecting Card Payment as payment method	Card details should be requested	Card Payment fields appear correctly shows the message.	Pass	Confirms Card Payment option is handled
Test_005f	Entering a valid amount and submitting payment	Payment request should be recorded (no actual deduction).	Payment request logged successfully	Pass	Confirms correct logging of dummy payment.
Test_005g	Clicking " Submit Payment " after all fields are filled	Payment request should be processed successfully	Payment request processed and confirmation displayed	Pass	Confirms successful finalization of dummy payment

Feedback Management

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comment
Test_006a	Valid feedback (rating: 4 stars, comment: "Great service!")	Feedback is successfully submitted and stored in the system.	The feedback is submitted and saved without errors.	Pass	Confirms that valid feedback is correctly stored.

Test_006b	Missing required fields (e.g., username or password not provided).	The system should prompt the user to complete all required fields.	The system displays an error message indicating missing information.	Pass	Ensures validation prevents account creation with incomplete information.
Test_006c	Invalid comment input (exceeds character limit).	The system should display an error message: "Comment exceeds character limit."	The system prevents submission and shows the error message.	Pass	Verifies proper handling of comment length validation.
Test_006d	Valid feedback submission (rating: 5 stars, comment: "Amazing product!").	Feedback is successfully submitted and displayed under the user's profile.	Feedback appears correctly in the user's profile.	Pass	Confirms successful feedback submission and visibility in user profile.
Test_006e	Admin reviews and responds to feedback.	Admin response is successfully saved and displayed under the user's feedback.	Admin's response is displayed with the feedback.	Pass	Confirms admin response functionality.
Test_006f	Admin deletes a user's feedback.	Feedback is removed from the database and the user's profile.	The feedback is deleted from the system.	Pass	Verifies feedback deletion functionality.

Chapter 5

Evaluation

Sweet Bliss, the custom cake ordering platform, was tested and evaluated in functional, user and performance testing via a total of 36 test cases to validate all the components of the system, even the alerts and notifications as well.

5.1. User feedback

When developing the platform, the user feedback and testing was an immense support for developing it accurately and as the client intended.

For this, a selected group of around 15 users tested and the platform for its accuracy, performance and usability rated as follows.

- **Performance** – 7 out of 10
- **Usability** – 8 out of 10
- **Accuracy** – 6 out of 10
- **User Experience** – 8.5 out of 10

5.2. Future Improvements

Even though the system is created as the client intended, there is still room for improvement. As for future improvements,

- A more user-friendly and well optimized search for the store page.
- Better trained AI image generator for more accurate cake designs.
- Better optimized filtration for the report generation.
- Better optimization to reduce the loading time.

5.3. Conclusion

Sweet Bliss is a custom cake ordering platform developed to streamline and modernize a mutually operated business run by a university student. The system automates order handling, inventory, payments and customer communication, addressing inefficiencies caused by manual processes. Built with the MERN stack, *Sweet Bliss* offers a practical, scalable and user-centric solution tailored specifically for the custom cake industry.

As for the features the system provides,

- Custom Order Management – Fully customizable order forms with AI-powered design suggestions and admin approval.
- Finance Management – Secure payment handling, automated invoicing, and refund processing.
- Inventory Management – Real-time stock tracking, low-stock alerts, and promotion tools.
- User Management – Role-based access, account recovery, activity logging, and personalized preferences.
- Feedback System – Review submission, sentiment analysis, admin moderation, and feedback-based improvement tools.
- External Adaptability – Considers university schedules and includes vacation mode to match the owner's availability.
- AI System Integration – Smart design recommendations based on user preferences and trends.
- Responsive UI – Clean and modern interface optimized for desktop and mobile usage.

References

- [1] ANM Cupcakes. (2023). *Cupcakes*. [Online]. Available: <https://anmcupcakes.com/product-category/cupcakes/> (accessed Feb. 10, 2025).
- [2] Automatic Inc. (2023). *WooCommerce*. [Online]. Available: <https://woocommerce.com/> (accessed Feb. 12, 2025).
- [3] Divine. (2023). *Divine*. [Online]. Available: <https://www.divine.lk/> (accessed Feb. 12, 2025).
- [4] The Cake Factory. (2023). *The Cake Factory*. [Online]. Available: <https://www.cakefactory.lk/> (accessed Feb. 12, 2025).
- [5] Trustpilot A/S. (2023). *Trustpilot*. [Online]. Available: <https://www.trustpilot.com/> (accessed Feb. 14, 2025).
- [6] Uber Technologies, Inc. (2023). *Uber Eats Sri Lanka*. [Online]. Available: <https://www.ubereats.com/lk> (accessed Feb. 14, 2025).
- [7] DoorDash, Inc. (2023). *DoorDash*. [Online]. Available: <https://www.doordash.com/> (accessed Feb. 14, 2025).
- [8] Zazzle Inc. (2023). *Zazzle*. [Online]. Available: <https://www.zazzle.com/> (accessed Feb. 14, 2025).
- [9] CustomMade, Inc. (2023). *CustomMade*. [Online]. Available: <https://www.custommade.com/> (accessed Feb. 16, 2025).

Appendix A

	Student Registration Number	Student Name	Task Performed
1	IT23396272	Kariyawasam L L N	<ul style="list-style-type: none"> • Inventory Management including Store, Cart Page with Store Items • Search Option in the Store • Admin level Management System with Addition, Updating and Deletion. • Admin Level Data Filtration • Inventory Notification Management
2	IT23183018	Hirusha D G A D	<ul style="list-style-type: none"> • Order Management Including Custom Cake Ordering Form with AI features, Cart Page with Custom Orders • Customer Orders Page • Admin level Customer Order Management System • Admin Level Report Generation • Admin Level Data Filtration • Order Notification Management
3	IT23179158	Budara H M K	<ul style="list-style-type: none"> • Customer Management Including Login and Register • User Level Order Tracking • Admin Level Order Tracking • Admin Level Report Generation • Admin Level Data Filtration • Customer Notification Management

4	IT23222236	Eranthnage T N	<ul style="list-style-type: none"> • Feedback View Option • Add New Feedback Option • Feedback Report Mechanism • Admin Level Feedback Management Section • Admin Level Feedback Resolving Mechanism • Data Filtration • Admin Level Report Generation
5	IT23222786	Budara V P R	<ul style="list-style-type: none"> • Implementing Payment Management System • Implementing Loyalty Wheel System • Discount Code Validation & Expiration System • Order Invoice Generation System • Admin Level Finance Tracking System • Admin Level Report Generation System

Appendix B

Member Name	IT Number	Contribution to the report
Kariyawasam L L N	IT23396272	<ul style="list-style-type: none"> • Abstract • Acknowledgement • Chapter 01 • Added necessary essentials, Use case scenarios, Test cases, UI designs Flow chart
Hirusha D G A D	IT23183018	<ul style="list-style-type: none"> • Finalized the report by formatting accordingly. • Table of Contents • Chapter 03

		<ul style="list-style-type: none"> • Added necessary essentials, Use case scenarios, Test cases, UI designs Use case Diagram Data Flow Diagram 	
Budara H M K	IT23179158	<ul style="list-style-type: none"> • Chapter 02 • Appendices • Added necessary essentials, Use case scenarios, Test cases, UI designs System Diagram 	
Eranthnage T N	IT23222236	<ul style="list-style-type: none"> • Chapter 05 • Conclusion • Added necessary essentials, Use case scenarios, Test cases, UI designs. Normalized schema. 	
Budara V P R	IT23222786	<ul style="list-style-type: none"> • Chapter 04 • Kanban board and the product backlog. • Added necessary, Use case scenarios, Test cases, UI designs ER diagram 	