Introduction to Software Engineering

Software Testing

The student team is required to complete the **Software Testing** documentation for the assigned course project, following the attached template.

Software Engineering Department
Faculty of Information and Technology
University of Science

Table of Contents

Software Testing

Objectives

This document focus on the following topics:

- ✔ Completing the Software Testing document with the following sections:
  - Test Plan
  - Test Cases
- ✔ Understanding the Software Testing document.

1 Member Contribution Assessment

ID	Name	Contribution (%)	Signature
23127128	Nguyễn Thành Tiến	100%	
23127157	Nguyễn Hoàng Gia Bảo	100%	
23127184	Lê Phạm Kiều Duyên	100%	
23127396	Lương Linh Khôi	100%	

2 Test plan

*This section outlines the testing strategy, scope, environment, and resources required to verify that the **SweetieBakery** system meets the specifications defined in the Software Requirements Specification (SRS) and functions correctly for end-users.*

2.1 Introduction

The purpose of this test plan is to define the testing approach for the **SweetieBakery – Food Delivery Website**. The primary goal is to identify defects, ensure the software quality, and verify that both functional and non-functional requirements are met before the final release.

2.1.1 Scope of Testing

In Scope:

- **Functional Testing:** Verification of all core modules including:
  - Authentication & Account Management (User/Admin).
  - Product Catalog, Search, and Filtering.
  - Shopping Cart & Checkout Process (Cash On Delivery).
  - Order Management (User Tracking & Admin Processing).
  - Admin Dashboard & Reporting.
  - Reviews & Feedback System.
- **Non-Functional Testing:**
  - **UI/UX Testing:** Responsiveness on different screen sizes and browser compatibility.
  - **Performance Testing:** Page load speed verification.
  - **Security Testing:** Authentication barriers, authorization checks (RBAC), and input validation.

Out of Scope:

- Live Payment Gateway processing (Real banking transactions are simulated via "Banking Payment" manual check).

- Heavy Load/Stress Testing (e.g., 10,000 concurrent users).

2.2 Test Strategy

The project will adopt a combination of **Manual Testing** and **Unit Testing**.

2.2.1 Testing Levels

1. **Unit Testing:** Conducted by developers during the implementation phase to verify individual components (React components, Node.js functions).
2. **Integration Testing:** Verifying the communication between the Frontend (React), Backend (Express API), and Database (MongoDB).
3. **System Testing:** End-to-end testing of complete user flows (e.g., A user registers -> searches for a cake -> adds to cart -> places an order -> Admin approves the order).
4. **Acceptance Testing (UAT):** Validating the system against the user requirements to ensure it meets the stakeholders' needs.

2.2.2 Testing Techniques

- **Black Box Testing:** Testing functionalities without looking at the internal code structure. Focus on Input/Output.
- **Positive Testing:** Verifying the system works as expected with valid data.
- **Negative Testing:** Verifying the system handles invalid data and errors gracefully (e.g., entering a wrong password, adding out-of-stock items).

2.3 Test Environment

The testing will be performed in the following environment:

Component	Specification
Hardware	Personal Laptops (Windows 10/11, macOS)
Browser	Google Chrome (Latest), Firefox, Microsoft Edge
Backend Server	Node.js (Localhost:5000 or Deployed on Render)

Frontend Host	React.js (Localhost:3000 or Deployed on Vercel)
Database	MongoDB Atlas (Cloud Database)
Tools	Postman (API Testing), Chrome DevTools (UI Debugging), Jira/Excel (Bug Tracking)

2.4 Test Resources & Responsibilities

The following team members are responsible for the testing phase:

Role	Member	Responsibilities
Test Lead	Nguyễn Thành Tiến	- Define test strategy and schedule. - Review test cases. - Final quality sign-off.
Testers	Lê Phạm Kiều Duyên Lương Linh Khôi Nguyễn Hoàng Gia Bảo	- Execute test cases. - Report bugs and defects. - Perform regression testing after fixes.
Developers	(All Members)	- Fix reported bugs. - Perform unit testing on their assigned modules.

2.5 Test Schedule

The testing phase is aligned with the project timeline:

Phase	Activity	Estimated Time
Phase 1	Unit Testing (Concurrent with Development)	Weeks 6-8
Phase 2	Integration Testing (API & Database connection)	Week 8-9

Phase 3	System Testing (Execution of Test Cases)	Week 9-10
Phase 4	Bug Fixing & Regression Testing	Week 10
Phase 5	Final Review & Report	Week 10

2.6 Defect Management

Defects found during testing will be logged (in Excel or Jira) with the following attributes:

- **Defect ID:** Unique identifier.
- **Description:** Summary of the error.
- **Steps to Reproduce:** How to trigger the bug.
- **Severity:**
  - Critical: System crash or data loss.
  - High: Major functionality broken (e.g., cannot checkout).
  - Medium: Minor functionality issue or validation error.
  - Low: UI/Cosmetic issues (typos, alignment).
- **Status:** Open -> Fixed -> Verified.

2.7 Entry and Exit Criteria

Entry Criteria (When to start testing):

- The development of the module is complete.
- The test environment is set up and accessible.
- Test cases are written and reviewed.

Exit Criteria (When to stop testing):

- 100% of defined test cases have been executed.
- All Critical and High severity bugs are fixed and verified.
- 95% of total test cases have a "Passed" result.

3 Test cases

3.1 List of test cases

Seq	Test case	Target	Description
1. Authentication & Account Management			
TC-001	User Registration - Successful	Auth Module	Verify that a new user can register successfully with valid email and password.
TC-002	User Registration - Duplicate Email	Auth Module	Verify that the system displays an error when registering with an existing email.
TC-003	User Registration - Weak Password	Auth Module	Verify validation error when the password does not meet complexity requirements.
TC-004	User Registration - Mismatched Password	Auth Module	Verify error message when 'Password' and 'Confirm Password' do not match.
TC-005	User Login - Valid Credentials	Auth Module	Verify that a registered user can log in and is redirected to the Homepage.
TC-006	User Login - Invalid Password	Auth Module	Verify that the system displays an error message for incorrect passwords.
TC-007	User Login - Empty Fields	Auth Module	Verify validation when clicking Login with empty email or password fields.
TC-008	Admin Login - Access Dashboard	Auth Module	Verify that an Admin account is redirected to the Admin Dashboard upon login.
TC-009	Forgot Password - Send Email	Auth Module	Verify that a reset link is sent to the user's email when requesting a password reset.
TC-010	Reset Password - Success	Auth Module	Verify that the user can login with the new password after a successful reset.
TC-011	User Logout	Auth Module	Verify that clicking Logout terminates the session and redirects to Home/Login.

TC-012	Update User Profile	User Module	Verify that the user can update personal details (Name, Phone, Address).
TC-013	Change Password - Success	User Module	Verify that the user can change their password by providing the correct old password.
TC-014	Change Password - Wrong Old Pass	User Module	Verify error message when the user enters an incorrect old password.
2. Product Catalog & Browsing			
TC-015	View Product List	Product Module	Verify that the product catalog displays products with images, names, and prices.
TC-016	Search Product - Valid Keyword	Product Module	Verify that the search bar returns products matching the entered keyword.
TC-017	Search Product - No Results	Product Module	Verify the "No products found" message when searching for a non-existent item.
TC-018	Filter Product by Category	Product Module	Verify that selecting a category (e.g., Cakes) displays only relevant products.
TC-019	Sort Products by Price	Product Module	Verify that products can be sorted by Price (Low to High / High to Low).
TC-020	Product Pagination	Product Module	Verify that pagination works correctly when there are many products.
TC-021	View Product Details	Product Module	Verify that the detail page shows correct description, stock, price, and images.
TC-022	Product Image Zoom	Product Module	Verify that hovering over the product image zooms in or shows details.
TC-023	Add to Wishlist	Wishlist Module	Verify that a logged-in user can add a product to their wishlist.
TC-024	Remove from Wishlist	Wishlist Module	Verify that a user can remove a product from their wishlist.
3. Shopping Cart System			
TC-025	Add Item to Cart	Cart Module	Verify that adding an item updates the cart icon count and stores the item.

TC-026	Add Item - Out of Stock	Cart Module	Verify that the system prevents adding more items than the available stock.
TC-027	Update Cart Quantity - Increase	Cart Module	Verify that the item subtotal and cart total increase when quantity is increased.
TC-028	Update Cart Quantity - Decrease	Cart Module	Verify that the item subtotal and cart total decrease when quantity is decreased.
TC-029	Remove Item from Cart	Cart Module	Verify that clicking the delete button removes the item and updates the Total.
TC-030	Apply Promotion Code - Valid	Cart Module	Verify that a valid coupon code applies the correct discount percentage/amount.
TC-031	Apply Promotion Code - Invalid	Cart Module	Verify error message when applying an expired or non-existent coupon code.
TC-032	Cart Persistence	Cart Module	Verify that cart items remain saved after refreshing the page or re-logging in.
4. Checkout & Order Processing			
TC-033	Checkout - Successful (COD)	Order Module	Verify successful order placement using Cash On Delivery method.
TC-034	Checkout - Validation	Order Module	Verify that the system requires all mandatory shipping fields (Address, Phone).
TC-035	Checkout - Empty Cart	Order Module	Verify that the user cannot access the checkout page if the cart is empty.
TC-036	Order Confirmation Email	Order Module	Verify that the user receives a confirmation email after placing an order.
TC-037	View Order History	Order Module	Verify that the new order appears in the User Profile > Order History list.
TC-038	Track Order Status	Order Module	Verify that the user can see the real-time status (New, Processing, Shipping).
TC-039	Cancel Order - Valid	Order Module	Verify that the user can cancel an order if the status is still "New".
TC-040	Cancel Order - Invalid	Order Module	Verify that the user cannot cancel an order once it is "Shipping" or "Completed".

5. Product Reviews & Feedback			
TC-041	Submit Review - Success	Review Module	Verify that a user who purchased a product can submit a rating and comment.
TC-042	Submit Review - Unpurchased	Review Module	Verify that a user cannot review a product they have not purchased.
TC-043	View Product Reviews	Review Module	Verify that approved reviews are displayed correctly on the product page.
6. Admin Management (Backend)			
TC-044	Admin - Dashboard Stats	Admin Module	Verify that the dashboard widgets (Revenue, Orders) display accurate data.
TC-045	Admin - Create Product	Admin Module	Verify that Admin can add a new product with image, description, and price.
TC-046	Admin - Create Product (Validation)	Admin Module	Verify error when Admin tries to create a product without a name or price.
TC-047	Admin - Update Product	Admin Module	Verify that Admin can edit product details (e.g., update price or stock).
TC-048	Admin - Delete/Hide Product	Admin Module	Verify that Admin can deactivate a product so it no longer appears to users.
TC-049	Admin - Filter Orders	Admin Module	Verify that Admin can filter the order list by status (e.g., Show only 'New').
TC-050	Admin - Update Order Status	Admin Module	Verify that Admin can change order status (New -> Processing -> Delivered).
TC-051	Admin - View User List	Admin Module	Verify that Admin can view a list of all registered users.
TC-052	Admin - Lock/Unlock User	Admin Module	Verify that Admin can lock a user account to prevent them from logging in.
TC-053	Admin - Manage Promotions	Admin Module	Verify that Admin can create, edit, or delete discount coupons.
TC-054	Admin - Reply to Review	Admin Module	Verify that Admin can reply to customer feedback/reviews.

TC-055	Admin - Export Report	Admin Module	Verify that Admin can export sales reports to CSV or PDF format.
TC-056	Admin - View Logs	System Module	Verify that Admin can view system logs (errors, activities) for debugging.
TC-057	Admin - Backup Database	System Module	Verify that Admin can trigger a database backup successfully.
7. Communication & Notifications			
TC-058	Newsletter Subscription	System Module	Verify that a guest can subscribe to the newsletter with a valid email.
TC-059	Chat with Support	Chat Module	Verify that the chat widget opens and messages are sent/received in real-time.
8. Non-Functional & Security			
TC-060	URL Authorization Check	Security	Verify that a normal user cannot access Admin pages by typing the URL directly.
TC-061	SQL/NoSQL Injection	Security	Verify that input fields (Login, Search) verify against code injection attacks.
TC-062	Performance - Page Load	System	Verify that the Home page and Product page load within 3 seconds.
TC-063	Responsive - Mobile View	UI/UX	Verify that the layout adjusts correctly on mobile devices (Hamburger menu, etc).
TC-064	Broken Link Check (404)	System	Verify that accessing a non-existent URL displays a friendly 404 Error page.
TC-065	Session Timeout	Security	Verify that the user is automatically logged out after a period of inactivity.

3.2 Test case specifications

[Students should present the specifications of a few (10-15) of the most important test cases]

3.2.1 Test case 1

Test case	**TC006 - Verify system reports error for invalid password**
Related Use case	**U001-2** (Authentication and Access Control)
Context	User is on the Login Page. A valid account exists with email: user@sweetie.com and password: Password123.
Input Data	Email: user@sweetie.com,Password: WrongPass999
Expected Output	1. The system prevents login. 2. An error message **"Invalid username or password. Please try again"** is displayed on the screen. 3. The user remains on the Login Page.
Test steps	1. Navigate to the **Login Page**. 2. Enter user@sweetie.com into the Email field. 3. Enter WrongPass999 into the Password field. 4. Click the **"Login"** button. 5. Observe the system response.
Actual Output	The system prevented login. The error message "Invalid username or password. Please try again" was displayed correctly.
Result	Passed

3.2.2 Test case 2

Test case	**TC018 - Verify product filtering by Category**
Related Use case	**U010** (Product Catalog Display, Search and Filtering)
Context	User is on the **Products / Shop Page**. The database contains mixed products (Cakes, Cookies, Bread, etc.).
Input Data	Filter Selection: **"Bánh Kem"** (Cakes)
Expected Output	1. The product grid updates immediately. 2. Only products belonging to the **"Bánh Kem"** category are displayed. 3. Products from other categories are hidden.

Test steps	1. Navigate to the **Products Page**. 2. Locate the **Category Filter** sidebar or dropdown. 3. Click on the **"Bánh Kem"** (Cakes) category. 4. Observe the displayed product list.
Actual Output	The product grid updated immediately. Only "Bánh Kem" products were visible, and others were hidden.
Result	Passed

3.2.3 Test case 3

Test case	**TC030 - Verify discount code is applied and deduction is correct**
Related Use case	**U011-1** (Manage Shopping Cart) / **U007** (Promotion)
Context	User is on the **Shopping Cart Page**. Cart Subtotal is **200,000 VND**. A valid voucher SALE10 exists (Fixed discount: **20,000 VND**).
Input Data	Voucher Code: SALE10
Expected Output	1.A success message/tag (e.g., **"SALE10 applied"**) is displayed. 2. The **"Discount"** field in the Order Summary updates to show **-20,000 VND**. 3. The **"Total"** is recalculated correctly (200,000 - 20,000 = **180,000 VND**).
Test steps	1. Add items to the cart so Subtotal is 200,000 VND. 2. Navigate to the **Shopping Cart Page**. 3. Locate the **"Mã giảm giá"** (Voucher Code) input field. 4. Enter SALE20. 5. Click the **"Áp dụng"** (Apply) button. 6. Check the calculations in the Order Summary.
Actual Output	The success message was displayed. The discount of 20,000 VND was applied, and the Total updated to 180,000 VND correctly.
Result	Passed

3.2.4 Test case 4

Test case	**TC016 - Verify search bar returns correct products by name**
Related Use case	**U010** (Product Catalog Display, Search and Filtering)
Context	User is on the **Products / Shop Page**. The catalog contains "Lemon Cake" and "Chocolate Cookie".
Input Data	Search Query: **"Lemon"**
Expected Output	1. The product list filters to show only the **"Lemon Cake"**. 2. Products not matching the keyword (e.g., "Chocolate Cookie") are hidden. 3. If no match is found, a "No products found" message is displayed.
Test steps	1. Navigate to the **Products Page**. 2. Click on the **Search Bar** input field. 3. Type **"Lemon"**. 4. Press Enter (or wait for the auto-search debounce). 5. Observe the results grid.
Actual Output	The product list filtered correctly, showing only "Lemon Cake" and hiding non-matching items.
Result	Passed

3.2.5 Test case 5

Test case	**TC027 - Verify Total amount updates when changing item quantity**
Related Use case	**U011-1** (Manage Shopping Cart)
Context	User is on the **Shopping Cart Page**. Cart contains: **1x Cupcake** (Price: **20,000 VND**). Current Total: **20,000 VND**.
Input Data	Action: Click **(+) Plus button** on the quantity selector.
Expected Output	1. The quantity input changes from **1** to **2**. 2. The **Item Subtotal** updates to **40,000 VND**.

	3. The **Order Total** updates to **40,000 VND** (assuming no shipping/discounts yet).
Test steps	1. Navigate to the **Shopping Cart Page** with the item in the cart. 2. Locate the **Quantity Selector** (Minus, Input, Plus). 3. Click the **(+) Plus** button. 4. Observe the **Total** price update.
Actual Output	The quantity increased to 2. The Order Total immediately updated to 40,000 VND.
Result	Passed

3.2.6 Test case 6

Test case	**TC021 - Verify Product Detail Page displays correct price, description, and image**
Related Use case	**U010** (Product Catalog Display) / **U003** (Product Management)
Context	User is on the **Home Page** or **Products Page**. Target Product: **"Red Velvet"** (Price: 50,000 VND, Image: red_velvet.jpg, Desc: "Classic red cake").
Input Data	Click action on the **"Red Velvet"** product card.
Expected Output	1. Redirects to the **Product Detail Page**. 2. Product Name is **"Red Velvet"**. 3. Price is displayed as **50,000 VND**. 4. Description matches **"Classic red cake"**. 5. The correct image (red_velvet.jpg) is loaded and visible.
Test steps	1. Navigate to the product listing. 2. Click on the image or name of **"Red Velvet"**. 3. Wait for the Detail Page to load. 4. Verify the Name, Price, Description text, and Image against the expected data.
Actual Output	Redirected to the Detail Page successfully. The Name, Price, Description, and Image matched the expected data exactly.
Result	Passed

3.2.7 Test case 7

Test case	**TC002 – Verify system reports error for duplicate email during registration**
Related Use case	**U001-1** (User Registration)
Context	User is on the Registration Page. An account already exists with email: **user@sweetie.com**.
Input Data	Email: **user@sweetie.com** Username: **user123** Password: **Password123** Confirm Password: **Password123**
Expected Output	1. The system prevents account creation. 2. An error message **"Email already exists. Please use another email."** is displayed. 3. The user remains on the Registration Page.
Test steps	1. Navigate to the Registration Page. 2. Enter "**user@sweetie.com"** into the Email field. 3. Enter "**user123"** into Username field. 4. Enter "**Password123** into the Password field. 5. Enter "**Password123"** into the Confirm Password field. 6. Click the **"Register"** button. 7. Observe the system response.
Actual Output	The system prevented registration. The error message **"Email already exists. Please use another email."** was displayed correctly.
Result	Passed

3.2.8 Test case 8

Test case	**TC-004 – Verify system reports error for invalid password**
Related Use case	**U001-2** (Authentication and Access Control)
Context	User is on the Login Page. A valid account exists with email: **user@sweetie.com** and password: **Password123**.
Input Data	Username: **WrongUsername** Password: **WrongPass999**
Expected Output	1. The system prevents login. 2. An error message "Invalid username or password. Please try again" is displayed. 3. The user remains on the Login Page.
Test steps	1. Navigate to the Login Page. 2. Enter "**WrongUsername"** into the Username field. 3. Enter **"WrongPass999"** into the Password field. 4. Click the Login button. 5. Observe the system response.
Actual Output	The system prevented login. The error message "Invalid username or password. Please try again" was displayed correctly.
Result	Passed

3.2.9 Test case 9

Test case	**TC-007 – Verify user can update profile information**
Related Use case	**U001-4** (User Profile Management)

Context	User is logged in and is on the **User Profile** page. The user account already exists with initial profile information.
Input Data	Fullname: **Nguyễn Văn A** Age: **22** Phone Number: **0123456789** Address: **123 Nguyễn Trãi, Hà Nội**
Expected Output	1. The system updates the user profile successfully. 2. A success message such as **"Profile updated successfully"** is displayed. 3. Updated information (Fullname, Age, Phone Number, Address) is saved and displayed correctly on the Profile page.
Test steps	1. Login with a valid user account. 2. Navigate to **Update User Profile** page. 3. Enter new values for Fullname, Age, Phone Number, and Address. 4. Click the **Save Profile** button. 5. Refresh the page or revisit the Profile page.
Actual Output	The system saved the updated profile information and displayed the success message. All updated fields were shown correctly after refresh.
Result	Passed

3.2.10 Test case 10

Test case	**TC-018 – Verify user leaves empty address when checkout**
Related Use case	**U011-2** (Proceed to Checkout and Confirm Order)
Context	User is logged in and has at least one product in the shopping cart. User is on the Checkout Page.

Input Data	Fullname: **Nguyễn Văn A** Phone Number: **0123456789** Address: *(left empty)* District: **10** City: **Ho Chi Minh City**
Expected Output	1. The system prevents order submission. 2. A validation message such as **"Shipping address is required"** is displayed. 3. The user remains on the Checkout Page.
Test steps	1. Navigate to the Checkout Page. 2. Enter "**Nguyễn Văn A**" in the Full Name field. 3. Enter "**0123456789**" in the Phone Number field. 4. Leave the Address field empty. 5. Enter "**10**" in the District field. 6. Enter "**Ho Chi Minh City**" in City field. 6. Click the **Next** button. 7. Observe the system response.
Actual Output	The system prevented checkout and displayed the message **"Shipping address is required"**.
Result	Passed

3.2.11 Test case 11

Test case	**TC-025 - Verify that adding an item updates the cart icon count and stores the item**
Related Use case	**U011.1 (Manage Shopping Cart)**
Context	User is logged in and viewing the Product Detail Page of "Tiramisu Cake". The Shopping Cart is currently empty (Cart count is 0). The product "Tiramisu Cake" is in stock.
Input Data	Product: Tiramisu Cake

	Quantity: 1 Action: Click "Thêm vào giỏ hàng" (Add to Cart)
Expected Output	1. A success message (e.g., "Đã thêm sản phẩm vào giỏ") is displayed temporarily. 2. The Cart Icon badge updates from 0 to 1. 3. The item "Tiramisu Cake" is stored in the cart database with the correct price and quantity.
Test steps	1. Navigate to the Product Detail Page of "Tiramisu Cake". 2. Click the "Thêm vào giỏ hàng" (Add to Cart) button. 3. Observe the notification message. 4. Click on the Cart Icon to verify the item is listed inside.
Actual Output	The success message was displayed. The Cart Icon count updated to 1. The "Tiramisu Cake" appeared correctly in the cart list.
Result	Passed

3.2.12 Test case 12

Test case	**TC-045 - Verify that Admin can add a new product with image, description, and price**
Related Use case	**U003-1 (Product Management System - Create)**
Context	Admin is successfully logged into the system. Admin is currently on the Products Management page.
Input Data	Product Name: Bánh Mousse Dâu Tây Price: 35,000 Stock Level: 50 (Positive integer required) Description: "Bánh mì giòn tan với sốt bơ tỏi đặc biệt." Image: “strawberry_mousse.jpg”
Expected Output	1. System saves the new product record to the database. 2. A confirmation message "Product **Bánh Mousse Dâu Tây** created successfully" is displayed. 3. The browser redirects the Admin to the product list or the product's detail page. 4. The new product appears in the catalog with correct details (Name, Price, Stock).
Test steps	1. Navigate to the Product Management list. 2. Click the "Add New Product" button.

	3. Fill in the required fields: Name ("**Bánh Mousse Dâu Tây**"), Price (45,000), Stock Level (20), and Description. 4. Upload the image "strawberry_mousse.jpg" 5. Click the "Save" or "Create Product" button.
Actual Output	The product was saved successfully. The message "Product **Bánh Mousse Dâu Tây** created successfully" appeared. The system redirected to the Product List where the new item was visible.
Result	Passed

3.2.13 Test case 13

Test case	**TC-054 - Verify that Admin can reply to customer feedback/reviews**
Related Use case	**U006-3 (Respond to Review)**
Context	Admin is logged in and is on the Feedback Management page. A customer (User B) has left a review: "Bánh Tiramisu rất thơm và mềm, không bị ngọt quá." regarding the product "Bánh Tiramisu". The review status is currently "Approved"
Input Data	Selected Review: Review from "Bánh Tiramisu". Response Content: "Cảm ơn bạn đã yêu thích món Bánh Tiramisu của tiệm! Hy vọng được phục vụ bạn lần tới ạ." Action: Click "Post Response" button.
Expected Output	1. The system validates the input and saves the response to the database 2. The original feedback's status updates to "Responded" 3. A success message "**Response posted successfully**" is displayed 4. The Admin's response appears on the interface linked to the original review
Test steps	1. Navigate to the Feedback Management list. 2. Locate the review for "Bánh Tiramisu". 3. Click the "Respond" button. 4. Type the content: "Cảm ơn bạn đã yêu thích món Bánh Tiramisu của tiệm! Hy vọng được phục vụ bạn lần tới ạ." 5. Click the "Post Response" button.
Actual Output	The success message "Response posted successfully" was displayed. The review status changed to "Responded", and the reply was visible under the customer's comment.
Result	Passed