



CIS 423 Web-Based System

HANAMI

FLORAL SHOP



Instructor: Mrs. Rawan Algamdi

2025-2026

Group members:

No.	Student name	ID number
1.	Manar Alqahtani (Leader)	2220002785
2.	Zahra Saeed Bahyan	2220009005
3.	Budoor Alturki	2210004707
4.	Layan Suliman Ghazwani	2220004241
5.	Munirah Aljindan	2210003442
6.	Fatima Alomir	2210002981
7.	Fatima Alyami	2220005758
8.	Fatimah Al Muhaissen	2210002838

8SF01 | Group no.4 | CS

Submission Date: May 4, 2025

Table of Contents

1. Work Division Between Members:	7
2. Database Design:	7
3. Web Application Interface:	10
1. Homepage	10
2. Expanded Product Details Page	12
3. Cart Page	13
4. Checkout Page	14
5. Admin Login Page	15
6. Manage Products (Admin Dashboard)	15
7. Products Page	18
	19
8. Contact Us Page	19
4. Fulfilled Requirements:	21
5. Testing Cases + Screenshots:	21
1. Homepage Testing	21
2. Product Details Page	21
3. Add to Cart Functionality	24
4. Shopping Cart Page	28
5. Checkout Page	32
6. Admin Login and Access	40
7. Manage Products (Admin Panel)	42
8. Contact Us Page	44
9. Past Purchases with Cookies	46
10. Form Validation	47
11. Accessibility Testing	49
12. Code Efficiency and Structure	49
1. Issues Faced by the Business (Floral Shop)	49
2. Technological Features Solving the Problems	49
3. New Challenges Introduced by the System	49
4. Tools That Helped Build the System	50

Table of Table

Table 1: Testing the Product Details Page Functionality	24
Table 2: Testing the Add to Cart Functionality	28
Table 3 Cart functionality,	32
Table 4 Checkout Functionality	39
Table 5 Admin Login Test.....	40
Table 6 Add product test	42
Table 7 Edit product test	42
Table 8 Delete product test	43
Table 9 Searching product test.....	43
Table 10 Contact Form	44
Table 11Cookies Functionality	46
Table 12 Form validation.....	47

Table of Figures

Figure 1 HANAMI database ER Diagram.....	7
Figure 2 Product Table Filled with Data.....	8
Figure 3 Product Table Structure	8
Figure 4 Admin Table Filled with Data	9
Figure 5 Admin Table Structure.....	9
Figure 6 DB Indexes.....	9
Figure 7 DB Overview.....	9
Figure 8 Design Model	10
Figure 9 Homepage of Hanami.....	12
Figure 10 Product details	12
Figure 11 Shopping Cart.....	13
Figure 12 Checkout page	14
Figure 13 Admin login.....	15
Figure 14 Admin dashboard.....	15
Figure 15 Adding new product	16
Figure 16 Deleting product.....	16
Figure 17 Editing products	17
Figure 18 search/filter options	17
Figure 19 Exploring products	18
Figure 20 Message after adding a product to cart.....	18
Figure 21in Product page a button showing “Max Quantity Reached” button when item limit is met	19
Figure 22 Contact us page	20
Figure 23 Homepage entry point to access the product catalog	22
Figure 24 shows all products with “View” and “Add to Cart” options	22
Figure 25 Displays product image, name, price, description, quantity input, “Add to Cart” button, and help popup link	23
Figure 26 Displays help window with answers to common questions about quantity, adding to cart, delivery, and support.....	23
Figure 27 : Product Listing with Add to Cart Options.....	25
Figure 28 Displays a validation message when a user enters a quantity greater than available stock	25
Figure 29 Shows an input of 10 when max is 5. Error tooltip: “Value must be less than or equal to 5	26
Figure 30Message After Adding Product to Cart	26
Figure 31 Displays a warning message when the user attempts to add more items than the available stock in the cart	29
Figure 32 Shows the cart page with the updated quantity and total after modifying the item quantity	29
Figure 33 Demonstrates the shopping cart with a single item after delete one item	30
Figure 34 Shows a pop-up confirmation when the user chooses to empty the cart.....	30
Figure 35 Displays the cart page with no items, confirming the cart is empty after deletion	31
Figure 36 Shows the billing details form on the checkout page, allowing users to input personal information	33
Figure 37 Shows the order summary page, including product details, total cost, and the option to complete the transaction using a payment method like PayPal.....	33
Figure 38 Shows the order summary page, including product details, total cost, and the option to complete the transaction using a payment method like PayPal.....	34
Figure 39 Displays the country selection dropdown, where users can choose their country from a predefined list of options	34

Figure 40 Depicts the form submission error message when the user attempts to proceed with incomplete or invalid fields, ensuring all required information is provided	35
Figure 41 Displays a validation error for the email field.....	35
Figure 42 A validation error indicating missing or incorrect phone number input during the checkout process, ensuring proper data entry before proceeding	36
Figure 43 A validation error indicating missing or incorrect phone number input during the checkout process, ensuring proper data entry before proceeding	36
Figure 44 Success message after a user has successfully completed their order.....	37
Figure 45 Presents the "Your Previous Purchases" section, which displays the user's previous orders for reference and potential re-ordering and the cart is empty after completing the order.....	37
Figure 46 The database shows 5 white tulip vases available before the customer places the order.....	38
Figure 47 The database updates successfully, reducing the white tulip vase quantity to 0 after the customer places the order.	38
Figure 48 The white tulips vase is now marked as "Out of Stock," indicating the database was updated successfully.....	39
Figure 49 An error message appears when the admin username is incorrect	40
Figure 50 An error message appears when the username is valid, but the password is incorrect	41
Figure 51Manage product page when the admin login successfully	41
Figure 52 Contact Us - Success Message	45
Figure 53 Contact Us-empty fields alert.....	45
Figure 54 Past Purchases Appearing on Homepage	46
Figure 55 Contact Us- email format	47
Figure 56 Admin Login: Empty Field alert	48
Figure 57 Checkout: Empty Field alert	48

1. Work Division Between Members:

Frontend Team

- **Manar Alqahtani:** Team leader. Divided the work, ensured everything stayed on track, and combined the individual contributions into a cohesive final product. Also designed the homepage layout and intuitive navigation menu.
- **Budoor Alturki:** Created dynamic product listing pages and bouquet customization UI.
- **Fatimah Al Muhaissen:** Developed the cart interface and order checkout flow.
- **Fatima Alyami:** Handled form validation with JavaScript and ensured responsive, accessible design.

Backend Team

- **Zahra Bahyan:** Designed the ER diagram and implemented user login system with session control.
- **Fatima Alomir:** Built the admin panel to manage inventory and orders (CRUD operations).
- **Munirah Aljindan:** Developed the core cart and order processing logic using PHP sessions.
- **Layan Ghazwani:** Added a “Past Purchases” feature using cookies and sessions and led end-to-end testing.

2. Database Design:

ER Diagram

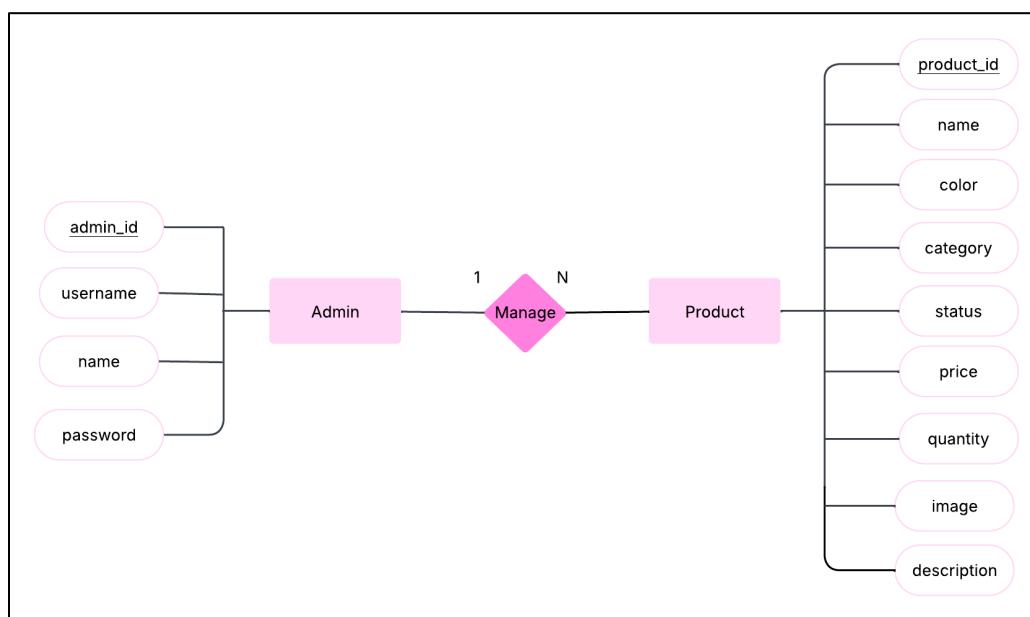


Figure 1 HANAMI database ER Diagram

Main Tables**Product Table**

<input type="checkbox"/>				#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
<input type="checkbox"/>				1	Tulip	White Tulips Vase	190.99 1.jpg	A peaceful display of pure white tulips in a simpl...	1	5	available	white			
<input type="checkbox"/>				2	Tulip	Orange and Yellow Tulips Vase	190.99 2.jpg	Bold orange tulips with yellow tips, elegantly arr...	1	5	available	orange yellow			
<input type="checkbox"/>				3	rose	Red Tulips in White Vase	190.99 3.jpg	Deep red tulips with lush green leaves tied with a...	1	2	available	red			
<input type="checkbox"/>				4	tulip	Pink Tulips in Glass Vase	219.99 4.jpeg	A harmonious blend of pink tulips in a clear vase,...	1	5	available	pink			
<input type="checkbox"/>				5	Tulip	Mixed Pink and Yellow Tulips Vase	159.99 5.jpg	Yellow and red tulips in a textured vase bring a c...	1	5	available	pink yellow			
<input type="checkbox"/>				6	Tulip	Purple Tulips Vase	219.99 6.jpg	Elegant lavender tulips tied with a jute band, add...	1	5	available	purple			
<input type="checkbox"/>				7	Tulip	Lavender Tulips Vase	359.99 7.jpg	A royal arrangement of abundant lavender tulips in...	1	5	available	lavender			
<input type="checkbox"/>				8	Astroemeria	White Astroemeria Vase	259.99 8.jpg	A full bouquet of soft white astroemeria blooms w...	2	5	available	white			
<input type="checkbox"/>				9	Astroemeria	Pink Astroemeria Vase	259.99 9.jpg	Bright pink astroemeria flowers in a round vase, ...	2	5	available	pink			
<input type="checkbox"/>				10	Mixed	Pink Tulips & White Astroemeria Vase	329.99 10.jpg	Pink tulips blended with delicate white astroemer...	2	0	unavailable	pink white			
<input type="checkbox"/>				11	Astroemeria	Red Astroemeria Vase	259.99 11.jpg	Vibrant red astroemeria flowers in a stylish glas...	2	5	available	red			
<input type="checkbox"/>				12	Rose	Red Roses Vase	229.99 12.jpg	An iconic bouquet of fresh red roses for your spec...	5	5	available	red			
<input type="checkbox"/>				13	Rose	White Roses Vase	229.99 13.jpg	A luxurious bouquet of white roses that radiates p...	5	5	available	white			
<input type="checkbox"/>				14	Rose	Pink Roses Vase	229.99 14.jpg	Delicate pink roses elegantly bundled in a simple ...	5	5	available	pink			
<input type="checkbox"/>				15	Rose	Red & Pink Roses Vase	229.99 15.jpg	Pink and red roses paired for the perfect romantic...	5	5	available	red pink			
<input type="checkbox"/>				16	Rose	Red & White Roses Vase	229.99 16.jpg	A passionate blend of red and white roses to say "...	5	5	available	red white			
<input type="checkbox"/>				18	tulip	Pink Tulips in White Vase	219.99 18.jpg	Fresh pink tulips tied with a red ribbon in a mode...	1	3	available	pink			

Figure 2 Product Table Filled with Data

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
1	product_id	int(11)		No	None		AUTO_INCREMENT			
2	category	varchar(50)	utf8mb4_general_ci	No	None					
3	name	varchar(50)	utf8mb4_general_ci	No	None					
4	price	decimal(10,2)		No	None					
5	image	varchar(200)	utf8mb4_general_ci	No	None					
6	description	varchar(300)	utf8mb4_general_ci	No	None					
7	admin_id	int(11)		No	None					
8	quantity	int(11)		No	0					
9	status	varchar(20)	utf8mb4_general_ci	No	available					
10	color	varchar(100)	utf8mb4_general_ci	Yes	NULL					

Figure 3 Product Table Structure

Admin Table

		admin_id	Name	username	password
<input type="checkbox"/>	Edit Copy Delete	1	Manar Alqahtani	admin_manar	manar362003
<input type="checkbox"/>	Edit Copy Delete	2	Layan Ghazwani	admin_layan	layan12345
<input type="checkbox"/>	Edit Copy Delete	3	Zahra Bahyan	admin_zahra	zahra2000
<input type="checkbox"/>	Edit Copy Delete	4	Budoor Alturki	admin_budor	budor123
<input type="checkbox"/>	Edit Copy Delete	5	Fatimah Al Muhaissen	admin_fatemahM	fatemah123
<input type="checkbox"/>	Edit Copy Delete	6	Fatima Alomir	admin_fatemahO	fatemah123
<input type="checkbox"/>	Edit Copy Delete	7	Fatima Alyami	admin_fatemahY	fatemah123
<input type="checkbox"/>	Edit Copy Delete	8	Munirah Aljindan	admin_Munirah	Munirah2020

Figure 4 Admin Table Filled with Data

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 admin_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 Name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 username	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 password	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More

Figure 5 Admin Table Structure

Indexes								
Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	product_id	17	A	No	
admin_id	BTREE	No	No	admin_id	8	A	No	

Figure 6 DB Indexes

The screenshot shows the phpMyAdmin interface for the 'hanami' database. The left sidebar lists databases and tables. The main area shows the 'Structure' tab for the 'admin' table, with columns: id, Name, username, and password. The 'Operations' tab is selected, showing options like Browse, Structure, Search, Insert, Empty, Drop, etc. Below the table structure, there's a summary of 2 tables and 25 rows.

Figure 7 DB Overview

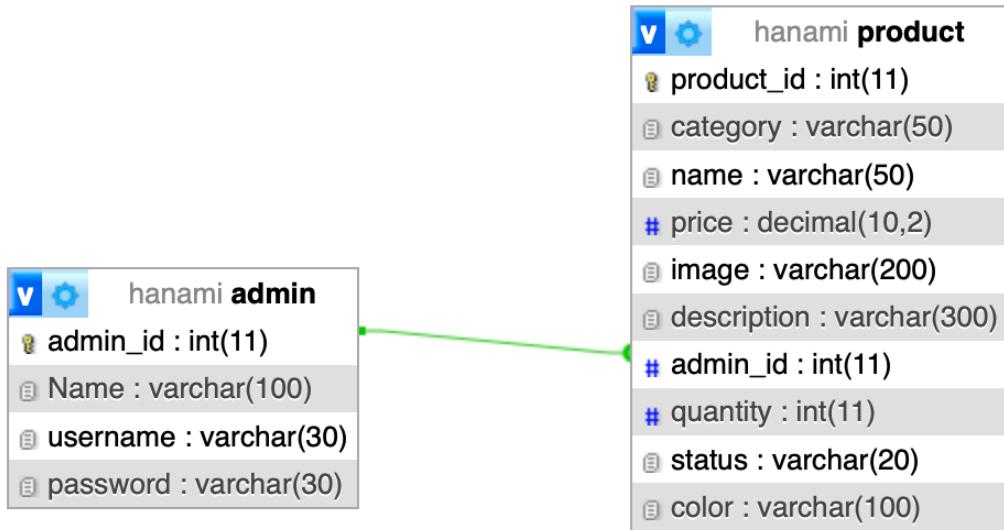


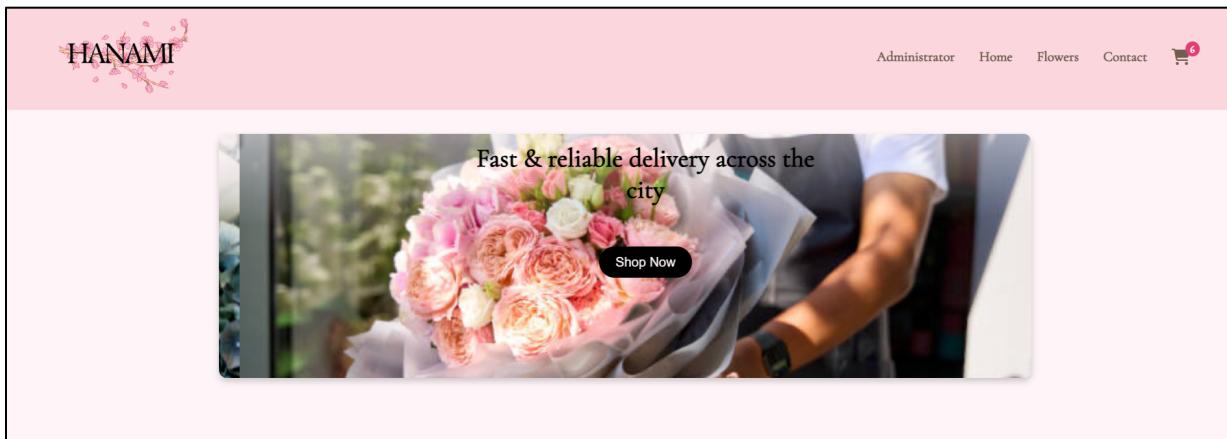
Figure 8 Design Model

3. Web Application Interface:

The HANAMI platform includes the following key interfaces:

1. Homepage

- Displays featured products and categories.
- Includes site navigation (Administrator, Home, Shop, Contact Us, Cart).



Flowers for the



Gifts

Surprise someone special with a beautiful bouquet that brings a smile.

[Shop Now](#)

Celebrate

Celebrate your special moments with elegant blooms full of joy and charm.

[Shop Now](#)

Categories



Tulips



Roses



Alstroemeria

About Us

We believe flowers speak louder than words. Our mission is to deliver the freshest and most beautiful floral arrangements to your doorstep with love and care.

The Talented Team Behind HANAMI

A team of highly-skilled experts in floristry, design, and customer service – working behind the scenes to create memorable moments for every occasion.



Manar Alqahtani



Zahra Bahyan



Layan Ghazwani



Fatimah Al Muhaissen
Customer Service Manager



Munirah Aljindan
Marketing Coordinator



Fatimah Alomair
Website Developer



Fatima Alyami
Creative Director



Budoor Alturki
Visual Stylist

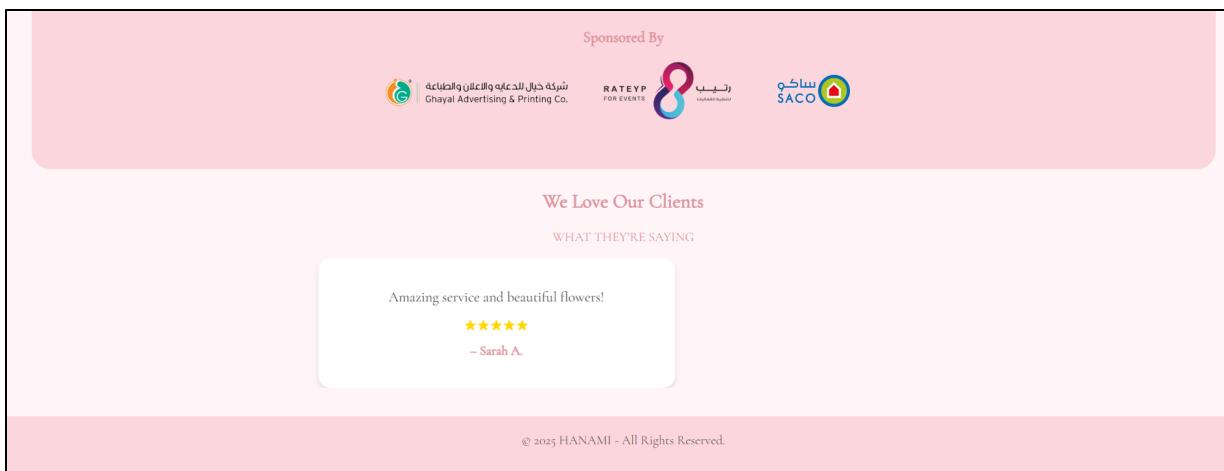


Figure 9 Homepage of Hanami.

2. Expanded Product Details Page

- Shows detailed view of each product.
- Includes description, image, price, and add-to-cart button.

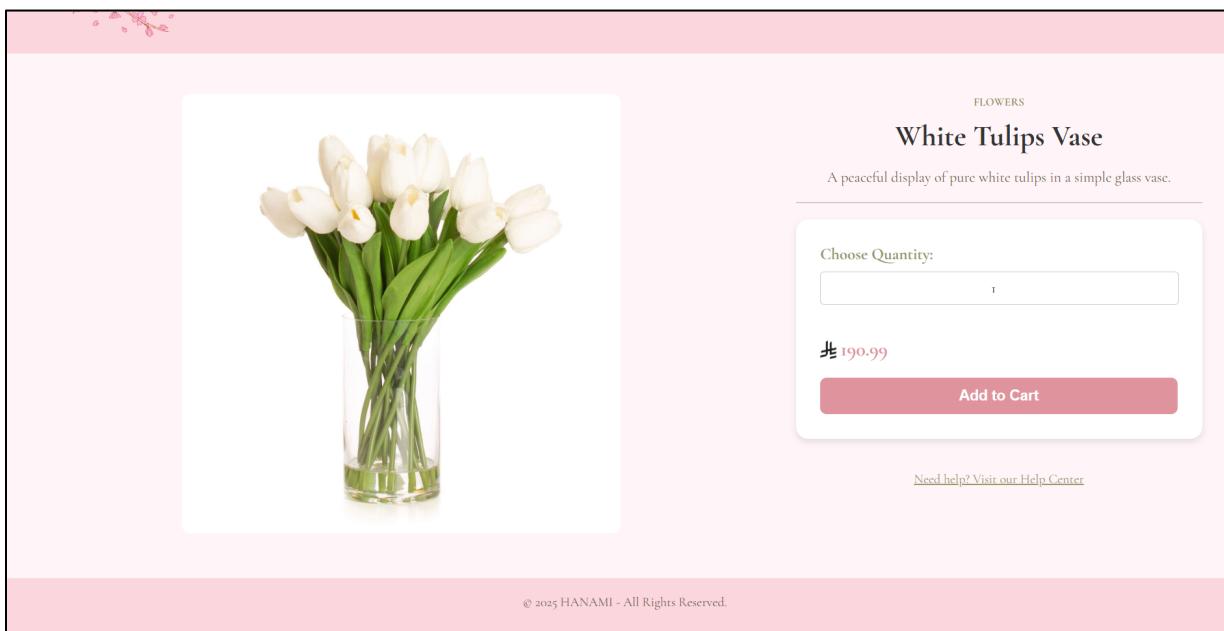


Figure 10 Product details

3. Cart Page

- Allows users to view, edit, and remove items.
- Dynamically updates total prices.

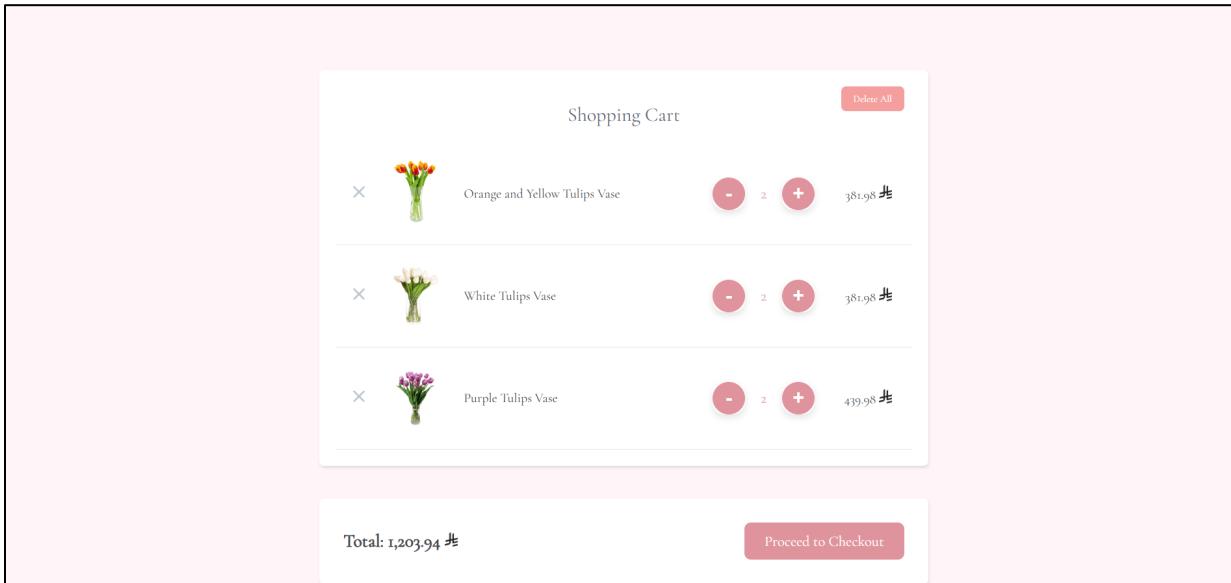


Figure 11 Shopping Cart

4. Checkout Page

- Users complete shipping and contact details.
- Displays final order summary and confirmation.

Have a coupon?

Your Order Summary		
Product		Total
Orange and Yellow Tulips Vase x 2		₼ 381.98
White Tulips Vase x 2		₼ 381.98
Purple Tulips Vase x 1		₼ 219.99
Cart Subtotal		₼ 983.95
Order Total		₼ 983.95

[PayPal](#)

Make your payment directly into our bank account. Use your Order ID as the payment reference. We will not ship until payment is received.

© 2025 HANAMI - All Rights Reserved.

Figure 12 Checkout page

5. Admin Login Page

- Secure login for administrators.
- Session-based access control.

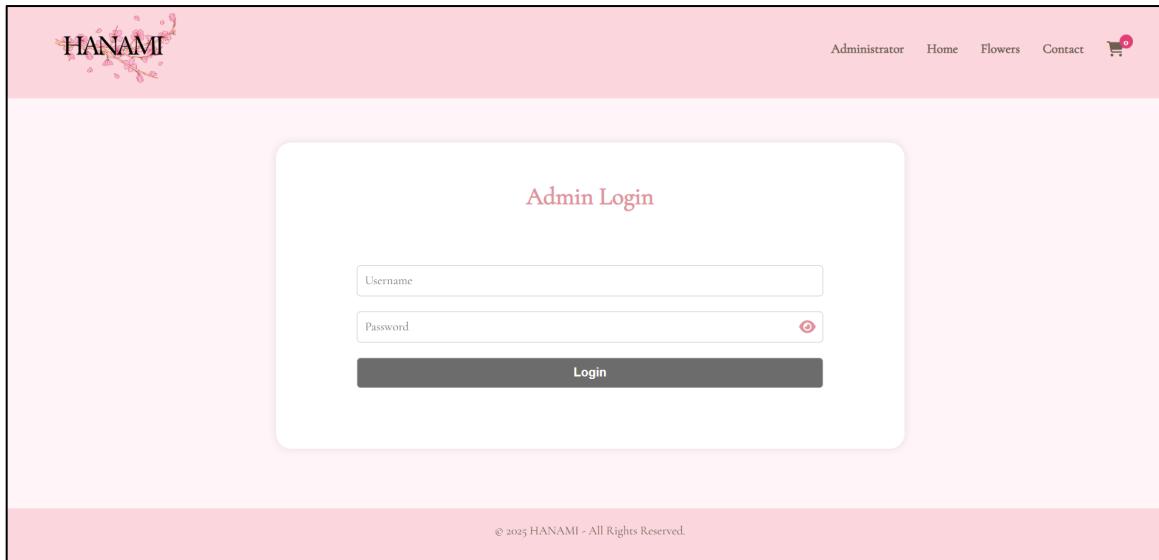


Figure 13 Admin login

6. Manage Products (Admin Dashboard)

- CRUD functionality: Add, update, or delete products.
- Product list table with search/filter options.

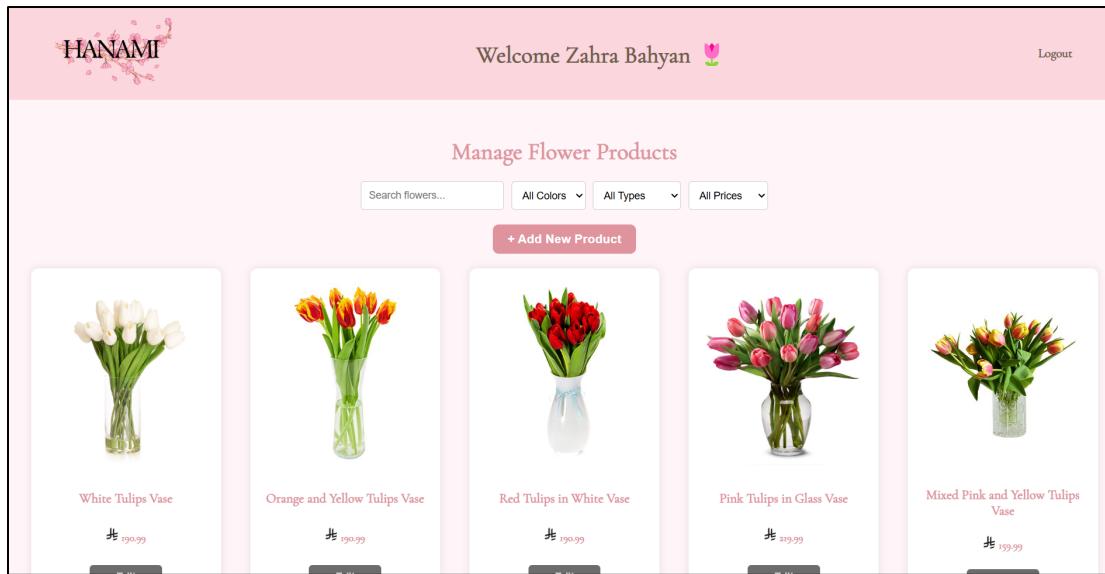


Figure 14 Admin dashboard

Welcome Munirah Aljindan

Logout

Manage Flower Products

Search flowers... All Colors All Types All Prices

+ Add New Product

White Tulips Vase ₦ 150.99	Orange and Yellow Tulips Vase ₦ 160.99	Red Tulips in White Vase ₦ 150.99	Pink Tulips in Glass Vase ₦ 150.99
<button>Edit Details</button> <button>Delete</button>	<button>Edit Details</button> <button>Delete</button>	<button>Edit Details</button> <button>Delete</button>	<button>Edit Details</button> <button>Delete</button>
White Roses Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>	Pink Roses Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>	Red & Pink Roses Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>	Red & White Roses Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>

Figure 15 Adding new product

Welcome Munirah Aljindan

Logout

Manage Flower Products

Search flowers... All Colors All Types All Prices

+ Add New Product

Luxurious Roses Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>	Pink Roses Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>	Red & Pink Roses Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>	Red & White Roses Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>	Luxurious Roses Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>	Pink Tulips in White Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>	Red & White Roses Vase ₦ 150.99 <button>Edit Details</button> <button>Delete</button>

Figure 16 Deleting product

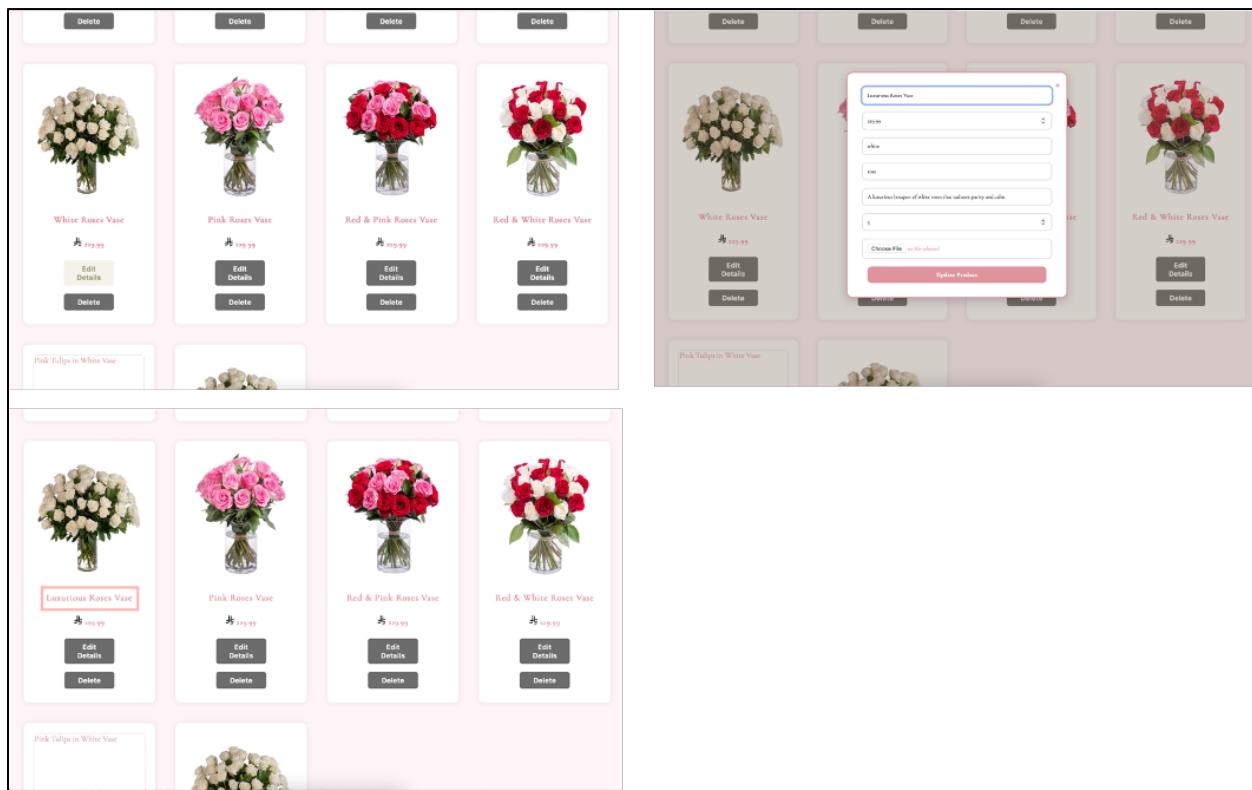


Figure 17 Editing products

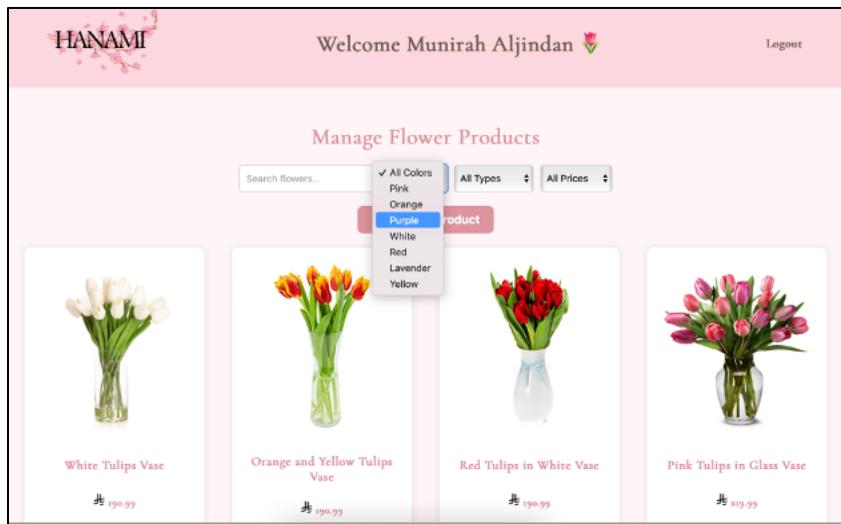


Figure 18 search/filter options

7. Products Page

- Grid view of all available bouquets and floral items.
- Linked to detail and customization interfaces.

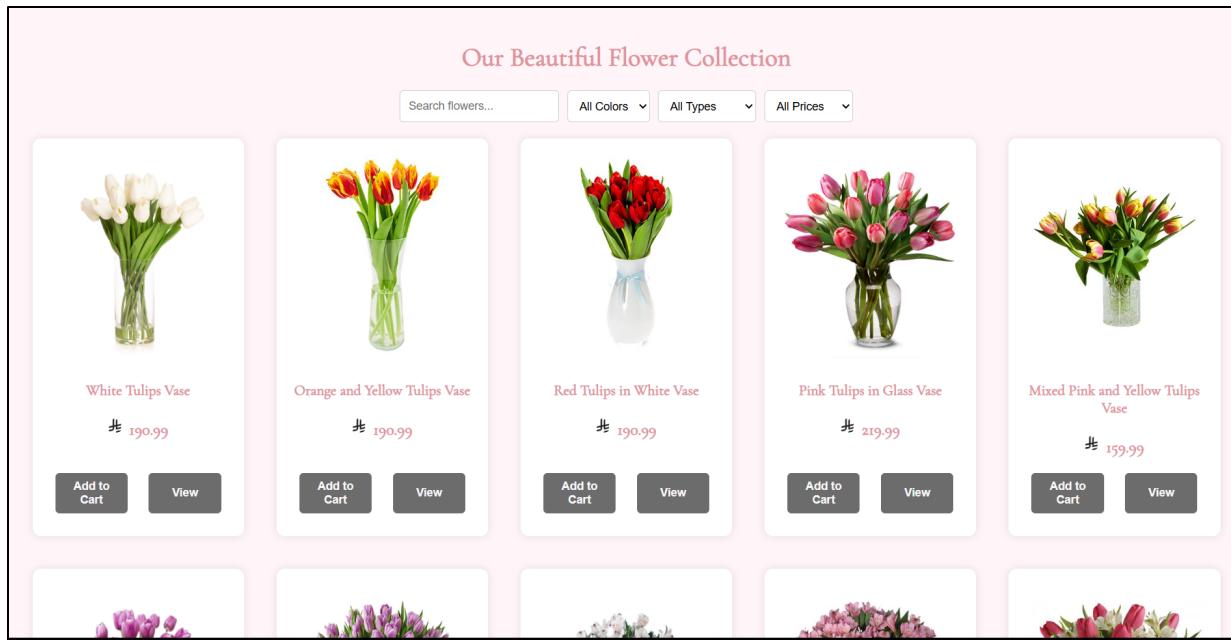


Figure 19 Exploring products

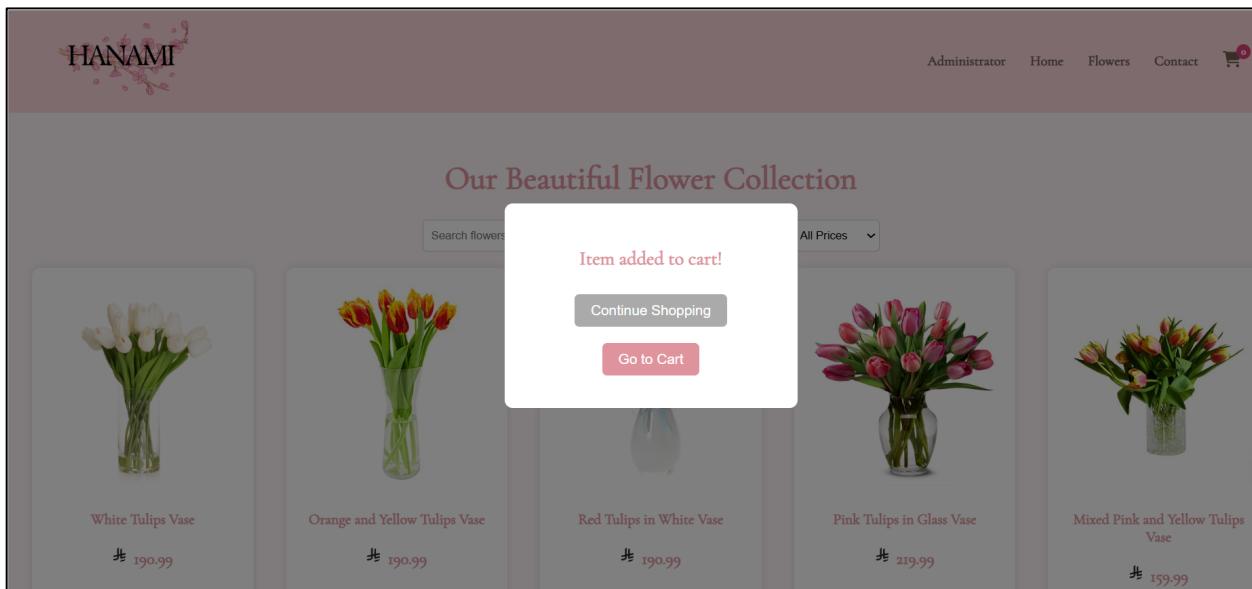


Figure 20 Message after adding a product to cart



Figure 21 in Product page a button showing “Max Quantity Reached” button when item limit is met

8. Contact Us Page

- o Includes a form to send feedback/inquiries.
- o JavaScript validation and basic anti-spam.

A screenshot of a contact us page titled "Get in Touch". The page features four input fields: "Name", "Email", "Subject", and a large "Message" area. Below these fields is a "Send A Message" button.

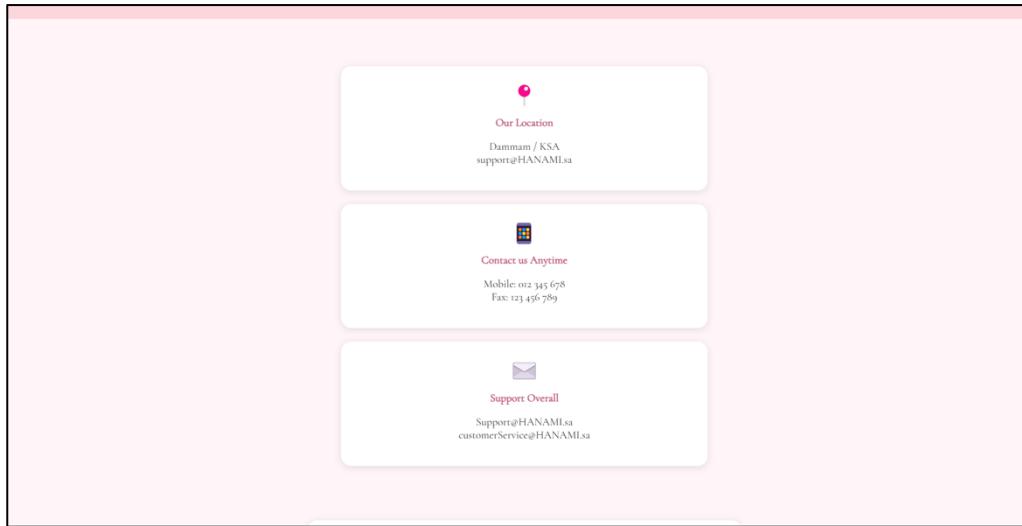


Figure 22 Contact us page

4. Fulfilled Requirements:

Tasks Completed

- User registration/login with session support
- Product listing and bouquet customization
- Cart management with real-time updates
- Checkout and order confirmation
- Admin dashboard with CRUD for products
- Past purchases tracked with cookies
- Responsive and accessible UI
- JavaScript validation

5. Testing Cases + Screenshots:

To ensure the HANAMI web application met all functionality requirements, we conducted comprehensive testing across all pages and features. Below is a detailed summary of the testing efforts, organized by feature.

1. Homepage Testing

The homepage was tested to ensure it correctly loads featured products and displays the navigation bar with proper links to all key sections of the website. All categories, images, and text were checked for accuracy and responsiveness. The test passed successfully.

2. Product Details Page

Clicking on any product from the homepage or product listing redirects users to a detailed product view. We tested this page to confirm it displays all product information such as the image, name, description, price, and available stock. The quantity input box and the “Add to Cart” button were also tested. The JavaScript pop-up help window on this page was triggered successfully. All functionalities worked as intended.

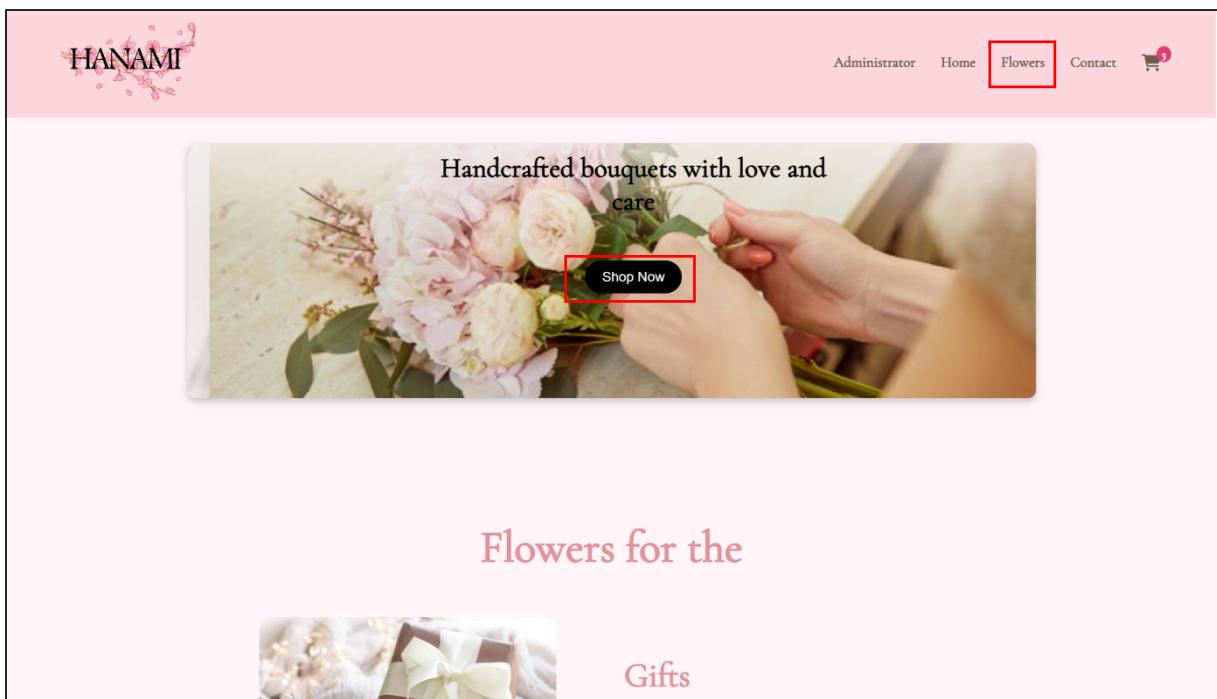


Figure 23 Homepage entry point to access the product catalog

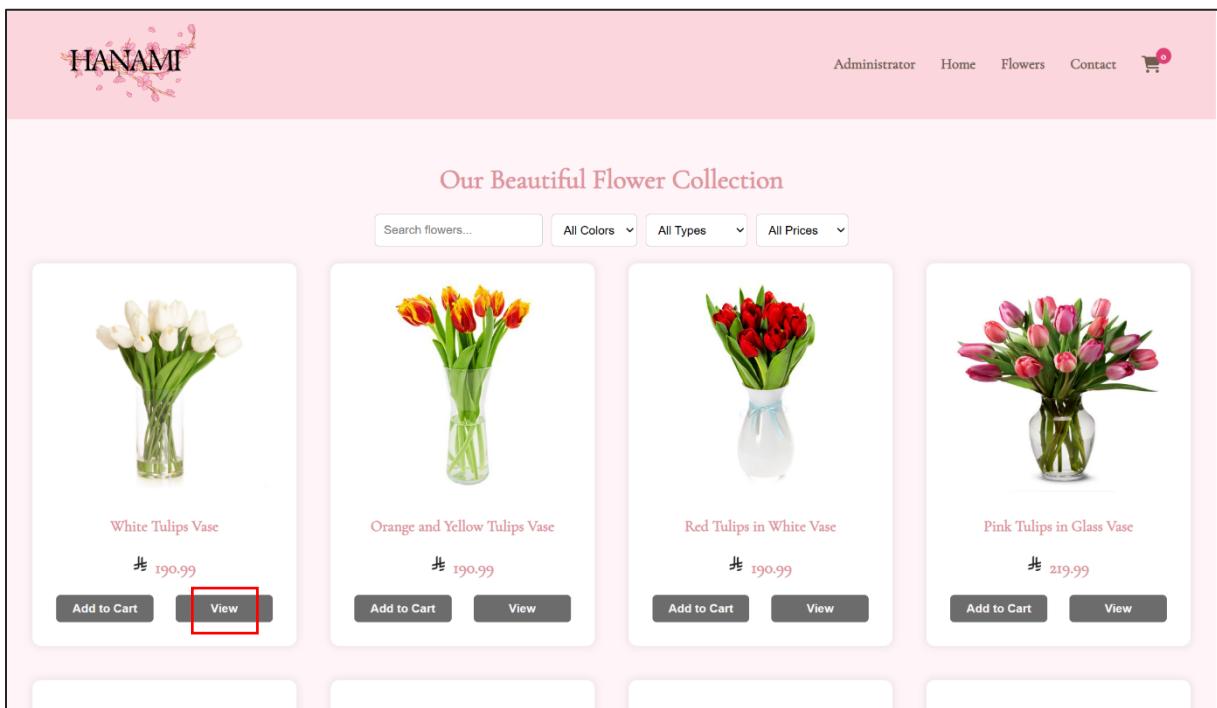


Figure 24 shows all products with "View" and "Add to Cart" options

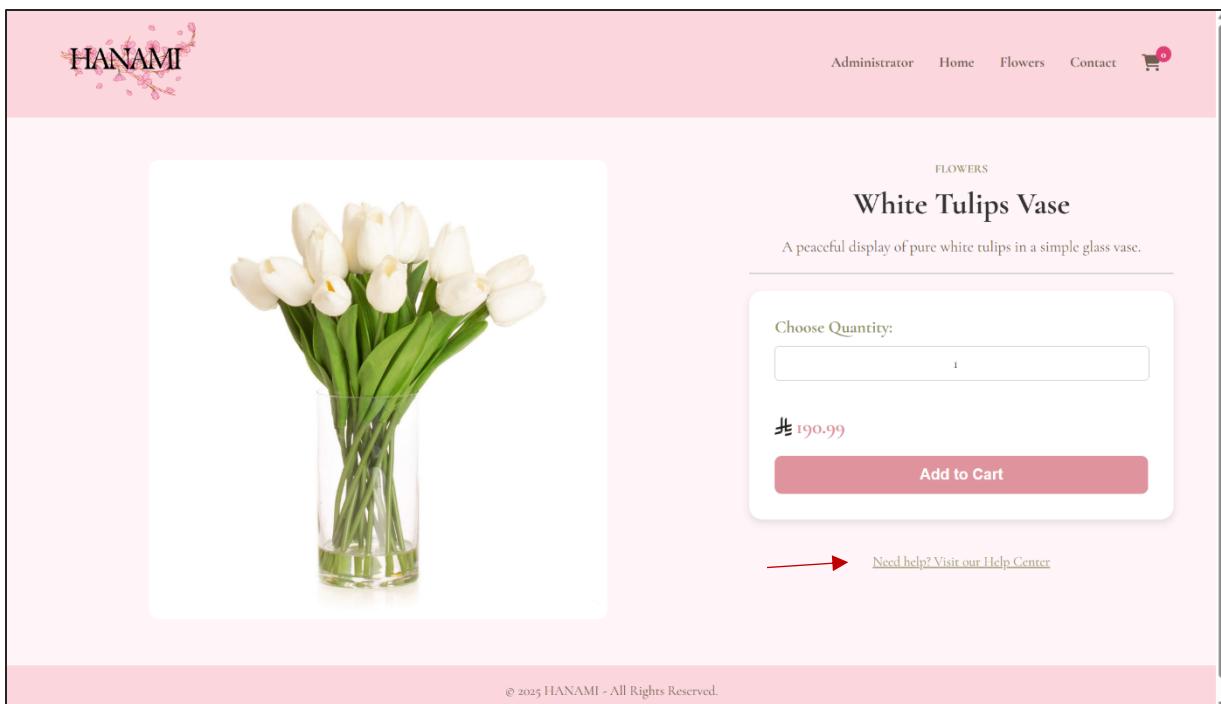


Figure 25 Displays product image, name, price, description, quantity input, “Add to Cart” button, and help popup link

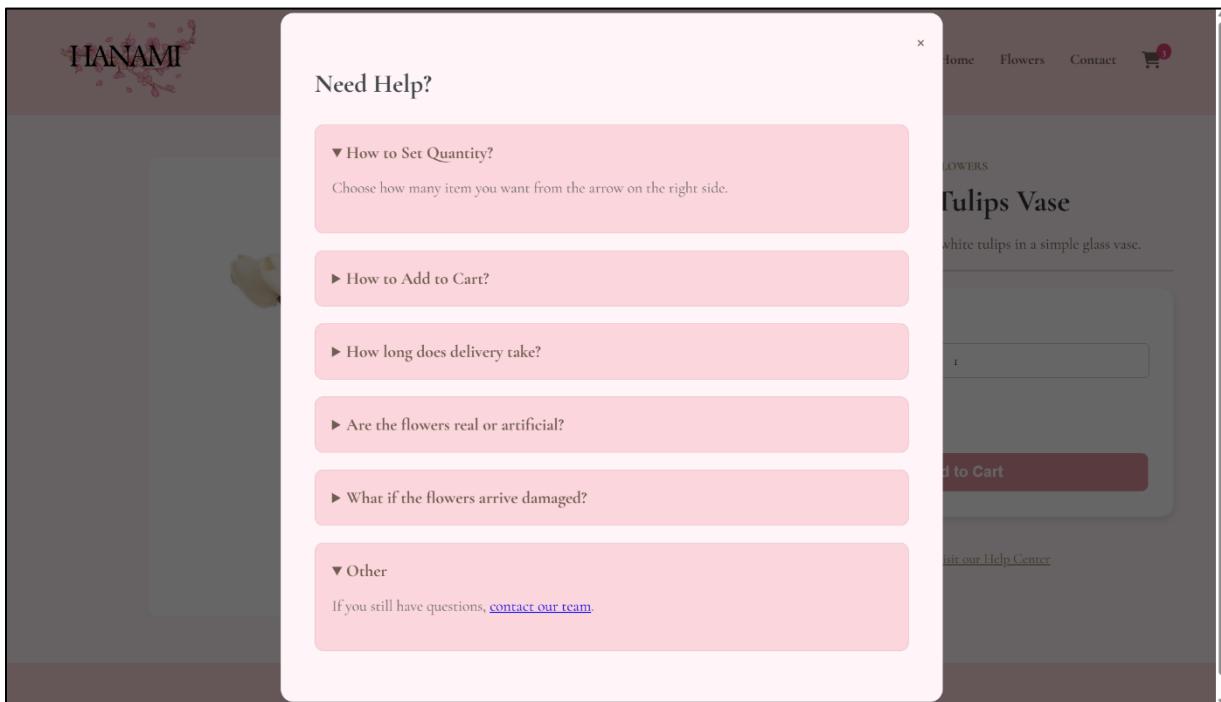


Figure 26 Displays help window with answers to common questions about quantity, adding to cart, delivery, and support.

Table 1: Testing the Product Details Page Functionality

Test ID	Product Details_01
Feature	View Product Details
Prerequisite	A product exists in the database
Test Procedure	<p>3. Access the product listing by either:</p> <ul style="list-style-type: none"> ▪ Clicking the “shop now” card on the home page ▪ Scrolling to the categories section and selection one in the homepage ▪ Clicking the “Flowers” link in the top navigation menu <p>4. Click the “View” button on any product</p> <p>5. Confirm that a new page opens displaying the full details of the selected product.</p> <p>6. Verify the product image, name, description, price, and quantity are displayed.</p> <p>7. Click the “help” link to open the JavaScript help popup</p>
Expected Result	Full product details are shown correctly, quantity input works, and help popup opens
Actual Result	Product data displayed correctly; quantity accepted; help popup triggered as expected
Verified (Yes/No)	Yes

3. Add to Cart Functionality

We verified that entering a valid quantity within the available stock successfully adds the product to the cart. When the user attempts to add more items than are in stock, an appropriate warning message is displayed. The total price per item (unit price multiplied by quantity) is accurately calculated. This functionality is powered by PHP sessions and was tested under multiple scenarios.

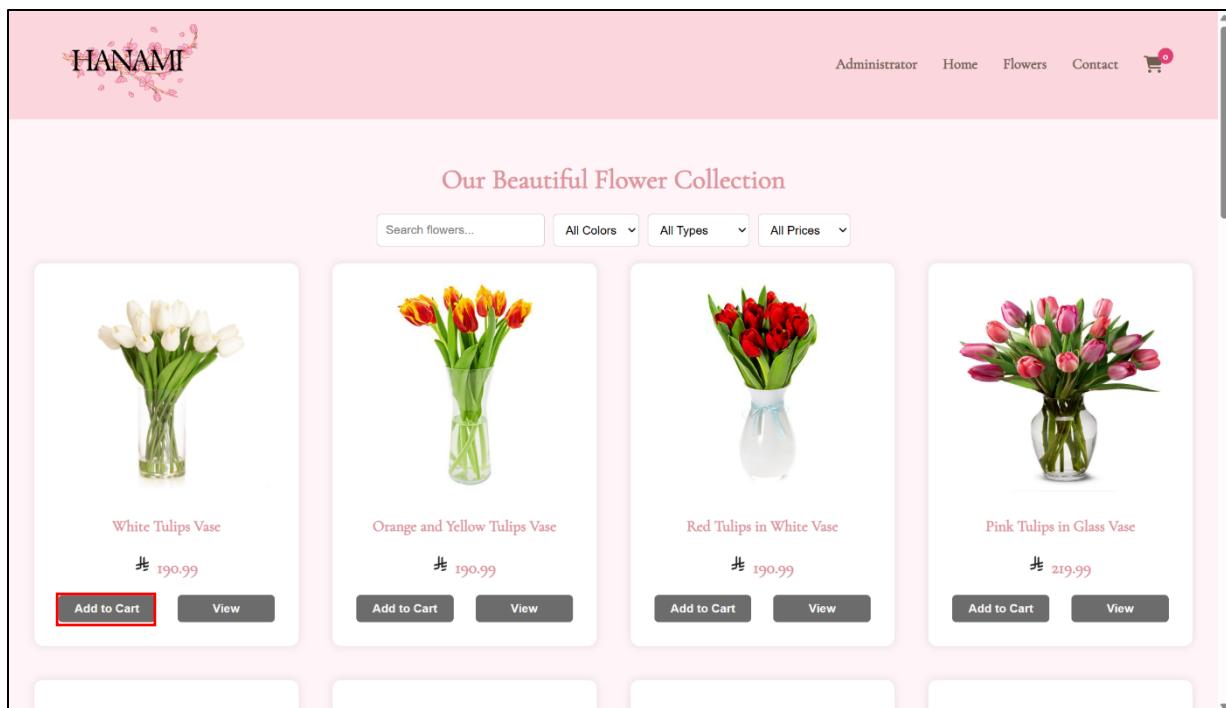


Figure 27: Product Listing with Add to Cart Options

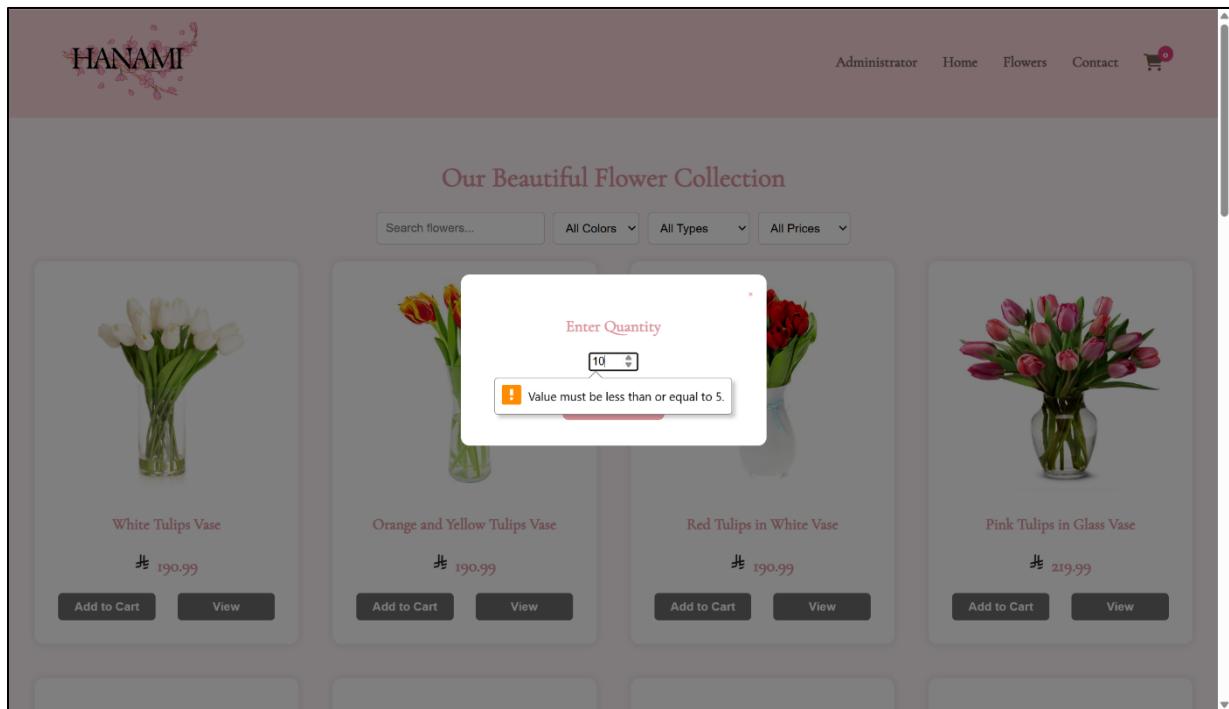


Figure 28 Displays a validation message when a user enters a quantity greater than available stock

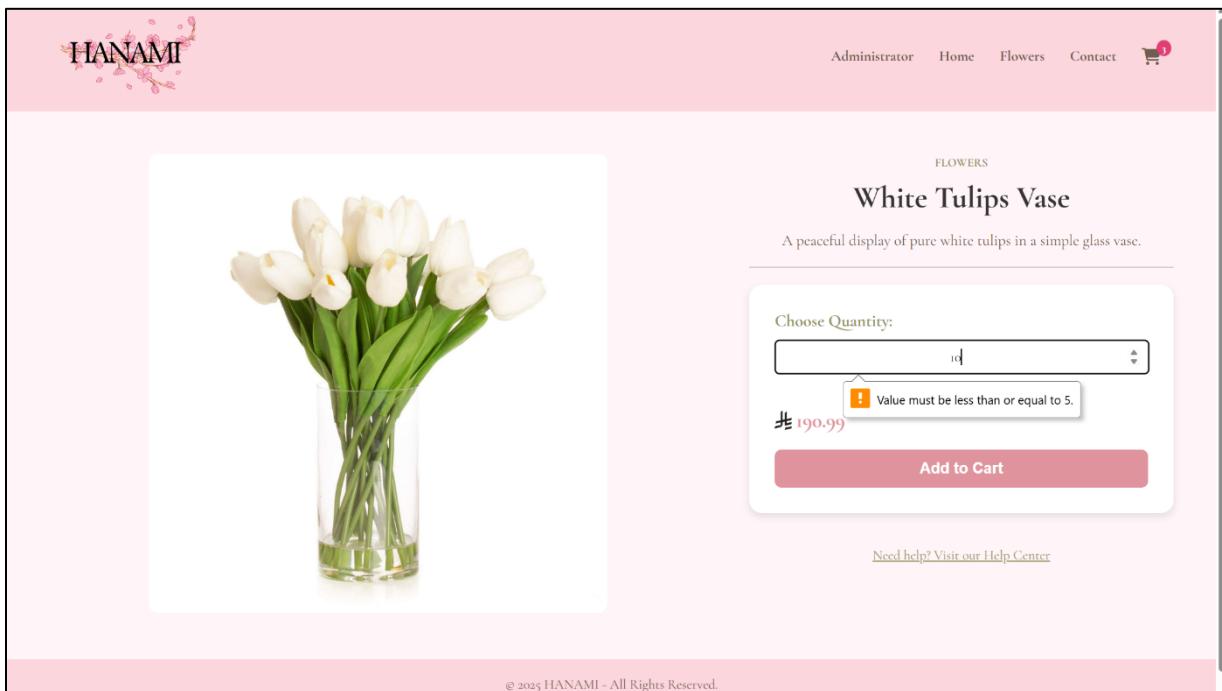


Figure 29 Shows an input of 10 when max is 5. Error tooltip: "Value must be less than or equal to 5"

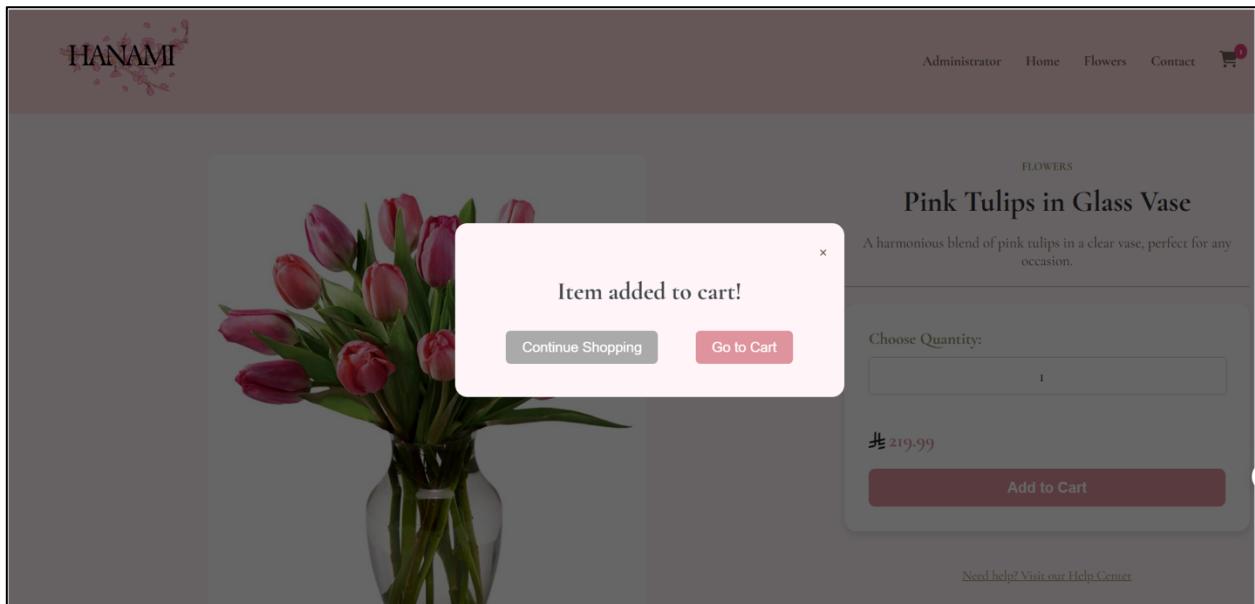
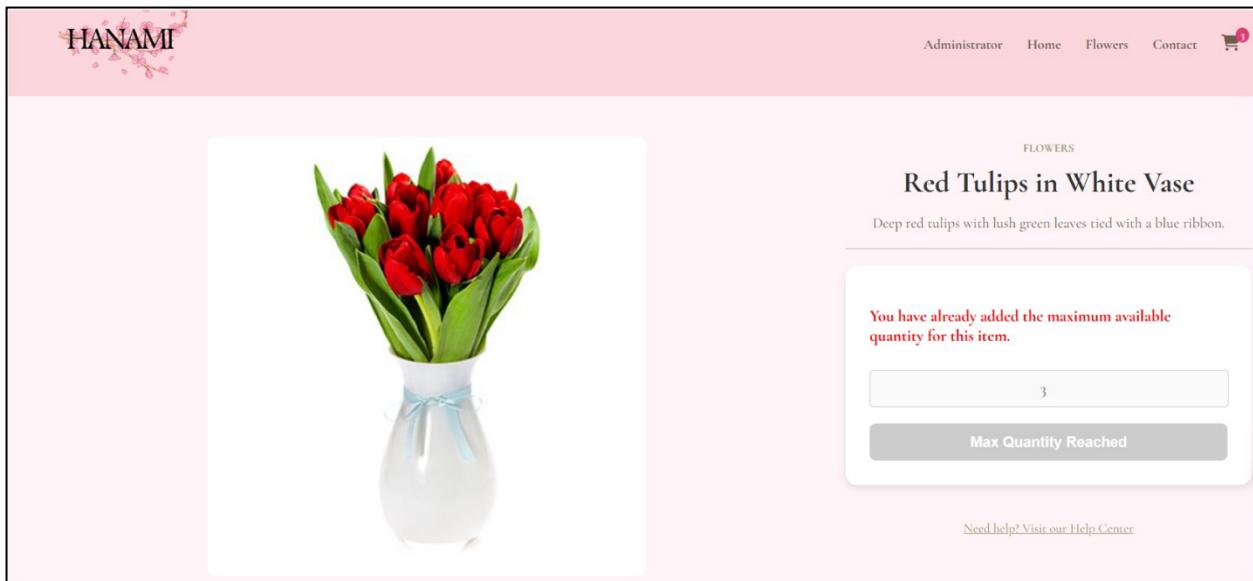


Figure 30 Message After Adding Product to Cart



Product detail page with alert and disabled input after reaching max quantity.

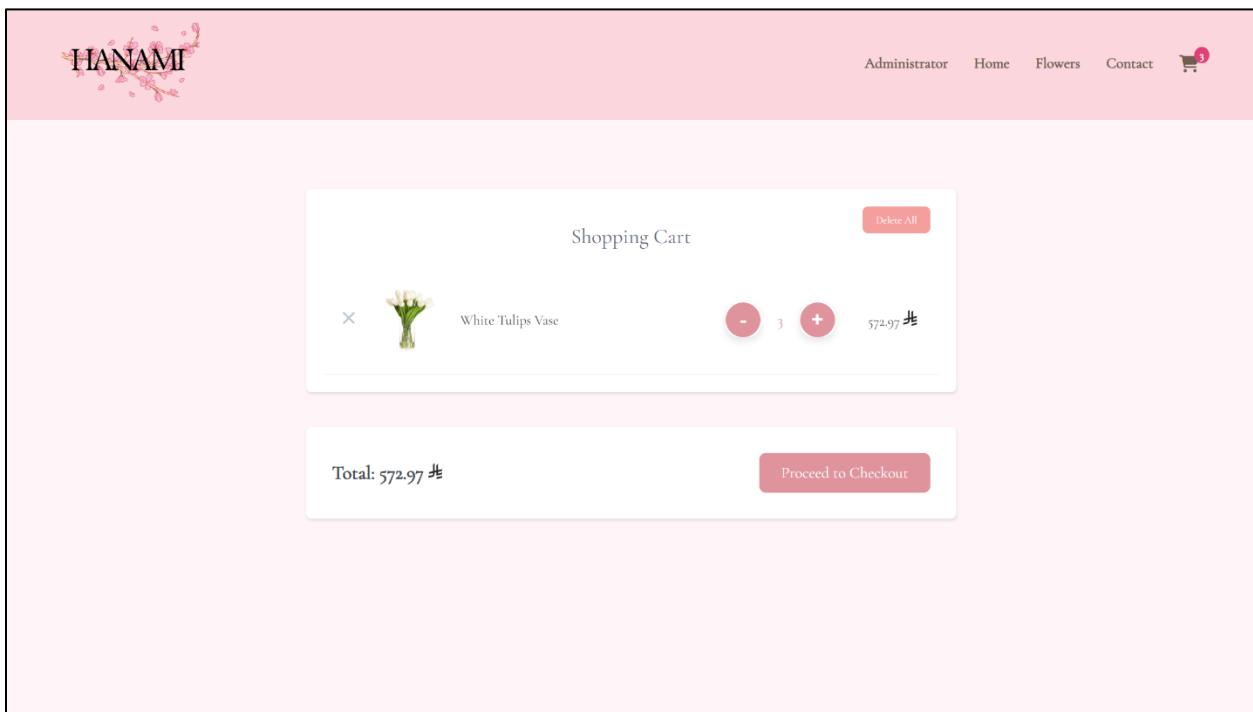


Figure 30: Confirms the correct quantity and total price ($3 \times 190.99 = 572.97$) are shown after adding to the cart

Table 2: Testing the Add to Cart Functionality

Test ID	Cart Functionality_01
Feature	Add Product to Cart
Prerequisite	Product is listed and have available stock
Test Procedure	<ol style="list-style-type: none"> Access the product listing by either: <ul style="list-style-type: none"> Clicking the “shop now” card on the home page Scrolling to the categories section and selection one in the homepage Clicking the “Flowers” link in the top navigation menu. You can add the product to the cart by: <ul style="list-style-type: none"> Selecting a quantity directly from the product listing. Or navigation to the product details page and entering the quantity there. Click “Add to Cart” Navigate to the Cart page by clicking on the symbol of the cart up in the menu Confirm that the product name, quantity, unit price, and total are correct Try entering a quantity greater than the available stock Observe whether a warning message appears (e.g., “<i>Value must be less than or equal to 5</i>”)
Expected Result	Product is added correctly; total is calculated; warning shown when exceeding stock
Actual Result	Cart updated with accurate totals; warning displayed for invalid quantity
Verified (Yes/No)	Yes

4. Shopping Cart Page

The cart page displays all selected items with options to update the quantity, remove individual items, or clear the entire cart. We tested modifying item quantities and deleting products, and confirmed the updates were reflected instantly. JavaScript validation prevents incorrect inputs, such as zero or negative values.

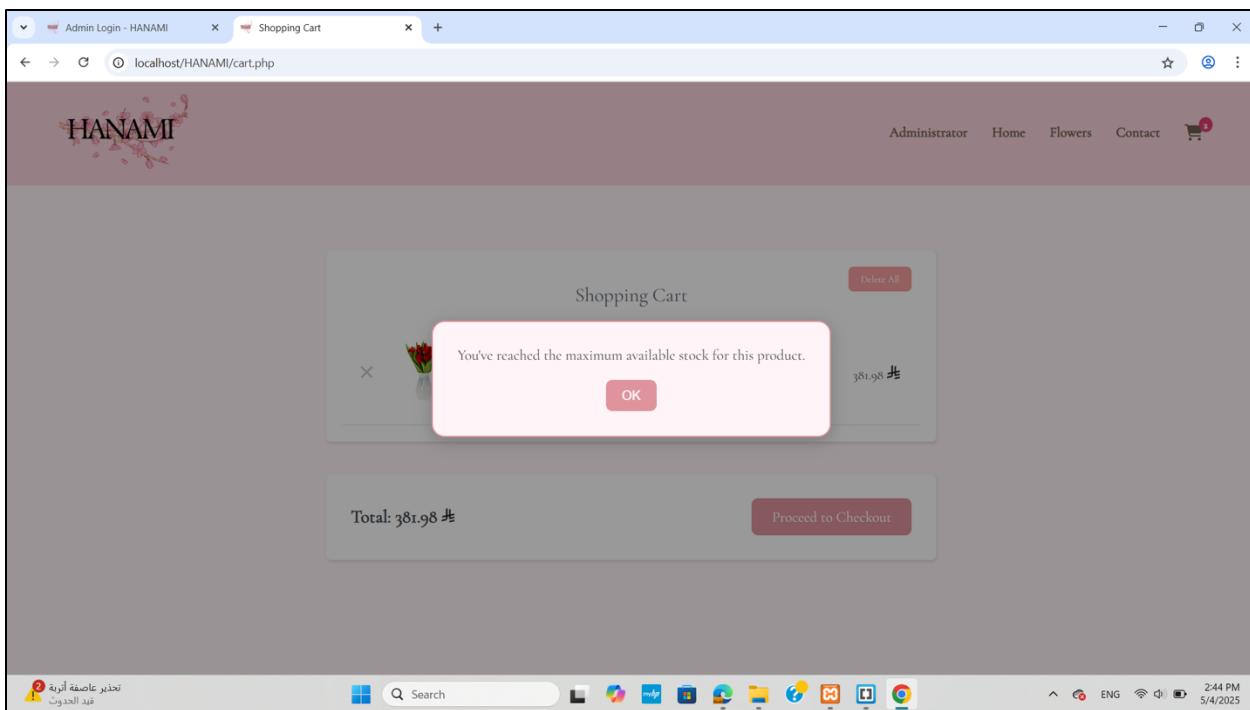


Figure 31 Displays a warning message when the user attempts to add more items than the available stock in the cart

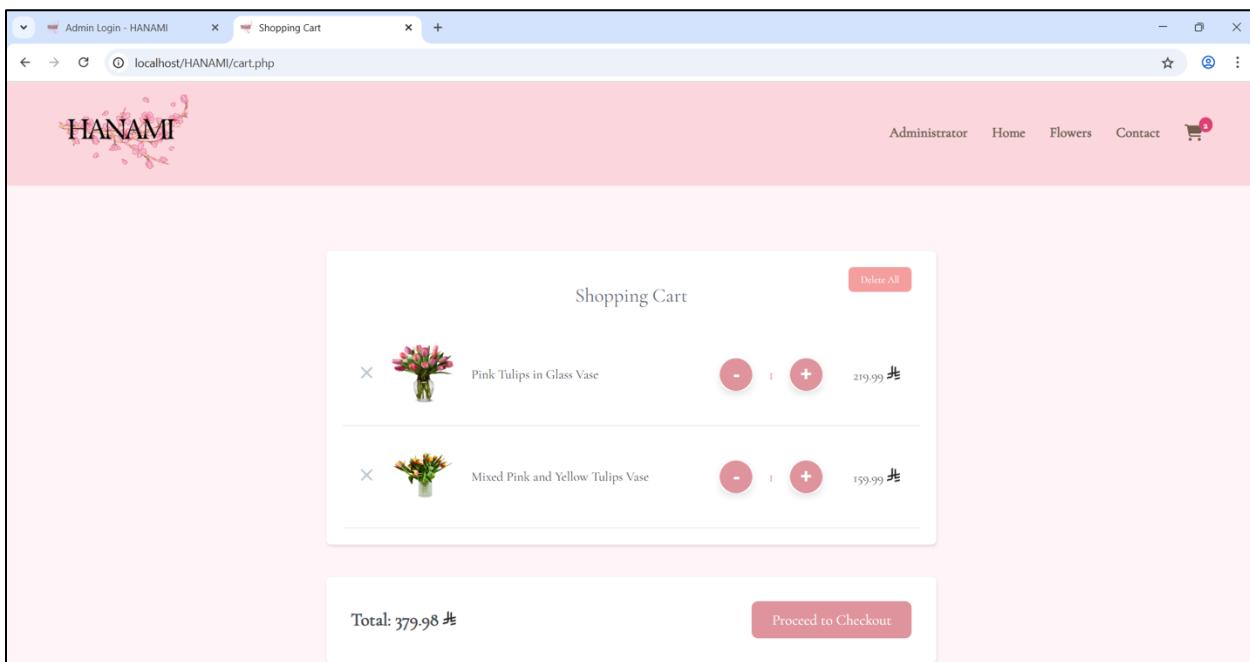


Figure 32 Shows the cart page with the updated quantity and total after modifying the item quantity

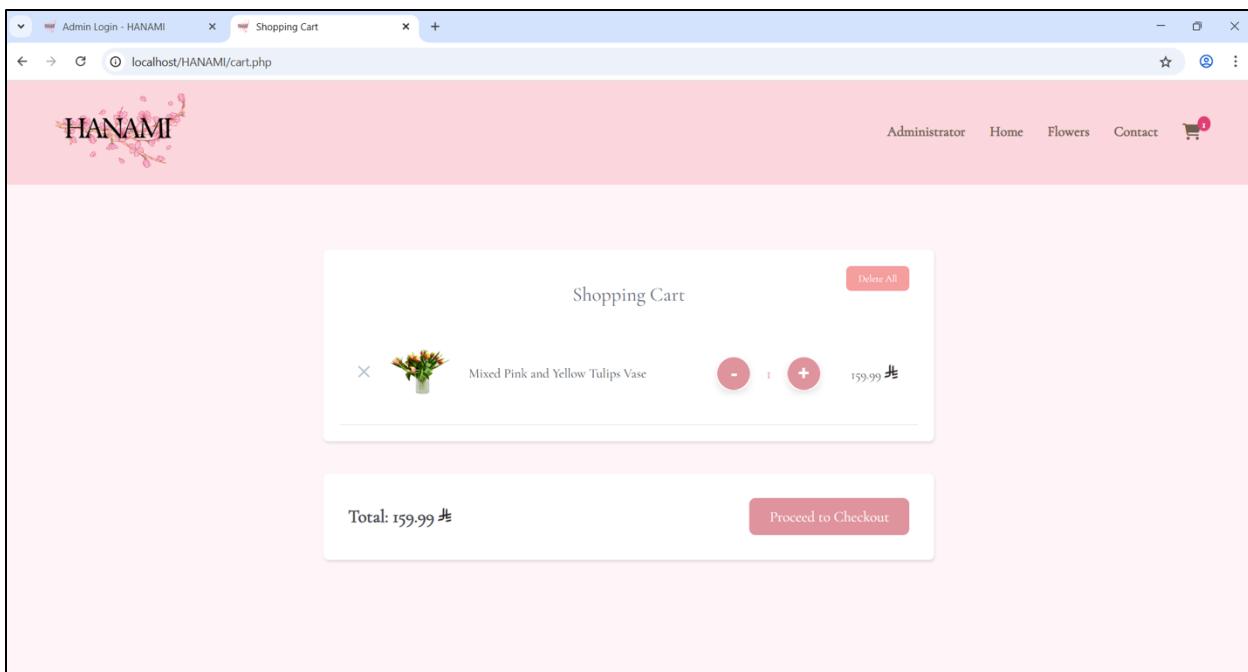


Figure 33 Demonstrates the shopping cart with a single item after delete one item

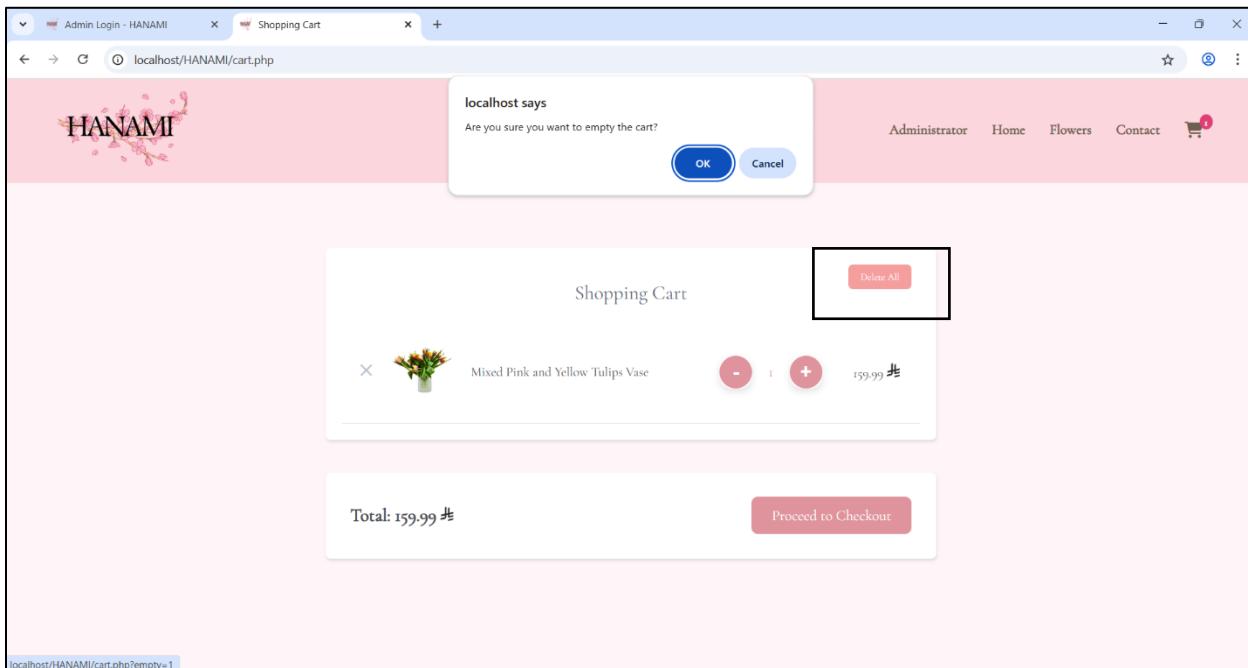


Figure 34 Shows a pop-up confirmation when the user chooses to empty the cart

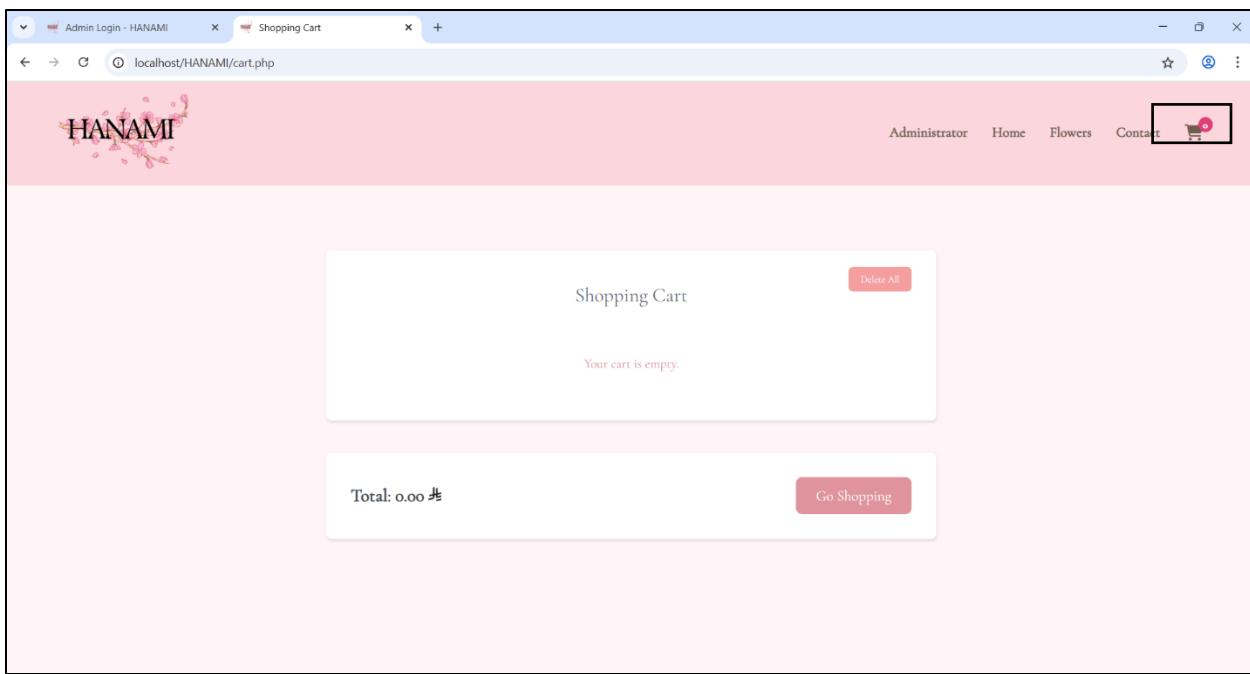


Figure 35 Displays the cart page with no items, confirming the cart is empty after deletion

Table 3 Cart functionality,

Test ID	Cart Functionality_01
Feature	Shopping Cart Operations
Prerequisite	Product added to the cart and available stock
Test Procedure	<ol style="list-style-type: none"> Access the shopping cart page by navigating through the website. Add the product to the cart by selecting the quantity and clicking "Add to Cart." Navigate to the Cart page by clicking the cart icon in the top navigation menu. Confirm that the product name, quantity, unit price, and total are correctly displayed. Test if the user can modify the quantity of the product in the cart. Test removing a product from the cart. Verify that the cart updates immediately, reflecting the modifications made. Test the behavior when the quantity exceeds available stock.
Expected Result	<ul style="list-style-type: none"> * The product is added to the cart with accurate details (name, price, quantity). * The total is updated correctly when the quantity is modified, or a product is removed. * If the quantity exceeds the available stock, a warning message is displayed (e.g., "Stock exceeds available quantity"). * The user can update quantities and delete items without issues.
Actual Result	<ul style="list-style-type: none"> * Cart is updated correctly after modification. * When an invalid quantity (greater than available stock) is entered, a warning is displayed. * The total price updates as expected.
Verified (Yes/No)	Yes

5. Checkout Page

We tested the checkout page by reviewing the complete order summary, verifying that users can finalize their purchases or continue modifying their cart. All order entries were correctly displayed with options to delete, update quantities, or proceed to buy. The “Buy” button was tested to confirm it updates the stock levels in the database and clears the cart after purchase.

The screenshot shows a light pink-themed checkout page. At the top right, there's a link 'Have a coupon?' with a text input field 'Enter your coupon code' and a button 'Apply Coupon'. Below this is a large white form area titled 'Billing Details'. It contains fields for 'Country *' (dropdown menu showing 'Saudi Arabia'), 'First Name *' and 'Last Name *' (text inputs), 'Address *' (text input), 'Town / City *' (text input), 'Email Address *' and 'Phone *' (text inputs), and a 'Place Order' button at the bottom.

Figure 36 Shows the billing details form on the checkout page, allowing users to input personal information

The screenshot shows an order summary page with a light pink header and footer. The main content is a white box titled 'Your Order Summary'. It contains a table with three rows:

Product	Total
Orange and Yellow Tulips Vase × 2	₼ 381.98
Cart Subtotal	₼ 381.98
Order Total	₼ 381.98

Below the table is a 'PayPal' button and a note: 'Make your payment directly into our bank account. Use your Order ID as the payment reference. We will not ship until payment is received.' The footer contains the copyright notice '© 2025 HANAMI - All Rights Reserved.'

Figure 37 Shows the order summary page, including product details, total cost, and the option to complete the transaction using a payment method like PayPal.

Your Order Summary

Product	Total
Orange and Yellow Tulips Vase * 2	₼ 381.98
Cart Subtotal	₼ 381.98
Order Total	₼ 381.98

[PayPal](#)

Make your payment directly into our bank account. Use your Order ID as the payment reference. We will not ship until payment is received.

© 2025 HANAMI - All Rights Reserved.

Figure 38 Shows the order summary page, including product details, total cost, and the option to complete the transaction using a payment method like PayPal

Have a coupon?

Enter your coupon code [Apply Coupon](#)

Billing Details

Country *

Saudi Arabia

Saudi Arabia

Bahrain

Kuwait

Oman

Qatar

United Arab Emirates

Town / City *

City

Email Address * Phone *

Email address Phone number

[Place Order](#)

Figure 39 Displays the country selection dropdown, where users can choose their country from a predefined list of options

A screenshot of a web browser showing a form submission error. A modal dialog box is centered over the page with the title "localhost says" and the message "Please fill in all required fields." An "OK" button is at the bottom right of the dialog. The main page contains a "Billing Details" form with the following fields:

- Country *: Saudi Arabia (selected in a dropdown)
- First Name *: Fatima
- Last Name *: Alomir
- Address *: Street address (empty input field)
- Town / City *: Dammam
- Email Address *: fm.alomir@gmail.com
- Phone *: 0599819307

A "Place Order" button is located at the bottom left of the form.

Figure 40 Depicts the form submission error message when the user attempts to proceed with incomplete or invalid fields, ensuring all required information is provided

A screenshot of a web browser showing a validation error for an email field. At the top, there is a coupon code input field with "Enter your coupon code" placeholder text and an "Apply Coupon" button. Below this is the same "Billing Details" form as Figure 40. In the Email Address field, the user has entered "5666". A validation error message is displayed below the field: "Please include an '@' in the email address. '5666' is missing an '@'." The rest of the form fields are filled with the same data as Figure 40.

Figure 41 Displays a validation error for the email field

A screenshot of a web-based checkout form titled "Billing Details". The form includes fields for Country (Kuwait), First Name (hanan), Last Name (salem), Address (Azizya), Town / City (Aldahiya), Email Address (hanan@gmail.com), and Phone (677). A red "Place Order" button is at the bottom. A modal window titled "localhost says" displays the message "Please enter a valid phone number (at least 10 digits)." with an "OK" button.

Figure 42 A validation error indicating missing or incorrect phone number input during the checkout process, ensuring proper data entry before proceeding

A screenshot of a web-based checkout form titled "Billing Details". The form includes fields for Country (Kuwait), First Name (hanan), Last Name (salem), Address (Azizya), Town / City (Aldahiya), Email Address (hanan@gmail.com), and Phone (677). A red "Place Order" button is at the bottom. A modal window titled "localhost says" displays the message "Please enter a valid phone number (at least 10 digits)." with an "OK" button.

Figure 43 A validation error indicating missing or incorrect phone number input during the checkout process, ensuring proper data entry before proceeding

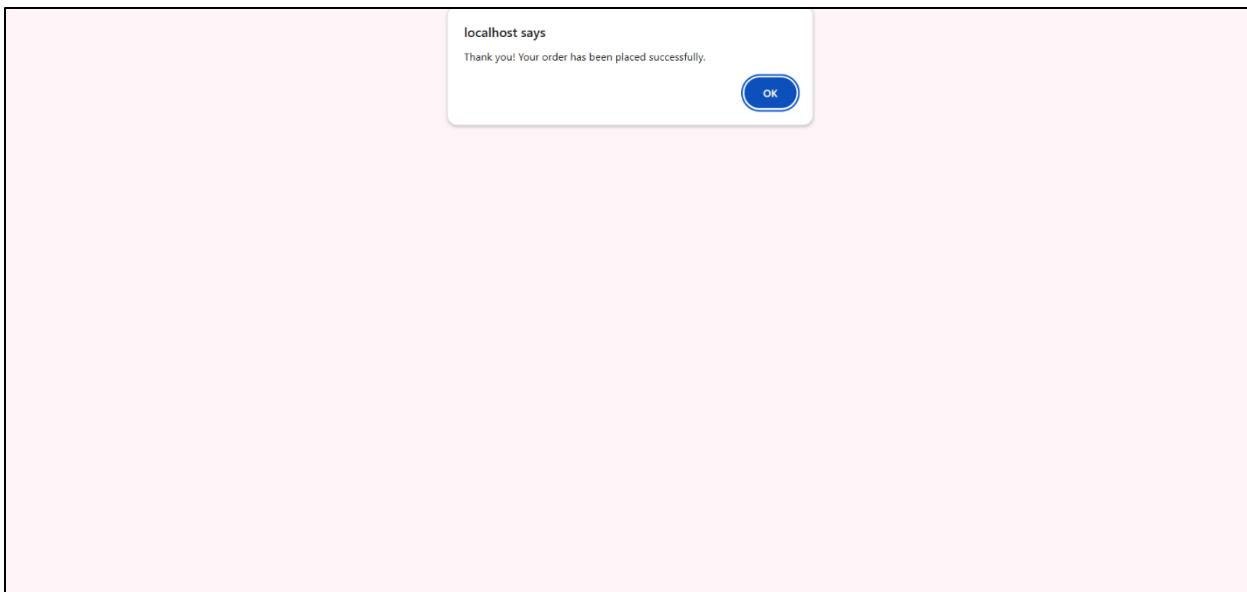


Figure 44 Success message after a user has successfully completed their order

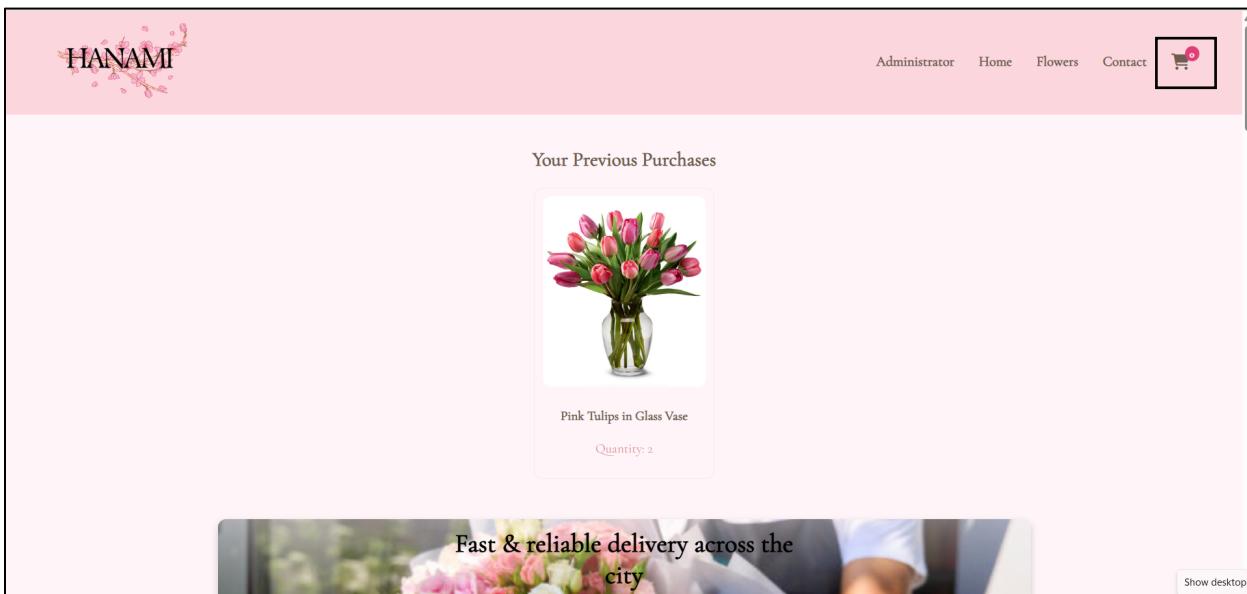
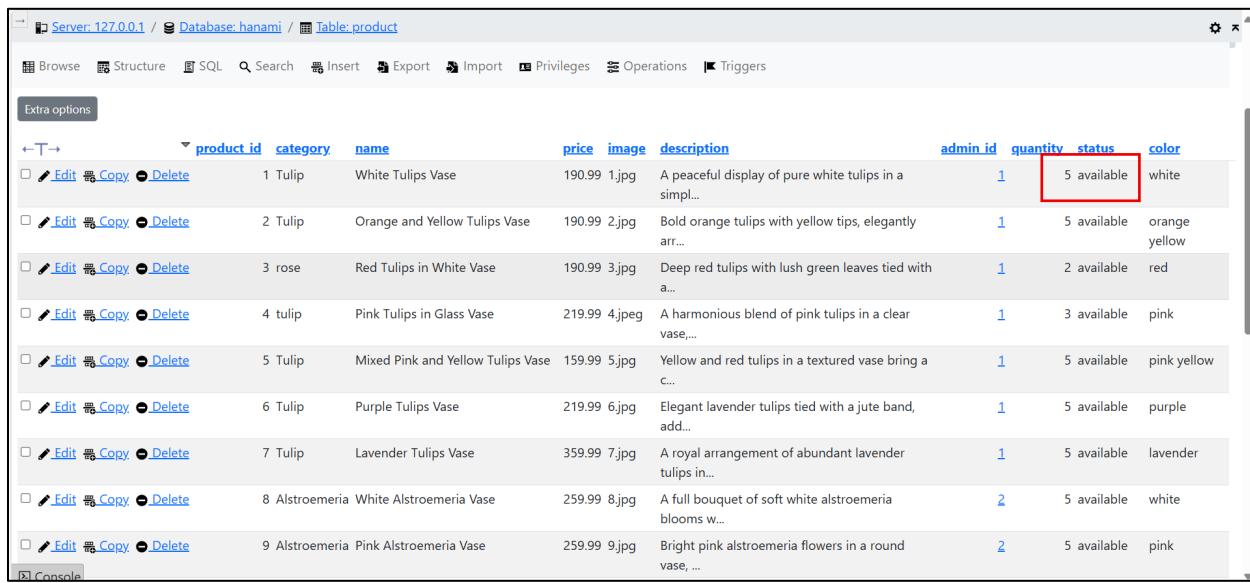


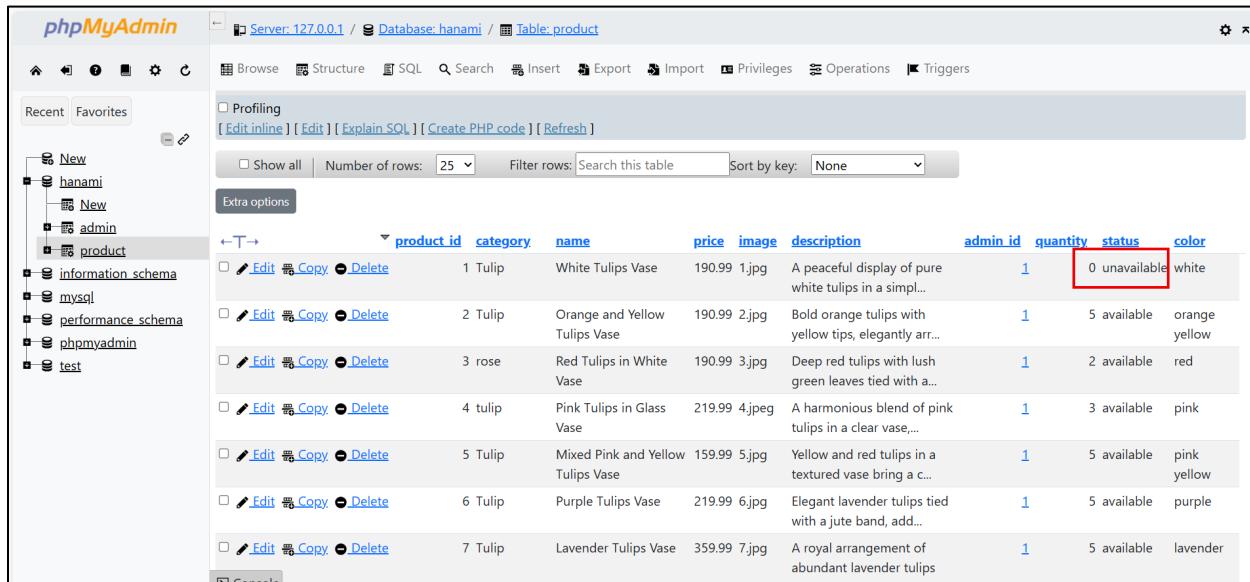
Figure 45 Presents the "Your Previous Purchases" section, which displays the user's previous orders for reference and potential re-ordering and the cart is empty after completing the order



The screenshot shows the phpMyAdmin interface for the 'product' table. The table has columns: product_id, category, name, price, image, description, admin_id, quantity, status, and color. There are 9 rows of data. The 1st row (White Tulips Vase) has a quantity of 5, which is highlighted with a red box. The 9th row (Pink Alstroemeria Vase) has an admin_id of 2.

	Edit Copy Delete	product_id	category	name	price	image	description	admin_id	quantity	status	color
<input type="checkbox"/>	Edit Copy Delete	1	Tulip	White Tulips Vase	190.99	1.jpg	A peaceful display of pure white tulips in a simpl...	1	5 available	white	
<input type="checkbox"/>	Edit Copy Delete	2	Tulip	Orange and Yellow Tulips Vase	190.99	2.jpg	Bold orange tulips with yellow tips, elegantly arr...	1	5 available	orange yellow	
<input type="checkbox"/>	Edit Copy Delete	3	rose	Red Tulips in White Vase	190.99	3.jpg	Deep red tulips with lush green leaves tied with a...	1	2 available	red	
<input type="checkbox"/>	Edit Copy Delete	4	tulip	Pink Tulips in Glass Vase	219.99	4.jpeg	A harmonious blend of pink tulips in a clear vase,...	1	3 available	pink	
<input type="checkbox"/>	Edit Copy Delete	5	Tulip	Mixed Pink and Yellow Tulips Vase	159.99	5.jpg	Yellow and red tulips in a textured vase bring a c...	1	5 available	pink yellow	
<input type="checkbox"/>	Edit Copy Delete	6	Tulip	Purple Tulips Vase	219.99	6.jpg	Elegant lavender tulips tied with a jute band, add...	1	5 available	purple	
<input type="checkbox"/>	Edit Copy Delete	7	Tulip	Lavender Tulips Vase	359.99	7.jpg	A royal arrangement of abundant lavender tulips in...	1	5 available	lavender	
<input type="checkbox"/>	Edit Copy Delete	8	Alstroemeria	White Alstroemeria Vase	259.99	8.jpg	A full bouquet of soft white alstroemeria blooms w...	2	5 available	white	
<input type="checkbox"/>	Edit Copy Delete	9	Alstroemeria	Pink Alstroemeria Vase	259.99	9.jpg	Bright pink alstroemeria flowers in a round vase, ...	2	5 available	pink	

Figure 46 The database shows 5 white tulip vases available before the customer places the order



The screenshot shows the phpMyAdmin interface for the 'product' table. The table has columns: product_id, category, name, price, image, description, admin_id, quantity, status, and color. There are 9 rows of data. The 1st row (White Tulips Vase) now has a quantity of 0, which is highlighted with a red box. The 9th row (Pink Alstroemeria Vase) has an admin_id of 2.

	Edit Copy Delete	product_id	category	name	price	image	description	admin_id	quantity	status	color
<input type="checkbox"/>	Edit Copy Delete	1	Tulip	White Tulips Vase	190.99	1.jpg	A peaceful display of pure white tulips in a simpl...	1	0 unavailable	white	
<input type="checkbox"/>	Edit Copy Delete	2	Tulip	Orange and Yellow Tulips Vase	190.99	2.jpg	Bold orange tulips with yellow tips, elegantly arr...	1	5 available	orange yellow	
<input type="checkbox"/>	Edit Copy Delete	3	rose	Red Tulips in White Vase	190.99	3.jpg	Deep red tulips with lush green leaves tied with a...	1	2 available	red	
<input type="checkbox"/>	Edit Copy Delete	4	tulip	Pink Tulips in Glass Vase	219.99	4.jpeg	A harmonious blend of pink tulips in a clear vase,...	1	3 available	pink	
<input type="checkbox"/>	Edit Copy Delete	5	Tulip	Mixed Pink and Yellow Tulips Vase	159.99	5.jpg	Yellow and red tulips in a textured vase bring a c...	1	5 available	pink yellow	
<input type="checkbox"/>	Edit Copy Delete	6	Tulip	Purple Tulips Vase	219.99	6.jpg	Elegant lavender tulips tied with a jute band, add...	1	5 available	purple	
<input type="checkbox"/>	Edit Copy Delete	7	Tulip	Lavender Tulips Vase	359.99	7.jpg	A royal arrangement of abundant lavender tulips in...	1	5 available	lavender	

Figure 47 The database updates successfully, reducing the white tulip vase quantity to 0 after the customer places the order.

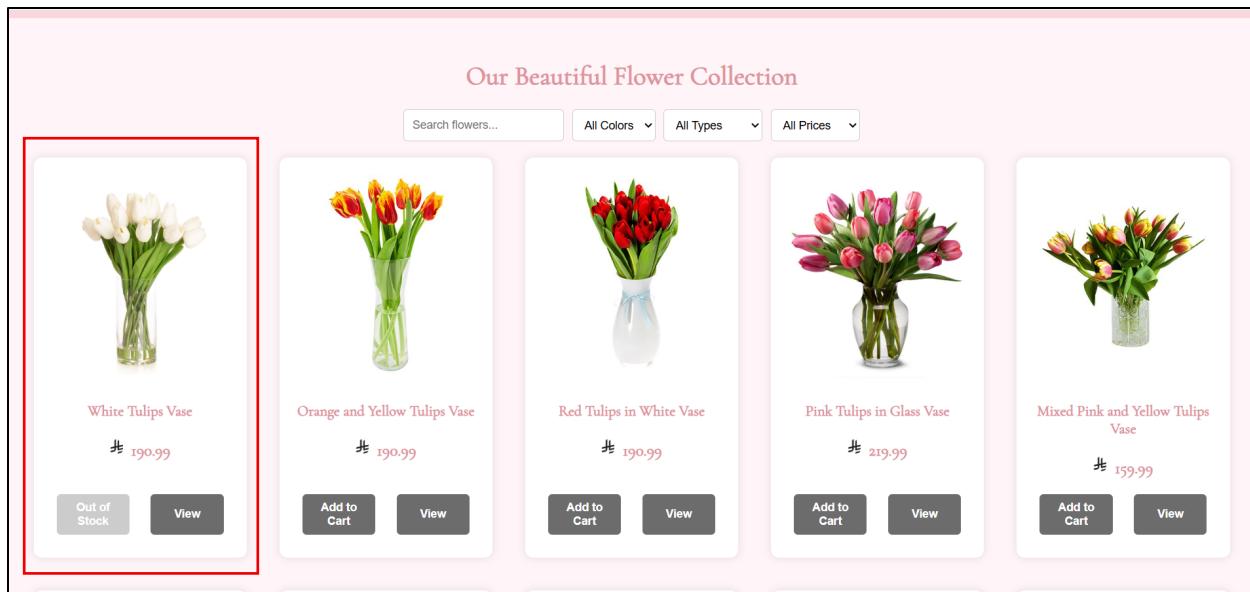


Figure 48 The white tulips vase is now marked as "Out of Stock," indicating the database was updated successfully

Table 4 Checkout Functionality

Test ID	Checkout Functionality_02
Feature	Checkout Process
Prerequisite	Cart contains selected items
Test Procedure	1. Review the order summary on the checkout page. 2. Verify that the cart details are accurate (products, quantity, price, and total). 3. Test modifying the cart on the checkout page (e.g., adjusting quantities or removing items). 4. Click on the "Buy" button to finalize the order. 5. Verify that after clicking "Buy", the cart is cleared, and the database stock levels are updated accordingly.
Expected Result	Order summary is displayed with accurate details. User can proceed to checkout and finalize the purchase. After clicking "Buy", the cart is cleared, and the stock levels in the database are updated.
Actual Result	Order summary was correct with no discrepancies. Cart was cleared after the purchase. Stock levels were updated in the database after completing the purchase.
Verified (Yes/No)	Yes

6. Admin Login and Access

The administrator login page was tested using both valid and invalid credentials. Successful logins redirected to the admin dashboard, while incorrect credentials displayed an appropriate error message. The database was queried securely for authentication.

Table 5 Admin Login Test

Test ID	Login 01
Feature	Admin Login and Access
Prerequisite	Admin account exists in database
Test Procedure	<ol style="list-style-type: none"> Access the Admin Login page Try logging in with different inputs: <ul style="list-style-type: none"> Leave both username and password fields empty, then click "Login" Enter a non-existent admin username with any password Enter a valid admin username with incorrect password Enter a valid admin username and correct password Observe the behavior after each attempt
Expected Result	Empty fields trigger a message: "Please fill out this field" Wrong username shows: "Admin not found" Wrong password shows: "Invalid password" Correct credentials redirect to the admin dashboard
Actual Result	All messages appeared as expected, and the admin was successfully redirected to the Manage Products page upon correct login
Verified (Yes/No)	Yes

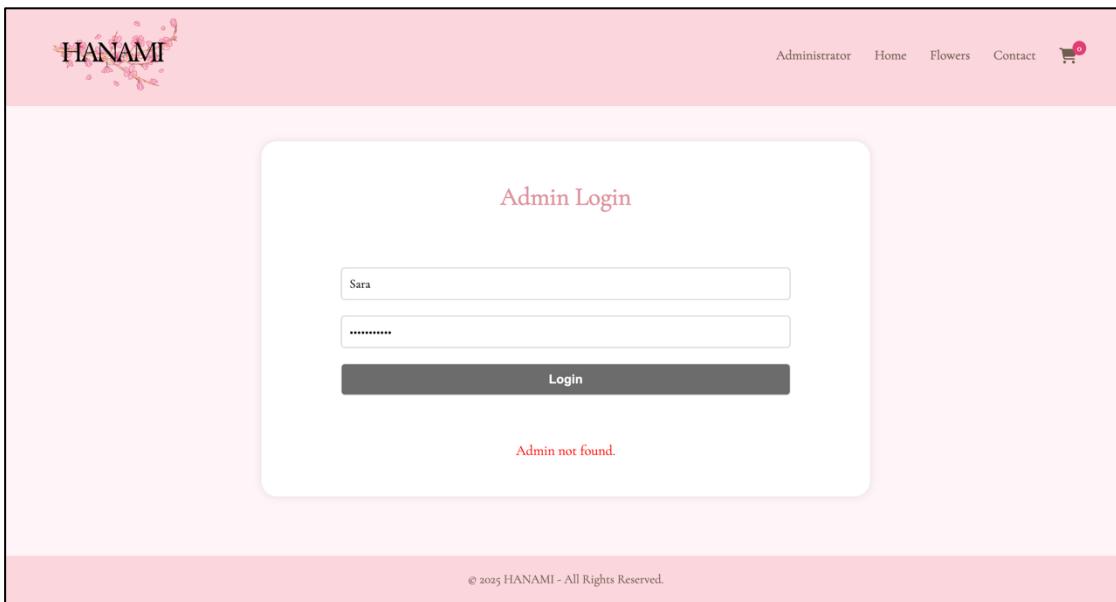

 A screenshot of a web browser displaying the HANAMI Admin Login interface. At the top, there's a pink header bar with the HANAMI logo on the left and navigation links for 'Administrator', 'Home', 'Flowers', 'Contact', and a shopping cart icon with a '0' notification on the right. Below the header is a light gray content area containing a white rectangular login form. The form has a title 'Admin Login' at the top center. It contains two input fields: the first is labeled 'Username' and has 'Sara' typed into it; the second is a password field with several dots. Below these fields is a dark gray 'Login' button. Underneath the button, a red error message 'Admin not found.' is displayed. At the very bottom of the page, there's a thin pink footer bar with the copyright notice '© 2025 HANAMI - All Rights Reserved.'.

Figure 49 An error message appears when the admin username is incorrect

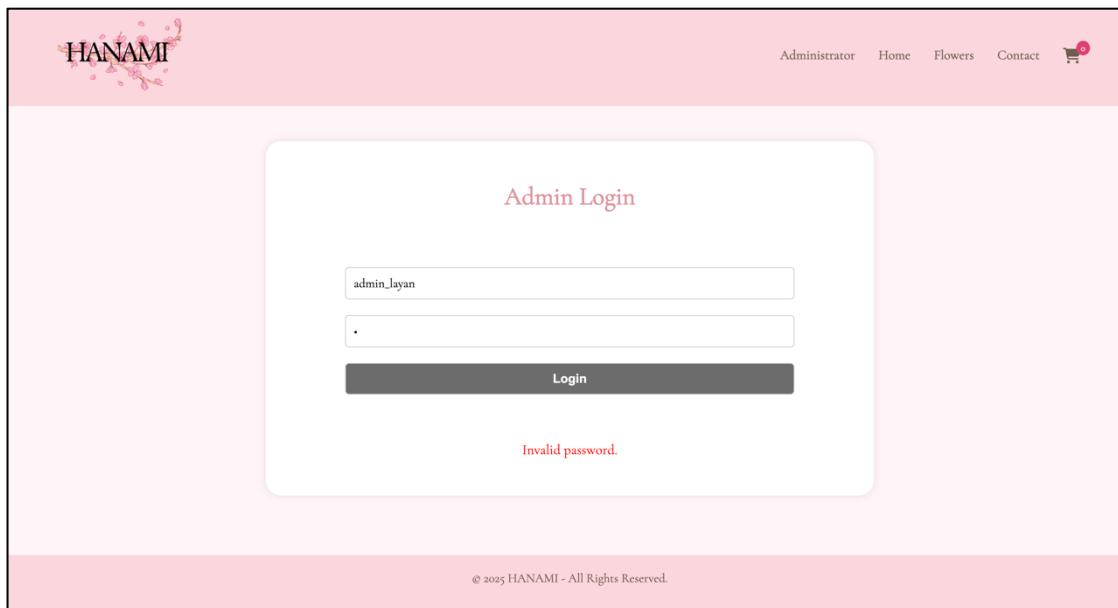


Figure 50 An error message appears when the username is valid, but the password is incorrect

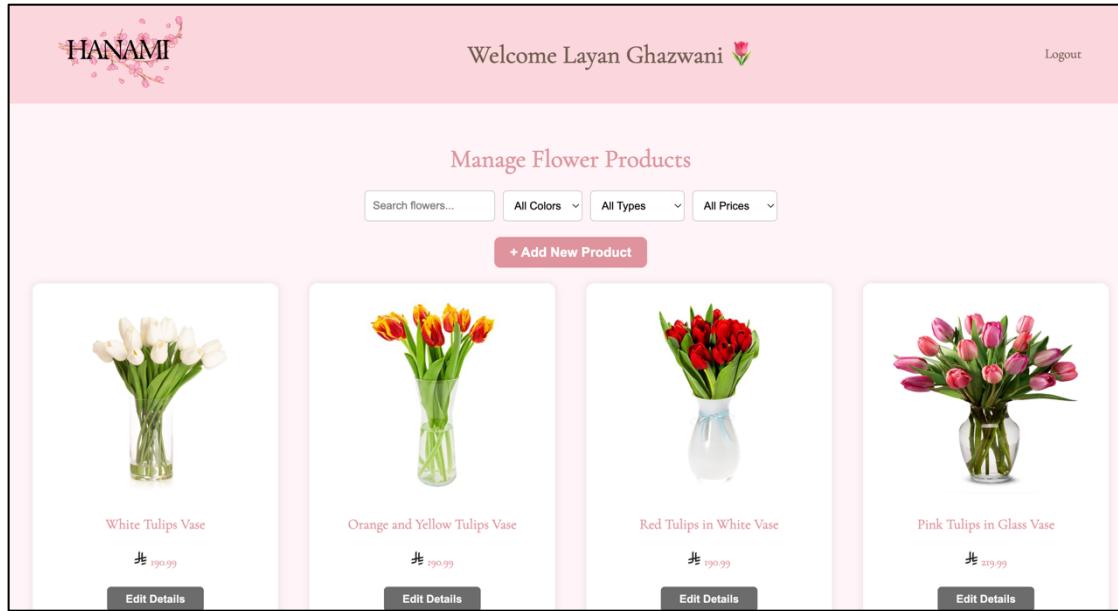


Figure 51 Manage product page when the admin login successfully

7. Manage Products (Admin Panel)

We tested the admin panel for adding, editing, and deleting products. When a new product was added, it appeared immediately on the products page. We also tested searching for existing products, modifying them (e.g., updating stock or price), and deleting them from the database. All changes reflected correctly in both the admin view and customer-facing pages.

Add product test:

Table 6 Add product test

Test ID	Add products
Prerequisite	Log in as an admin, click on “add new product” button
Test procedure	Enter product details then click on “save product” then the product will immediately appear on the products page.
Actual Result	A pop-up message saying "thank you, product has been saved successfully"
Verified (Yes/No)	yes

Edit product:

Table 7 Edit product test

Test ID	edit products
Prerequisite	Log in as an admin , click on “edit” button
Test procedure	The products will be displayed, each with "Edit" and "Delete" buttons. The admin can click "Edit" under the desired product to modify its details, then click "Save" to apply the changes.
Actual Result	A pop up message saying“thank you, product has been saved successfully”
Verified (Yes/No)	yes

Delete products:*Table 8 Delete product test*

Test ID	delete products
Prerequisite	Log in as an admin, click on "delete" button
Test procedure	The products will be displayed, each with "Edit" and "Delete" buttons. The admin can click "delete" under the desired product to delete it
Actual Result	A pop-up message saying "the product deleted successfully"
Verified (Yes/No)	yes

Search products test:*Table 9 Searching product test*

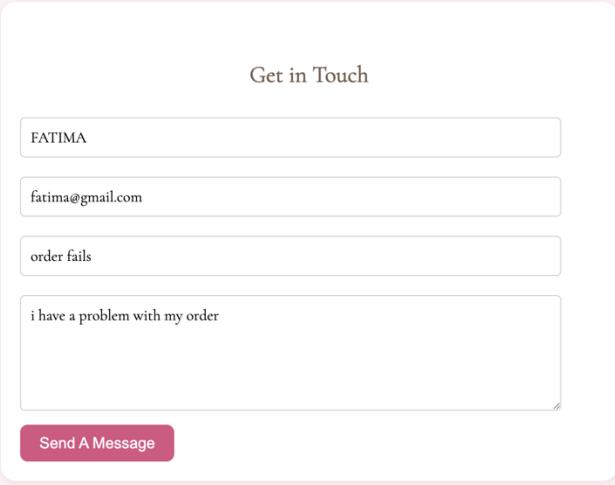
Test ID	search products
Prerequisite	Log in as an admin, then use the search bar to search by name or apply filters.
Test procedure	Use the search bar to search by name or apply filters. Matching results will automatically appear as you type.
Actual Result	The admin gets the products that match the search or filters, if there is no matching a message "You have great taste! Unfortunately, no products match your search. Try adjusting your filters." Will be shown
Verified (Yes/No)	yes

8. Contact Us Page

This page was tested to ensure the form includes JavaScript validation for all required fields. The embedded Google Map successfully displayed the shop's physical location. Form submissions triggered confirmation messages after validation.

Table 10 Contact Form

Test ID	Contact_Form
Feature	Contact Us Page – Form Validation and Map Display
Prerequisite	Contact Us page is accessible; JavaScript is enabled
Test Procedure	1. Navigate to the Contact Us page 2. Leave required fields empty and click Submit 3. Observe if a pop-up alert appears requiring user action 4. Enter an invalid email format (e.g., "user@") and submit 5. Verify if an error message appears or submission is blocked 6. Fill in all fields correctly and click Submit 7. Check for a confirmation message saying "Thank you! Your message has been sent." 8. Confirm that the embedded Google Map is visible and accurate
Expected Result	<ul style="list-style-type: none"> - Pop-up alert appears when fields are left empty - Invalid email is rejected - Form submits only with valid input - Success message is shown - Google Map is displayed correctly
Actual Result	<ul style="list-style-type: none"> - Alert shown for empty fields - Invalid email blocked - Success message: "Thank you! Your message has been sent." shown on valid input - Google Map loaded correctly
Verified (Yes/No)	Yes



A screenshot of a contact form titled "Get in Touch". The form contains four input fields: "FATIMA", "fatima@gmail.com", "order fails", and "i have a problem with my order". Below the form is a red button labeled "Send A Message". At the bottom of the page, a success message says "Thank you! Your message has been sent."

Figure 52 Contact Us - Success Message

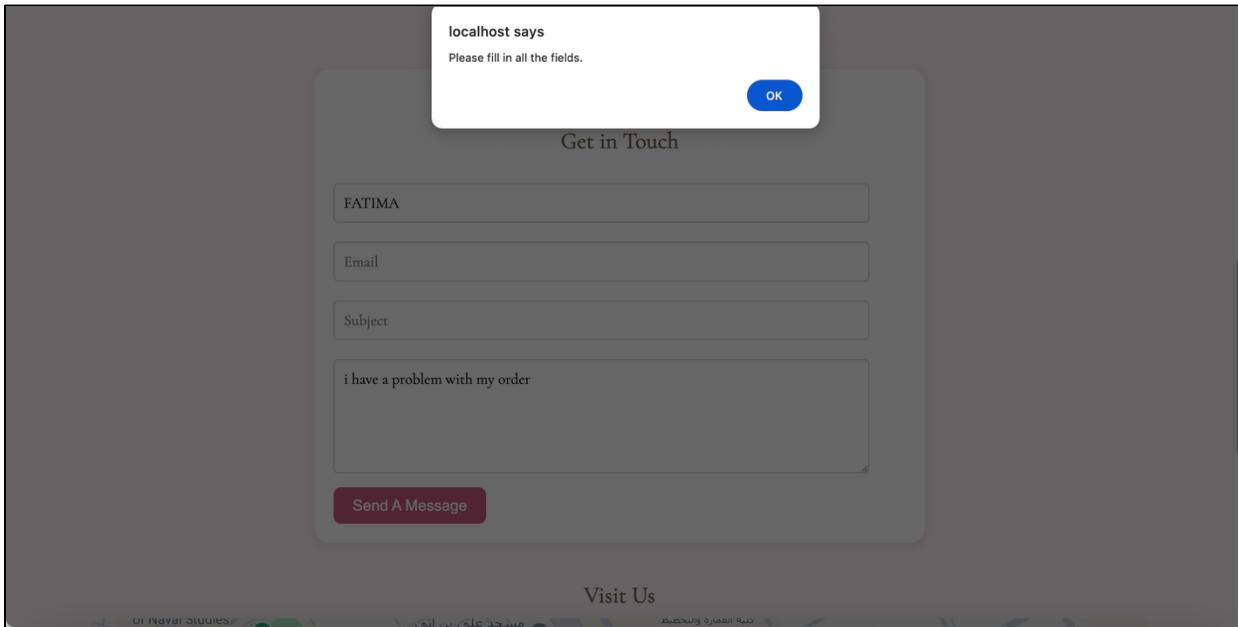


Figure 53 Contact Us-empty fields alert

9. Past Purchases with Cookies

We verified that past purchases are tracked using browser cookies. Returning customers were able to see a history of their previous orders directly on the homepage. This feature enhanced the personalized shopping experience.

Table 11 Cookies Functionality

Test ID	Cookies_Functionality
Feature	Track and Display Past Purchases Using Cookies
Prerequisite	User completes a purchase; cookies must be enabled in the browser
Test Procedure	1. Add one or more products to the cart 2. Proceed to checkout and complete the order 3. Ensure a cookie named previous_purchases is set 4. Return to the homepage or relevant page where past purchases are displayed
Expected Result	A list of previously purchased product names appears automatically for returning users in the home page
Actual Result	Past purchases were displayed correctly on the homepage.
Verified (Yes/No)	Yes

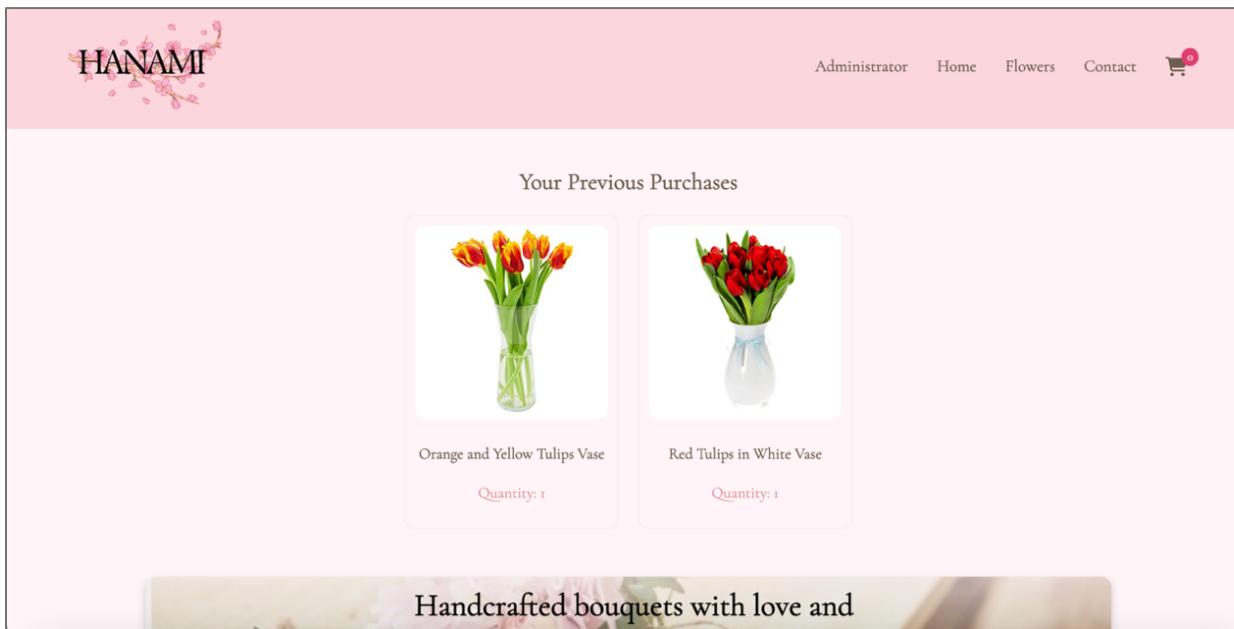


Figure 54 Past Purchases Appearing on Homepage

10. Form Validation

All forms throughout the website (login, checkout, contact us, product addition) were tested for validation. JavaScript was used to ensure all required fields are filled in correctly before submission. Errors were clearly displayed when inputs were missing or invalid.

Table 12 Form validation

Test ID	Form_Validation
Feature	Form Validation (Login, Checkout, Contact Us, Product Addition)
Prerequisite	JavaScript is enabled; user has access to all relevant forms
Test Procedure	1. Navigate to each form: Login, Checkout, Contact Us, and Product Addition 2. Attempt to submit each form with one or more required fields left empty 3. Observe if error messages or alert boxes appear 4. Input invalid data (e.g., wrong email format) 5. Submit again 6. Correct all inputs and submit to verify that forms proceed only when all fields are valid
Expected Result	<ul style="list-style-type: none"> - JavaScript validation blocks form submission if required fields are missing or invalid - Error messages or alerts are clearly displayed - Forms submit successfully only with correct inputs
Actual Result	<ul style="list-style-type: none"> - Validation prevented incomplete/incorrect submissions - Clear error messages or alert dialogs were shown - Valid forms submitted without issues
Verified (Yes/No)	Yes

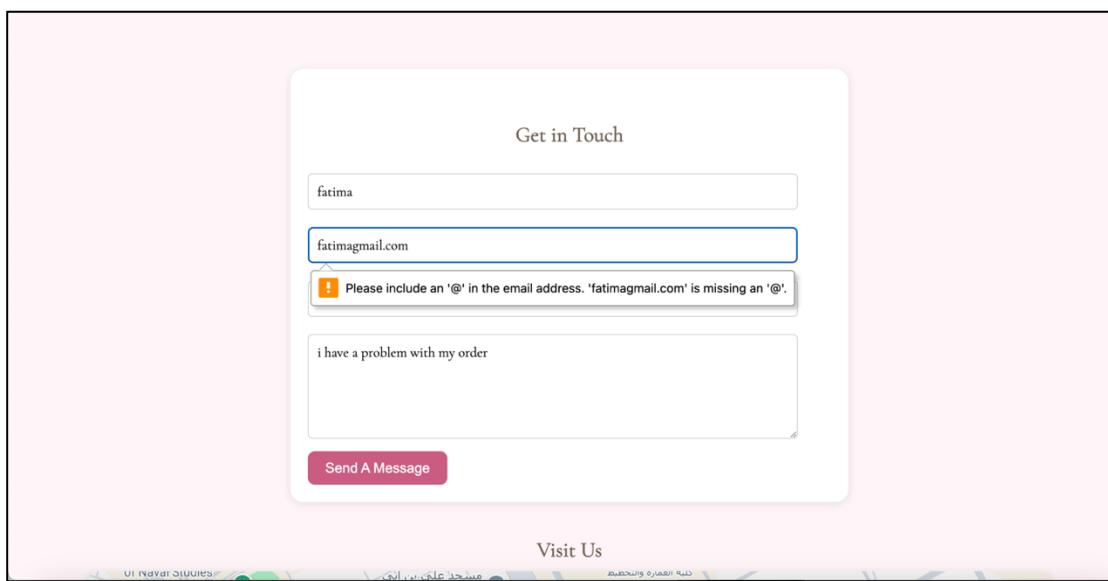


Figure 55 Contact Us- email format

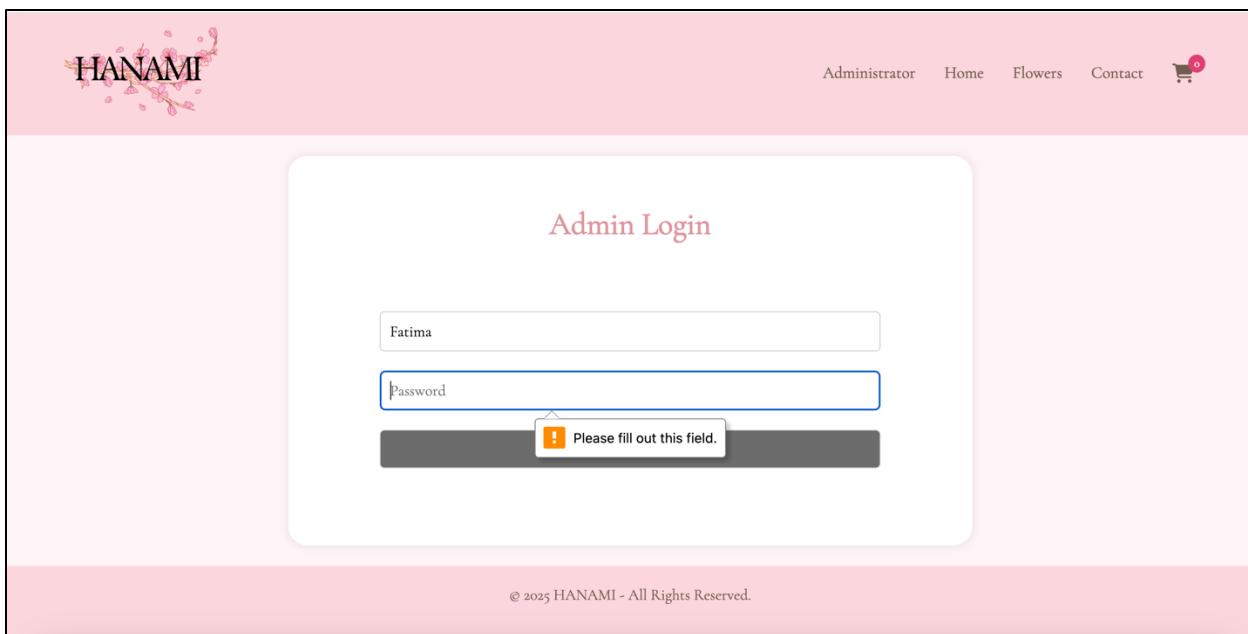


Figure 56 Admin Login: Empty Field alert

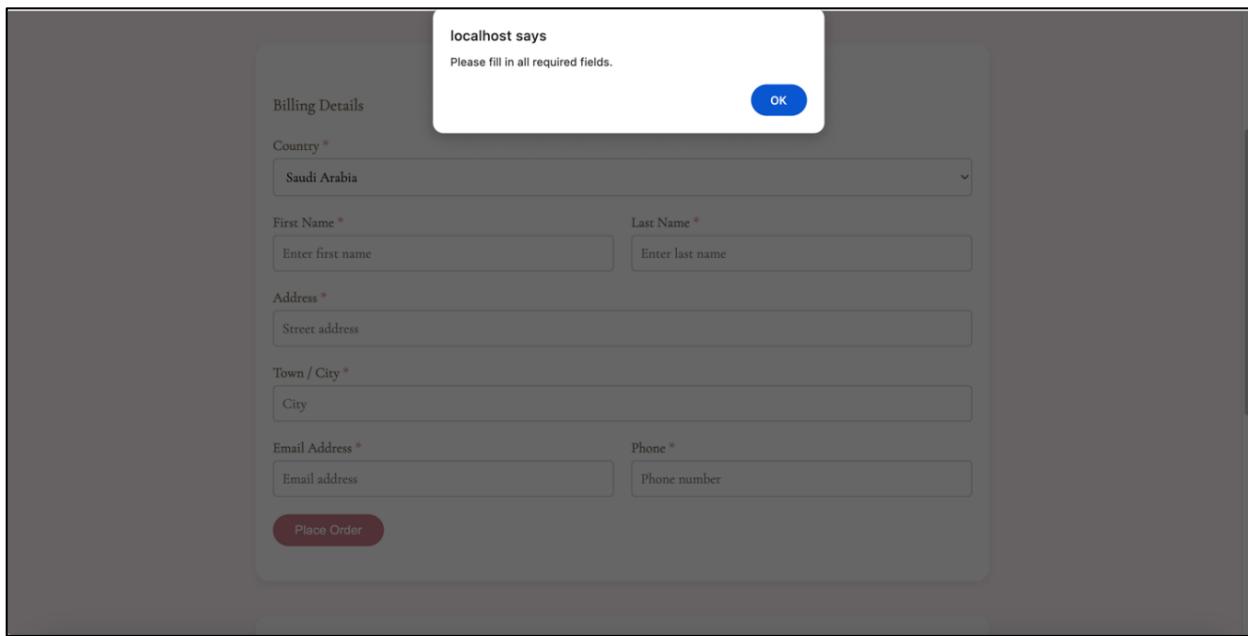


Figure 57 Checkout: Empty Field alert

11. Accessibility Testing

The website was reviewed to ensure it is accessible to users with disabilities. We verified that images include alt text, all form elements have associated labels, and navigation is possible via keyboard. Contrast and font sizes were also reviewed for readability.

12. Code Efficiency and Structure

The backend PHP code was tested for efficiency. We ensured reusable functions were used instead of repetitive code. PHP include statements helped separate layout and logic, improving maintainability. Overall performance was consistent with minimal load time.

8. Project Implications:

1. Issues Faced by the Business (Floral Shop)

- Manual ordering process and limited online presence.
- Lack of custom bouquet creation options.
- Difficulty tracking inventory and managing orders.
- No system for storing past purchases or user feedback.

2. Technological Features Solving the Problems

- **E-Commerce Integration:** Online ordering and checkout with cart system.
- **Customizable Bouquets:** Product detail page with flexible options.
- **Admin Panel:** Manage stock and orders easily without technical expertise.
- **Authentication:** Secure login for admin access and customer sessions.
- **Cookies:** Show past orders to improve customer experience.
- **Accessibility Enhancements:** Inclusive design for all users.
- **Validation & Efficiency:** JavaScript for front-end checks, PHP includes and modular structure for backend maintainability.

3. New Challenges Introduced by the System

- Dependency on internet and device compatibility.
- Required training for non-technical staff to manage the admin panel.
- Future need for scalability (e.g., adding payment gateways or multi-language support).
- Data privacy concerns (e.g., cookies, sessions, order history).

4. Tools That Helped Build the System

- **HTML/CSS/JavaScript:** Structured, styled, and validated the UI.
- **PHP:** Enabled server-side logic, sessions, and database operations.
- **MySQL & phpMyAdmin:** Simplified database design and testing.
- **XAMPP:** Local server for testing and integration.
- **VS Code:** Unified development environment with Git support and debugging extensions.