**PHASE 5**

**Project Documentation & Submission**

NAME : DEVADHARSHINI R S

NAAN MUDHALVAN ID : au920821104013

PHASE : Project Documentation & Submission

**E-Commerce Application on IBM Cloud Foundry**

**Project Overview:**

1. Objective: Create a secure and scalable e-commerce platform on IBM Cloud Foundry.

2. Features: User management, product catalogic, cart/checkout, payment integration, order tracking, and robust security.

3.Technology Stack: Frontend (HTML/CSS/JS, React), Backend (Node.js, Express), Database (PostgreSQL), Hosting (IBM Cloud Foundry), Security (HTTPS, encryption).

4. Phases: Planning, Development, Deployment, Testing, Maintenance, and Scaling.

5. Success Metrics: High user engagement, conversion rates, secure transactions, minimal downtime.

6. Team: Project manager, developers, UI/UX designers, DB administrators, QA testers, and DevOps engineers.

**Design Thinking:**

1. Platform Design: Design the platform layout with sections for product categories, individual product pages, shopping cart, checkout, and payment.

2. Product Showcase: Create a database to store product information such as images, descriptions, prices, and categories.

3. User Authentication: Implement user registration and authentication features to enable artisans and customers to access the platform.

4. Shopping Cart and Checkout: Design and develop the shopping cart functionality and a smooth checkout process.

5. Payment Integration: Integrate secure payment gateways to facilitate transactions.

6. User Experience: Focus on providing an intuitive and visually appealing user experience for both artisans and customers.

**Incorporating features like product reviews, wish lists, and personalized recommendations to enhance user engagement and satisfaction.**

**PRODUCT REVIEWS:**

The product review feature is a game-changer. Being able to read reviews from other shoppers helps everyone make informed decisions. Plus, the ability to leave my own reviews adds a sense of community to the shopping experience.

* **User-Generated Reviews:** Allow users to leave reviews and ratings for products they have purchased or used. Ensure that the review system is easy to use and accessible.
* **Moderation:** Implement a moderation system to filter out inappropriate content and spam. This helps maintain the quality and trustworthiness of reviews.
* **Sorting and Filtering:** Enable users to sort and filter reviews based on criteria such as most recent, highest rated, lowest rated, or most helpful. This makes it easier for shoppers to find relevant information.
* **Response Mechanism:** Allow the store or product owners to respond to user reviews, addressing concerns or providing additional information. This shows your commitment to customer satisfaction.

**WISHLISTS:**

A well-implemented Wishlist feature allows users to save items they are interested in for future reference and can also serve as a tool for marketing and personalized recommendations.

* **User-Friendly Wishlist Creation:** Make it easy for users to create and manage Wishlist. This can be done through a simple click or tap on a "Add to Wishlist" button next to each product.
* **Privacy Options:** Give users the option to keep their wish lists private, share them with specific individuals, or make them public. This caters to different preferences.
* **Notifications:** Implement notifications to alert users when a product in their wish list goes on sale or becomes available again. This can encourage them to make a purchase.
* **Integration with Shopping Cart:** Allow users to easily move items from their wish list to their shopping cart when they decide to make a purchase.

**PERSONALISED RECOMMENDATIONS:**

Personalized recommendations can help users discover relevant products, increase their average order value, and keep them coming back for more.

* **User Profiling:** Collect data on user behaviour, such as browsing history, purchase history, and user preferences, to create user profiles.
* **Recommendation Algorithms:** Utilize recommendation algorithms (collaborative filtering, content-based filtering, or hybrid approaches) to suggest products tailored to each user's preferences.
* **Dynamic Content:** Display personalized product recommendations on the homepage, product pages, and in email marketing campaigns.
* **A/B Testing:** Continuously refine your recommendation algorithms through A/B testing to improve accuracy and relevance.

**Design the platform layout and create a database to store product information:**

Designing a platform layout and creating a database to store product information involves several steps.

PLATFORM LAYOUT DESIGN:

* Start by designing the user interface for your platform. Consider the overall look and feel, user experience, and navigation.
* Create wireframes or mock-ups to visualize the layout.
* Implement a user authentication system for user registration and login.
* Consider using industry-standard authentication protocols like OAuth or OpenID for enhanced security.
* Design a dashboard where users can access various features and functions of the platform.
* Organize it logically, such as with sections for products, user profiles, settings, and more.
* Create a page to display a list of products. Include features like search, filters, and sorting options.
* Design a detailed view of a product that shows information like product name, description, price, and images.
* If the platform is for sellers, create an admin panel for managing products, orders, and user accounts.
* Implement a shopping cart where users can add and manage the products they want to purchase.
* Design a checkout page for users to enter their shipping and payment information.

DATABASE DESIGN:

User Table:

Fields: UserID (Primary Key), Username, Email, Password (hashed), etc.

Product Table:

Fields: ProductID (Primary Key), ProductName, Description, Price, StockQuantity, CategoryID (Foreign Key), SellerID (Foreign Key), etc.

Category Table:

Fields: CategoryID (Primary Key), CategoryName, Description, etc.

Order Table:

Fields: OrderID (Primary Key), UserID (Foreign Key), OrderDate, TotalAmount, ShippingAddress, etc.

OrderItem Table:

Fields: OrderItemID (Primary Key), OrderID (Foreign Key), ProductID (Foreign Key), Quantity, UnitPrice, Subtotal, etc.

Seller Table (if applicable):

Fields: SellerID (Primary Key), SellerName, SellerDescription, ContactInfo, etc.

Image Table (for product images):

Fields: ImageID (Primary Key), ProductID (Foreign Key), ImageURL, etc.

Review Table:

Fields: ReviewID (Primary Key), ProductID (Foreign Key), UserID (Foreign Key), Rating, ReviewText, ReviewDate, etc.

**CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Artisanal E-Commerce</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<h1>Welcome to Artisanal Marketplace</h1>

<nav>

<ul>

<li><a href="#home">Home</a></li>

<li><a href="#products">Products</a></li>

<li><a href="#cart">Cart</a></li>

</ul>

</nav>

</header>

<section id="home">

<h2>Discover Unique Handcrafted Items</h2>

<p>Explore our collection of artisan-made products.</p>

</section>

<section id="products ">

<h2>Featured Products</h2>

<div class="product-gallery">

<!-- Display product images, names, and prices -->

<div class="product">

<img src="product1.jpg" alt="Product 1">

<h3>Artistic Jewelry</h3>

<p>$50.00</p>

<button>Add to Cart</button>

</div>

<!-- Add more products -->

</div>

</section>

<section id="cart" >

<h2>Your Shopping Cart</h2>

<!-- Display cart items and total -->

<div class="cart-items">

<!-- Display cart items -->

</div>0

<p>Total: $[Total Amount]</p>

<button>Proceed to Checkout</button>

</section>

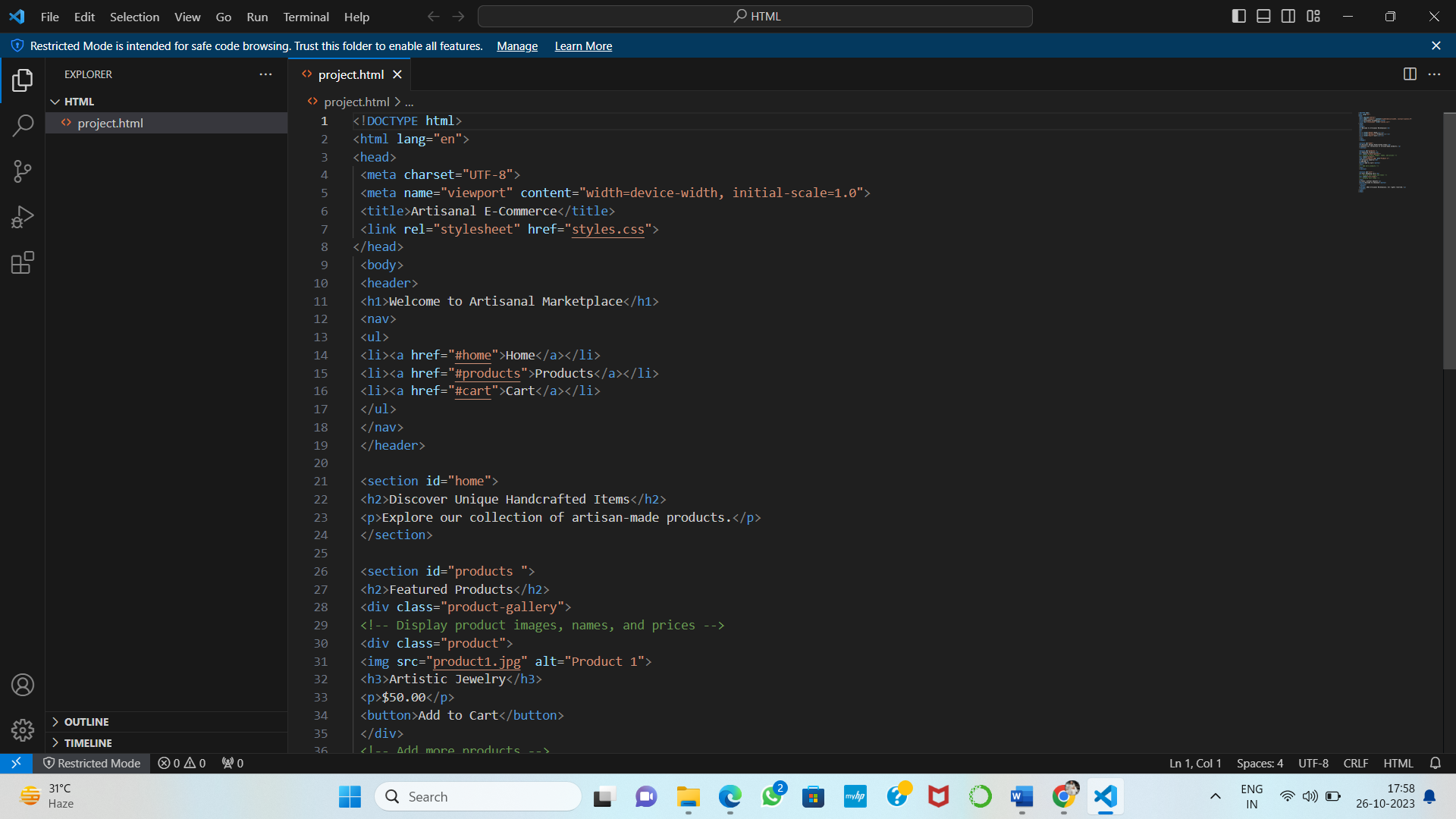
<footer>

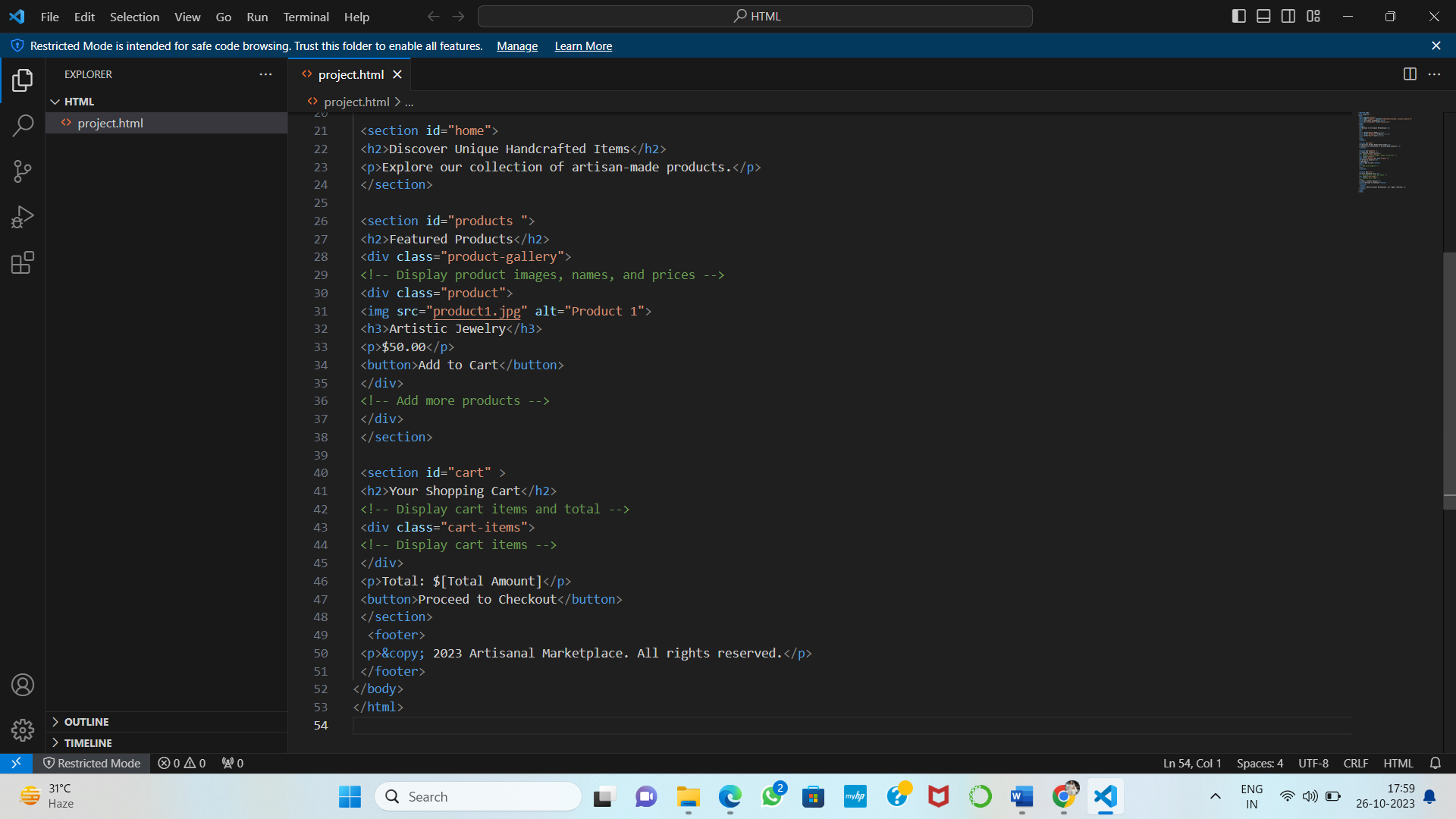
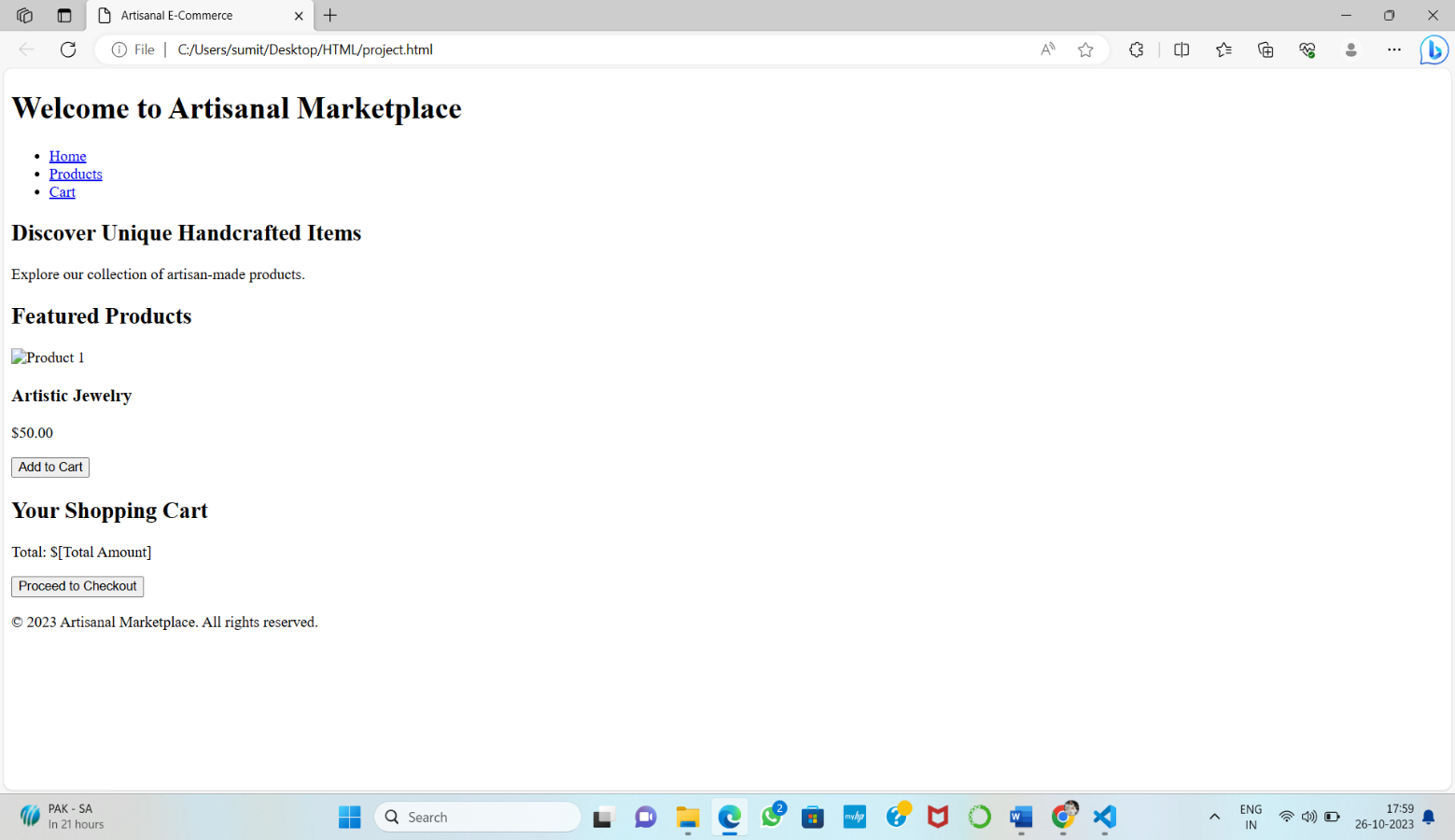
<p>&copy; 2023 Artisanal Marketplace. All rights reserved.</p>

</footer>

</body>

</html>



OUTPUT:

Building the e-commerce platform by implementing user authentication, shopping cart, and checkout functionality:

**User Authentication:**

* Implement user registration and login functionality.
* Use authentication libraries like Passport.js (for Node.js) or Devise (for Ruby on Rails) to handle user authentication.
* Secure user data and sessions using techniques like password hashing and JSON Web Tokens (JWT).
* Implement password reset functionality with email verification.

**HTML Code:**

<center>

    <h1>Artizan</h1>

    <p>

        Sign in to Arizan.com using your Artizan Account

    </p>

    <h2>

        Sign in

    </h2>

    <table border="1">

        <tr>

           <td> Email (phone for mobile accounts)

            </td>

        <tr>

          <td>  passord

            </td>

        </tr>

    </table>

    <h3>

        Sign in

    </h3>

    <table border="1">

        <tr>

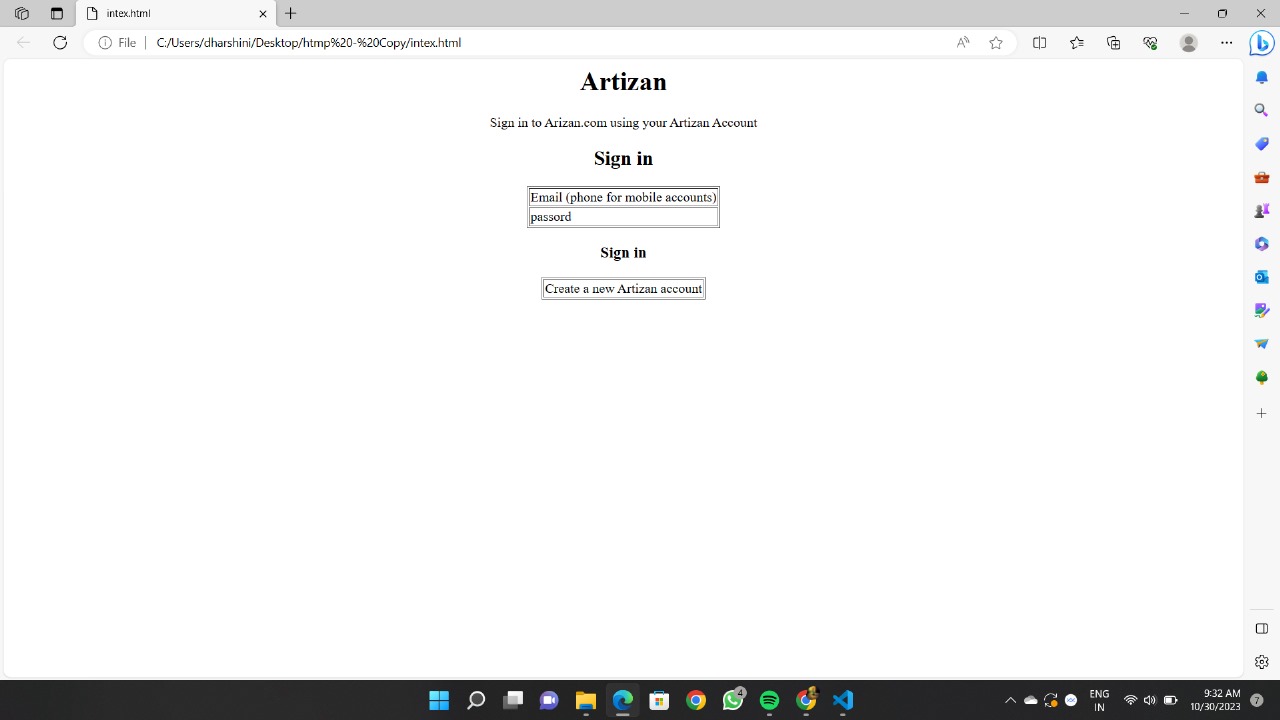
           <td>Create a new Artizan account

            </td>

        </tr>

    </table>

    </center>

**Output:**

**Shopping Cart:**

* Create a data structure to represent the shopping cart (e.g., a session or a database table).
* Allow users to add, update, and remove items from their cart.
* Calculate the total cost of items in the cart.
* Use cookies or session management to associate a cart with a user.

**HTML Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Artizan</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="header">

      <nav>

        <h1>Artizan</h1>

        <ul>

            <li>Home|</li>

            <li>About us|</li>

            <li>Contact us</li>

        </ul>

      </nav>

    </div>

    <div class="searchbar">

        <input placeholder="seach">

    </div>

    <div class="products">

        <div class="box">

       <img src="images.jpeg">

       <p>Black Forum Baby Reborn Toddler Baby Dolls Silicone Full Body girl22 inch Long Hair Anatomically Correct Waterproof Dolls Reborn Pink Dress Birthday Gift Set for Girls</p>

    </div>

</div>

    <div class="about">

        <h1 style="text-align: center;">Thank You</h1>

        </div>

</body>

</html>

**CSS Code:**

\*{

    margin:0;

    padding:0;

}

nav{

    background-color: black;

    color: white;

}

li,h1,ul{

    display:inline;

}

ul{

    margin-left: 60%;

}

li{

    color: gray;

}

li:hover{

    color: white;

    cursor: pointer;

}

input{

    width: 60%;

    padding: 15px;

}

.searchbar{

    padding:50px;

    text-align: center;

}

.box{

    border-color: black;

    border-width: 2px;

    border-style: solid;

    display: inline-block;

    width: 220px;

}

.products{

    text-align: center;

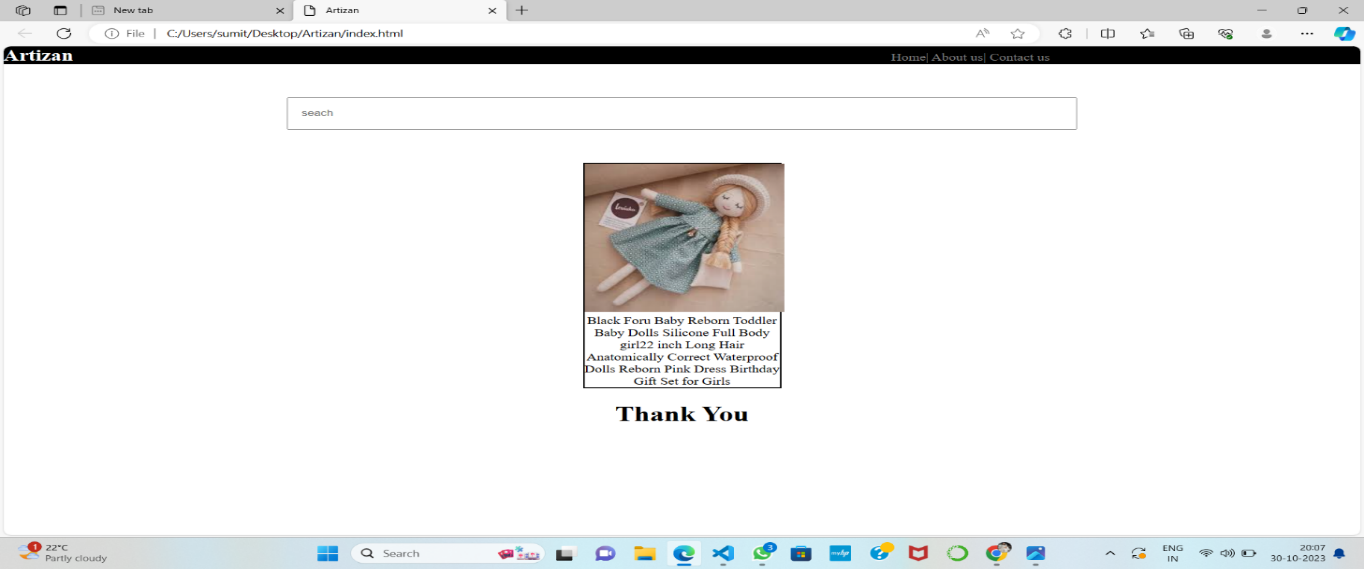
}

.about{

    text-align: center;

    margin-top: 20px;

}

**Output:**

**Checkout Functionality:**

* Create a secure payment gateway integration (e.g., Stripe, PayPal, or other preferred providers).
* Allow users to enter shipping and billing information.
* Implement order confirmation and receipt generation.
* Handle order fulfilment, including inventory management and order status updates.

It's essential to plan meticulously, invest in robust security measures, and continuously adapt to meet the changing needs and expectations of your users.

**Conclusion:**

In conclusion, deploying an e-commerce application on the cloud offers numerous advantages, including scalability, cost-efficiency, and global accessibility. Leveraging cloud services can streamline operations, enhance customer experiences, and facilitate rapid growth. However, it's crucial to carefully plan, secure, and maintain the application to fully realize these benefits. Ultimately, embracing cloud technology can empower e-commerce businesses to thrive in a competitive digital marketplace.