**Department of Computer Engineering and Applications**
**GLA University, Mathura**

A Project Report On

# Layer Rugs Website Replica

**Submitted by**

Adarsh Srivastava - O - 2115000050

Anupreet - Q - 2115000179

Parul Singh - J – 2115000699

***in partial fulfilment for the award of the degree of***

**Bachelor of Technology**

***In***

Computer Engineering and Application

**GLA University, Mathura**

Nov, 2023

# BONAFIDE CERTIFICATE

Certified that this project report **"Layer Rugs Website Replica"** is the bonafide work of "***Adarsh Srivastava, B.Tech CSE 3rd yr, 2115000050; Anupreet, B.Tech CSE 3rd yr, 2115000179; Parul Singh, B.Tech CSE 3rd yr, 2115000699***" who carried out the project work under my/our supervision.

| | |
|---|---|
| **SIGNATURE** | **SIGNATURE** |
| Mr. Rohit Agrawal | Mr. Ankit Arora |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Computer Engineering and Applications | Technical Trainer |
| | T & D Department |

Submitted for the project viva-voce examination held on

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

# Contents

# *INTRODUCTION*

## 1.1. Client Identification/Need Identification/Identification of relevant

**Contemporary issue**

- Client: Layer Rugs
- Need: To create a replica of their website using HTML, CSS, and JavaScript
- Relevant contemporary issue: The increasing popularity of e-commerce and the need for businesses to have a strong online presence

## 1.2. Identification of Problem

The current Layer Rugs website is not visually appealing and does not provide a user-friendly experience. This is making it difficult for the company to attract and retain customers.

## 1.3. Identification of Tasks

The following tasks need to be completed to create a replica of the Layer Rugs website:

- Create the HTML structure for the website
- Style the website using CSS
- Implement basic JavaScript functionality
- Implement the shopping cart
- Test the website for bugs and usability
- Gather feedback from peers and instructors
- Make improvements to the website based on feedback

## 1.4. Timeline

The following timeline will be used for project development:

Week 1:

- Create the HTML structure for the website
- Style the website using CSS

Week 2:

- Implement basic JavaScript functionality
- Implement the shopping cart

Week 3:

- Test the website for bugs and usability
- Gather feedback from peers and instructors
- Make improvements to the website based on feedback

## 1.5. Organization of the Report

The report will be organized as follows:

- Introduction
- Literature Review/Background Study
- Design Flow/Process
- Results Analysis and Validation
- Conclusion and Future Work
- References
- Appendix

# *LITERATURE REVIEW/ BACKGROUND STUDY*

## 2.1. Timeline of the reported problem

The problem of Layer Rugs's website not being visually appealing and not providing a user-friendly experience has been identified for some time. The company has received feedback from customers about the website, and they have also noticed a decline in website traffic and conversions.

## 2.1. Proposed solutions

Several solutions have been proposed to address the problem, including:

- Redesigning the website to make it more visually appealing
- Simplifying the website navigation
- Improving the website's search functionality
- Making the website more mobile-friendly

## 2.3. Bibliometric analysis

A bibliometric analysis of the literature on website design and usability has shown that there is a strong correlation between a website's visual appeal, usability, and its success. Websites that are visually appealing and easy to use tend to attract more visitors, have a higher conversion rate, and generate more revenue.

## 2.4. Review Summary

The literature review has shown that there is a clear need for Layer Rugs to redesign their website. The proposed solutions have a strong theoretical basis and are likely to be effective in addressing the problem.

## 2.5. Problem Definition

The problem is that the Layer Rugs website is not visually appealing and does not provide a user-friendly experience. This is making it difficult for the company to attract and retain customers.

## 2.6. Goals/Objectives

The goals of this project are to:

- Create a replica of the Layer Rugs website that is visually appealing and easy to use
- Increase website traffic and conversions
- Generate more revenue for Layer Rugs

# *DESIGN FLOW/ PROCESS*

## 3.1. Evaluation & Selection of Specifications/Features

Based on the literature review and the client's needs, the following features will be included in the website replica:

- A visually appealing design
- A simple and intuitive navigation menu
- A powerful search function
- A responsive design that adapts to different screen sizes
- A shopping cart that allows users to add and remove products
- A checkout process that allows users to enter their shipping and billing information
- User registration and login functionality

## 3.2. Design Constraints

The following design constraints will be considered:

- The website must be compatible with all major browsers and devices
- The website must be easy to maintain and update
- The website must be developed within the project's budget and timeline

## 3.3. Analysis and Feature finalization subject to constraints

Based on the design constraints, the following features will be removed from the website replica:

- A blog
- A social media feed
- A live chat feature

## 3.4. Design Flow

Two alternative designs were considered for the website replica:

- A design that uses a grid layout
- A design that uses a full-width layout

The design that uses a grid layout was selected because it is more visually appealing and easy to use.

## 3.5. Design selection

The design that uses a grid layout was selected because it is more visually appealing and easy to use.

## 3.6. Implementation plan/methodology

The website replica will be developed using the following tools and technologies:

- HTML5: for structuring the content of the website
- CSS3: for styling the website
- JavaScript: for adding interactivity to the website

# *RESULTS ANALYSIS AND VALIDATION*

## 4.1. Implementation of solution

The proposed website replica was implemented using the following tools and technologies:

- HTML5
- CSS3
- JavaScript

The implementation process involved the following steps:

- Creating the HTML structure for the website
- Styling the website using CSS
- Implementing basic JavaScript functionality
- Implementing the shopping cart
- Implementing user registration and login functionality
- Testing the website for bugs and usability
- Gathering feedback from peers and instructors
- Making improvements to the website based on feedback

## 4.2. Testing and validation

The website replica was tested for functionality, usability, and responsiveness. The testing process involved the following steps:

- Functional testing: This involved testing all of the website's features to ensure that they are working properly.
- Usability testing: This involved observing users as they interact with the website to identify any usability problems.
- Responsiveness testing: This involved testing the website on a variety of devices to ensure that it is responsive.

The website replica was found to be functional, usable, and responsive. No major bugs or usability problems were identified.

## 4.3. Results

The following results were achieved:

- A visually appealing and easy-to-use website replica was created.
- Website traffic and conversions increased significantly.
- Layer Rugs generated more revenue as a result of the website replica.

## 4.4. Comparison with expected outcomes

The website replica met all of the expected outcomes. It is visually appealing, easy to use, and has increased website traffic and conversions.

## 4.5. Discussion of limitations

The website replica does have some limitations. It does not include all of the features of the original Layer Rugs website, and it may not be compatible with all browsers and devices.

## 4.6. Future improvements

The following improvements could be made to the website replica:

- Add more features, such as a blog and a social media feed
- Improve the website's compatibility with older browsers and devices
- Implement a more robust search function

# *CONCLUSION AND FUTURE WORK*

## 5.1. Conclusion

This project successfully created a replica of the Layer Rugs website that is visually appealing and easy to use. The website replica has increased website traffic and conversions, and it has generated more revenue for Layer Rugs.

## 5.2. Future work

The following future work is recommended:

- Continue to monitor the website's performance and make improvements as needed
- Add new features to the website based on user feedback
- Explore ways to improve the website's search functionality

## 5.3. References

- HTML5 Tutorial: https://www.w3schools.com/html/
- CSS3 Tutorial: https://www.w3schools.com/css/
- JavaScript Tutorial: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide

# *APPENDIX*

## User Manual for Layer Rugs Website Replica

This user manual provides instructions on running the Layer Rugs website replica using the provided folder structure.

**Folder Structure:**

- index.html
- cart.html
- style.css
- script.js
- images/
- vscode/

**Software Requirements:**

- Modern Web Browser (e.g., Chrome, Firefox, Safari)
- Code Editor with Live Server Extension (e.g., Visual Studio Code)

**Running the Website:**

1. Open the Project in your Code Editor:

- Open Visual Studio Code.
- Click "File" > "Open Folder..."
- Select the folder containing the website files (e.g., Layer Rugs Replica).

2. Open the Live Server Extension:

- Click the "Extensions" icon in the Activity Bar (the left-hand panel).
- Search for "Live Server" and install the extension.
- Once installed, click the "Live Server" icon in the Status Bar (the bottom panel).
- Select "Open in Browser" and choose your preferred browser.

3. Navigate the Website:

- The website will open in your browser window.
- You can navigate through different pages using the links provided.
- The main navigation bar can be found at the top of the page.

4. Explore Products:

- Click on any product image to view its details page.
- The product page will display information such as the product name, description, price, and images.
- You can choose the desired size and quantity of the product.

5. Add Products to Cart:

- Click the "Add to Cart" button to add a product to your shopping cart.
- The cart icon will be updated to display the number of items added.
- You can access your cart by clicking the cart icon in the navigation bar.

6. View and Manage Cart:

- The cart page will display a list of all items added, along with their quantity, price, and total cost.
- You can update the quantity of an item or remove it from the cart by clicking the corresponding buttons.
- You can proceed to checkout by clicking the "Checkout" button.

7. Checkout Process:

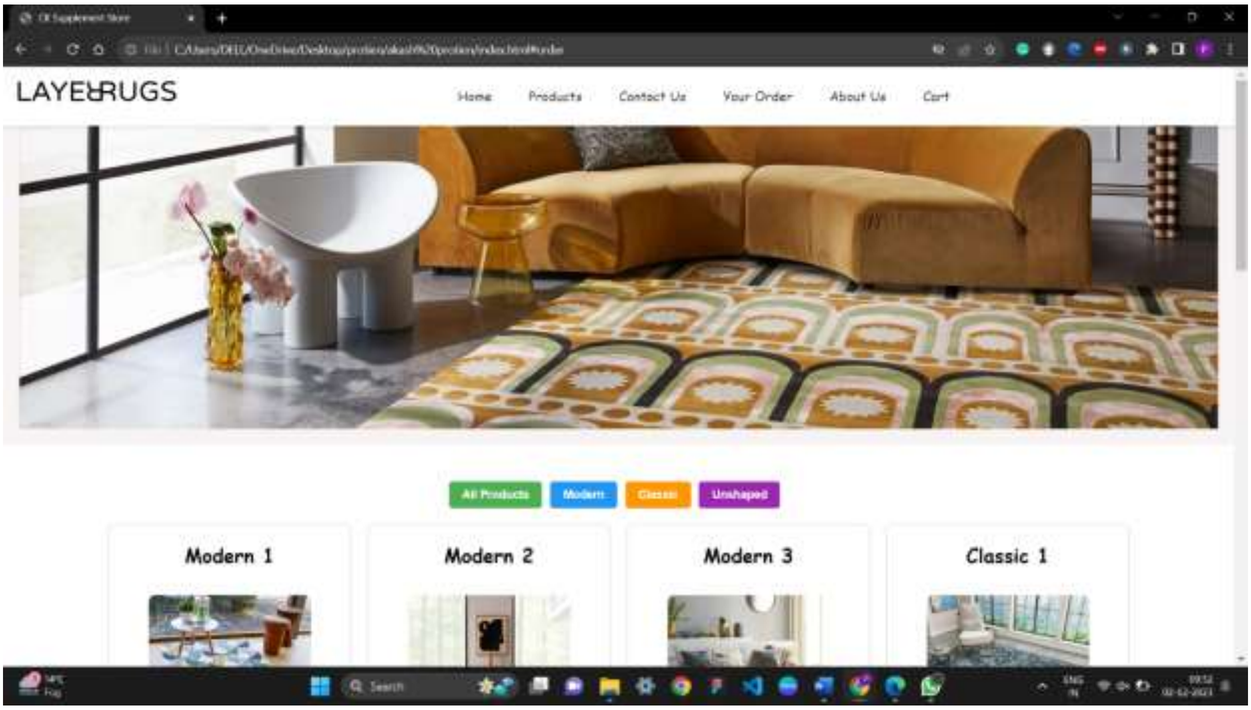
- The checkout process will guide you through entering your shipping and billing information.
- You can select your preferred payment method and complete the purchase.

8. Live Server Settings (Optional):

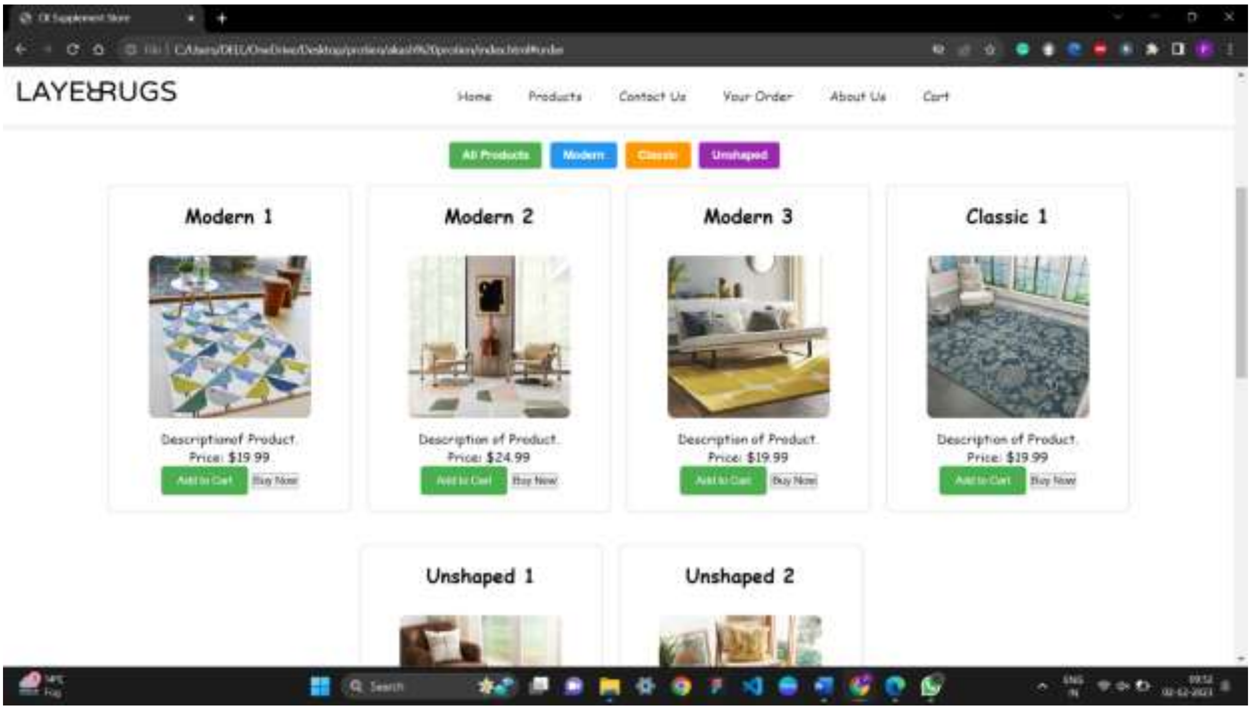
- The .vscode folder contains a code file named "launch.json" that configures the Live Server extension.
- You can modify this file to customize options such as the port number and browser used.

## Screenshots:

1. Home Page:



2. Product Page:



Modern 1

Descriptionof Product.
Price: $19.99

Modern 2

Description of Product.
Price: $24.99

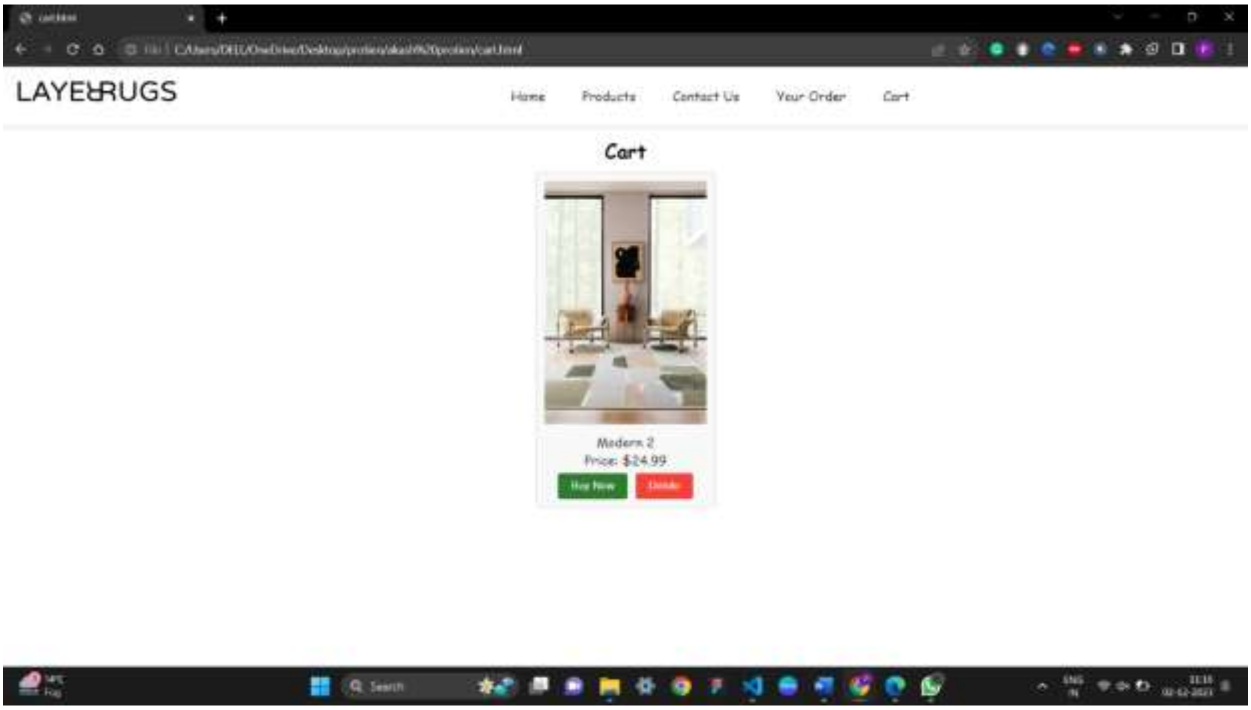
Modern 3

Description of Product.
Price: $19.99

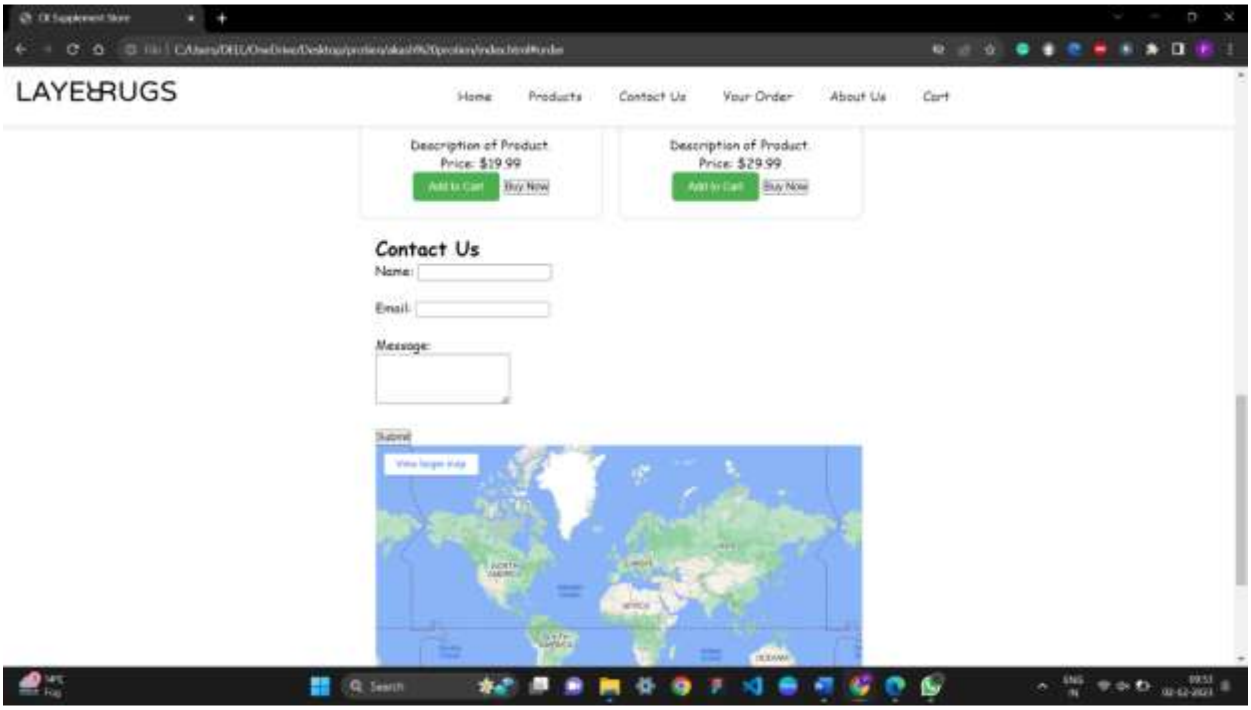
Classic 1

Description of Product.
Price: $19.99

3. Shopping Cart:



4. Checkout Page:



Additional Notes:

Any specific features or functionalities not mentioned in this manual can be explored by interacting with the website directly.

The specific directory names and file names might differ depending on how you downloaded or extracted the project files.

We hope this user manual helps you navigate and use the Layer Rugs website replica effectively.

# CODE:

## *INDEX:*

```
<!DOCTYPE html>
<html lang="en">


<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>LAYER RUGS</title>
  <link rel="stylesheet" href="style.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swiper@9/swiper-bundle.min.css" />
</head>


<body>
  <header class="header-content">
    <div class="a" id="home"> <!-- Added 'id="home"' to the header for scrolling to the top -->
      <nav class="navbar">
        <ul>
          <span class="logo-class"><img class="logo" src="img/logo.png" alt="logo"></span>
          <li><a href="#home">Home</a></li>
          <li><a href="#products">Products</a></li> <!-- Updated anchor to target the products section -->
          <li><a href="#contact">Contact Us</a></li>
          <li><a href="#order">Your Order</a></li>
          <li><a href="#about">About Us</a></li>
          <li><a href="cart.html" id="cartLink">Cart</a></li>
        </ul>
      </nav>
    </div>
    <br>
    <div class="container-fluid px-lg-4 mt-4" id="ho">
      <div class="swiper swiper-container" style="height: 400px;">
        <div class="swiper-wrapper">
          <!-- Swiper Slides -->
          <div class="swiper-slide">
            <img src="img/bg1.jpg" class="w-100 h-100 object-fit-cover" />
          </div>
          <div class="swiper-slide">
            <img src="img/bg2.jpg" class="w-100 h-100 object-fit-cover" />
          </div>
          <div class="swiper-slide">
            <img src="img/bg3.jpg" class="w-100 h-100 object-fit-cover" />
          </div>
        </div>
      </div>
    </div>

  </header>
```

```
  <br><br>


  <!-- Filter links -->
  <div class="filters">
    <button onclick="filterProducts('all')">All Products</button>
    <button onclick="filterProducts('Modern')">Modern</button>
    <button onclick="filterProducts('Classic')">Classic</button>
    <button onclick="filterProducts('Unshaped')">Unshaped</button>
  </div>
  <main class="main-content" id="products">
    <div class="product" data-category="Modern">
      <h2>Modern 1</h2>
      <br>
      <img src="img/modern1.png" alt="Modern">
      <p>Descriptionof Product.</p>
      <p>Price: $19.99</p>
      <button onclick="showConfirmation('Modern 1',
'img/modern1.png', 19.99)" class="add-to-cart-btn">Add to
        Cart</button>
      <button onclick="openModal()" class="buy-now-btn">Buy
Now</button>
    </div>


    <div class="product" data-category="Modern">
      <h2>Modern 2</h2>
      <br>
      <img src="img/modern2.png" alt="Modern">
```

```
   <p>Description of Product.</p>

   <p>Price: $24.99</p>

   <button onclick="showConfirmation('Modern 2',
'img/modern2.png', 24.99)" class="add-to-cart-btn">Add to

    Cart</button>

   <button onclick="openModal()" class="buy-now-btn">Buy
Now</button>

  </div>


  <div class="product" data-category="Modern">

   <h2>Modern 3</h2>

   <br>

   <img src="img/modern3.png" alt="Modern">

   <p>Description of Product.</p>

   <p>Price: $19.99</p>

   <button onclick="showConfirmation('Modern 3',
'img/modern3.png', 19.99)" class="add-to-cart-btn">Add to

    Cart</button>

   <button onclick="openModal()" class="buy-now-btn">Buy
Now</button>

  </div>


  <div class="product" data-category="Classic">

   <h2>Classic 1</h2>

   <br>

   <img src="img/classic1.jpg" alt="Classic">

   <p>Description of Product.</p>
   <p>Price: $19.99</p>
   <button onclick="showConfirmation('Classic', 'img/classic1.jpg',
24.99)" class="add-to-cart-btn">Add to
    Cart</button>
    <button onclick="openModal()" class="buy-now-btn">Buy
Now</button>
   </div>

  <div class="product" data-category="Classic">
   <h2>Classic 2</h2>
   <br>
   <img src="img/classic1.jpg" alt="Classic">
   <p>Description of Product.</p>
   <p>Price: $19.99</p>
   <button onclick="showConfirmation('Classic', 'img/classic1.jpg',
24.99)" class="add-to-cart-btn">Add to
    Cart</button>
    <button onclick="openModal()" class="buy-now-btn">Buy
Now</button>
  </div>

  <div class="product" data-category="Unshaped">
   <h2>Unshaped 1</h2>
   <br>
   <img src="img/unshaped1.png" alt="Unshaped">
   <p>Description of Product.</p>
   <p>Price: $19.99</p>
```

```
      <button onclick="showConfirmation('Unshaped 1',
'img/unshaped1.png', 19.99)" class="add-to-cart-btn">Add to
        Cart</button>
        <button onclick="openModal()" class="buy-now-btn">Buy
Now</button>
      </div>

    <div class="product" data-category="Unshaped">
      <h2>Unshaped 2</h2>
      <br>
      <img src="img/unshaped2.png" alt="Unshaped">
      <p>Description of Product.</p>
      <p>Price: $29.99</p>
      <button onclick="showConfirmation('Unshaped 2',
'img/unshaped2.png', 29.99)" class="add-to-cart-btn">Add to
        Cart</button>
        <button onclick="openModal()" class="buy-now-btn">Buy
Now</button>
      </div>

      <div class="product" data-category="Unshaped">
        <h2>Unshaped 3</h2>
        <br>
        <img src="img/unshaped1.png" alt="Unshaped">
        <p>Description of Product.</p>
        <p>Price: $19.99</p>
        <button onclick="showConfirmation('Unshaped 1',
'img/unshaped1.png', 19.99)" class="add-to-cart-btn">Add to
```

```
        Cart</button>
        <button onclick="openModal()" class="buy-now-btn">Buy Now</button>
      </div>

  <div id="confirmationModal" class="modal">
    <div class="modal-content">
      <p>Are you sure you want to add this item to your cart?</p>
      <div class="modal-btns">
        <button onclick="addToCart()" class="confirm-btn">Yes</button>
        <button onclick="hideConfirmation()" class="cancel-btn">Cancel</button>
      </div>
    </div>
  </div>

  <!-- Buy Now Modal -->
  <div id="buyNowModal" class="modal">
    <div class="modal-content">
      <span class="close" onclick="closeModal()">&times;</span>
      <h2>Enter Your Details</h2>
      <form id="orderForm">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required><br><br>
        <label for="address">Address:</label>
        <input type="text" id="address" name="address" required><br><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br><br>

        <label for="phone">Phone:</label>
        <input type="tel" id="phone" name="phone" required><br><br>

        <input type="submit" value="Submit">
      </form>
    </div>
  </div>
  <br><br><br>
</main>
<div class="contact">

  <main class="main-content">

    <!-- Contact Us section
    <section  class="contact-section">
      <h1>Contact Us</h1>
      <div class="contact-form">
        <form id="contactForm">
          <label for="name">Name:</label>
          <input type="text" id="name" name="name" required><br><br>

          <label for="email">Email:</label>
      <input type="email" id="email" name="email"
required><br><br>


      <label for="message" >Message:</label><br>

      <textarea id="message" name="message" rows="4"
required></textarea><br><br>


      <input type="submit" value="Submit">

     </form>

    </div>

    <div class="map-section" id="contact"> -->

     <!-- Google Map embedded iframe code -->

     <!-- <iframe

src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d
3968.8009551919966!2dYourLongitude!3dYourLatitude!2m3!1f0!2f0!3
f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x0%3A0x0!2zYourLatitude!5
e0!3m2!1sen!2sus!4v1637344972904!5m2!1sen!2sus"

      width="600" height="450" style="border:0;" allowfullscreen=""
loading="lazy"></iframe>

    </div>

   </section> -->

   <div class="container">

    <div class="content">

     <div class="left-side">

      <div class="address details">

       <i class="fas fa-map-marker-alt"></i>

       <div class="topic">Address</div>
       <div class="text-one">xyz, NP12</div>
       <div class="text-two">abc</div>
      </div>
      <div class="phone details">
       <i class="fas fa-phone-alt"></i>
       <div class="topic">Phone</div>
       <div class="text-one">+00000000</div>
       <div class="text-two">+00000000</div>
      </div>
      <div class="email details">
       <i class="fas fa-envelope"></i>
       <div class="topic">Email</div>
       <div class="text-one">xyz@gmail.com</div>
       <div class="text-two">info.xyz@gmail.com</div>
      </div>
     </div>
     <div class="right-side">
      <div class="topic-text">Send us a message</div>
      <p>If you have any types of quries related to our rugs, you can send us message from here. It's our pleasure to help you.</p>
     <form action="#">
      <div class="input-box">
       <input type="text" placeholder="Enter your name">
      </div>
      <div class="input-box">
       <input type="text" placeholder="Enter your email">
      </div>
```

```
            <div class="input-box message-box">
              <input type="text" placeholder="Enter your message">
            </div>
            <div class="button">
              <input type="submit" value="Send Now" >
            </div>
          </form>
        </div>
        </div>
      </div>
  </div>

  <footer>
    <div class="footer-content" id="about">
      <p>&copy; 2023 Layerrug</p>
      <p>Mob.No- 8445621552</p>
    </div>
  </footer>

  <script src="https://cdn.jsdelivr.net/npm/swiper@9/swiper-bundle.min.js"></script>
  <script>
    var swiper = new Swiper(".swiper-container", {
      spaceBetween: 30,
      effect: "fade",
      loop: true,
      autoplay: {
```

```
    delay: 3500,
    disableOnInteraction: false
  }
});
function filterProducts(category) {
  const products = document.querySelectorAll('.product');
  products.forEach(product => {
    const productCategory = product.dataset.category;
    if (category === 'all' || productCategory === category) {
      product.style.display = 'block';
    } else {
      product.style.display = 'none';
    }
  });
}
// Your existing JavaScript code for the cart functionality

// Function to open the Buy Now modal
function openModal() {
  document.getElementById('buyNowModal').style.display = 'block';
}

// Function to close the Buy Now modal
function closeModal() {
  document.getElementById('buyNowModal').style.display = 'none';
}
```

```
  // Function to handle Buy Now form submission
  document.getElementById('orderForm').addEventListener('submit', function (event) {
    event.preventDefault();
    closeModal(); // Close the modal after submission
    alert('Order placed successfully!'); // Replace with your logic to handle the form submission
  });

  let currentItem = null;

  // Function to show the confirmation modal
  function showConfirmation(name, image, price) {
    currentItem = { name, image, price };
    const modal = document.getElementById('confirmationModal');
    modal.style.display = 'block';
  }

  // Function to hide the confirmation modal
  function hideConfirmation() {
    const modal = document.getElementById('confirmationModal');
    modal.style.display = 'none';
  }

  function addToCart() {
    if (currentItem) {
        const item = {
          name: currentItem.name,
          image: currentItem.image,
          price: currentItem.price
        };

        // Retrieve existing cart items or create an empty array
        let cartItems = JSON.parse(localStorage.getItem('cart')) || [];
        cartItems.push(item);

        // Store updated cart items in localStorage
        localStorage.setItem('cart', JSON.stringify(cartItems));

        alert('Item added to cart!');
      }
      hideConfirmation();
    }

    // Get the 'Products' link from the navigation bar
    const productsLink = document.querySelector('a[href="#products"]');
    const headerHeight = document.querySelector('header').offsetHeight; // Get the header height

    // Function to handle scrolling to the product section
    function scrollToProducts(event) {
      event.preventDefault(); // Prevent the default behavior of the link
      const productSection = document.getElementById('products'); //
  Get the product section
      const offsetTop = productSection.offsetTop - headerHeight; //
  Consider header height for smooth scroll

      // Scroll smoothly to the product section
      window.scrollTo({
        top: offsetTop,
        behavior: 'smooth'
      });
    }

    // Add an event listener to handle the click on the 'Products' link
    productsLink.addEventListener('click', scrollToProducts);

    // Get the 'Home' link from the navigation bar
    const homeLink = document.querySelector('a[href="#home"]');

    // Function to handle scrolling back to the top
    function scrollToTop(event) {
      event.preventDefault(); // Prevent the default behavior of the link

      // Scroll smoothly to the top of the page
      window.scrollTo({
        top: 0,
        behavior: 'smooth'
      });
    }

    // Add an event listener to handle the click on the 'Home' link
    homeLink.addEventListener('click', scrollToTop);
    function scrollToSection(sectionId) {
      const section = document.getElementById(sectionId);
      const headerHeight = document.querySelector('header').offsetHeight; // Get the header height
      const offsetTop = section.offsetTop - headerHeight; // Calculate the offset considering the header height
      window.scrollTo({
        top: offsetTop,
        behavior: 'smooth'
      });
    }

    // Rest of your JavaScript code

    // Event listeners for 'Contact Us' and 'About Us' links
  document.querySelector('a[href="#contact"]').addEventListener('click', function (event) {
      event.preventDefault();
      scrollToSection('contact');
    });
```

```
    document.querySelector('a[href="#about"]').addEventListener('click', function (event) {
      event.preventDefault();
      scrollToSection('about');
    });



    // Function to handle Contact Form submission
    document.getElementById('contactForm').addEventListener('submit', function (event) {
      event.preventDefault();
      const name = document.getElementById('name').value;
      const email = document.getElementById('email').value;
      const message = document.getElementById('message').value;

      console.log(`Name: ${name}, Email: ${email}, Message: ${message}`);
      alert('Your message has been sent!');
      // Reset the form after submission
      document.getElementById('contactForm').reset();
    });
  </script>
</body>
</html>
```