



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Spring 24-25

Section: B

Software Quality Assurance and Testing

Pastry Shop Management System

A Report submitted

By

SN	Student Name	Student ID
1	PROTTOY SAHA	21-45640-3
2	FARDIN ABU UBAID	21-44863-2
3	MUHTASIB IBTIDA KOUSIK	21-44864-2
4	MD. REZUAN HUSSAIN	21-45389-3

Checked By Industry Personnel (Optional)

Name: Upoma Ahmed

Designation: QA Team Lead

Company: Workspace Infotech Limited

Sign:

Date: 26-06-2025

Software Test Plan

for

Pastry Shop Management System

Version 1.0 approved

Prepared by Prottoy Saha, Fardin Abu Ubaid, Muhtasib Ibtida Kousik, Md. Rezuan Hussain

American International University-Bangladesh

26/05/25

Table of Contents

Revision History	3
1. TEST PLAN IDENTIFIER:RS-MTP01.3	4
2. REFERENCES.....	4
3. INTRODUCTION.....	4
Background to the Problem.....	4
Solution to the Problem (Need to be updated in google Docx)	5
4. REQUEIREMNT SPECIFICATION	7
4.1 System Features.....	7
4.2 System Quality Attributes	10
4.3 System Interface	11
4.4 Project Requirements.....	17
5. FEATURES NOT TO BE TESTED.....	19
○ Third-Party Payment Gateway Integration	19
○ Hosting Server Performance and Reliability.....	19
○ Customer-Side Spreadsheet or Report Processing.....	19
○ Browser Compatibility for Outdated Browsers	19
6. TESTING APPROACH	20
6.1 Testing Levels	20
6.2 Test Tools	20
6.3 Meetings	21
7. TEST CASES/TEST ITEMS	22
8. ITEM PASS/FAIL CRITERIA	36
9. TEST DELIVERABLES	36
10. STAFFING AND TRAINING NEEDS	36
11. RESPONSIBILITIES	37
12. TESTING SCHEDULE.....	37
13. PLANNING RISKS AND CONTINGENCIES	38
14. APROVALS.....	38

Revision History

Revision	Date	Updated by	Update Comments
0.1	2025.05.26	Prottoy Saha	Initial Document
0.2	2025.06.01	Fardin Abu Ubaid	First Draft
0.3	2025.06.07	Muhtasib Ibtida Kousik	Second Draft
0.4	2025.06.13	Md. Rezuan Hussain	Updated Requirement (first Version)
0.5	2025.06.18	Fardin Abu Ubaid	Updated Requirement (second Version)
0.6	2025.06.22	Prottoy Saha	Initial Test Cases
0.7	2025.06.25	Md. Rezuan Hussain	Updated Test Cases
0.8	2025.06.26	Muhtasib Ibtida Kousik	Final Revision

1. TEST PLAN IDENTIFIER:RS-MTP01.3

2. REFERENCES

- Software Quality and Testing Course PowerPoint Slides
- Software Requirements Specification (SRS) Document

3. INTRODUCTION

Background to the Problem

- Companies in a variety of industries are adopting technology in the rapidly changing digital world to improve customer satisfaction and expedite operations. However, a lot of small businesses, including local bakery shops, still do their daily activities primarily by hand. Product management, staff coordination, order processing, client interaction, and sales tracking are some of these duties. Ineffective operations, inconsistent data, poor communication, and lost business opportunities are frequently the outcomes of operating without a centralized, digital infrastructure. The goal of the Pastry Shop Management System project is to close this digital divide by implementing a methodical, web-based platform that makes running a pastry store easier and more effective. This project is related to the retail and service automation issue domain. Where there is a growing need for user-friendly, role-based, customized solutions. All stakeholders' admins, staff, and clients are brought together by the system on a single platform that represents their unique operating requirements and rights. The project aims to reduce the drawbacks of conventional, disconnected management techniques by clearly organizing and automating tasks including user administration, sales monitoring, inventory management, and order placement. The creation of this system gives all project participants—developers, stakeholders, users, and testers a common knowledge of the goals, the problem's background, and the functional requirements. It guarantees that everyone involved is working toward the same goal of increasing the modern pastry business's scalability, efficiency, and transparency.
- The absence of a centralized, automated, and role-based digital management system designed especially for pastry shops' operational requirements is the main source of the problem. Many small and medium-sized bakery businesses still manage orders, staff, goods, and customer data using disconnected manual techniques such handwritten records, verbal communication, and isolated spreadsheets. Data duplication, order fulfillment failures, inventory mismanagement, and trouble tracking performance indicators are just a few of the problems caused by this antiquated method. In addition, clients frequently receive poor service in the absence of a specialized system because of delayed processing, ambiguous order tracking, and a lack of interaction opportunities. In the absence of a clear process backed by digital technologies, employees also struggle to manage their jobs effectively. The lack of real-time sales data and system-wide visibility prevents administrators from making well-informed choices. Customer happiness, operational correctness, and corporate agility are crucial success elements in today's

competitive business market, making this issue worth considering. Ignoring this problem may result in less loyal customers, lower sales, and delays in the expansion of the company. As a result, putting in place a systematic and scalable management system is crucial for enhancing present operations as well as guaranteeing long-term viability and digital preparation for future growth.

Solution to the Problem

- This project suggests creating a web-based pastry shop management system to successfully handle the difficulties small-scale pastry shops confront. Three main user roles admin, employee, and customer can communicate and coordinate operations more easily thanks to this solution's unified digital platform architecture. By providing each user type with tailored access and functionality according to their roles, the system lowers complexity and boosts productivity in all facets of corporate operations. A role-based registration and login process is part of the suggested solution, which guarantees safe access and customized system interaction. Admins have the capabilities necessary to monitor product inventory, manage system users, monitor sales performance, and oversee high-level operations. Employees can take care of product related tasks, maintain customer data, and help with order fulfillment. A user-friendly interface is provided to customers so they can explore items, place orders, pay, and maintain their profiles. By automating repetitive procedures and guaranteeing real-time data accessibility, our solution specifically targets the fundamental operational inefficiencies present in traditional pastry shop management. The system is particularly ideal for small and medium-sized enterprises with limited IT infrastructure because it is constructed with widely used and lightweight web technologies like HTML, CSS, JavaScript, PHP, AJAX, and MySQL, which guarantee cost-effective deployment, scalability, and ease of maintenance. The goal of increasing customer pleasure, streamlining internal processes, and offering data-driven insights for improved decision-making are all ideally aligned with the solution's business viability. It decreases human mistakes, gets rid of the need for manual procedures, and lets the company grow without sacrificing service quality. In addition to streamlining daily operations, the system's digitization of key procedures positions the company for future digital development and market competitiveness.
- Using a role-based access model, the Pastry Shop Management System is an online software program created to automate and oversee the essential operations of a pastry shop. The system was created with HTML, CSS, JavaScript, PHP, AJAX, and MySQL with the goal of giving three different user types—Admin, Employee, and Customer—a safe, responsive, and easy-to-use interface. According to their roles, each user is given system access, guaranteeing that communication is efficient and duties are well-defined. This software's main objectives are to increase operational effectiveness, decrease manual labor, and raise the standard of service in pastry shop settings. The program guarantees precision, consistency, and dependability in business operations by automating routine tasks including order processing, payment processing, client administration, and product catalog handling. Centralized management over all operational modules, better data quality and less administrative labor, quicker and more dependable customer service, real-time access to sales and inventory data, and increased staff responsibility and productivity are some of the system's main advantages. a platform that can grow to accommodate new

features in the future. Implementing a safe, role-based login and registration process and giving administrators complete control over system users and goods are the system's main goals. In order enable staff members to effectively handle customer and product records, to enable interactive product browsing, ordering, and payment for clients online. The project's major objective is to provide a reliable, reasonably priced software solution that fits the requirements of pastry companies and keeps them competitive, customer-focused, and operationally optimized in a market that is becoming more and more digital.

- A few open-source and commercial software programs have surfaced in recent years to meet the growing need for digital business management in the retail and food sectors. Features including product listing, order processing, inventory control, and sales reporting are frequently provided by these systems. Many current systems, however, are either too generic for small-scale pastry shops or do not have the role-specific features needed to facilitate internal processes among various user types, including administrators, staff, and clients. Shopify, a cloud-based e-commerce platform that lets companies create online storefronts, control inventory, and take payments, is one popular platform. Despite its great flexibility, Shopify requires monthly subscriptions and additional plugin purchases to expand capabilities, which tiny local pastry companies with tight budgets might not be able to afford. Similar features are provided by WooCommerce, a WordPress plugin that facilitates online ordering, product administration, and customer interaction. WooCommerce is primarily designed for online businesses, though, and until third-party solutions are included, it does not provide internal job separation between administrators and staff. Point-of-sale systems like Square POS and Lightspeed offer robust inventory and sales capabilities in addition to order management and customer relationship management (CRM) connections. However, these systems are primarily intended for in-store usage and sometimes come with service costs and hardware requirements, which limits their accessibility for small or freshly opened bakeries. A few specially designed bakery or restaurant management systems have been created for instructional or experimental reasons in academic and open-source environments. These, however, are frequently too small in scope, do not support multi-user roles or contemporary web technologies, and cannot be implemented in a real-world corporate setting without extensive adaptation. As a result, even with the availability of several platforms, there is still a need for an affordable, lightweight, and role-specific management system that is suited to the operational requirements and workflow of a pastry business. By providing a targeted solution that facilitates role-based interactions, digital order processing, and administrative control inside a single, integrated web application, the suggested Pastry Shop Management System seeks to close this gap.

4. REQUIREMENT SPECIFICATION

4.1 System Features

1. Registration

- The system shall prompt users to select a role (Admin, Employee, Customer) before registration.
- Based on the selected role, the system shall redirect users to a corresponding role-based registration form.
- The system shall validate all required fields (e.g., name, email, password, contact number) before account creation.
- Upon successful registration, the system shall display a confirmation message with a clickable login link.

Priority Level: High

Precondition: User must access the registration page with correct data.

2. System Login

- The system allows users (Admin, Employee, Customer) to login using a registered email and password.
- The system shall allow users to select their role from a dropdown before logging in.
- If login credentials are entered incorrectly more than three times, the system shall generate a random verification code for the user to retry login.
- If login attempts exceed five incorrect tries, the system shall block the user account for one hour.

Priority Level: High

Precondition: User has a valid email and password.

3. Add Customer (Employee)

- The system shall allow an Employee to register a new customer by entering details like name, email, contact number, and address.
- The system shall validate customer input data before saving.

Priority Level: Medium

Precondition: Employee must be logged in.

4. Update Customer (Employee)

- The system shall allow an Employee to update existing customer records.
- The system shall display a confirmation message after a successful update.

Priority Level: Medium

Precondition: Employee must be logged in and select a customer.

5. Delete Customer (Employee)

- The system shall allow an Employee to delete an existing customer record from the system.
- The system shall display a confirmation prompt before deletion.

Priority Level: Medium

Precondition: Employee must be logged in.

6. Add Product (Admin & Employee)

- The system shall allow Admins and Employees to add a new product with fields like name, category, price, quantity, and image.
- The system shall validate all product fields before saving.

Priority Level: High

Precondition: Admin or Employee must be logged in.

7. View All Products (All Roles)

- The system shall allow all users to view the list of available products with details.

Priority Level: High

Precondition: User must be logged in.

8. Update Product (Admin Only)

- The system should allow Admins to update product details such as name and price.
- The system shall validate the Inputs like name, category, manufacturing and expiration date.

Priority Level: High

Precondition: Admin must be logged in.

9. Delete Product (Admin Only)

- The system allows Admins to delete a product from the catalog.

Priority Level: High

Precondition: Admin must be logged in.

10. Add to Cart (Customer)

- The system shall allow Customers to select products and add them to a shopping cart.
- The system shall allow customers to adjust quantity before checkout.

Priority Level: High

Precondition: Customer must be logged in.

11. Order Products (Customer)

- The system shall allow customers to place an order for the items in their cart.
- The system shall display an order summary with product details and total cost before final submission.

Priority Level: High

Precondition: Customer must be logged in with items in cart.

12. Make Payment (Customer)

- The system should provide a payment module for customers to complete transactions.
- The system shall display payment success/failure status after completion.

Priority Level: High

Precondition: Customer has submitted an order.

13. View Sales (Admin & Employee)

- The system shall allow Admins and Employees to view a dashboard of completed sales with order IDs, customer info, total amount, and date.
- The system shall properly calculate the sales based on daily sales.

Priority Level: Medium

Precondition: User must be logged in as Admin or Employee.

14. Profile Management (All Roles)

- The system allows all users to view their personal profile information.
- The system allows Admins and Customers to update editable fields in their profiles.

Priority Level: Medium

Precondition: User must be logged in.

15. Software Help (All Roles)

Functional Requirements:

- The system should provide a "Help" or "Support" option accessible to all users (Admin, Employee, Customer) from their respective dashboards.
- The system shall display role-specific help content based on the logged-in user's role to guide them in navigating and using system functionalities.
- The help section shall include frequently asked questions (FAQs), usage instructions, and contact details for technical support.

Priority Level: Medium

Precondition: User must be logged in and, on the dashboard, or main navigation interface.

16. Software Logout (All Roles)

- The system shall provide a "Logout" option on the user interface that allows any logged-in user to safely end their session.
- Upon logout, the system shall terminate the current session and redirect the user to the login or home page.
- The system shall ensure that after logging out, no previous session data can be accessed by using the browser's back button or cached links.

Priority Level: High

Precondition: User must be logged in to the system.

4.2 System Quality Attributes

- **Usability**

A trained user shall be able to complete a product order, including browsing, cart management, and payment, within an average of five and a maximum of seven minutes. New users shall be able to register and log in to the system without assistance, provided they follow the on-screen instructions.

- **Performance**

The system shall be able to handle up to 100 concurrent users without degradation in page loading time (response time should not exceed 2 seconds per request). Search and retrieval of product listings shall occur in less than 1.5 seconds on average.

- **Reliability**

The system should be available and functional 99% of the time during business hours (8 AM to 10 PM). In case of unexpected system crashes, session data shall be auto saved and retrievable upon system recovery.

- **Security**

User passwords shall be stored in the database using secure hashing (e.g., SHA-256 or equivalent). The system shall automatically log out users after 10 minutes of inactivity to prevent unauthorized access. Role-based access control shall restrict features according to user type (Admin, Employee, Customer).

- **Maintainability**

The system should be modular and documented in such a way that future updates or feature additions can be made with minimal disruption. Developers shall be able to troubleshoot and resolve system-level bugs within two hours of reported downtime.

- **Scalability**

The system shall support expansion into additional user roles or modules (e.g., loyalty points, delivery tracking) with minimal structural changes. Database structure and queries shall be optimized for handling future growth in transaction volume.

- **Interoperability**

The system should be designed in a way that allows future integration with third-party services such as online payment gateways (e.g., PayPal, SSLCommerz) or delivery tracking systems. The application shall use standard communication protocols (HTTP/HTTPS) and data formats (JSON) for any external interaction.

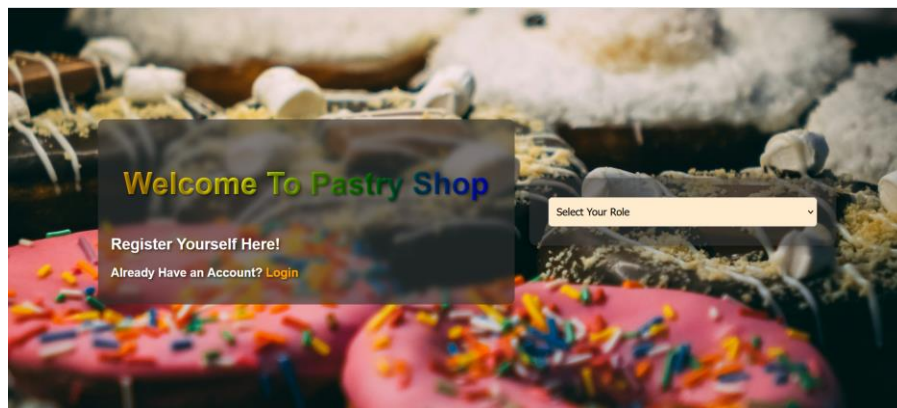
- **Availability**

The system shall ensure a minimum up time of 98% on a 24/7 basis. Routine maintenance or system updates shall be scheduled during off-peak hours with prior notice to users.

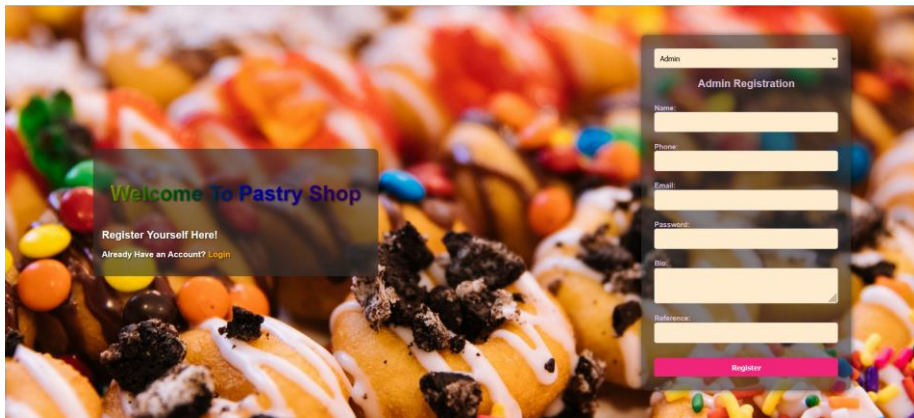
- **Portability**
The software shall be accessible through standard web browsers (Google Chrome, Mozilla Firefox, Microsoft Edge) across different platforms (Windows, macOS, Linux). The system's UI shall adapt responsively to desktop, tablet, and mobile screen resolutions.
- **Recoverability**
The system shall perform automatic backups of critical data (user accounts, orders, and sales) once every 24 hours. In case of data corruption or server failure, recovery procedures shall allow data restoration within a maximum of 2 hours.
- **Auditability**
The system shall log into user activities such as login attempts, profile updates, product management actions, and order placements for monitoring and accountability. Admin users shall have access to view selected audit logs through a secure interface.
- **Accessibility**
The interface shall support basic accessibility features such as keyboard navigation and appropriate color contrast for visually impaired users. All form elements shall include placeholder text and labels to assist users with screen readers.
- **Extensibility**
The system architecture shall allow developers to add new features (e.g., promo codes, order tracking, inventory alerts) without modifying existing core modules. New user roles or access privileges shall be defined through configuration changes and minimal code updates.

4.3 System Interface

○ Registration



Admin Registration Pannel



The Admin Registration Panel is overlaid on a background of assorted pastries. It features a dropdown menu set to 'Admin' and a title 'Admin Registration'. The form includes fields for Name, Phone, Email, Password, and Bio, followed by a Register button.

Welcome To Pastry Shop

Register Yourself Here!
Already Have an Account? [Login](#)

Admin

Admin Registration

Name

Phone

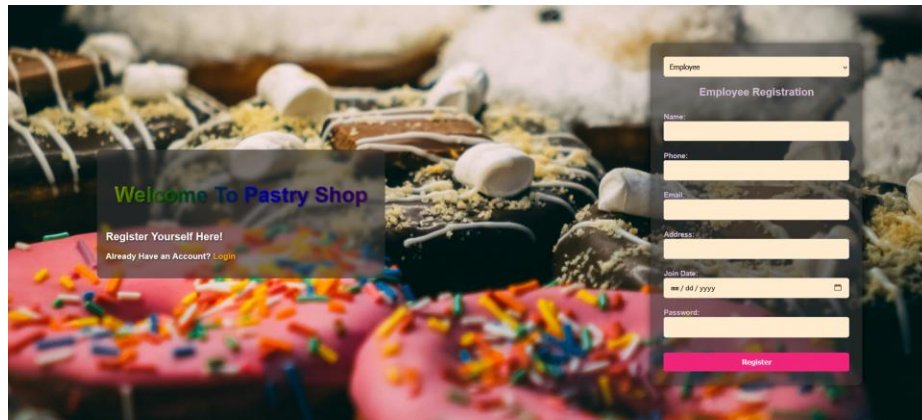
Email

Password

Bio

Register

Employee Registration Pannel



The Employee Registration Panel is overlaid on a background of pink donuts. It features a dropdown menu set to 'Employee' and a title 'Employee Registration'. The form includes fields for Name, Phone, Email, Address, Join Date (with a calendar icon), and Password, followed by a Register button.

Welcome To Pastry Shop

Register Yourself Here!
Already Have an Account? [Login](#)

Employee

Employee Registration

Name

Phone

Email

Address

Join Date

Register

Customer Registration Pannel



The Customer Registration Panel is overlaid on a background of a bakery display case filled with various pastries. It features a dropdown menu set to 'Customer' and a title 'Customer Registration'. The form includes fields for Name, Age, Gender (with a dropdown menu), Permanent Address, Delivery Address, Email, and Password, followed by a Register button.

Welcome To Pastry Shop

Register Yourself Here!
Already Have an Account? [Login](#)

Customer

Customer Registration

Name

Age

Gender

Permanent Address

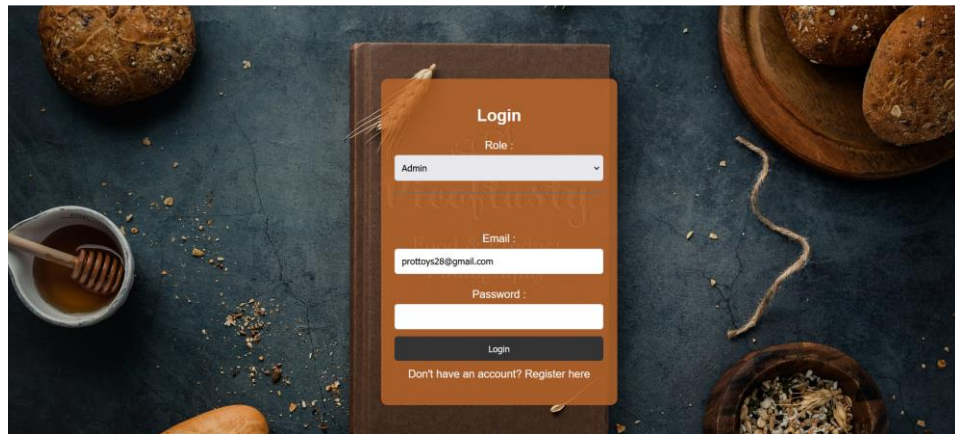
Delivery Address

Email

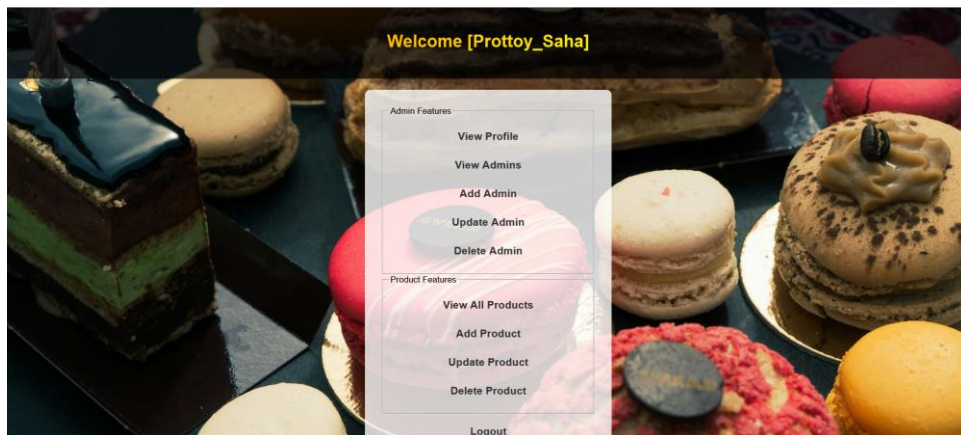
Password

Register

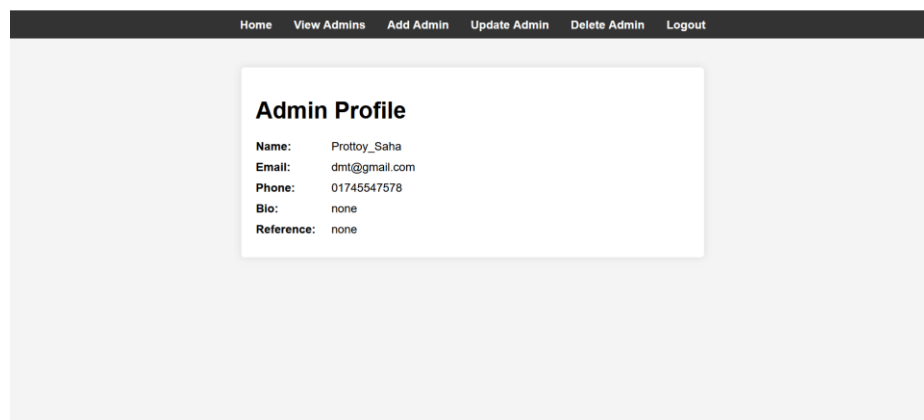
- **Login**



- **Admin Dashboard**



- **View Profile**



- **View Admins**

Name	Phone	Email	Bio	Reference
Pronoy Saha	01759705036	babitaroy336616@gmail.com	nai	nai
Sanjib Saha	01710475192	sanjib336616saha@gmail.com	not-applicable	not-applicable
Ibtida Kousik	01615664070	kousik55@gmail.com	not-applicable	not-applicable
MaCroDmT	01745547578	prottoys28@gmail.com	none	nai
Fardin Ubaid	01770366022	fardinkenway@gmail.com	Nothing	No ref
Fardin Ubaid	01770366022	fardinkenway@gmail.com	Nothing	No ref
Prottoysaha	01745547578	prottoys28@gmail.com	naiiiiiiii	nnnnnnnnoooooooooooo
Prottoy_Saha	01745547578	dmt@gmail.com	none	none

- **Add Admin**

[Home](#) [View Admins](#) [Update Admin](#) [Delete Admin](#) [Admin Profile](#) [Logout](#)

New Admin Registration

Admin Name

:

Email

:

Password

:

Phone Number

:

Reference

:

Bio

:

Add Admin

- **Update Admin**

[illegible]

- **Delete Admin**

		Home	View Admins	Add Admin	Update Admin	Admin Profile	Logout	
Name	Phone	Email	Bio	Reference	Actions			
Pronoy Saha	01759705036	babitaray336616@gmail.com	nai	nai	Delete			
Sanjib Seha	01710475192	sarjib.33661fcsaho@gmail.com	not-applicable	not-applicable	Delete			
Ibdida Kousik	01615664070	kousk55@gmail.com	not-applicable	not-applicable	Delete			
MacCroDmT	01745547578	prottoys28@gmail.com	none	nai	Delete			
Fardin Ubaid	01770366022	fardinkerway@gmail.com	Nothing	No ref	Delete			
Fardin Ubaid	01770366022	fardinkerway@gmail.com	Nothing	No ref	Delete			
ProttoySaha	01745547578	prottoys28@gmail.com	naaaaaaa	rnnnnnnnnnoooooooooooooo	Delete			
Protoy_Saha	01745547578	dmt@gmail.com	none	none	Delete			

- **View All Products**

[Home](#)[Add Product](#)[Update Product](#)[Delete Product](#)[Admin Profile](#)[Logout](#)

View All Products

Product Name	Category	Price	Production Date	Expire Date
red_velevet_pastry	pastry	250	2025-01-03	2025-01-04
rupchanda_cake	dophy	500000	2024-12-07	2025-01-11
MR BAGU_cake	BAGU	10000	2024-12-14	2025-01-10
chicken_pasta	pasta	55	2025-01-01	2025-01-03
Opera Pastry	Premium Pastry	300	2025-05-30	2025-06-03

- **Add Product**

[Home](#) [View All Products](#) [Add Product](#) [Update Product](#) [Delete Product](#) [Admin Profile](#) [Logout](#)

New Product Registration

Product Name

:

Category


:

Price

:


Production Date

:



Expire Date

:



Add Product

○ Update Product

HomeView All ProductsAdd ProductUpdate ProductDelete ProductAdmin ProfileLogout

Update Product

Name	Category	Price	Production Date	Expire Date	Select
red_velevet_pastry	pastry	250	2025-01-03	2025-01-04	Select
rupchanda_cake	dophy	500000	2024-12-07	2025-01-11	Select
MR.BAGU_cake	BAGU	10000	2024-12-14	2025-01-10	Select
chicken_pasta	pasta	55	2025-01-01	2025-01-03	Select
Opera Pastry	Premium Pastry	300	2025-05-30	2025-06-03	Select

Edit Product

Name:

Category:

Price:

Production Date:

mm / dd / yyyy

Expire Date:

mm / dd / yyyy

Update Product

○ Delete Product

HomeView All ProductsAdd ProductUpdate ProductAdmin ProfileLogout

Delete Product

Product Name	Category	Price	Production Date	Expire Date	Action
red_velevet_pastry	pastry	250	2025-01-03	2025-01-04	Delete
rupchanda_cake	dophy	500000	2024-12-07	2025-01-11	Delete
MR.BAGU_cake	BAGU	10000	2024-12-14	2025-01-10	Delete
chicken_pasta	pasta	55	2025-01-01	2025-01-03	Delete
Opera Pastry	Premium Pastry	300	2025-05-30	2025-06-03	Delete

○ Employee Dashboard

Welcome, Employee!

- [View Orders](#)
- [Take Order](#)
- [Update Order](#)
- [View Menu](#)
- [Logout](#)

Dashboard

Here you can view and manage all orders.

© 2025 Pastry Shop. All rights reserved.

- **Customer Dashboard**

Welcome, prottoys28@gmail.com!

- [View Products](#)
- [Buy Product](#)
- [View Cart](#)
- [Profile](#)
- [Logout](#)

4.4 Project Requirements

- **Budget**

Category	Estimated Cost (BDT)
Developer Salary (2 devs x 4 mo)	1,60,000
UI/UX Design & Graphics	30,000
Testing & Quality Assurance	25,000
Domain & Hosting (1 year)	10,000
Software Tools & Licenses	15,000
Documentation & Reporting	20,000
Maintenance & Support (6 mo)	50,000
Contingency Reserve (10%)	50,000
Total	3,60,000 BDT

- **Time Constraints**

Phase	Duration
Requirement Analysis	1 week
System Design (UI + Architecture)	3 weeks
Database Design	1 week
Development (Frontend + Backend)	5 weeks
Testing and Debugging	3 weeks
Deployment and Review	1 week
User Training & Documentation	1 week
Evaluation	1 week
Total Duration	16 weeks

- **Resource Constraints**

- **Human Resources:**
 - Project Manager
 - Web Developers (Frontend and Backend)
 - QA Engineer
 - UI/UX Designer
 - Documentation/Report Writer
- **Technical Resources:**
 - Laptops or Desktop Computers with Development Tools
 - Internet Access
 - Code Repository (e.g., GitHub)
 - Localhost Server (XAMPP or LAMP Stack)

- **Environmental Constraints**

- The system must be developed using HTML, CSS, JavaScript, PHP, AJAX, and MySQL.
- The software must be deployable in a shared or cloud hosting environment.
- The final system must be compatible with modern browsers (Google Chrome, Mozilla Firefox, Microsoft Edge).
- Development and testing shall be conducted in a local development environment, followed by deployment to a live hosting server.

5. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect because of other testing efforts. For example:

- **Third-Party Payment Gateway Integration**

Although the software may be designed to integrate with a payment gateway (e.g., PayPal or SSLCommerz), the functionality of the external payment service itself will not be tested. Only the integration point (data handoff, redirection, and return confirmation) will be verified from within the system. Testing of the gateway's internal operations, security, and performance is the responsibility of the payment provider.

- **Hosting Server Performance and Reliability**

The system is expected to be deployed in a web hosting environment. However, testing of server-side performance metrics such as CPU load, uptime reliability, or network bandwidth is not within the scope of this project. The performance of the hosting infrastructure is the responsibility of the hosting provider.

- **Customer-Side Spreadsheet or Report Processing**

Any analysis or reporting activities carried out by customers or stakeholders using exported data (e.g., sales reports in CSV format) are outside the scope of system testing. While the system will be tested for correct data export functionality, testing of third-party applications such as Microsoft Excel or Google Sheets used to manipulate this data will not be conducted. End users are responsible for validating their own data usage.

- **Browser Compatibility for Outdated Browsers**

Testing will not include outdated or legacy browsers such as Internet Explorer, as the system is intended to support only modern, standards-compliant browsers (e.g., Chrome, Firefox, Edge). Any issues resulting from deprecated browser use are considered unsupported.

6. TESTING APPROACH

6.1 Testing Levels

- The testing for the SMS project will consist of Unit, System/Integration (combined) and Acceptance test levels. It is hoped that there will be at least one full-time independent test person for system/integration testing. However, with the budget constraints and timeline established; most testing will be done by the test manager with the development teams' participation.
- UNIT Testing will be done by the developer and will be approved by the development team leader. Proof of unit testing (test case list, sample output, data printouts, defect information) must be provided by the programmer to the team leader before unit testing will be accepted and passed on to the test person. All unit test information will also be provided to the test person.
- SYSTEM/INTEGRATION Testing will be performed by the test manager and development team leader with assistance from the individual developers as required. No specific test tools are available for this project. Programs will enter System/Integration test after all critical defects have been corrected. A program may have up to two Major defects if they do not impede testing of the program (I.E. there is a work around for the error).
- ACCEPTANCE Testing will be performed by the actual end users with the assistance of the test manager and development team leader. The acceptance test will be done in parallel with the existing manual ZIP/FAX process for a period of one month after completion of the System/Integration test process.

6.2 Test Tools

- To ensure consistent and efficient testing of the Pastry Shop Management System, Selenium IDE has been selected as the primary testing tool. It is a browser-based, open-source automation tool that supports record-and-playback of user interactions, making it ideal for functional and regression testing.
Selenium IDE will be used to automate key workflows such as role-based login, customer registration, product management, order placement, and viewing reports. Its ability to replay recorded actions, set assertions, and handle waiting conditions helps reduce manual effort, ensures test accuracy, and improves overall test coverage. As a lightweight, browser-integrated tool, Selenium IDE is well-suited for testing web-based systems like this project, making the testing process more reliable and time-efficient.

6.3 Meetings

Effective communication among team members is essential for successful testing and smooth project execution. To facilitate this, a structured meeting plan will be followed during the testing phase of the Pastry Shop Management System. The testing team will hold weekly internal meetings to monitor ongoing test activities, share findings, and ensure that all team members are aligned with current priorities. These sessions will provide a platform to openly discuss any challenges encountered during testing, identify recurring errors, and decide on next steps for resolution. Such regular check-ins will help maintain momentum and ensure issues are addressed early. In addition to these internal reviews, the test team leader will meet bi-weekly with both the development team and the project manager. These meetings are designed to ensure that feedback from testing is properly integrated into the development workflow and that progress remains aligned with project goals. They also serve as a forum for discussing cross-functional concerns and coordinating bug fixes or updates. These two types of meetings weekly within the test team and bi-weekly with external teams will be scheduled for alternating weeks to avoid overlapping and ensure dedicated focus. Unplanned or emergency meetings may be called when critical bugs or unexpected system behaviors need to be resolved immediately to prevent delays. This approach ensures a collaborative environment, promotes transparency, and enables proactive issue management throughout the software testing lifecycle.

7. TEST CASES/TEST ITEMS

Admin Registration

Project Name: Pastry Shop Management System			Test Designed by: Prottoy	
Test Case ID: FR_1			Test Designed date: 30.05.2025	
Test Priority (Low, Medium, High): High			Test Executed by: Prottoy	
Module Name: Registration Session			Test Execution date: 05.06.2025	
Test Title: Register new user with valid email password and other desired criteria				
Description: Test the Registration page				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website 2. Press Registration Button 3. Select Role : Admin 4. Enter Name 5. Enter Phone Number 6. Enter Email address 7. Enter Password 8. Enter Bio 9. Enter Reference 10. Click Register	Name: Prottoy Saha Email: prottoy24@gmail.com Password: abc123 Bio: ABC Reference: ABC	If all the provided input fields are valid then the user should be able to register into the system	As expected	Pass
Post Condition: All the user data should be encrypted and saved into the database. After that a user should be able to login to his/her account				

Employee Registration

Project Name: Pastry Shop Management System		Test Designed by: Fardin		
Test Case ID: FR_2		Test Designed date: 30.05.2025		
Test Priority (Low, Medium, High): High		Test Executed by: Fardin		
Module Name: Registration Session		Test Execution date: 05.06.2025		
Test Title: Register new user with valid email password and other desired criteria				
Description: Test the Registration page				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website 2. Press Registration Button 3. Select Role as Employee 4. Enter Name 5. Enter Phone Number 6. Enter Email address 7. Enter Address 8. Enter Join Date 9. Enter Password 10. Click Register	Name: Prottoy Saha Phone Number: 01745547578 Email: prottoy24@gmail.com Address: Nikunja-2, Dhaka Join Date: 06/22/2025 Password: abc123	If all the provided input fields are valid then the user should be able to register into the system	As expected	Pass
Post Condition: All the user data should be encrypted and saved into the database. After that a user should be able to login to his/her account				

Customer Registration

Project Name: Pastry Shop Management System		Test Designed by: Kousik		
Test Case ID: FR_3		Test Designed date: 30.05.2025		
Test Priority (Low, Medium, High): High		Test Executed by: Kousik		
Module Name: Registration Session		Test Execution date: 05.06.2025		
Test Title: Register new user with valid email password and other desired criteria				
Description: Test the Registration page				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website 2. Press Registration Button 3. Select Role as Customer 4. Enter Name 5. Enter Age 6. Select Gender 7. Enter Permannet Address 8. Enter Delivery Address 9. Enter Phone Number 10. Enter Email address 11. Enter Password 12. Click Register	Name: Prottoy Saha Age: 23 Gender: Male Permanent address: Dhaka Delivery Address: Nikunja-2, Dhaka Phone Number: 01745547578 Email: prottoysaha@gmail.com Password: abc123	If all the provided input fields are valid then the user should be able to register into the system	As expected	Pass
Post Condition: All the user data should be encrypted and saved into the database. After that a user should be able to login to his/her account				

Login

Project Name: Pastry Shop Management System		Test Designed by: Rezuan		
Test Case ID: FR_4		Test Designed date: 30.05.2025		
Test Priority (Low, Medium, High): High		Test Executed by: Rezuan		
Module Name: Login Session		Test Execution date: 05.06.2025		
Test Title: verify login with valid username and password				
Description: Test website login page				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website 2. Enter Email 3. Enter password 4. Click Login	Username: prottoys28@gmail.com Password: wemethey4	User should login into the application	As expected	Pass
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				

View Profile

Project Name: Pastry Shop Management System		Test Designed by: Prottoy		
Test Case ID: FR_5		Test Designed date: 02.06.2025		
Test Priority (Low, Medium, High): High		Test Executed by: Prottoy		
Module Name: DashBoard		Test Execution date: 07.06.2025		
Test Title: Verification of View Profile Feature				
Description: Test website’s View Profile page				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website after successful login 2. Click on the "View Profile" or Profile tab/button 3. Check if all personal data fields are displayed properly 4. Verify that data displayed matches the stored user data	Name: Prottoy Saha Email: prottoys28@gmail.com Phone: 01745547578 Bio: none Reference: nai	Dashboard loads correctly after login Profile page shows fields like Name, Email, Phone, Role, and other user details	As expected	Pass
Post Condition: The user’s profile data is successfully retrieved from the database. No unauthorized or incorrect data is shown. The session remains active, and the user can continue using the system.				

View Admin

Project Name: Pastry Shop Management System		Test Designed by: Fardin		
Test Case ID: FR_6		Test Designed date: 02.06.2025		
Test Priority (Low, Medium, High): High		Test Executed by: Fardin		
Module Name: Dashboard		Test Execution date: 07.06.2025		
Test Title: Verification of View Profile Feature				
Description: Test website's View Profile page				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as an admin user 2. Click on the View Admins button from the dashboard's menu 3. Verify that a table of admin details is shown 4. Check if data matches entries from the database	Email: prottoys28@gmail.com Password: wenethey4	System should redirect to a page displaying the list of existing admins Admins are displayed in a structured format table with fields like Name, Email, Role, and Action buttons	As expected	Pass
Post Condition: The admin successfully views all registered admin records. The data is correctly pulled from the database without any unauthorized exposure or formatting errors. The admin remains authenticated in the current session.				

Add Admin

Project Name: Pastry Shop Management System			Test Designed by: Kousik	
Test Case ID: FR_7			Test Designed date: 02.06.2025	
Test Priority (Low, Medium, High): High			Test Executed by: Kousik	
Module Name: Admin Management			Test Execution date: 07.06.2025	
Test Title: Verification of Add Admin Functionality				
Description: Validate that the system allows an existing admin to successfully add/register a new admin user with all required input fields.				
Precondition (If any): User must be logged in with Admin credentials and must be on the “Add Admin” page.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as an admin user 2. Click on the Add Admin button from the dashboard’s menu 3. Enter Admin Name 4. Enter Email 5. Enter Password 6. Enter Phone 7. Enter Reference 8. Enter Bio 9. Click Add Admin Button 10. Automatically redirected into View Admins page and verify that the new admin appears in the table	Admin Name: Saha Email: sahaprottoy@admin.com Password: Admin@1234 Phone: 01745547578 Reference: Senior Admin Bio: Handles marketing section	Admin Registration form is displayed with all input fields Form accepts data and performs front-end validations Admin data is submitted, validated and finally stored in database and redirected to the View Admins page Recently added admin entry is visible in the admin list	As expected	Pass
Post Condition: A new admin record is inserted into the database. The system validates all input fields before submission. As confirmation, after inserting an new admin the Add Admin page should be redirected to the View Admins Page automatically.				

Update Admin

Project Name: Pastry Shop Management System			Test Designed by: Rezuan	
Test Case ID: FR_8			Test Designed date: 02.06.2025	
Test Priority (Low, Medium, High): High			Test Executed by: Rezuan	
Module Name: Dashboard			Test Execution date: 07.06.2025	
Test Title: Verification of View Profile Feature				
Description: Test website's View Profile page				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as an admin user 2. Click on the Update Admin button from the dashboard's menu 3. Click on any "Edit" button next to an admin entry 4. Modify data in one or more fields (for example: change Bio to "Sales Manager", Reference to "HR Lead") 5. Click the "Update Admin" button 6. Confirm that the updated data is reflected in the left-hand sides Admin table	New Bio: "Sales Manager" Reference: "HR Lead"	Dashboard loads correctly after the login System should redirect to a page displaying the list of existing admins and with each admin rows an Edit button should be shown. Corresponding admin's data should populate in the right-hand form fields Form accepts the updated values System updates the data in the database Changes appear in the list immediately	As expected	Pass
Post Condition: The selected admin's updated information is stored in the database. UI reflects the changes made after successful update. No duplication or unintended field overwrite occurs.				

Delete Admin

Project Name: Pastry Shop Management System			Test Designed by: Prottoy	
Test Case ID: FR_9			Test Designed date: 05.06.2025	
Test Priority (Low, Medium, High): High			Test Executed by: Prottoy	
Module Name: Admin Management – Delete Admin			Test Execution date: 09.06.2025	
Test Title: Verification of Delete Admin Functionality				
Description: Validate that an admin can delete another admin from the list and that the data is removed from both UI and the database				
Precondition (If any): User must be logged in as an Admin				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
<div>1. Log in as an admin user</div> <div>2. Click the "Delete" button next to a specific admin (e.g., Saha)</div> <div>3. Confirm the deletion</div> <div>4. Verify that the deleted admin is no longer in the list</div> <div>5. Click on any "Edit" button next to an admin entry</div> <div>6. Modify data in one or more fields (for example: change Bio to "Sales Manager", Reference to "HR Lead")</div> <div>7. Click the "Update Admin" button</div>	<div>Delete the admin with</div> <div>Admin email: saha@gmail.com</div>	<div>Admin list with Delete buttons is visible</div> <div>Admin is removed from list and database</div>	As expected	Pass
Post Condition: The selected admin's record is removed permanently from the database. The user interface reflects the deletion immediately.				

View Product

Project Name: Pastry Shop Management System		Test Designed by: Fardin		
Test Case ID: FR_10		Test Designed date: 05.06.2025		
Test Priority (Low, Medium, High): High		Test Executed by: Fardin		
Module Name: Product Management		Test Execution date: 09.06.2025		
Test Title: Verification of View All Products Feature				
Description: Ensure that users can view the complete list of available products with accurate details.				
Precondition (If any): User must be logged in as an Admin				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
<div>1. Login to system and go to "View All Products"</div> <div>2. Check if product table is visible</div> <div>3. Verify that product fields are shown: Name, Category, Price, Production Date, Expire Date</div> <div>4. Confirm listed data matches the database</div>	<div>Check if this item is shown in the product table:</div> <div>red_velevet_pastry = 250</div>	<div>Product list page opens</div> <div>Table displays correctly</div> <div>Products are visible (read-only)</div>	As expected	Pass
Post Condition: Product information is retrieved correctly from the database. All users can view product listings based on their access level (read-only or editable).				

Add Product

Project Name: Pastry Shop Management System		Test Designed by: Kousik		
Test Case ID: FR_11		Test Designed date: 05.06.2025		
Test Priority (Low, Medium, High): High		Test Executed by: Kousik		
Module Name: Product Management		Test Execution date: 09.06.2025		
Test Title: Verification of View Add Product Functionality				
Description: Verify that admin or employee can add a new product to the system with all required fields.				
Precondition (If any): User must be logged in as an Admin and must access the Add Product page.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
<div>1. Navigate to the "Add Product" page</div> <div>2. Enter Name</div> <div>3. Enter Category</div> <div>4. Enter Price</div> <div>5. Enter Production Date</div> <div>6. Enter Expire Date</div> <div>7. Click Add Product Button</div> <div>8. Check if product table is visible</div> <div>9. Verify that product fields are shown: Name, Category, Price, Production Date, Expire Date</div> <div>10. Confirm that after clicking website redirected to the View All Product page</div>	<div>Name: Opera Pastry</div> <div>Category: Premium Pastry</div> <div>Price: 300</div> <div>Production Date: 2025-06-01</div> <div>Expire Date: 2025-06-05</div>	<div>Product form is displayed, and Data is accepted in form fields</div> <div>Products are added to database.</div> <div>Recently added product appears into the View All Products Table</div>	As expected	Pass
Post Condition: Product is inserted into the database with valid attributes. Data is visible in the product listing section for all roles.				

Update Product

Project Name: Pastry Shop Management System			Test Designed by: Rezuan	
Test Case ID: FR_12			Test Designed date: 05.06.2025	
Test Priority (Low, Medium, High): High			Test Executed by: Rezuan	
Module Name: Product Management - Update Product			Test Execution date: 09.06.2025	
Test Title: Verification of View Update Product Functionality				
Description: Ensure that the system allows a valid user to select a product, edit its information, and save the updated record successfully.				
Precondition (If any): User must be logged in as an Admin or Employee and must access to the Update Product page.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the "Update Product" page 2. Click "Select" beside a product 3. Update one or more fields 4. Click "Update Product" 5. Verify the updated product appears in the left sided product list	Name; chicken_pasta New Price: 70 New Category: Savory	List of products is shown After Clicking into the Select button Product details are auto filled into the form Updated values are shown in the table	As expected	Pass
Post Condition: Product data is successfully updated and stored in the database. User interface reflects the changes without errors or duplication.				

Delete Product

Project Name: Pastry Shop Management System			Test Designed by: Prottoy	
Test Case ID: FR_13			Test Designed date: 10.06.2025	
Test Priority (Low, Medium, High): High			Test Executed by: Prottoy	
Module Name: Product Management - Delete Product			Test Execution date: 13.06.2025	
Test Title: Verification of View Delete Product Functionality				
Description: Ensure that an admin can successfully delete a product from the system and that it is removed from the product list and database.				
Precondition (If any): User must be logged in as an Admin and must access the Delete Product page.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login and go to product list page 2. Click "Delete" button beside a product row 3. Update one or more fields 4. Click "Update Product" 5. Verify the deleted product is not being shown in the left sided product list	Product Name: chicken_pasta	Product list loads Products are deleted from database and removed from table	As expected	Pass
Post Condition: The selected product is permanently deleted from the database. Product no longer appears in the product list.				

Log Out

Project Name: Pastry Shop Management System			Test Designed by: Kousik	
Test Case ID: FR_14			Test Designed date: 10.06.2025	
Test Priority (Low, Medium, High): High			Test Executed by: Kousik	
Module Name: Authentication of Logout			Test Execution date: 13.06.2025	
Test Title: Verification of the Logout Functionality				
Description: Ensure that when a user clicks the logout button, the session is properly terminated, and the user is redirected to the login page				
Precondition (If any): User must be logged in with a valid session.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as a valid user 2. Click the Logout button from the navbar/menu	Product Name: chicken_pasta	User is logged out and redirected to login page	As expected	Pass
Post Condition: User session is destroyed. Access to restricted pages is denied until the user logs in again.				

ITEM PASS/FAIL CRITERIA

For the Pastry Shop Management System, item pass/fail criteria are defined to assess the overall quality and readiness of the system's individual features and modules. A functionality will be considered as a pass if it performs according to the defined requirements and uses case expectations across all user roles Admin, Employee, and Customer. This includes accurate registration and login processes with proper role-based redirection, correct data retrieval in modules like "View Profile" or "View Products," successful record operations such as Add, Update, and Delete for both Admin and Product modules, seamless cart management and ordering functionality for customers, and stable logout with proper session termination. Furthermore, each module must maintain data integrity with the MySQL database, follow proper input validation, and handle edge cases gracefully (e.g., invalid logins or empty product entries). A failure status will be assigned to any feature that results in inconsistent behavior, incorrect database operations, UI malfunction, lack of expected confirmation or feedback messages, missing validations, or improper role-specific access. Failure to load dynamic content via AJAX, malfunctioning CRUD operations, or broken navigation links will also qualify as failure. Overall, a module must not only meet functional correctness but also demonstrate reliability, usability, and responsiveness under expected conditions to be marked as passed. The test process will be completed once the initial set of distributors have successfully sent in reassigned.

8. TEST DELIVERABLES

- Requirement Documentation
- Use Case
- System Interface Documentation
- Test Plan Document
- Unit Test Plans and Reports
- Integration Test Cases
- System Test Cases
- Test Execution Reports
- Defect Log
- Final Test Summary Report

9. STAFFING AND TRAINING NEEDS

A project management team will be formed first. This team will be responsible for everything in the project. Every major decision should be approved by the project management team before acting.

A HR team will be formed among the project management team for hiring human resources. The project management team will inform them how many and what type of resources they need, like system interface designer, software development team, quality assurance team, system testing team. Every member will be trained in preparing for the development and testing process as per requirements.

10. RESPONSIBILITIES

	TM	PM	Dev Team	Test Team	Client
Test cases documentation	X	X	X	X	
Test Procedures and rules	X		X	X	
Unit test documentation & execution			X	X	
Integration test Documentation & Execution	X		X	X	
System test Documentation & Execution		X		X	
System Design Reviews	X	X	X	X	X
Details Design Reviews	X	X	X	X	
Screen & Report prototype reviews	X	X		X	X
Change Control and regression testing	X	X	X	X	X
Acceptance test Documentation & Execution	X	X		X	X

11. TESTING SCHEDULE

Gantt Chart for Pastry Shop Management System

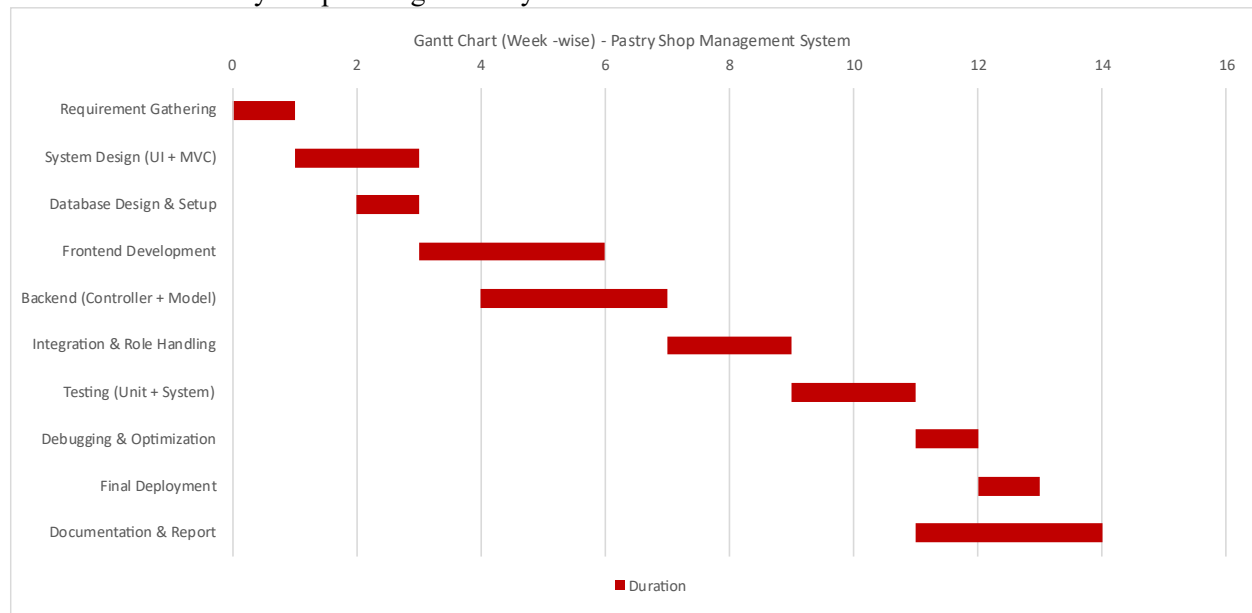


Fig: Gantt Chart

12. PLANNING RISKS AND CONTINGENCIES

Risk	Probability	Impact	Mitigation
Error in Database	Low	Low	Testing the database related frequently and maintaining the daily update
Error in function	Low	Medium	Testing the website frequently and maintaining the daily update.
Error while giving invalid input	Medium	High	The users or testers must test the input fields regularly by giving correct input data.
Loss of Data	High	High	Maintaining security checkup regularly.

13. APPROVALS

Project Sponsor	AIUB
QA Lead	Upoma
Team Manager	Protttoy Saha
Development Manager	Fardin Abu Ubaid
Project Manager	Muhtasib Ibtida Kousik
Business Analyst	Rezuan Hussain
Test Manager	Protttoy Saha