ANUDIP FOUNDATION

**Bakery Store Management System**

**By**

**Aisha Shaikh – AF0481878**

**Poonam kapase – AF0482060**

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Topic** | **Page no** |
| 1 | Title of Project | 1 |
| 2 | Acknowledgement | 3 |
| 3 | Abstract | 4 |
| 4 | Introduction | 5 |
| 5 | Objective | 6 |
| 6 | System Analysis | 7 |
| 7 | System Design | 22 |
| 8 | Screenshots | 32 |
| 9 | Coding | 39 |
| 10 | Testing | 46 |
| 11 | Report | 47 |
| 12 | Future Scope | 48 |
| 13 | Conclusion | 48 |
| 14 | Bibliography | 49 |
| 15 | References | 49 |

**Acknowledgement**

The project **“Bakery Store Management System”** is the Project work carried out by

|  |  |
| --- | --- |
| **Name** | **Enrollment No** |
| **Aisha Shaikh –** | **AF0481878** |
| **Poonam kapase –** | **AF0482060** |

Under the Guidance.

Rajshri Thete

We are thankful to my project guide for guiding me to complete the Project.

His suggestions and valuable information regarding the formation of the Project Report have provided me a lot of help in completing the Project and its related topics.

We are also thankful to my family member and friends who were always there to provide support and moral boost up.

## Abstract

The **Bakery Store Management System** is a comprehensive software solution developed to automate and manage the various processes involved in running a bakery. This system is designed to handle core operations such as inventory management, order processing, sales tracking, and customer data management. It replaces traditional manual methods with a digital interface, improving accuracy, efficiency, and productivity.

The system enables bakery staff to easily add and update product information, monitor stock levels, process customer orders, and generate sales reports. It also includes user authentication features to ensure secure access to administrative functionalities. By leveraging technology, this system reduces human error, streamlines operations, and enhances the overall customer experience.

This project is built using [insert technologies, e.g., Python, MySQL, HTML/CSS], and is aimed at small to medium-sized bakery businesses seeking to modernize their management processes. The result is a reliable and scalable solution that supports the smooth operation and growth of bakery enterprises.

## Introduction

The **Bakery Store Management System** is a software application designed to streamline and automate the daily operations of a bakery store. It addresses the common challenges faced in managing inventory, tracking sales, handling customer orders, and maintaining records of products and employees. By integrating all essential functions into a single system, it aims to improve efficiency, reduce manual errors, and enhance customer satisfaction.

This system provides a user-friendly interface that allows staff to easily manage product listings, update stock levels, generate bills, and monitor sales performance. Additionally, it offers features such as customer order management, report generation, and user authentication to ensure data security.

The implementation of the Bakery Store Management System helps in reducing workload, improving service quality, and supporting data-driven decision-making, making it an essential tool for modern bakery operations.

**1.1 Objective of the Present Work**

The primary objective of the **Bakery Store Management System** is to automate and simplify the daily operations of a bakery store to ensure efficient and error-free management. This system aims to:

* **Streamline bakery operations** such as product management, inventory control, sales tracking, and billing.
* **Reduce manual workload** by replacing traditional paper-based processes with a digital solution.
* **Enhance customer service** by enabling quick order processing and accurate billing.
* **Maintain accurate records** of products, orders, customers, and sales for better decision-making.
* **Improve inventory management** by tracking stock levels and alerting for low inventory.
* **Ensure data security** through user authentication and restricted access to administrative features.
* **Generate reports** that help in analyzing sales trends and business performance.
* **Provide a user-friendly interface** for both admin and staff to operate the system with ease.

Overall, the system is designed to increase productivity, reduce operational costs, and support the growth and efficiency of the bakery business.

**System analysis**

**3.1 PROBLEM DEFINITION**

In many small and medium-sized bakery stores, daily operations are managed manually using paper-based records or basic tools like spreadsheets. This traditional approach often leads to various challenges, including:

* **Inaccurate inventory tracking**, resulting in overstocking or stockouts.
* **Time-consuming billing and order processing**, leading to longer customer wait times.
* **Difficulty in maintaining sales records**, making it hard to analyze business performance.
* **High chances of human error** in calculations, record-keeping, and order handling.
* **Lack of centralized data**, causing inefficiency in managing products, customers, and sales.
* **Limited data security**, as manual records are prone to loss or unauthorized access.

These issues reduce operational efficiency, increase workload, and negatively impact customer satisfaction. Therefore, there is a need for an automated system that can manage all core bakery functions in a more reliable, efficient, and secure manner.

The **Bakery Store Management System** is designed to address these problems by providing a digital solution that simplifies operations, improves accuracy, and enhances overall business management.

**3.2 Preliminary Investigation**

Before developing the **Bakery Store Management System**, a preliminary investigation was conducted to understand the current challenges faced by bakery stores and to determine the feasibility and scope of implementing an automated solution.

### 1. ****Problem Identification****

Most bakery businesses rely on manual methods for handling inventory, sales, and customer orders. This often leads to issues such as slow service, stock mismanagement, human error, and lack of real-time data. These problems reduce overall efficiency and customer satisfaction.

### 2. ****Fact Finding****

To gather relevant information, the following methods were used:

* **Interviews** with bakery owners and employees to understand their daily operations and challenges.
* **Observation** of existing workflows in a typical bakery environment.
* **Review** of existing software systems used in similar small retail environments to identify gaps.

### 3.3 ****Feasibility Study****

A feasibility study was conducted to determine the practicality of developing and implementing the system:

* **Technical Feasibility:** The required tools and technologies (e.g., Python, MySQL, HTML/CSS) are readily available and well-supported.
* **Economic Feasibility:** Development costs are minimal, and the long-term benefits in terms of time and cost savings justify the investment.
* **Operational Feasibility:** The system is easy to use and can be operated by staff with basic computer skills, ensuring smooth adoption.

### 4. ****Scope of the Project****

The system will cover:

* Product and inventory management
* Customer order processing
* Billing and sales tracking
* Report generation
* User login and role-based access

### 5. ****Conclusion****

The preliminary investigation confirms that there is a clear need for an automated bakery management solution. The proposed system is both feasible and beneficial, offering significant improvements in operational efficiency and service quality.

3.4 Project Planning

Project planning is a crucial phase in the development of the **Bakery Store Management System**, as it helps define the project's scope, objectives, timelines, resources, and deliverables. Effective planning ensures that the system is developed efficiently and meets user requirements.

**1. Preliminary Investigation**

* + Identify the need for a bakery management system.
  + Conduct a feasibility study (technical, economic, operational).

**2. System Analysis**

* + Conduct stakeholder interviews and surveys.
  + Analyze current bakery operations and workflows.
  + Document functional and non-functional requirements.
  + Identify potential risks and mitigation strategies.

**3. System Design**

* + Design system architecture (e.g., client-server, MVC).
  + Create data models and database schemas.
  + Design user interfaces and user experience flows.
  + Plan for scalability and integration with other systems.

**4. Coding (Development)**

* + Set up the development environment using Python (Django).
  + Develop modules for inventory management, order processing, billing, etc.
  + Implement sentiment analysis features if required.
  + Conduct code reviews and maintain version control.

**5. Security Implementation**

* + Implement user authentication and authorization mechanisms.

**6. Testing**

* + Perform unit testing on individual components.
  + Conduct integration testing to ensure modules work together.
  + Carry out system testing for overall functionality.

**7. Implementation (Deployment)**

* + Prepare deployment plans and rollback strategies.
  + Set up the production environment and deploy the application.
  + Train end-users and provide documentation.
  + Monitor the system post-deployment for any issues

### 5. ****Resources Required****

* Software: Python, MySQL, HTML/CSS, and optionally Django or Flask
* Hardware: Computer system, internet access
* Human resources: Developers, testers, users (for feedback)

### 6. ****Risk Management****

Potential risks and mitigation strategies:

* **Scope creep:** Maintain clear and agreed-upon requirements.
* **Technical issues:** Use familiar and stable technologies.
* **User resistance:** Offer training and clear user guides.

### 7. ****Deliverables****

* Functional Bakery Store Management System
* User manual and technical documentation
* Deployment report and final presentation

3.5 Software Requirement Specification (SRS)

#### 1. Purpose

The purpose of this Software Requirement Specification (SRS) document is to define the functional and non-functional requirements for the **Bakery Store Management System**. This document serves as a guide for the development, design, and validation of the system.

#### 2. Scope

The Bakery Store Management System is designed to manage and automate various bakery store operations such as product management, inventory tracking, customer order handling, billing, and sales reporting. The system will ensure efficient business management and improved customer satisfaction.

#### 3 Intended Audience

* Developers
* Project Managers
* Bakery Owners/Staff
* Testers
* Documentation Team

### 2. ****Overall Description****

#### 1 Product Perspective

This system is a standalone web/desktop application that digitizes the current manual processes of a bakery store. It interacts with a database to manage real-time data for products, inventory, and transactions.

#### 2 Product Features

* Add, update, and delete bakery products
* Manage inventory stock levels
* Handle customer orders and billing
* Generate daily, weekly, and monthly sales reports
* User authentication and role-based access

#### 3 User Classes and Characteristics

* **Admin**: Full access to all modules including product, inventory, sales, and user management.
* **Cashier/Staff**: Limited access to order processing, billing, and viewing reports.

#### 4 Operating Environment

* Front-end: HTML/CSS/JavaScript
* Back-end: Python (Django)
* Database: MySQL
* Operating System: Windows

#### 5 Design and Implementation Constraints

* The system must run on basic hardware with standard OS.
* The application should be responsive and easy to navigate.
* Data must be securely stored in a relational database.

3.6 Functional Requirements

The **Functional Requirements** define the specific behavior and functionalities that the **Bakery Store Management System** must provide to fulfill its intended purpose. These requirements are grouped based on the core modules of the system.

### 1. ****User Authentication and Access Control****

* **FR1.1:** The system shall allow users to register (admin only) and log in with a username and password.
* **FR1.2:** The system shall support different user roles (Admin, Staff) with role-based access.
* **FR1.3:** The system shall restrict access to sensitive modules based on the user role.

### 2. ****Product Management****

* **FR2.1:** The admin shall be able to add new bakery products with details such as name, category, price, and quantity.
* **FR2.2:** The admin shall be able to update product information.
* **FR2.3:** The admin shall be able to delete products from the system.
* **FR2.4:** The system shall display a list of all available products for order processing.

### 3. ****Inventory Management****

* **FR3.1:** The system shall automatically update inventory quantities when a product is sold.
* **FR3.2:** The system shall notify the admin when a product stock level falls below a predefined threshold.
* **FR3.3:** The admin shall be able to manually update stock levels (e.g., after restocking).

### 4. ****Order Management****

* **FR4.1:** The system shall allow staff to select products and quantities to create a customer order.
* **FR4.2:** The system shall automatically calculate the total price, including applicable taxes.
* **FR4.3:** The system shall generate a printable bill or receipt for each order.
* **FR4.4:** The system shall store order details in the database for future reference.

### 5. ****Sales and Reporting****

* **FR5.1:** The system shall generate daily, weekly, and monthly sales reports.
* **FR5.2:** The reports shall include product-wise sales, total revenue, and quantity sold.
* **FR5.3:** The admin shall be able to view and export reports (PDF or CSV).

### 6. ****Customer Management (Optional)****

* **FR6.1:** The system may allow storing basic customer information (name, contact) for returning customers.
* **FR6.2:** The system may track customer order history for future reference or loyalty programs.

### 7. ****System Backup and Maintenance****

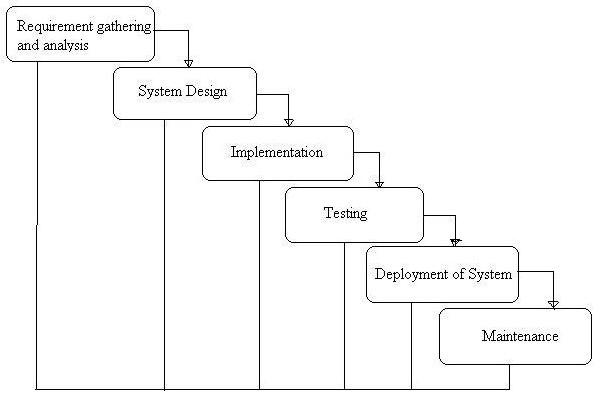
* **FR7.1:** The system shall provide a way to back up the database periodically.
* **FR7.2:** The system shall allow the admin to restore data from a backup if needed.

3.7 Software Engineering Paradigm

The development of the **Bakery Store Management System** follows a structured and disciplined approach based on established software engineering principles. The selected paradigm ensures that the system is reliable, maintainable, and meets user requirements effectively.

### 1. ****Paradigm Used: Waterfall Model****

For this project, the **Waterfall Model** has been chosen due to its linear and sequential nature, which is suitable for small to medium-scale systems with well-defined requirements.



### 2. ****Phases of the Waterfall Model Applied****

#### a. ****Requirement Analysis****

* Detailed interaction with bakery staff and management to gather all system requirements.
* Identification of functional and non-functional requirements.
* Documentation of the Software Requirement Specification (SRS).

#### b. ****System Design****

* Designing the system architecture.
* Creating Entity Relationship Diagrams (ERDs), Data Flow Diagrams (DFDs), and database schema.
* Designing user interfaces and navigation flow.

#### c. ****Implementation (Coding)****

* Actual coding of the system using appropriate technologies (e.g., Python, MySQL, HTML/CSS).
* Modular development of different components (product, inventory, billing, reports).

#### d. **Testing**

* Unit testing of individual modules.
* Integration testing to ensure modules work together correctly.
* System testing to validate the complete system against requirements.

#### e. ****Deployment****

* Installation of the system in the target environment.
* Configuration and setup based on the client’s hardware/software environment.

### ➤ ****Level 0 – Context Diagram****

The Level 0 DFD (Context Diagram) shows the system as a single process with interactions between external entities.

#### ****External Entities****:

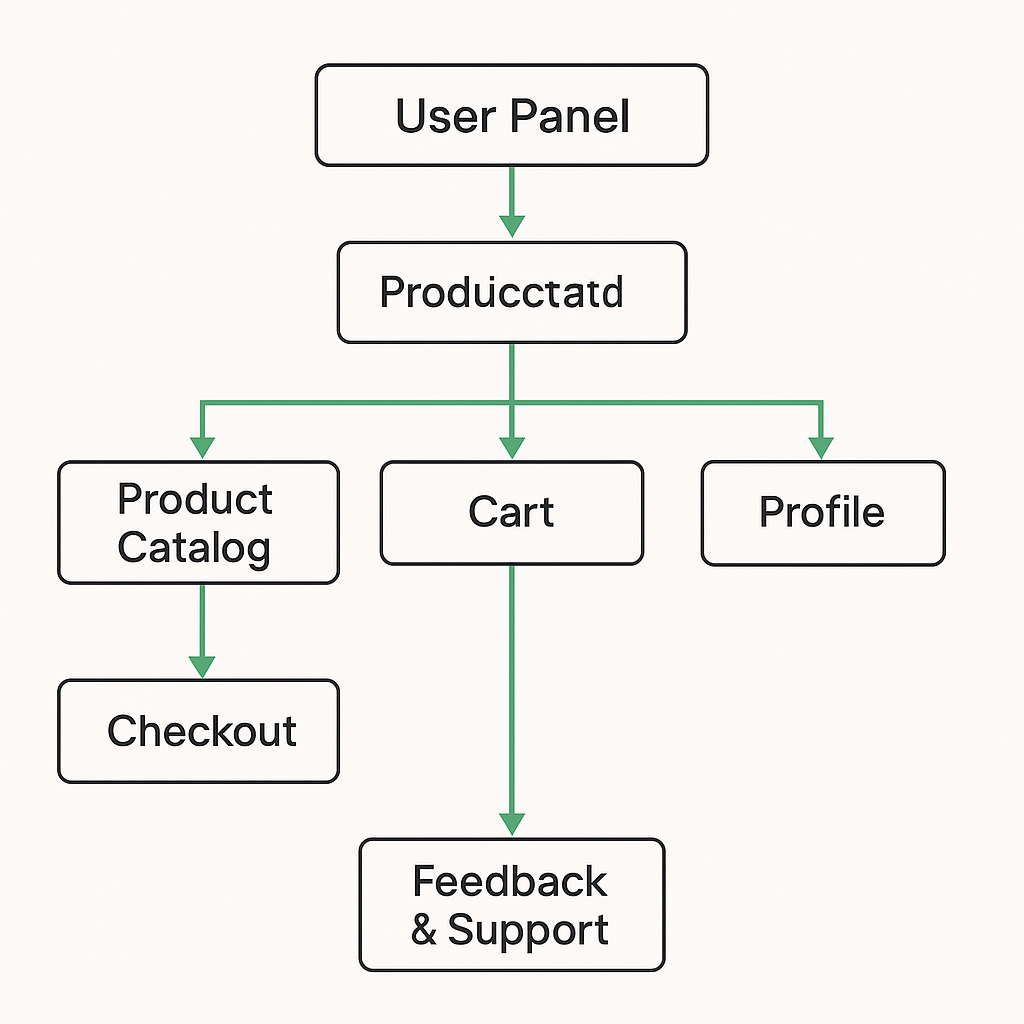
* **Admin**
* **Staff**
* **Customer**

#### ****Main Process****:

* Bakery Store Management System

#### ****Data Flows****:

* Admin inputs product, inventory, and user data
* Staff processes orders and updates sales
* Customers place orders
* System provides bills, inventory status, and reports



3.8 Data Flow Diagram:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

**The following observations about DFDs are essential:**

1. All names should be unique. This makes it easier to refer to elements in the DFD.
2. Remember that DFD is not a flow chart. Arrows is a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.
3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represents decision points with multiple exists paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.
4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.

**Processes:**

1. **Manage Products**: This process involves adding, updating, and removing bakery products from the system. It ensures that the product catalog is current and reflects the available items.
2. **Manage Inventory**: Responsible for tracking raw materials and finished goods. It updates stock levels, manages suppliers, and ensures that inventory is replenished as needed.
3. **Process Orders**: Handles customer orders from initiation to completion. This includes order validation, billing, and updating inventory to reflect sold items.
4. **Generate Reports**: Compiles data from various processes to produce reports on sales, inventory status, and financial summaries. These reports aid in decision-making and performance analysis.
5. **User Authentication**: Manages user login and access control, ensuring that only authorized personnel can access specific system functionalities.

**Data Stores:**

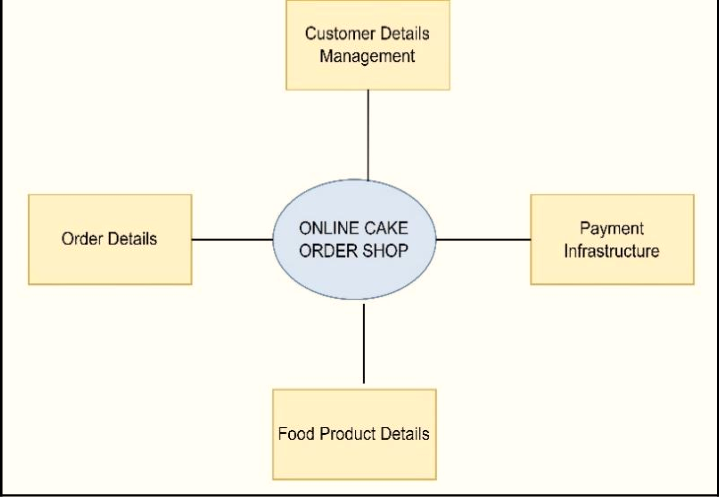
* **Product Database**: Holds information about bakery products, including names, descriptions, prices, and categories.
* **Inventory Database**: Contains records of raw materials and finished goods, including quantities, suppliers, and reorder levels.
* **Order Database**: Stores customer orders, including order details, customer information, and payment status.
* **User Database**: Maintains user credentials and roles to manage access control within the system.
* **Sales Reports**: Archives generated reports for historical reference and analysis.[ConceptDraw+3ConceptDraw+3Wikipedia+3](https://conceptdraw.com/examples/data-flow-diagram-bakery-management-system?utm_source=chatgpt.com)

**External Entities:**

* **Admin**: Oversees system operations, manages users, and ensures smooth functioning of the bakery store.
* **Staff**: Includes employees who interact with the system to manage products, process orders, and update inventory.
* **Customer**: Individuals who place orders, make payments, and receive products.[Codebun+6Visual Paradigm+6GeeksforGeeks+6](https://online.visual-paradigm.com/diagrams/features/dfd-maker/?utm_source=chatgpt.com)

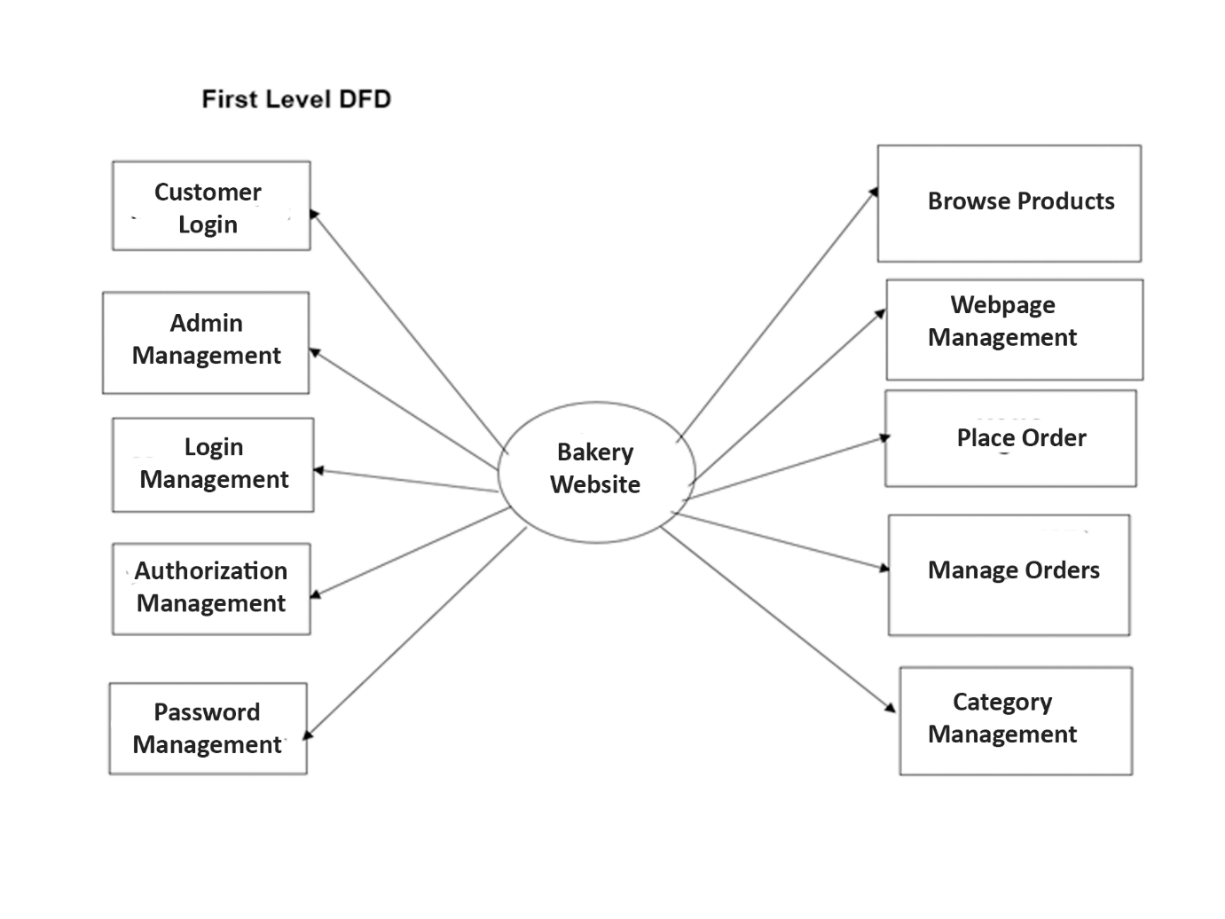
**Data Flows:**

* **Admin to Manage Products**: Admin provides product details to be added or updated in the system.[Codebun](https://codebun.com/uml-diagrams-for-online-bakery-shop-system-project/?utm_source=chatgpt.com)
* **Manage Products to Product Database**: New or updated product information is stored in the product database.
* **Customer to Process Orders**: Customers submit their orders, including product selections and quantities.
* **Process Orders to Order Database**: Order details are recorded in the order database for processing and tracking.
* **Process Orders to Inventory Database**: Inventory levels are updated based on the items sold in the order.
* **Generate Reports to Sales Reports**: Generated reports are stored for future reference and analysis.
* **User Authentication to User Database**: User credentials are validated against the user database to grant access.

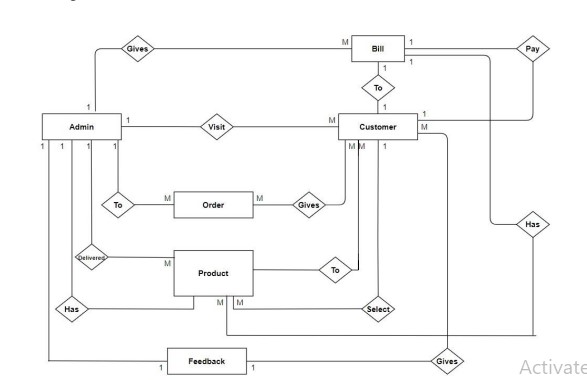


**Zero Level Data Flow Diagram (DFD)**

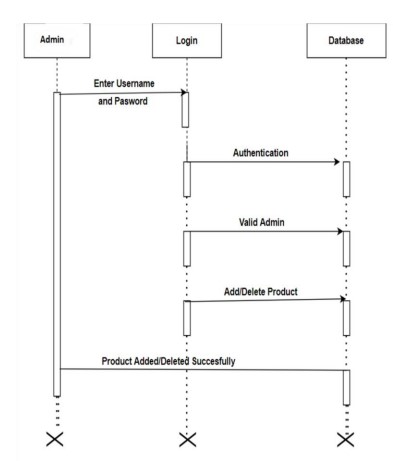
The **First-Level DFD**



ER diagram

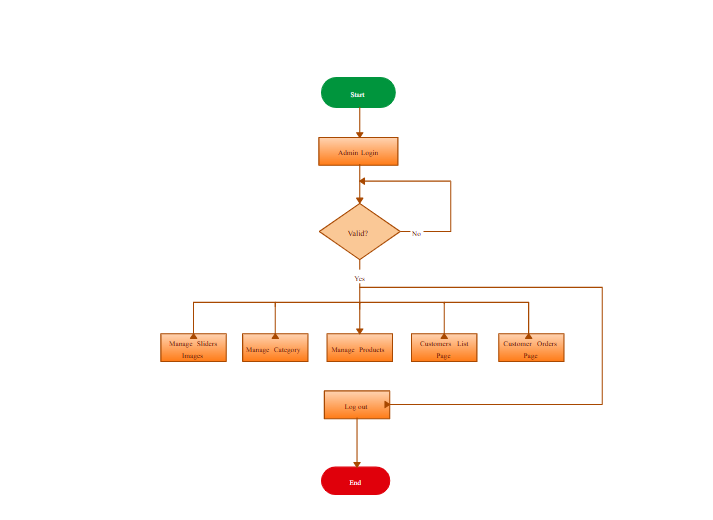
****

**Admin**

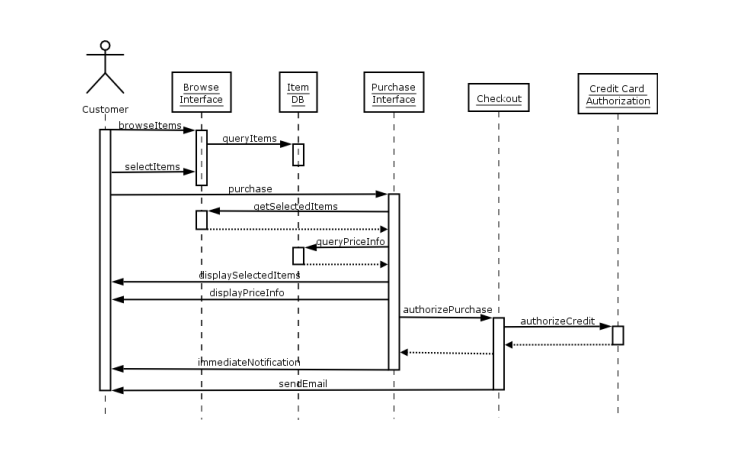
****

**Admin Flow Chart**

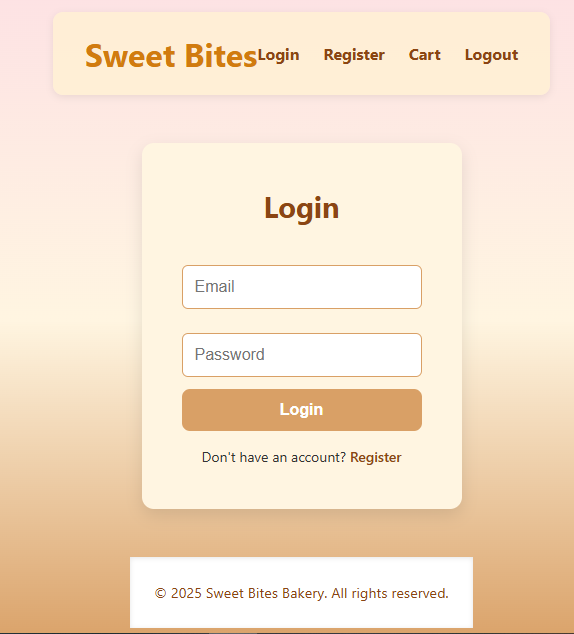
**Admin product diagram**

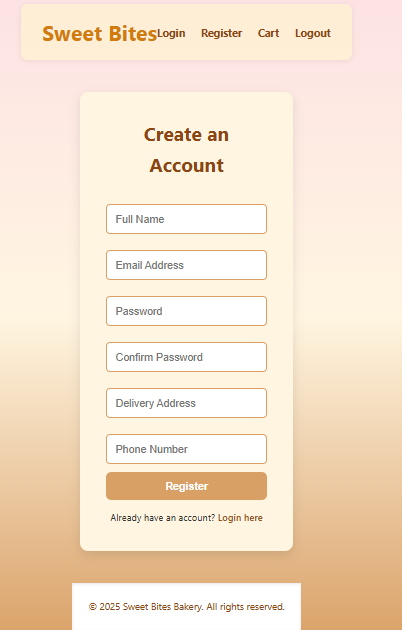
****

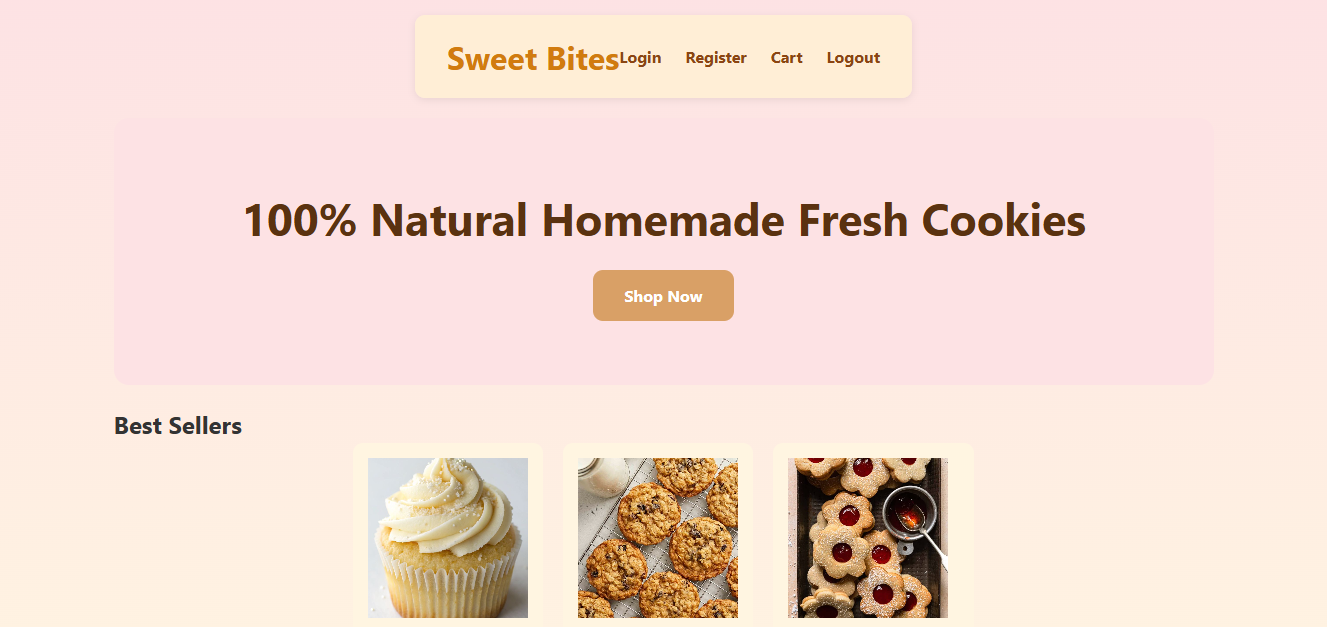
**Customer sequence diagram**

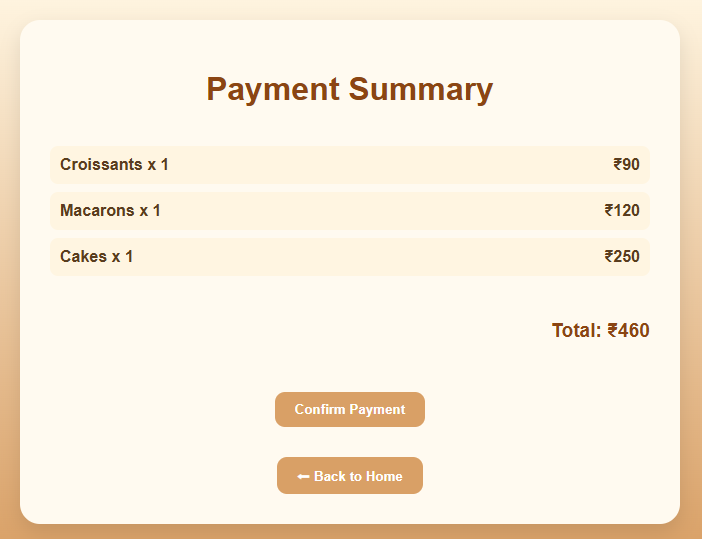
****

**Screenshots**

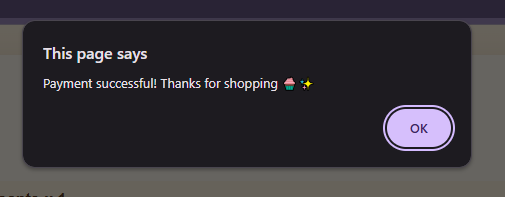
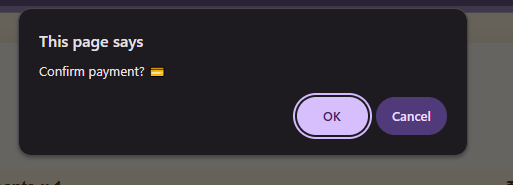
****

****

****

****

****

****

**Coding**

Layout

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>{% block title %}Sweet Bites Bakery{% endblock %}</title>

    <link rel="stylesheet" href="{% static 'css/style-v2.css' %}">

</head>

<body>

    <!-- Navbar -->

    <nav>

        <h1>Sweet Bites</h1>

        <ul>

            <li><a href="{% url 'login' %}">Login</a></li>

            <li><a href="{% url 'register' %}">Register</a></li>

            <li><a href="{% url 'cart' %}">Cart</a></li>

            <li><a href="{% url 'logout' %}">Logout</a></li>

        </ul>

    </nav>

    <!-- Main Content Block -->

    <main>

        {% block content %}

        {% endblock %}

    </main>

    <!-- Footer -->

    <footer>

        <p>&copy; 2025 Sweet Bites Bakery. All rights reserved.</p>

    </footer>

</body>

</html>

Main Front page

{% extends 'store/layout.html' %}

{% load static %}

{% block title %}Sweet Bites Bakery{% endblock %}

{% block content %}

<section class="hero">

  <div class="hero-content">

    <h2>100% Natural Homemade Fresh Cookies</h2>

    <a href="" class="btn-link">Shop Now</a>

  </div>

</section>

<section class="best-sellers">

  <h2>Best Sellers</h2>

  <div class="items">

    <div class="item">

      <img src="{% static 'images/white cupcake.jpg' %}" alt="Cupcake" width="150">

      <p>White Chocolate Bites</p>

      <p class="price">Rs.120</p>

    </div>

    <div class="item">

      <img src="{% static 'images/Oatmeal Cookies.jpeg' %}" alt="Cookie" width="150">

      <p>Oatmeal Bites</p>

      <p class="price">Rs.70</p>

    </div>

    <div class="item new">

      <img src="{% static 'images/Jam Cookies.jpg.jpeg' %}" alt="Jam Cookie" width="150">

      <p>Jam & Poppy Seed Bites</p>

      <p class="price">Rs.55</p>

    </div>

  </div>

</section>

<section class="gallery">

  <h2>Follow Me for More Bites</h2>

  <div class="grid">

    <img src="{% static 'images/Chocolate Covered Dates.jpg' %}" alt="Chocolate Nuts">

    <img src="{% static 'images/Cheesecake.jpg' %}" alt="Cheese Cake">

    <img src="{% static 'images/Strawberry Macaron Shells.jpg' %}" alt="Macarons">

    <img src="{% static 'images/cookies.jpg' %}" alt="Cookies">

    <img src="{% static 'images/Blueberry mini Pie.jpg' %}" alt="Mini Pies">

    <img src="{% static 'images/Croissant.jpg' %}" alt="Croissants">

  </div>

</section>

<section class="services">

  <div class="service">

    <h3>New Pick Up Service</h3>

    <p>Grab it fresh.</p>

  </div>

  <div class="extra-services">

    <div class="service">

      <h3>Free Delivery</h3>

      <p>Delivered to your door.</p>

    </div>

    <div class="service">

      <h3>Special Gift Wraps</h3>

      <p>Send a sweet surprise.</p>

    </div>

  </div>

</section>

{% endblock %}

Login

{% extends 'store/layout.html' %}

{% block title %}Login - Sweet Bites Bakery{% endblock %}

{% block content %}

<div class="login-container">

    <h2>Login</h2>

    <form method="POST">

        {% csrf\_token %}

        <input type="email" name="email" placeholder="Email" required>

        <input type="password" name="password" placeholder="Password" required>

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

    </form>

    <div class="register-link">

        Don't have an account? <a href="{% url 'register' %}">Register</a>

    </div>

</div>

{% endblock %}

Register

{% extends 'store/layout.html' %}

{% block title %}Register - Sweet Bites Bakery{% endblock %}

{% block content %}

<div class="register-container">

    <h2>Create an Account</h2>

    <form method="POST" >

        {% csrf\_token %}

        <input type="text" name="full\_name" placeholder="Full Name" required>

        <input type="email" name="email" placeholder="Email Address" required>

        <input type="password" name="password" placeholder="Password" required>

        <input type="password" name="confirm\_password" placeholder="Confirm Password" required>

        <input type="text" name="address" placeholder="Delivery Address" required>

        <input type="tel" name="phone" placeholder="Phone Number" required>

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

    </form>

    <div class="login-link">

        <p>Already have an account? <a href="{% url 'login' %}">Login here</a></p>

    </div>

</div>

{% endblock %}

**Future Scope**

The Bakery Store Management System is designed to streamline day-to-day operations such as order processing, inventory control, billing, and report generation. While the current version meets essential operational needs, there is significant potential for future enhancements to improve functionality, scalability, and customer engagement.

### 1. ****Online Ordering Integration****

* Develop an online ordering system (website or mobile app) to allow customers to place orders remotely.
* Real-time integration with the inventory and order management modules.
* Online payment gateway support.

### 2. ****Advanced Security Features****

* Implement role-based access control with detailed permissions.
* Use two-factor authentication (2FA) for user login.
* Enable audit trails to track system usage and changes.

### 3. ****Mobile Application****

* Develop a mobile app for staff to manage orders and inventory on the go.
* Push notifications for stock alerts and order updates.
* Customer-facing app for viewing menu, placing orders, and checking order history.

### 4. ****Business Analytics and Insights****

* Introduce dashboards with visual analytics using charts and graphs.
* Predictive analytics to forecast product demand based on historical data.
* AI-driven recommendations for upselling and product bundling

### 5. ****GST and Tax Management****

* Automatically apply regional tax regulations (e.g., GST).
* Generate tax-compliant invoices and summaries.

### 6. ****Barcode / QR Code Integration****

* Generate barcodes for products.
* Use barcode/QR scanners for fast billing and inventory updates.

### 7. ****Customer Feedback and Loyalty Program****

* Add customer feedback forms after order completion.
* Implement loyalty rewards, discounts, and personalized offers.

**CONCLUSION:-**

The transition from manual to computerized systems in bakery management brings significant improvements in efficiency, accuracy, and overall business operations. Manual processes often lead to time-consuming tasks and are susceptible to human errors, especially in data entry and inventory management. These errors can result in discrepancies, stockouts, or overstocking, affecting customer satisfaction and profitability.

Implementing a computerized system streamlines operations by automating routine tasks such as order processing, inventory tracking, and billing. This automation reduces the likelihood of errors, ensures real-time data updates, and enhances decision-making capabilities. For instance, bakery management software offers features like inventory management, sales tracking, and customer relationship management, which collectively contribute to smoother workflows and better resource allocation

Moreover, computerized systems facilitate quick retrieval of information, enabling staff to access customer orders, inventory levels, and sales reports promptly. This agility not only improves service delivery but also fosters a more responsive business environment. Additionally, the ability to generate detailed reports aids in monitoring performance metrics, identifying trends, and making informed strategic decisions.

In conclusion, adopting a computerized system in bakery management is a proactive step towards enhancing operational efficiency, minimizing errors, and providing a better experience for both customers and staff. It represents a commitment to modernizing business practices and staying competitive in an increasingly digital marketplace.

**BIBLIOGRAPHY**

1) <http://www.google.com/>

2) <https://www.w3schools.com>

3) <https://www.tutorialspoint.com/>

4) http://www.wikepedia.com/

**References**

 W3Schools. (n.d.). HTML, CSS, JavaScript Tutorials. Retrieved from <https://www.w3schools.com>  
➤ Used for frontend development references.

 Tutorialspoint. (n.d.). Django Framework Basics. Retrieved from https://www.tutorialspoint.com/django  
➤ Helped understand backend implementation for bakery systems.

 GeeksforGeeks. (n.d.). Inventory Management System Project Guide. Retrieved from <https://www.geeksforgeeks.org>  
➤ Provided insights on managing stock, billing, and order processes.