

The platform I created is called Blooming bouquet, an online flower shop. This platform allows you to order flowers bouquets from the various options provided by the shop. It also allows you to select how many bouquets you want to buy and shows you the total price of your order after selecting the bouquets. There is a contact information page which allows you to send messages by entering your name, email address, and message you want to send. This platform has processcontact.php, processorder.php, and eight pages which are Home, Our Bouquets, Halloween, Occasions, Best-selling, Order Now, Contact us, and About us to enhance the platform's usability and navigation capabilities.

The Home page includes the code that renders brief information about the services provided by the shop paired with images. In the code, the body section includes a header, navigation menu, main section, and a footer. It's where the visible content goes. Header introduces the title of the shop whereas footer has links for contact and about us page, and copyright notice. The main section explains the service offered by the shop. It has a table with two rows which feature images and description of some of the works done by the shop. Lastly, the navigation menu, which is the same for all pages, contains links to various pages of the website such as Home, Our Bouquet, Halloween, Occasions, Best-selling, and Order Now. Each menu item is wrapped in a div with a class called menuitem, and links are created using the <a> tag.

The Our Bouquets page displays most of the shop's products with beautifully arranged bouquets images and their names. This page has a code that includes a header, navigation menu, main section, and a footer. Header and footer are the same as every other page with a header having the shop name and footer with links for contact and about us page, and copyright notice. The navigation menu is the same as others page, with links with names to other service pages

and a home page. The main section has a heading named our products. There is a table with three rows and four columns, each having the products image link and name.

Next up is the Halloween page that shows you the curated bouquets for Halloween occasion. The code also includes a header, navigation menu, main section, and a footer. Header, footer, and navigation menu are the same as every other page. The main section includes a table with images of the bouquets and their names.

The occasions page shows you the bouquets for any special occasion such as birthday, valentine, anniversary, and so on. Similarly, the best-selling page shows you the popular bouquets. These two pages have the same header, navigation menu, and a footer as other pages.

The order now page also has the same header, footer, and navigation menu as the other pages. The main section of the page has a form which allows users to submit data. The `action="processorder.php"` attribute tells the browser to send the form data to `processorder.php` for processing when the form is submitted. The `method="POST"` is a secure way for sending sensitive data. `<label for="name">Your Name:</label>` is a label for the first input field. The `for="name"` matches the `id="name"` of the input field below. Labels improve accessibility and help screen readers identify the field. `<input type="text" id="name" name="name" required>` is a text input field for users to enter their name. The required attribute makes it a mandatory field. The `id="name"` matches the label's `for` attribute, and `name="name"` will be used when the form is submitted.

`<label for="product">Select Bouquet:</label>` asks the user to choose a bouquet from the dropdown menu. `<select id="product" name="product" required>` is a dropdown menu where the user can select one of the bouquets. The required attribute ensures that the user must choose one

option. Inside the `<select>` tag, there are `<option>` tags that have different bouquet options, each with a name and price. `<label for="quantity">Quantity:</label>` ask the user to choose how many bouquets they want to order. `<input type="number" id="quantity" name="quantity" min="1" required>` is the field where the users can tell the number of bouquets they want to buy. The `min="1"` ensures the users they can't choose a number less than 1. `<button type="submit">Place Order</button>` is a submit button that sends the form data to the server when clicked.

The `processorder.php` processes an order submitted by the user. It collects the user's name, selected bouquet, and quantity from the form. It also calculates the total cost based on the quantity ordered. `if ($_SERVER["REQUEST_METHOD"] === "POST")` checks if the form is submitted using the POST method. `$name = htmlspecialchars(trim($_POST['name']));` in this line, `$_POST['name']` retrieves the user's name from the form submission, `trim()` removes any leading or trailing spaces from the name, `htmlspecialchars()` converts any special characters into HTML entities to protect from any malicious code. `$product = htmlspecialchars(trim($_POST['product']));` retrieves the selected bouquet's name from the form, trims extra spaces, and sanitizes it for security. `$quantity = (int)$_POST['quantity'];` retrieves the quantity input from the form. `(int)` ensures that the input is treated as the number for calculations.

`$prices = ["Blooming beauty" => 79.99, "Happy Fall" => 79.99, "Majestic Meadow" => 69.99, "Roses for her" => 89.99, "Autumn Splendor" => 59.99, "Autumn Bouquet" => 59.99, "Mix Bouquet" => 79.99, "Bunch of Daffodil" => 49.99, "Holiday Greetings" => 59.99, "Tulips!" => 79.99, "Ghost Bouquet" => 59.99, "Mystery Bouquet" => 79.99, "Spooky Bouquet" => 79.99];` creates an array called `$prices`. The keys of the array are the name of the bouquets, and the values are the corresponding prices for those bouquets.

`$total_price = $prices[$product] * $quantity;` in this code, `$prices[$product]` retrieves the bouquet's price. The `$product` variable stores the name of the selected bouquet. `* $quantity` multiplies the price by the quantity selected by the user to calculate the total price.

`echo "<h1>Order Confirmation</h1>";` displays the main heading for the order confirmation page. `echo "<p>Thank you, $name!</p>";` displays a thank you message including the user's name. `echo "<p>You ordered $quantity x $product.</p>";` shows the quantity and the name of the bouquet ordered. `echo "<p>Total Price: \$$total_price</p>";` shows the total price of the order.

```
else {  
  
    echo "Invalid request method."  
  
}
```

this block is executed if the form was not submitted using the POST method , and shows the message Invalid request method.

The `processcontact.php` is used for the contact forms where a user sends a data to the server. `if ($_SERVER["REQUEST_METHOD"] == "POST")` ensures that the form is submitted using the POST method. `$name = htmlspecialchars(trim($_POST['name']));` collects the user's name. `$email = htmlspecialchars(trim($_POST['email']));` collects the user's email address. `$message = htmlspecialchars(trim($_POST['message']));` collects the message the user wrote in the contact form. All these codes have `trim()` and `htmlspecialchars()` function. The `trim()` removes any unnecessary spaces. `htmlspecialchars()` converts the special characters like `<`, `>`, `&` into HTML entities to protect from the malicious code.

echo "<h1>Message Sent!</h1>"; shows "Message Sent!" to tell the user that the form has been submitted successfully. echo "<p>Thank you, \$name! Your message has been received.</p>"; shows personalized message with the user's name.

```
else {  
  
    echo "Invalid request method."  
  
}
```

This code block is executed if the form is not submitted using the POST method and will display the message "Invalid request method."

In the style.css code, the header and footer have black background with white text and text is centered. H1 and h2 have white text aligned left with the font size to 24 points, using Arial font, and the headers are placed to the top of their container. The navigation bar has a black background color and text is aligned to the left. Menu items have white text with the font of 12 points and aligned to the left. The .menuitems are aligned to the top of their container with 5px padding and width of 150px. The .menutext is aligned vertically to the top of its container. The paragraph of the main section is black in color, with font of 12 points, text is justified and uses Arial font. The footer has white text color with a font of 9 points, text is centered and bold, uses Arial font, and has 6px padding. The main section has a space of 20px around the content. The table cell paragraph has text aligned to the center, headers are centered, and image have a width of 500px, height of 400px, and fills the space maintaining its aspect ratio.