



SWEET DELIGHT BAKERY.

INTRODUCTION

“Craving a slice of heaven? Look no further! .Sweet Delight Bakery is your one -stop shop for the most delectable cakes in Mauritius .From classic to contemporary , we have something to satisfy every sweet tooth.”

hp

Student name: Mitchell Dzapasi

Student ID:2412069

Student name: Ghuness
Appadoo

Student ID: 2414109

Screenshots and brief descriptions of the website.

1. Navigation and Layout

This section explains the structure and functionality of the navigation system and layout across the website, including responsive design features for different screen sizes.

Header & Navigation bar

- The website features a sticky header that remains at the top of the page as users scroll. The navigation bar includes links to different sections (Home, About Us, Products, Contact) and is equipped with a responsive hamburger menu for mobile devices.
- The logo and title are clickable, allowing users to return to the homepage.
- Responsive Design: The header's responsiveness is achieved through CSS media queries and the usage of Flexbox. On smaller screens, the hamburger icon triggers the navigation menu, ensuring a user-friendly experience.

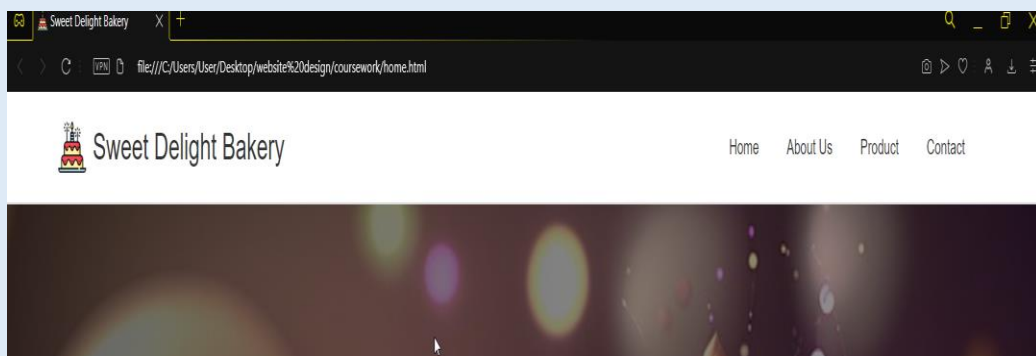


Figure 1 Header & Navbar

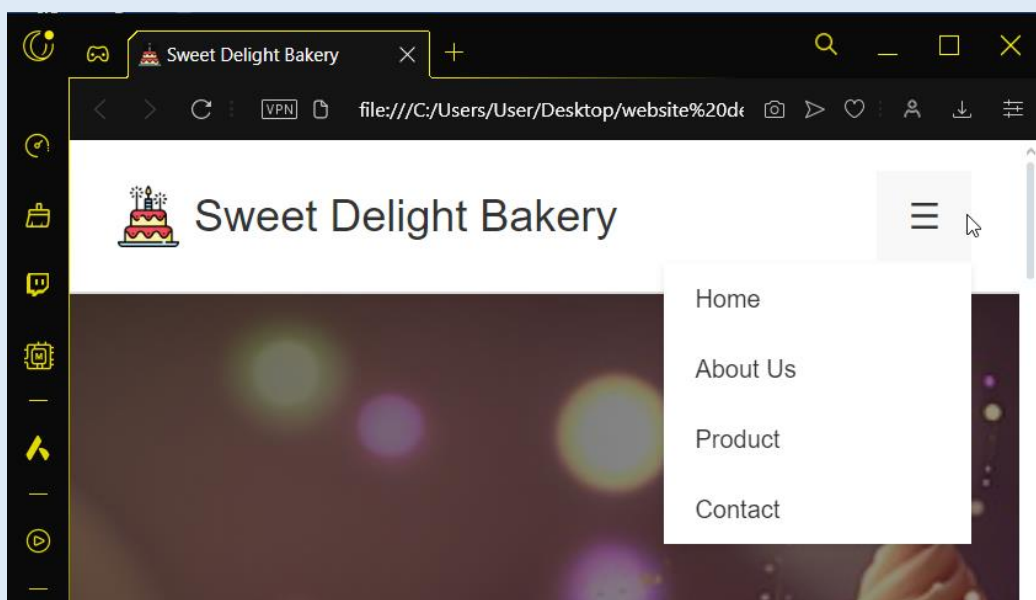


Figure 2 Hamburger for smaller screen

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Sweet Delight Bakery</title>
7      <link rel="stylesheet" href="home_styles.css">
8      <link rel="icon" href="images/birthday-cake.png" type="image/png">
9      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">
10 </head>
11 <body>
12     <header>
13         <div class="container">
14             <a href="home.html" class="logo">
15                 
16                 Sweet Delight Bakery
17             </a>
18             <nav class="navbar">
19                 <div class="hamburger" id="hamburger">
20                     &#9776; <!-- Hamburger icon -->
21                 </div>
22                 <ul id="navbar-links">
23                     <li><a href="home.html">Home</a></li>
24                     <li><a href="about.html">About Us</a></li>
25                     <li><a href="product.html">Product</a></li>
26                     <li><a href="contact.html">Contact</a></li>
27                 </ul>
28             </nav>
29         </div>
30     </header>
31
32

```

Figure 3 Snippet of html structure for header & Navbar

```

15  /* Header Styles */
16  header {
17      background: #ffff;
18      padding: 20px 0;
19      border-bottom: 2px solid #ddd;
20      position: sticky;
21      top: 0;
22      z-index: 1000;
23  }
24
25  .container {
26      width: 90%;
27      margin: 0 auto;
28      display: flex;
29      justify-content: space-between;
30      align-items: center;
31  }
32
33  .logo {
34      display: flex;
35      align-items: center;
36      font-size: 1.8em;
37      color: #333;
38      flex-grow: 1;
39      text-decoration: none;
40  }
41
42  .logo:hover {
43      color: #c87c6b;
44  }

```

Figure 4 part of css code for header

```

52  /* Navbar Styles */
53  .navbar {
54      display: flex;
55      align-items: center;
56      justify-content: space-between;
57      background-color: #fff;
58      padding: 10px 20px;
59      transition: background-color 0.3s;
60      position: relative;
61  }
62
63  /* Hover effect for navbar */
64  .navbar:hover {
65      background-color: #f8f8f8;
66  }
67
68  .hamburger {
69      display: none; /* Hidden by default */
70      font-size: 1.5em;
71      cursor: pointer;
72      color: #333;
73  }
74
75  /* Navbar Links */
76  #navbar-links {
77      list-style: none;
78      display: flex; /* Default display */
79      gap: 20px;
80  }
81  #navbar-links.active {
82      display: flex; /* Show when active */
83  }

```

Figure 5 part of css code for navbar

Hero section

The homepage consists of a hero image and a 'Discover our offers' button. The button links to the promotions section, improving the user journey by highlighting key offerings immediately upon visiting the homepage

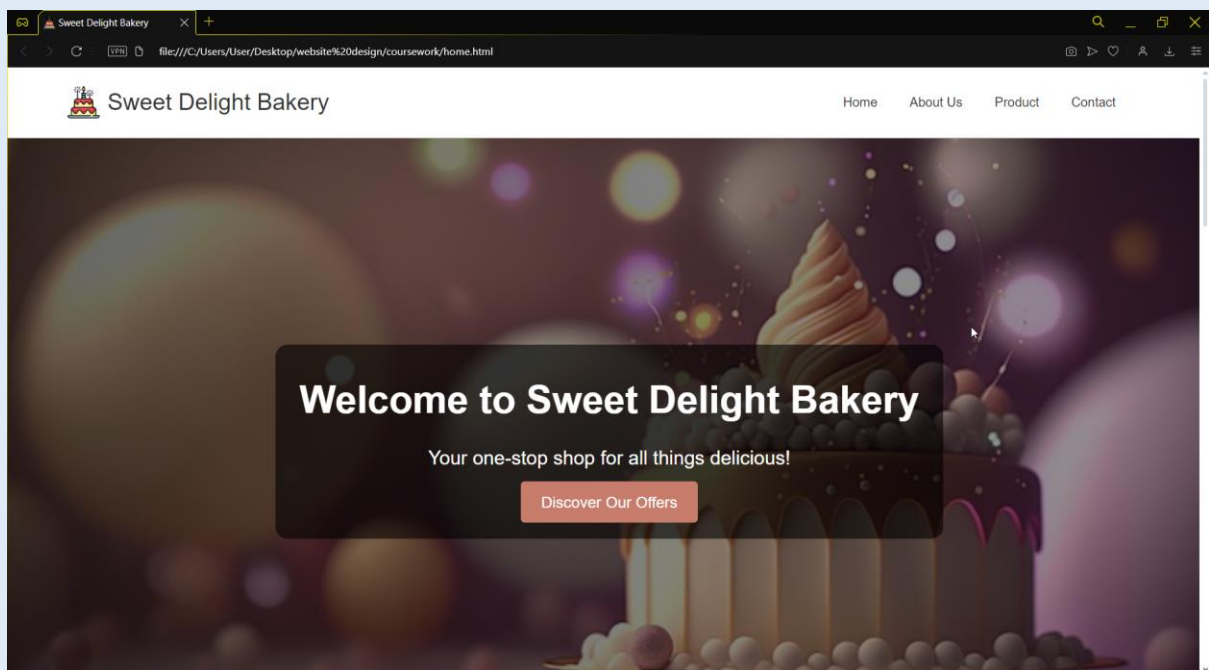


Figure 6 Hero image

```

33
34     <section id="hero" class="hero">
35         <div class="hero-content">
36             <h2>Welcome to Sweet Delight Bakery</h2>
37             <p>Your one-stop shop for all things delicious!</p>
38             <a href="#promotions" class="learn-more-btn">Discover Our Offers</a>
39         </div>
40     </section>
41

```

Figure 7 html code for hero image

```

97  /* Hero Section Styles */
98  #hero {
99      position: relative;
100     height: 100vh;
101     width: 100vw;
102     background: linear-gradient(to right, rgba(0, 0, 0, 0.6), rgba(0, 0, 0, 0.4)), url('images/hero-bg.jpg');
103     display: flex;
104     justify-content: center;
105     align-items: center;
106     color: #fff;
107 }
108
109 .hero-content {
110     text-align: center;
111     z-index: 1;
112     padding: 30px;
113     background: rgba(0, 0, 0, 0.6); /* Darker overlay for better text contrast */
114     border-radius: 15px;
115 }
116
117 .hero-content h2 {
118     font-size: 3em;
119     margin-bottom: 15px;
120     font-weight: 700;
121 }
122
123 .hero-content p {
124     font-size: 1.5em;
125     margin-bottom: 20px;
126 }
127

```

Figure 8 part of css code for hero section

Carousel image slider

- The website features a carousel on the homepage to showcase the best-selling cakes.
- Functionality: The carousel allows users to navigate between featured items using next and previous buttons. Each image slides in and out of view with smooth transitions.
- JavaScript is used to power the carousel's functionality, providing users with a smooth experience while navigating through featured products
- Responsive: The carousel adapts to different screen sizes.

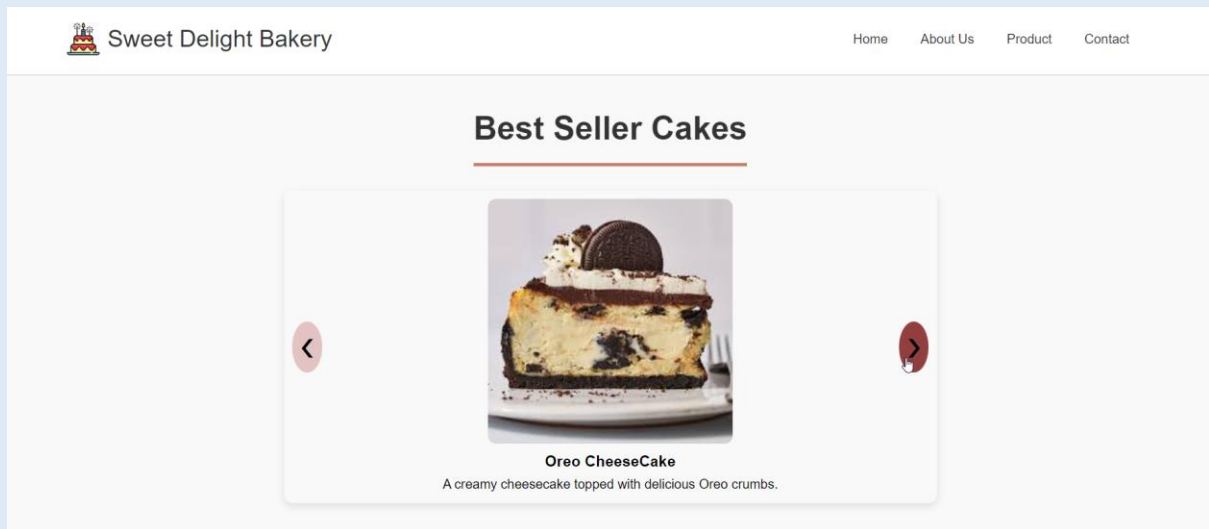


Figure 9 Carousel image slider

```
<section id="best-sellers">
  <h2>Best Seller Cakes</h2>
  <div class="carousel">
    <div class="carousel-track">
      <div class="carousel-item">
        
        <h3>Chocolate Cake</h3>
        <p>Rich and decadent chocolate cake topped with ganache.</p>
      </div>
      <div class="carousel-item">
        
        <h3>Red Velvet Cake</h3>
        <p>Classic red velvet cake with cream cheese frosting.</p>
      </div>
      <div class="carousel-item">
        
        <h3>Oreo Cheesecake</h3>
        <p>A creamy cheesecake topped with delicious Oreo crumbs.</p>
      </div>
      <div class="carousel-item">
        
        <h3>Vegan fruitcake</h3>
        <p>A rich and moist cake packed with assorted fruits, perfect for everyone.</p>
      </div>
      <div class="carousel-item">
        
        <h3>Honeybee Cupcake</h3>
        <p>Honey is the star ingredient in these cupcakes, from the batter to the buttercream.</p>
      </div>
    </div>
  </div>
```

Figure 10 Html code for carousel slider

```

425 .carousel {
426     position: relative;
427     overflow: hidden;
428     width: 100%;
429     max-width: 800px; /* Maximum width for larger screens */
430     margin: auto;
431     border-radius: 10px;
432     box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
433 }
434
435 .carousel-track {
436     display: flex;
437     transition: transform 0.3s ease-in-out;
438 }
439
440 .carousel-item {
441     min-width: 100%; /* Each item takes full width */
442     box-sizing: border-box;
443     padding: 10px;
444 }
445
446 .carousel-item img {
447     width: 100%;
448     height: auto; /* Maintain aspect ratio */
449     max-width: 300px; /* Set a maximum width for images */
450     border-radius: 10px;
451     object-fit: cover; /* Ensure images fill the container */
452 }
453
454 .carousel-button {
455     position: absolute;

```

Figure 11 part of css code for carousel slider

```

10
11 document.addEventListener('DOMContentLoaded', () => {
12     const track = document.querySelector('.carousel-track');
13     const items = document.querySelectorAll('.carousel-item');
14     const nextButton = document.querySelector('.next');
15     const prevButton = document.querySelector('.prev');
16
17     let currentIndex = 0;
18
19     function updateCarousel() {
20         const itemWidth = items[0].clientWidth;
21         track.style.transform = `translateX(${ -currentIndex * itemWidth }px)`;
22     }
23
24     nextButton.addEventListener('click', () => {
25         currentIndex = (currentIndex + 1) % items.length;
26         updateCarousel();
27     });
28
29     prevButton.addEventListener('click', () => {
30         currentIndex = (currentIndex - 1 + items.length) % items.length;
31         updateCarousel();
32     });
33 });
34

```

Figure 12 JavaScript code for carousel slider

Footer section

- This section provides quick access to social media and important links, as well as a newsletter subscription option, improving user engagement.
- Font-awesome library is used for icons of social links.

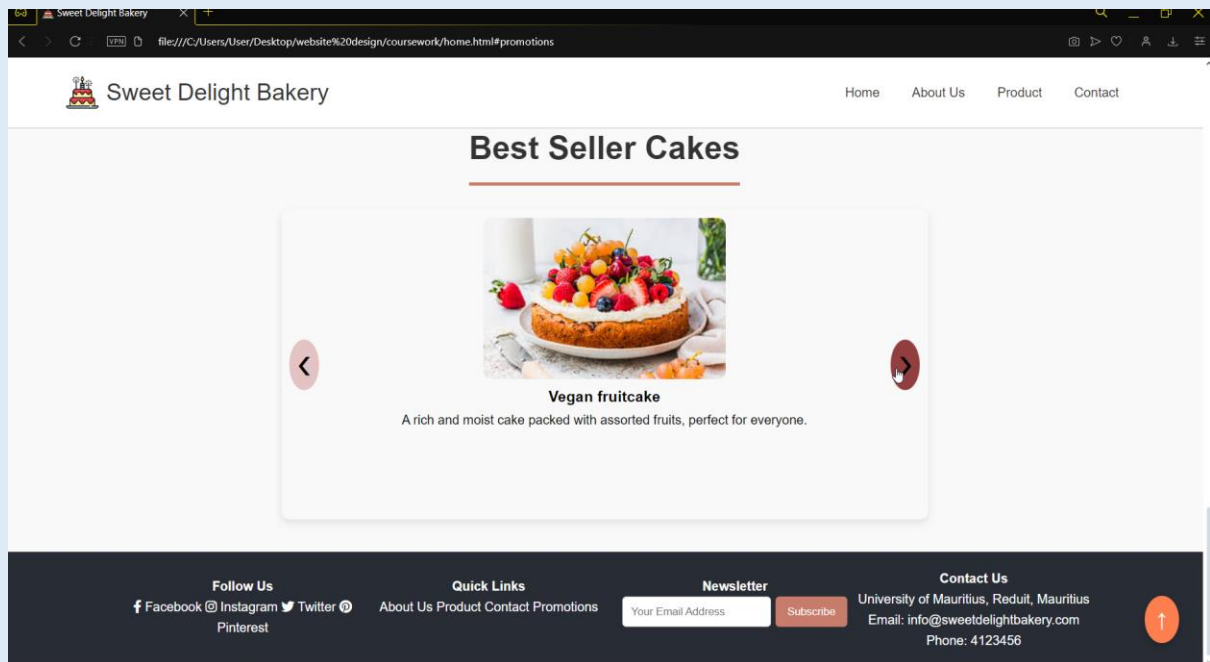


Figure 13 Footer section


```

<footer>
  <div class="footer-container">
    <div class="footer-section">
      <h4>Follow Us</h4>
      <div class="social-links">
        <a href="https://www.facebook.com"><i class="fab fa-facebook-f"></i> Facebook</a>
        <a href="https://www.instagram.com"><i class="fab fa-instagram"></i> Instagram</a>
        <a href="https://twitter.com"><i class="fab fa-twitter"></i> Twitter</a>
        <a href="https://www.pinterest.com"><i class="fab fa-pinterest"></i> Pinterest</a>
      </div>
    </div>
    <div class="footer-section">
      <h4>Quick links</h4>
      <div class="quick-links">
        <a href="about.html">About Us</a>
        <a href="product.html">Product</a>
        <a href="contact.html">Contact</a>
        <a href="#promotions">Promotions</a>
      </div>
    </div>
    <div class="footer-section footer-newsletter">
      <h4>Newsletter</h4>
      <form>
        <input type="email" placeholder="Your Email Address" required>
        <button type="submit">Subscribe</button>
      </form>
    </div>
    <div class="footer-info">
      <h4>Contact Us</h4>
    </div>
  </div>

```

Figure 14 html code for footer section

```

275  /* Footer Styles */
276  footer {
277      background: #282c34;
278      color: #ffffff;
279      padding: 20px 0;
280      text-align: center;
281  }
282
283  .footer-container {
284      max-width: 1200px; /* Max width for larger screens */
285      margin: 0 auto;
286      display: flex;
287      flex-wrap: wrap;
288      justify-content: space-between;
289  }
290
291  .footer-section {
292      flex: 1 1 200px; /* Responsive layout */
293      margin: 10px;
294  }
295
296  .footer h4 {
297      color: #c87c6b;
298  }
299
300  /* Footer Links */
301  .footer-section a {
302      color: #ffffff;
303      text-decoration: none;
304      padding: 5px 0; /* Padding for touch targets */
305      transition: color 0.3s; /* Smooth transitions */
306  }

```

Figure 15 Css code for footer

Scroll-to-top button

Scroll Detection: A scroll-to-top button appears when the user scrolls more than 300 pixels down the page. This button provides a quick way to return to the top of the page with a smooth scrolling effect when clicked.

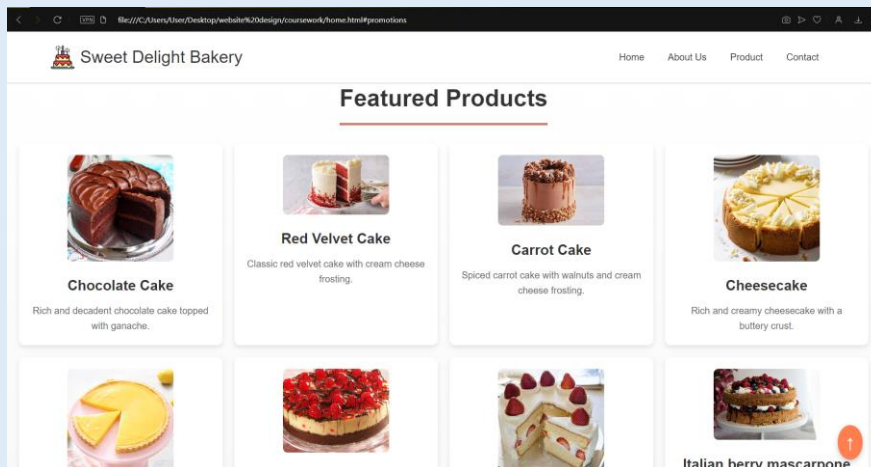


Figure 16 scroll to top button

```

188
189     <a href="#top" class="scroll-to-top" id="scrollToTopBtn">↑</a>

```

Figure 17 html code for scroll to top button

```

487
488     .scroll-to-top {
489         position: fixed;
490         bottom: 30px; /* Distance from the bottom */
491         right: 30px; /* Distance from the right */
492         background-color: #ff7f50;
493         color: white;
494         padding: 10px 15px;
495         border-radius: 50%;
496         text-decoration: none;
497         font-size: 1.5em;
498         box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
499         transition: background-color 0.3s, transform 0.3s;
500         opacity: 0; /* Initially hidden */
501         visibility: hidden; /* Initially hidden */
502         <element class="scroll-to-top visible"> /*
503         Selector Specificity: (0, 2, 0)
504     .scroll-to-top.visible {
505         opacity: 1; /* Show when visible */
506         visibility: visible; /* Show when visible */
507     }
508
509
510     .scroll-to-top:hover {
511         background-color: #ff6347;
512         transform: scale(1.1); /* Slightly enlarge on hover */
513     }
514
515

```

Figure 18 Css code for scroll to top button

```

35     // Scroll to Top Button
36     const scrollToTopBtn = document.getElementById('scrollToTopBtn');
37
38     window.addEventListener('scroll', () => {
39         if (document.body.scrollTop > 300 || document.documentElement.scrollTop > 300) {
40             scrollToTopBtn.classList.add('visible');
41         } else {
42             scrollToTopBtn.classList.remove('visible');
43         }
44     });
45
46     // Smooth Scroll Effect
47     scrollToTopBtn.addEventListener('click', (e) => {
48         e.preventDefault();
49         window.scrollTo({
50             top: 0,
51             behavior: 'smooth' // Smooth scrolling effect
52         });
53     });
54

```

Figure 19 JavaScript code for scroll to top button

About us page

The about us page displays information about the bakery

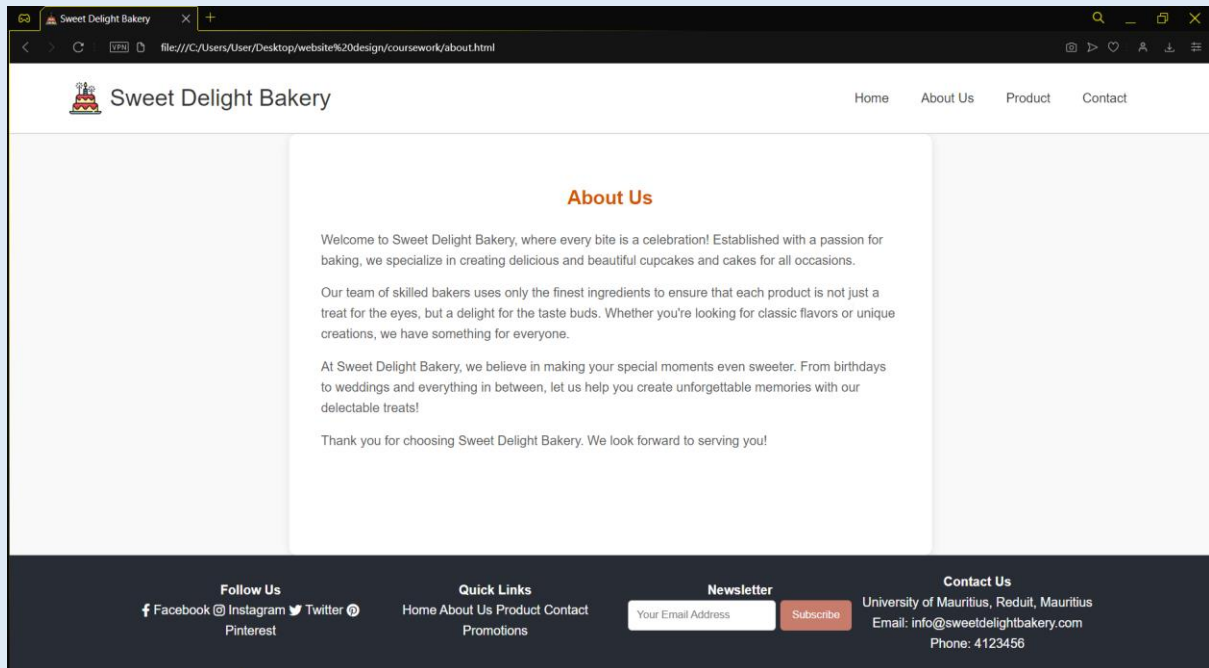


Figure 20 About us page

Product page

The product page displays different category of cakes.

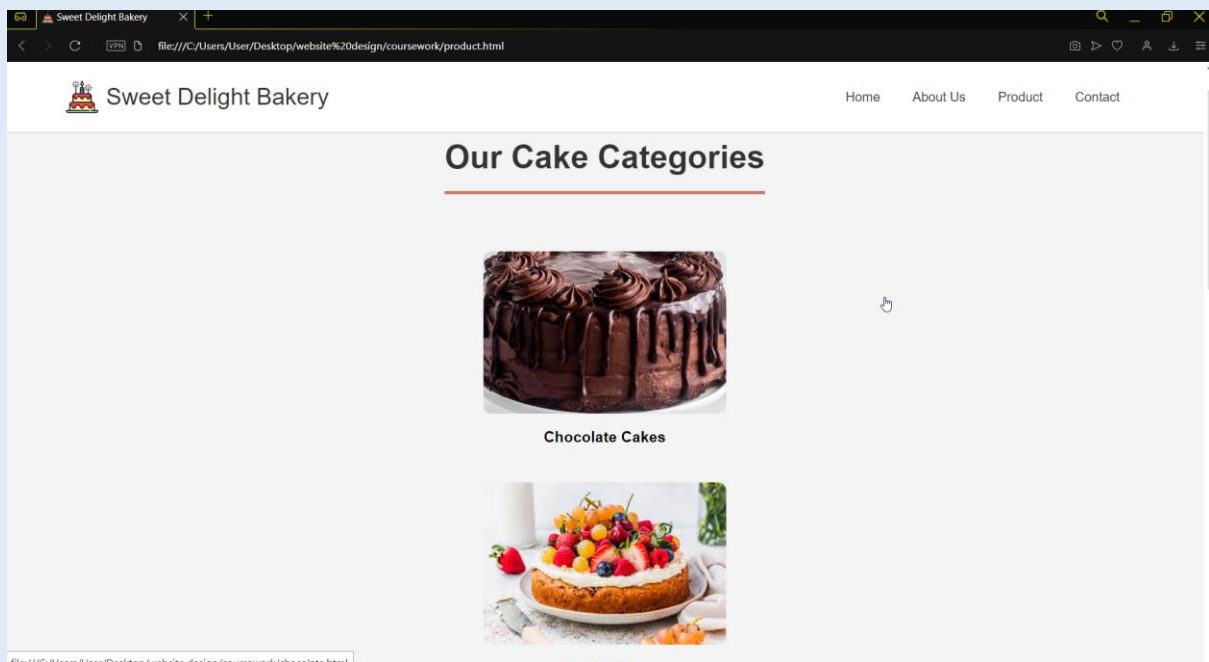


Figure 21 product page

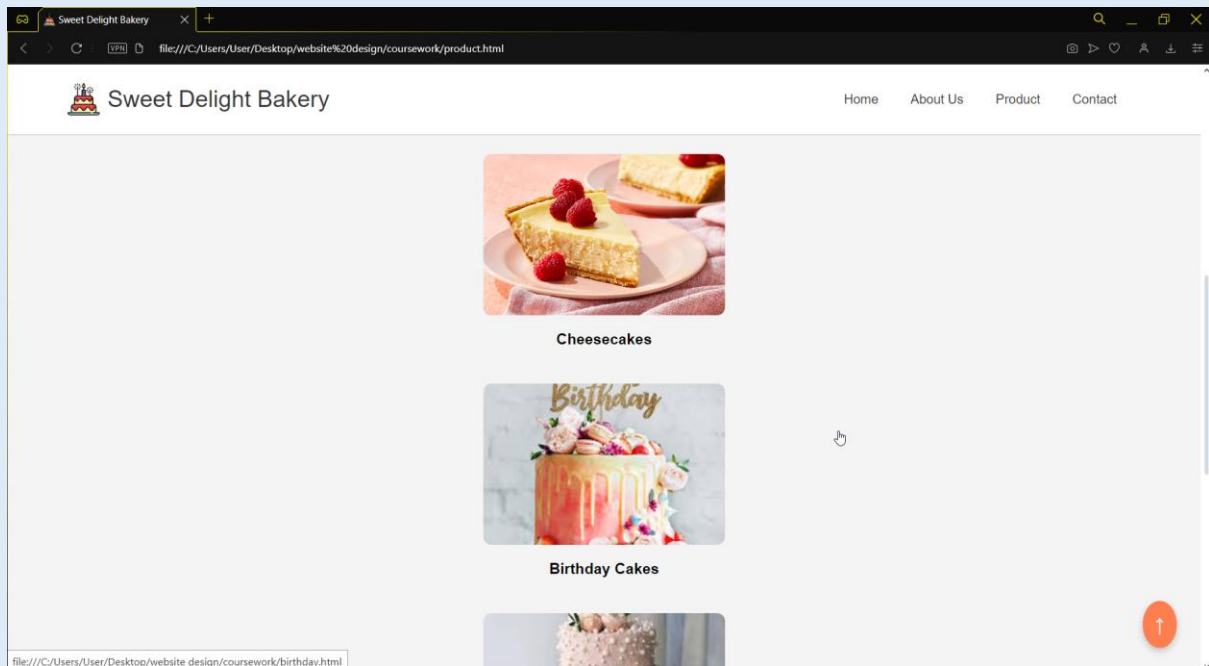


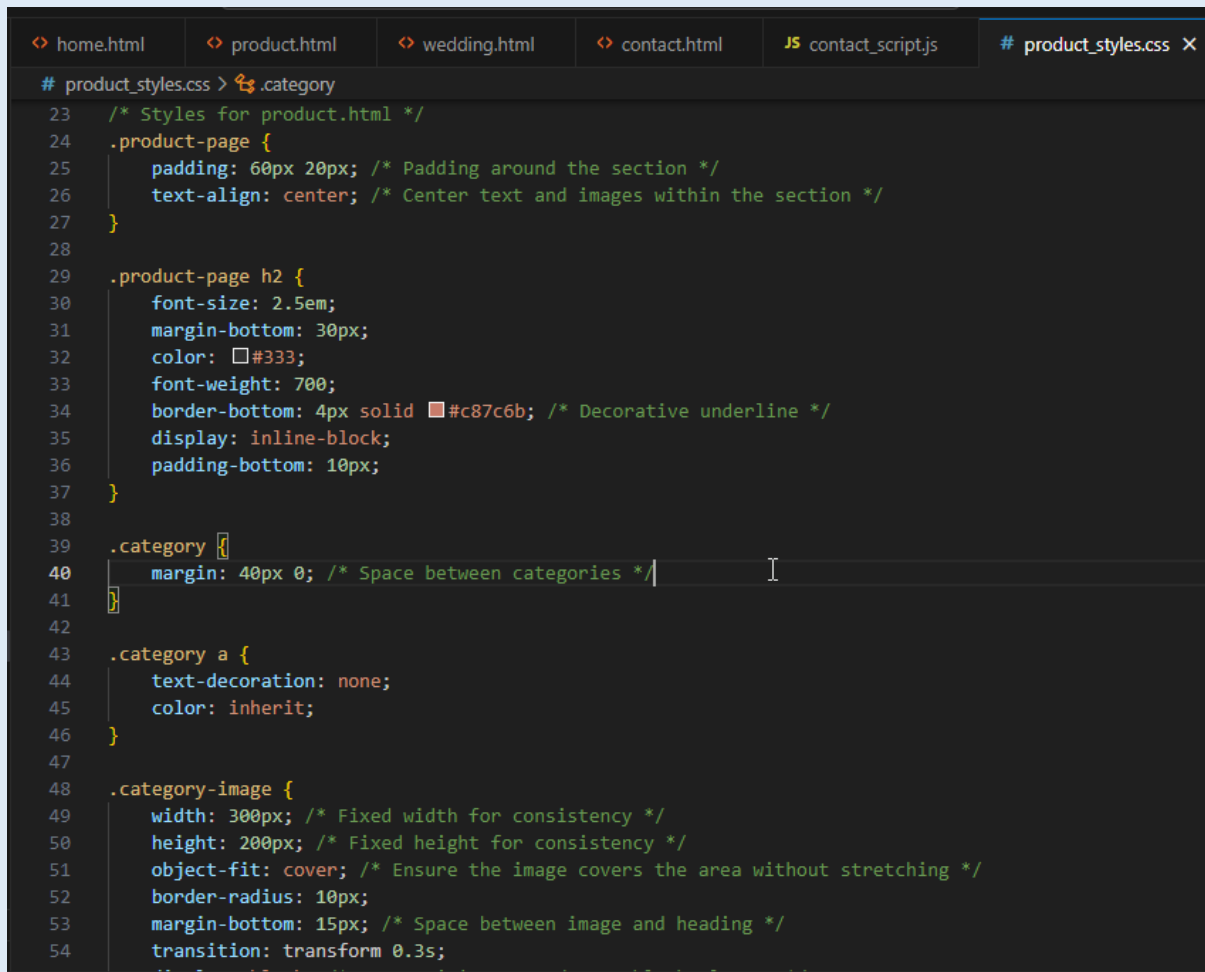
Figure 22 product page

```

home.html  product.html x  wedding.html  contact.html  JS contact_script.js  # product_styles.css  JS home_script.js
product.html > html > body > section.product-page
2  <html lang="en">
12 </body>
34 <section class="product-page">
35   <h2>Our Cake Categories</h2>
36   <div class="category">
37     <a href="chocolate.html">
38       
39       <h3>Chocolate Cakes</h3>
40     </a>
41   </div>
42   <div class="category">
43     <a href="fruitcake.html">
44       
45       <h3>Fruit Cakes</h3>
46     </a>
47   </div>
48   <div class="category">
49     <a href="cheesecake.html">
50       
51       <h3>Cheesecakes</h3>
52     </a>
53   </div>
54   <div class="category">
55     <a href="birthday.html">
56       
57       <h3>Birthday Cakes</h3>
58     </a>
59   </div>
60   <div class="category">
61     <a href="wedding.html">
62       
63       <h3>Wedding Cakes</h3>

```

Figure 23 html code for product page



```
# product_styles.css > .category
23  /* Styles for product.html */
24  .product-page {
25      padding: 60px 20px; /* Padding around the section */
26      text-align: center; /* Center text and images within the section */
27  }
28
29  .product-page h2 {
30      font-size: 2.5em;
31      margin-bottom: 30px;
32      color: #333;
33      font-weight: 700;
34      border-bottom: 4px solid #c87c6b; /* Decorative underline */
35      display: inline-block;
36      padding-bottom: 10px;
37  }
38
39  .category {
40      margin: 40px 0; /* Space between categories */
41  }
42
43  .category a {
44      text-decoration: none;
45      color: inherit;
46  }
47
48  .category-image {
49      width: 300px; /* Fixed width for consistency */
50      height: 200px; /* Fixed height for consistency */
51      object-fit: cover; /* Ensure the image covers the area without stretching */
52      border-radius: 10px;
53      margin-bottom: 15px; /* Space between image and heading */
54      transition: transform 0.3s;
```

Figure 24 css code for product page

Cake category

Each cake category has its own page. (e.g chocolate.html)

- Each product category page displays items in a grid format. Users can browse through different categories of cakes and pastries with product images, descriptions, and "Add to Cart" buttons. **(Add to cart button functionality not yet implemented)**
- Product Grid: The grid is built using Flexbox, allowing items to wrap and resize for various screen sizes. Each product card has hover effects, displaying product details with animations.
- Product Page Styling: All products are centered and consistently styled using Flexbox. Hovering over product images triggers a zoom effect for better user interaction.
- Each category page has a "back to all cake" button that points to the product.html page

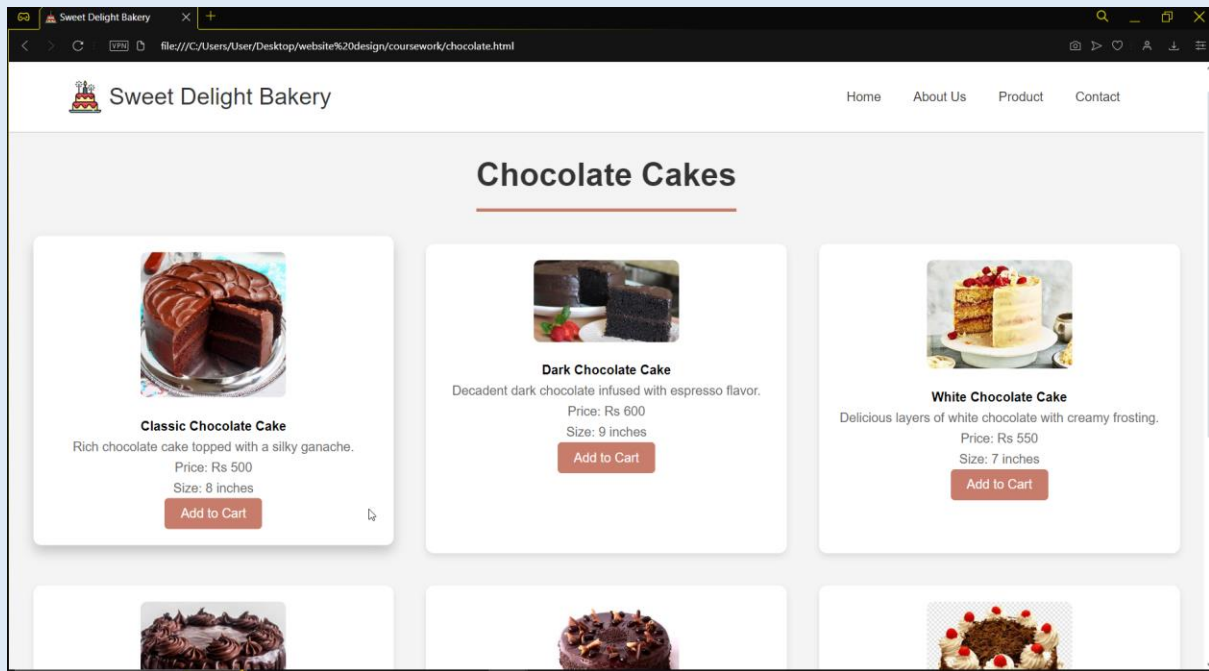


Figure 25 Chocolate.html page

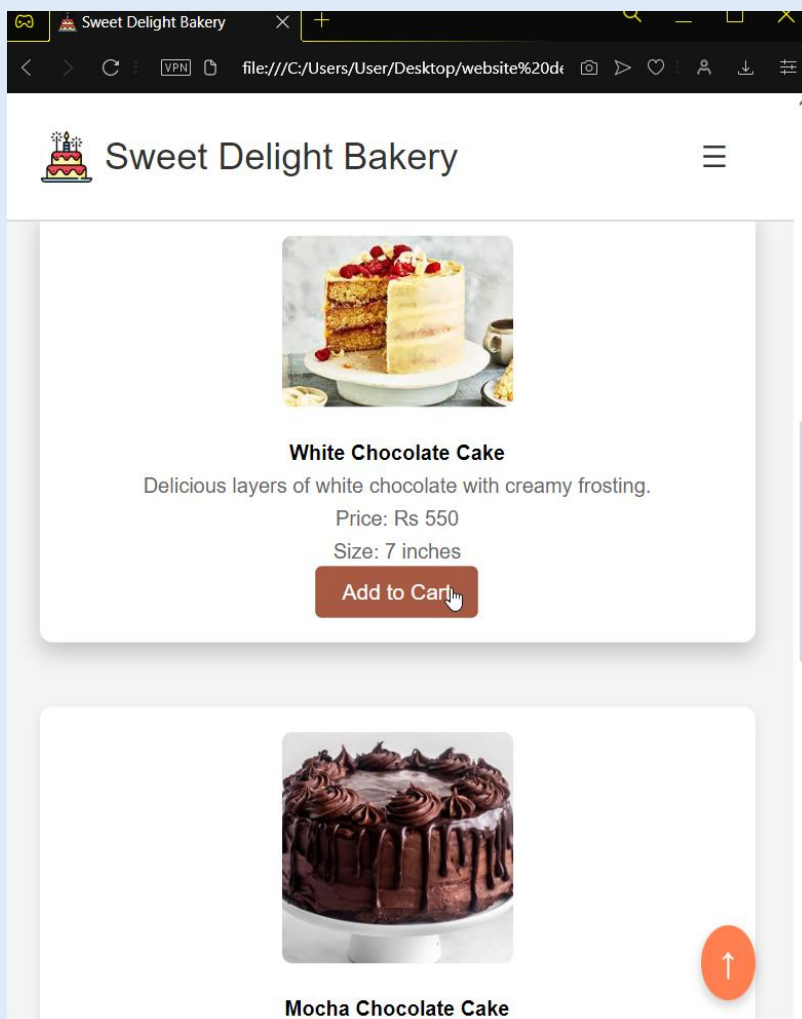


Figure 26 Grid layout adjusted for smaller screen

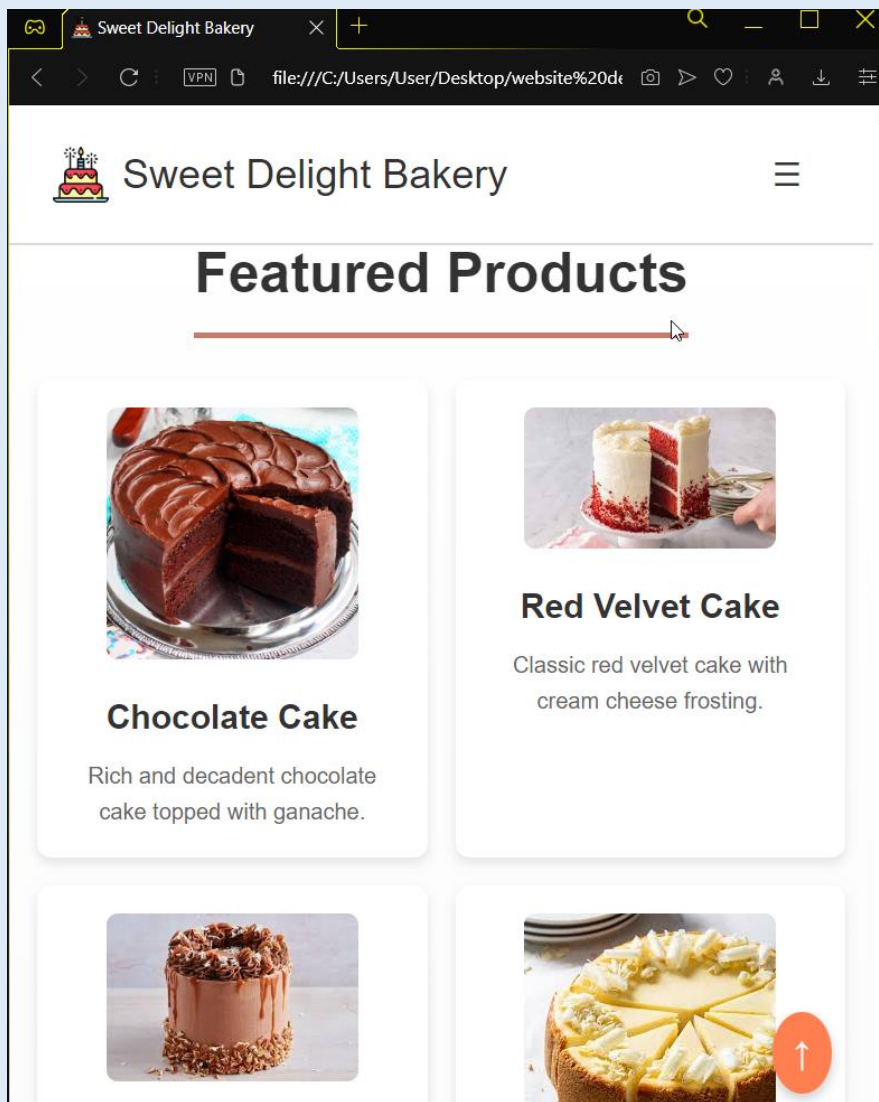


Figure 27 Grid layout adjusted for smaller screen

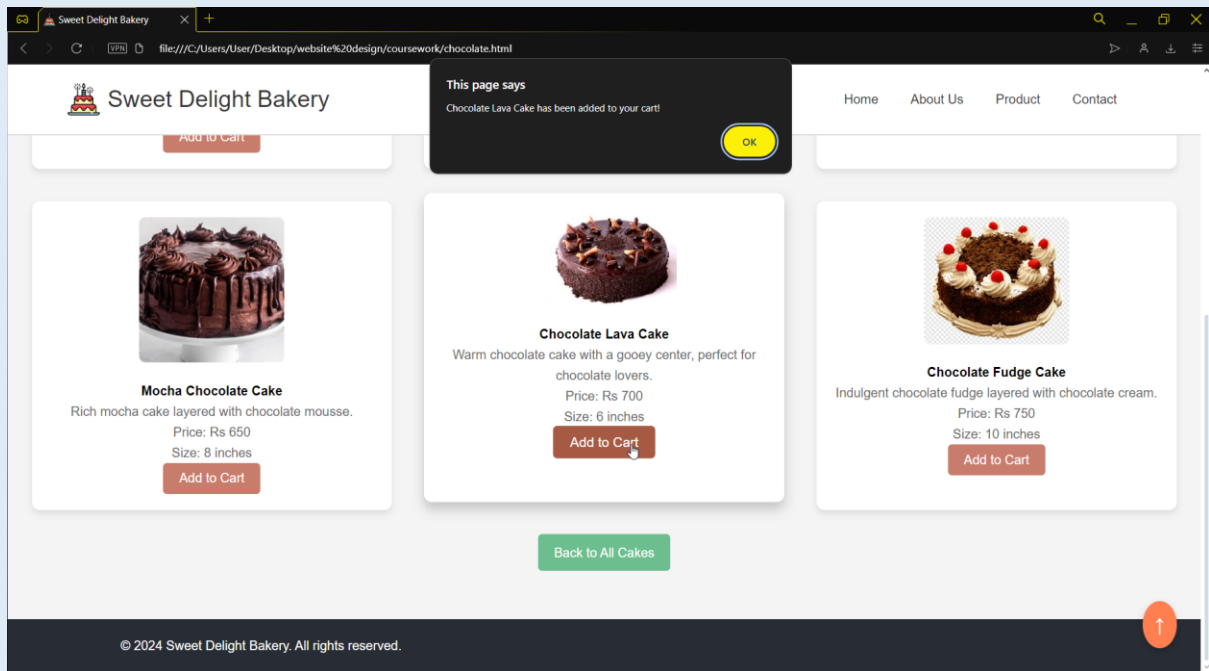


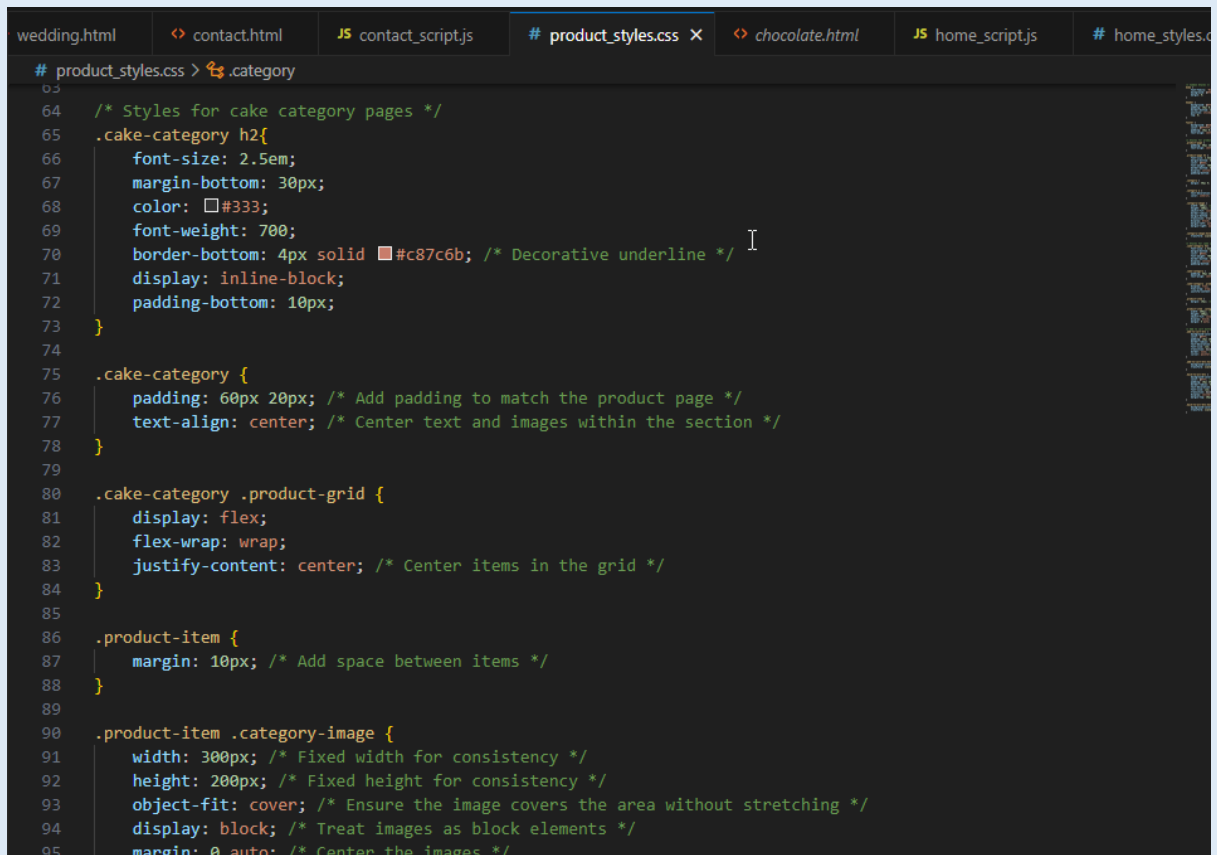
Figure 28 Message alert when add to cart button is triggered

```

home.html product.html wedding.html contact.html JS contact_script.js # product_styles.css chocolate.html x
chocolate.html > html > body > a#scrollToTopBtn.scroll-to-top
2 <html lang="en">
13 <body>
36 <section class="cake-category">
37
38 <h2>Chocolate Cakes</h2>
39 <div class="product-grid">
40 <div class="product-item">
41 
42 <h4>Classic Chocolate Cake</h4>
43 <p>Rich chocolate cake topped with a silky ganache.</p>
44 <p>Price: Rs 500</p>
45 <p>Size: 8 inches</p>
46 <button class="add-to-cart-btn" onclick="addToCart('Classic Chocolate Cake')>Add to Cart</button>
47 </div>
48 <div class="product-item">
49 
50 <h4>Dark Chocolate Cake</h4>
51 <p>Decadent dark chocolate infused with espresso flavor.</p>
52 <p>Price: Rs 600</p>
53 <p>Size: 9 inches</p>
54 <button class="add-to-cart-btn" onclick="addToCart('Dark Chocolate Cake')>Add to Cart</button>
55 </div>
56 <div class="product-item">
57 
58 <h4>White Chocolate Cake</h4>
59 <p>Delicious layers of white chocolate with creamy frosting.</p>
60 <p>Price: Rs 550</p>
61 <p>Size: 7 inches</p>
62 <button class="add-to-cart-btn" onclick="addToCart('White Chocolate Cake')>Add to Cart</button>
63 </div>
64 <div class="product-item">
65 

```

Figure 29 html code for chocolate.html page

A screenshot of a web development code editor with multiple tabs at the top: 'wedding.html', 'contact.html', 'JS contact_script.js', '# product_styles.css' (active), 'chocolate.html', 'JS home_script.js', and '# home_styles.d'. The main editor area shows CSS code for the 'category cake' page. The code includes comments in green and CSS rules in white. The rules define styles for the '.cake-category' section, including font size, margin, color, font weight, border, display, and padding. It also defines styles for the '.cake-category .product-grid' and '.product-item' elements, including display, flex-wrap, justify-content, margin, width, height, object-fit, and display. The code is as follows:

```
# product_styles.css > .category
64  /* Styles for cake category pages */
65  .cake-category h2{
66      font-size: 2.5em;
67      margin-bottom: 30px;
68      color: #333;
69      font-weight: 700;
70      border-bottom: 4px solid #c87c6b; /* Decorative underline */
71      display: inline-block;
72      padding-bottom: 10px;
73  }
74
75  .cake-category {
76      padding: 60px 20px; /* Add padding to match the product page */
77      text-align: center; /* Center text and images within the section */
78  }
79
80  .cake-category .product-grid {
81      display: flex;
82      flex-wrap: wrap;
83      justify-content: center; /* Center items in the grid */
84  }
85
86  .product-item {
87      margin: 10px; /* Add space between items */
88  }
89
90  .product-item .category-image {
91      width: 300px; /* Fixed width for consistency */
92      height: 200px; /* Fixed height for consistency */
93      object-fit: cover; /* Ensure the image covers the area without stretching */
94      display: block; /* Treat images as block elements */
95      margin: 0 auto; /* Center the images */
```

Figure 30 part of css code for category cake

```
> product.html < wedding.html < contact.html JS contact_script.js # product_styles.css X JS hor
# product_styles.css > .back-to-all-btn

98  /* Add to cart button styling */
99  .add-to-cart-btn {
100    background-color: #c87c6b;
101    color: #fff;
102    padding: 10px 20px;
103    border-radius: 5px;
104    text-decoration: none;
105    font-size: 1em;
106    transition: background-color 0.3s, transform 0.3s;
107    border: none;
108    cursor: pointer; /* Change cursor on hover */
109  }
110
111  .add-to-cart-btn:hover {
112    background-color: #a65a43;
113    transform: scale(1.05);
114  }
115
116  .back-to-all-btn {
117    background-color: #6dbe8f;
118    color: #fff;
119    padding: 10px 20px; /* Padding around the text */
120    border-radius: 5px;
121    text-decoration: none;
122    font-size: 1em;
123    transition: background-color 0.3s, transform 0.3s; /* Transition effects */
124    display: inline-block;
125    margin-top: 20px; /* Space above the button */
126  }
127
128  .back-to-all-btn:hover {
129    background-color: #a65a43; /* Darker shade on hover */
```

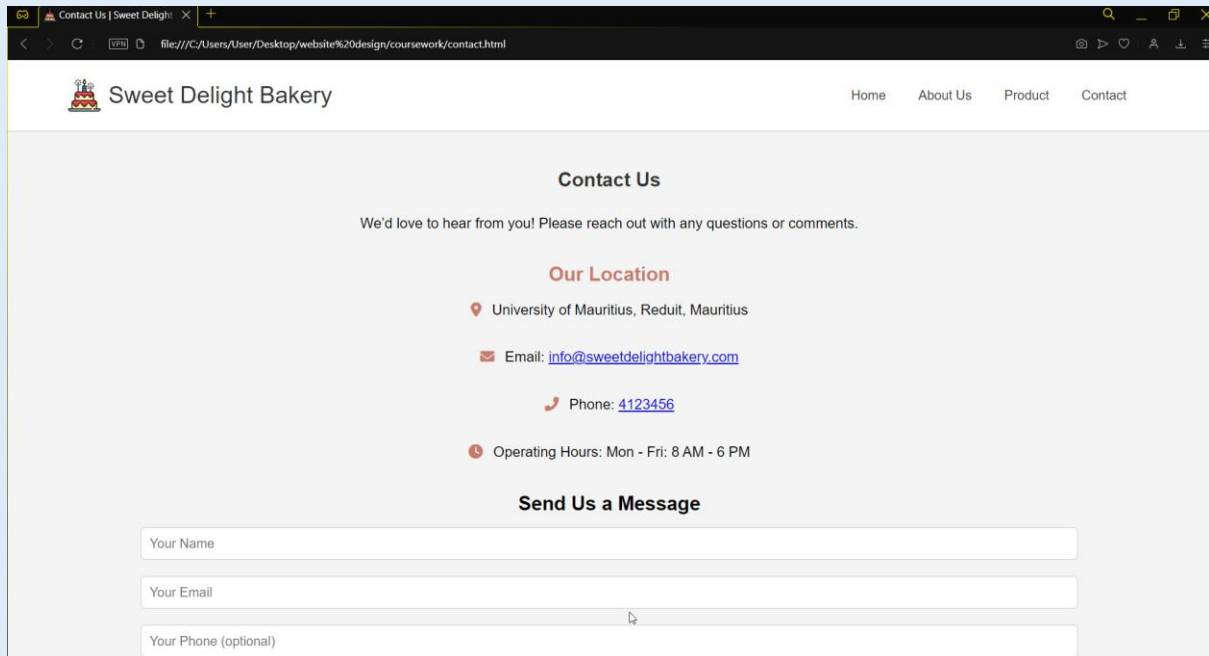
Figure 31 css code for button styling

```
< home.html < product.html < wedding.html < chocolate.html X < contact.html JS contact_script.js
< chocolate.html > html > body > a#scrollToTopBtn.scroll-to-top
2  <html lang="en">
13 <body>
36 <section class="cake-category">
89   <a href="product.html" class="back-to-all-btn">Back to All Cakes</a>
90
91 </section>
92
93 <footer>
94   <div class="footer-container">
95     <p>© 2024 Sweet Delight Bakery. All rights reserved.</p>
96   </div>
97 </footer>
98 <a href="#top" class="scroll-to-top" id="scrollToTopBtn">↑</a>
99 </body>
100
101 <script>
102 function addToCart(cakeName) {
103   alert(cakeName + " has been added to your cart!");
104 }
105 </script>
106
107 <script src="home_script.js"></script>
108 </html>
109
```

Figure 32 JavaScript for message alert

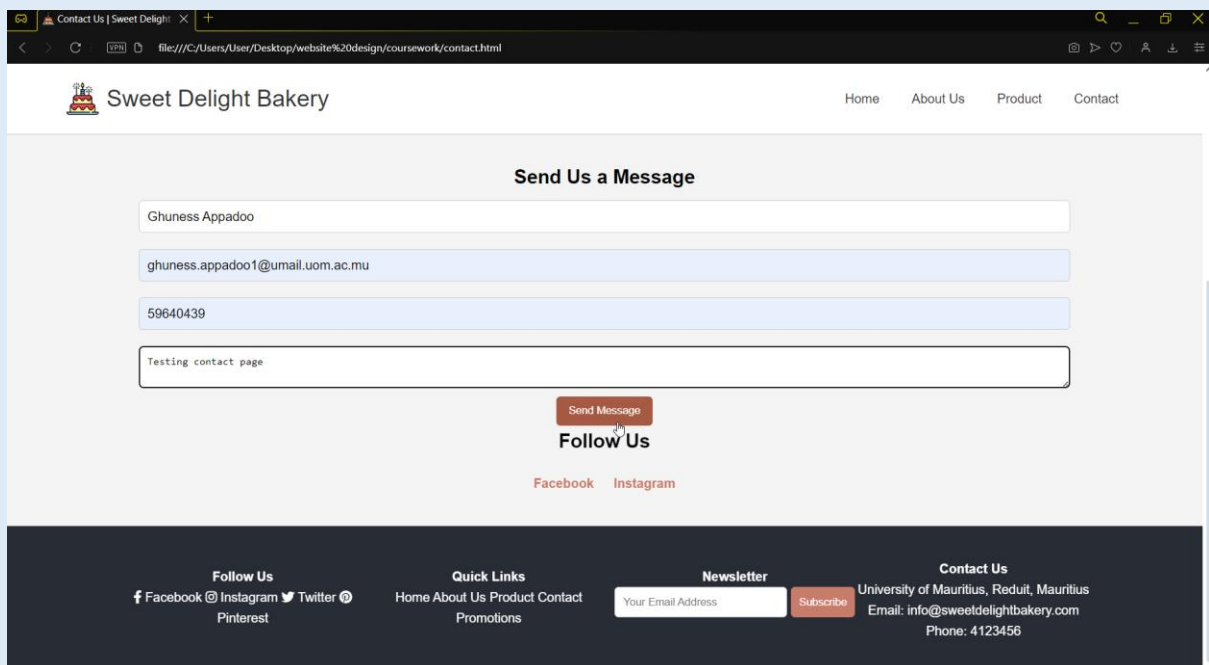
Contact page

- The contact page allows users to submit queries directly to the bakery via a contact form, which captures input data and uses EmailJS to send messages seamlessly without a backend server.
- Error Handling: If the email fails to send, the user is alerted with an error message, requesting them to try again.



The screenshot shows the 'Contact Us' page for Sweet Delight Bakery. The page has a navigation bar with links to Home, About Us, Product, and Contact. The main content area is titled 'Contact Us' and includes a welcome message: 'We'd love to hear from you! Please reach out with any questions or comments.' Below this is the 'Our Location' section, which lists the bakery's address (University of Mauritius, Reduit, Mauritius), email (info@sweetdelightbakery.com), phone (4123456), and operating hours (Mon - Fri: 8 AM - 6 PM). At the bottom of this section is a 'Send Us a Message' form with three input fields: 'Your Name', 'Your Email', and 'Your Phone (optional)'.

Figure 33 contact page



The screenshot shows the 'Send Us a Message' form filled out with test data. The form fields contain: 'Ghuness Appadoo' for the name, 'ghuness.appadoo1@uom.ac.mu' for the email, and '59640439' for the phone. Below the form is a 'Send Message' button. Below the button is a 'Follow Us' section with links to Facebook and Instagram. At the bottom of the page is a dark footer with four columns: 'Follow Us' (with social media icons), 'Quick Links' (with links to Home, About Us, Product, Contact, and Promotions), 'Newsletter' (with a 'Your Email Address' input field and a 'Subscribe' button), and 'Contact Us' (with the bakery's address, email, and phone number).

Figure 34 Testing contact page

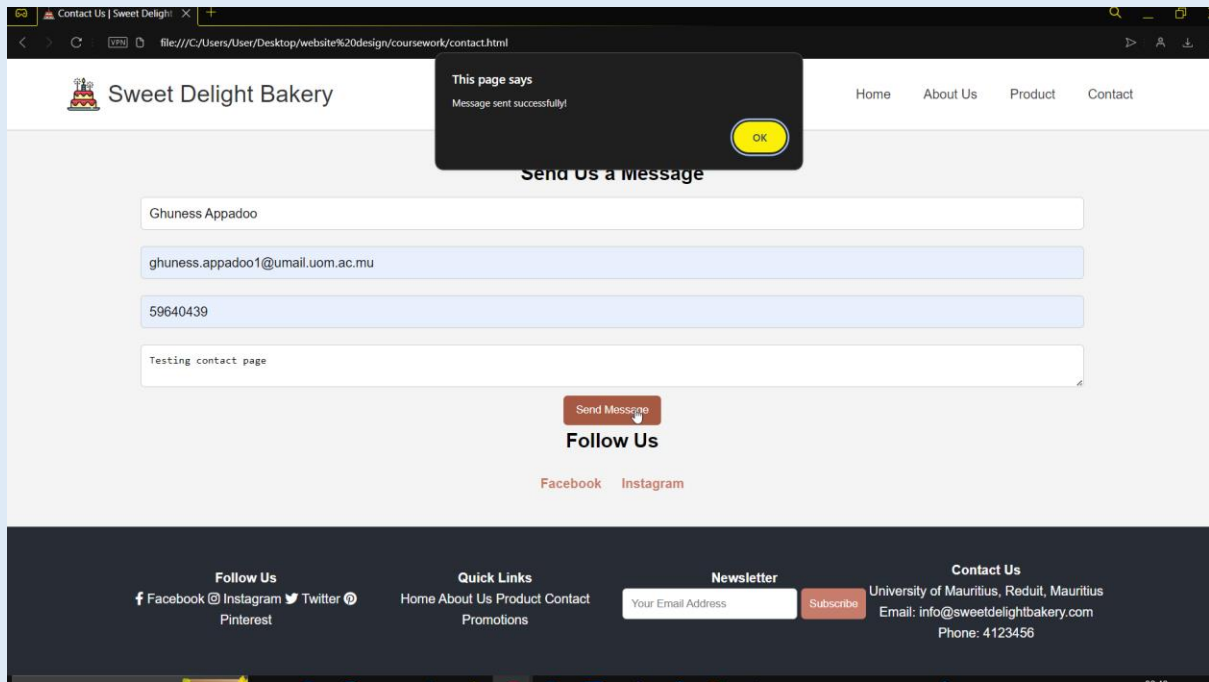


Figure 35 Message sent successfully

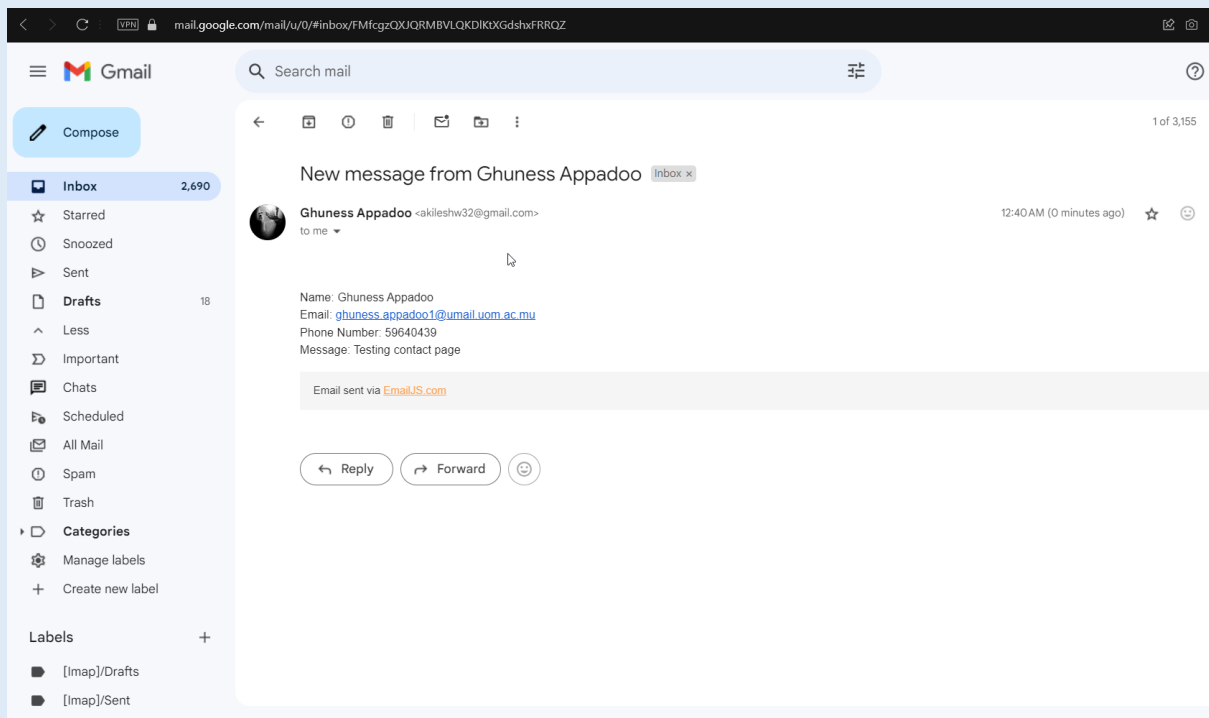


Figure 36 Message received on the email address set as administrator of the webpage through emailjs

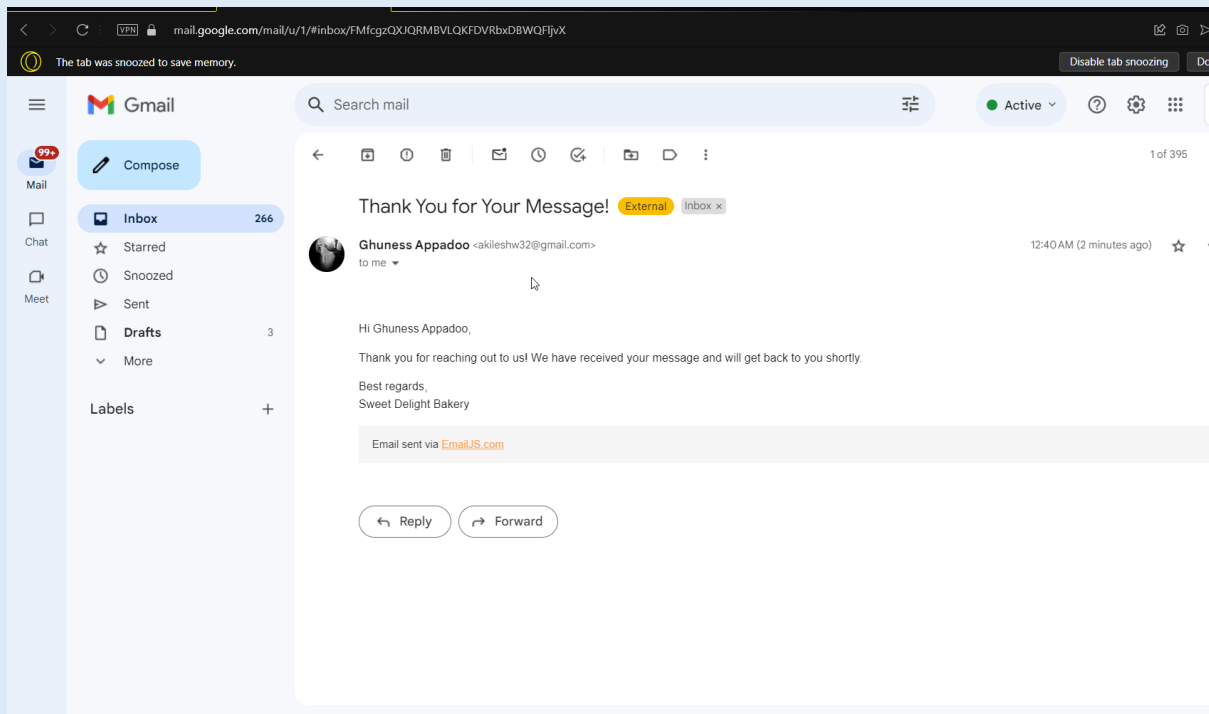


Figure 37 auto reply email sent to acknowledge the message

```

home.html  product.html  wedding.html  chocolate.html  contact.html X  JS contact_script.js  # product_styles.css
contact.html > html > body > header > div.container > nav.navbar > ul#navbar-links
  2  <html lang="en">
 14  <body>
 15    <header>
 34    </div>
 35    </header>
 36
 37    <section id="contact">
 38      <h1>Contact Us</h1>
 39      <p>We'd love to hear from you! Please reach out with any questions or comments.</p>
 40
 41      <div class="contact-info">
 42        <h2>Our Location</h2>
 43        <p><i class="fas fa-map-marker-alt"></i> University of Mauritius, Reduit, Mauritius</p>
 44        <p><i class="fas fa-envelope"></i> Email: <a href="mailto:akileshw32@gmail.com">info@sweetdelightbak</a>
 45        <p><i class="fas fa-phone-alt"></i> Phone: <a href="tel:+4123456">4123456</a></p>
 46        <p><i class="fas fa-clock"></i> Operating Hours: Mon - Fri: 8 AM - 6 PM</p>
 47      </div>
 48
 49      <h2>Send Us a Message</h2>
 50      <form id="contact-form">
 51        <input type="text" id="name" placeholder="Your Name" required>
 52        <input type="email" id="email" placeholder="Your Email" required>
 53        <input type="tel" id="phonenumber" placeholder="Your Phone (optional)">
 54        <textarea id="message" placeholder="Your Message" required></textarea>
 55        <button type="submit">Send Message</button>
 56      </form>
 57
 58
 59      <h2>Follow Us</h2>
 60      <div class="social-media">
 61        <a href="https://www.facebook.com" target="_blank">Facebook</a>
 62        <a href="https://www.instagram.com" target="_blank">Instagram</a>
 63      </div>
 64    </section>
 65  </body>
 66 </html>

```

Figure 38 html code for contact page

```

home.html  product.html  wedding.html  chocolate.html  contact.html X  JS contact_script.js  # product
contact.html > html > body > section#contact > form#contact-form > textarea#message
2  <html lang="en">
14 <body>
66   <footer>
67     <div class="footer-container">
87       <div class="footer-section footer-newsletter">
92         </div>
93       </div>
94       <div class="footer-info">
95         <h4>Contact Us</h4>
96         <p>University of Mauritius, Reduit, Mauritius</p>
97         <p>Email: info@sweetdelightbakery.com</p>
98         <p>Phone: 4123456</p>
99       </div>
100     </div>
101   </footer>
102
103   <script src="home_script.js"></script>
104   <script src="contact_script.js"></script>
105   <script type="text/javascript"
106     src="https://cdn.jsdelivr.net/npm/emailjs/browser@4/dist/email.min.js">
107 </script>
108   <script type="text/javascript">
109   (function(){
110     emailjs.init({
111       publicKey: "yFNJZrWwHD5f5NPjc",
112     });
113   })();
114 </script>
115 </body>
116 </html>
117

```

Figure 39 script to integrates EmailJs and to initialize the EmailJs service using the public key

```

home.html  product.html  wedding.html  chocolate.html  contact.html  JS contact_script.js X  # product_styles.css  JS  ..
JS contact_script.js > document.addEventListener('DOMContentLoaded') callback > contactForm.addEventListener('submit') callback
1  document.addEventListener('DOMContentLoaded', () => {
2    const contactForm = document.getElementById('contact-form');
3
4    if (contactForm) {
5      contactForm.addEventListener('submit', function (event) {
6        event.preventDefault(); // Prevent default form submission
7
8        // Get input values
9        const name = contactForm.elements[0].value;
10       const email = contactForm.elements[1].value;
11       const phonenumber = contactForm.elements[2].value;
12       const message = contactForm.elements[3].value;
13
14       // Send the email using EmailJS
15       emailjs.send("service_qsqrt6d", "template_h8b2hra", {
16         name: name,
17         email: email,
18         phonenumber: phonenumber,
19         message: message
20       });
21       .then(function(response) {
22         console.log('SUCCESS!', response.status, response.text);
23         alert('Message sent successfully!');
24         contactForm.reset(); // Reset the form
25       }, function(error) {
26         console.log('FAILED...', error);
27         alert('Error sending message. Please try again later.');

```

Figure 40 contact_script

This JavaScript code handles the functionality of the **Contact Form** on the website, allowing users to submit messages. Here is an explanation of its main components:

1. **DOMContentLoaded Event:**

- The code is wrapped inside a DOMContentLoaded event listener to ensure that the form is fully loaded before any JavaScript interactions occur. This prevents errors if the form elements are not yet available.

2. **Form Submission Handling:**

- The code targets the form with the ID contact-form and adds a submit event listener to it.
- **Prevent Default Submission:** When the form is submitted, the default form submission (which reloads the page) is prevented using event.preventDefault().

3. **Input Values:**

- The script retrieves values from the input fields: name, email, phone number (optional), and message. These values are extracted from the form fields using the elements property and stored in variables.

4. **Sending Emails with EmailJS:**

- **EmailJS** is used to send the form data as an email. The service and template IDs (service_qsqr6d and template_h8b2hra) are passed to the emailjs.send() function along with the form data. (public key, service and template IDs are unique for each user when sign up with EmailJS)
- The email service sends the form content to a predefined email address without requiring a backend server.

5. **Success & Error Handling:**

- Upon successful submission, the user is alerted with a **"Message sent successfully!"** message, and the form is reset for future submissions.
- If the email fails to send, an error message is logged, and the user is alerted with a message saying **"Error sending message. Please try again later."**

This code ensures seamless interaction by allowing users to send messages directly through the website and receive immediate feedback, improving user experience.


```
home.html product.html wedding.html contact.html JS contact_script.js # contact_styles.css # product_styles.css
# contact_styles.css > #contact h1
1 #contact {
2   padding: 40px 20px;
3   background: #f4f4f4;
4   text-align: center;
5 }
6
7 #contact h1 {
8   color: #333;
9   margin-bottom: 20px;
10  font-family: 'Arial', sans-serif;
11 }
12
13 #contact p {
14   font-size: 1.1em;
15   margin-bottom: 30px;
16 }
17
18 .contact-info {
19   margin-bottom: 30px;
20 }
21
22 .contact-info h2 {
23   color: #c87c6b;
24   margin-bottom: 10px;
25 }
26
27 .contact-info p {
28   margin: 5px 0;
29   font-size: 1em;
30 }
31
32 .contact-info i {
```

Figure 41 part of css code for contact page

dashboard.emailjs.com/admin/templates/vgm63ck

EmailJS 169 requests left Welcome, Ghuness Appadoo Docs Support Sign Out

Email Services

Email Templates

Email History

Suppressions beta

Contacts

Events

Statistics

Team Members

Account

Requests received 31 / 200

Resets on Oct 20

Increase request limit

Template 1

Content Auto-Reply Attachments Contacts Settings

Subject *

New message from {{name}}

Content *

Desktop Mobile Edit Content

Name: {{name}}
Email: {{email}}
Phone Number: {{phonenumber}}
Message: {{message}}

To Email *

akileshw32@gmail.com

From Name

Ghuness Appadoo

From Email *

☒ Use Default Email Address

Reply To

{{email}}

Bcc

Cc

Figure 42 EmailJS template1

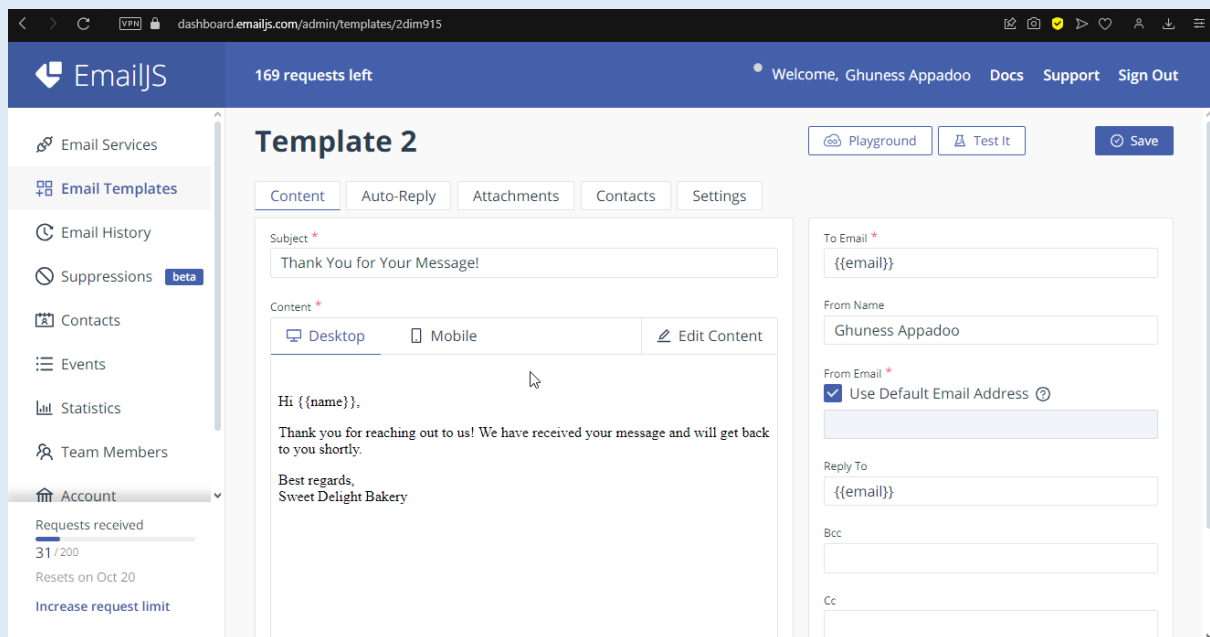


Figure 43 EmailJS template2 for autoreply

Responsive Design

1) Mobile Responsiveness

- **Hamburger Menu:** On small screens, the navigation menu is replaced with a hamburger icon that toggles the visibility of the menu items when clicked.
- **Product Grid:** The layout of product cards adjusts to smaller screens, showing fewer products per row for better readability.
- **Flexible Images:** All images, including product images and carousel images, resize according to the screen size to prevent distortion. The use of `object-fit: cover` ensures that images maintain their aspect ratio.

2) Media Queries

- Media queries ensure the site looks and functions well on different screen sizes (desktop, tablet, mobile). The grid layout adjusts to show different numbers of products based on the screen width.

User Interactivity

1. Add to Cart Button

- **Interactive Buttons:**
 - Each product card contains an "Add to Cart" button. This button changes color and slightly enlarges when hovered to signal interactivity.
 - **User Experience:** The button uses smooth CSS transitions to enhance the experience and make interactions more engaging.

2. Back to All Products Button

- **Return Navigation:**
 - On product detail pages, there is a "Back to All Products" button that allows users to return to the main product listing.
 - **Hover Effects:** The button enlarges and changes color when hovered over, using the same transition properties as the "Add to Cart" button for consistency.

3. Scroll-to-Top Button

- **Scroll Detection:** A scroll-to-top button appears when the user scrolls more than 300 pixels down the page. This button provides a quick way to return to the top of the page with a smooth scrolling effect when clicked.

4. Hover Effects

- **Images:** Product images and category images have a slight zoom-in effect when hovered over, enhancing interactivity.
- **Buttons:** Both the "Add to Cart" and "Back to All Products" buttons change color and size when hovered to signal that they are clickable.

Technical Overview

- I. **Technologies Used**
- II. **HTML5:** Used for structuring the content in a semantic and accessible way, ensuring compatibility with modern web standards.
- III. **CSS3:** Utilized for styling the layout and ensuring a responsive design across devices using Flexbox and media queries.
- IV. **JavaScript:** Provides interactivity for various UI components such as the contact form, carousel, and scroll-to-top button.
- V. **EmailJS:** Allows form submissions to be sent via email without the need for a backend server, simplifying the process of user communication.
- VI. **Font Awesome:** Used to add scalable vector icons for social media links and other navigation components.

Third-Party Integrations

- **EmailJS:** Used to handle email submissions from the contact form.
- **Font Awesome:** Adds icons for social media and navigation, enhancing the UI.

Future Improvements

Future improvements could include adding functionality to the 'Add to Cart' button, allowing users to fully engage with the e-commerce capabilities of the site.

Conclusion

In conclusion, the Sweet Delight Bakery website offers a user-friendly, responsive design with functional components such as a contact form, interactive product displays, and smooth navigation across devices. By leveraging modern web technologies like EmailJS, JavaScript, and CSS3, the site delivers an engaging and efficient user experience.