

PROJECT: “THE HOODIES SHOP”

(A SHOPPING WEBPAGE INTERFFACE
USING HTML, CSS AND JAVASCRIPT)

Contents

1. Abstract:

1.1 Introduction

1.2 Objective

2. Features

3. Implementation

4. Source code

4.1 Frontend (HTML, CSS, JS)

4.1.1 HTML

4.1.2 CSS

4.1.3 JS

5. Output

INTRODUCTION:

In the fast-paced world we live in, e-commerce has truly revolutionized the way we shop and conduct business. With just a few clicks, we can purchase products from the comfort of our homes and have them delivered straight to our doorstep. As such, it's essential for businesses to have a strong online presence, and a well-designed e-commerce website is a crucial component of that.

To create a successful e-commerce website, it's important to pay close attention to the user experience. The website should be easy to navigate, with intuitive menus and clear calls-to-action. A visually appealing design is also essential, as it can capture users' attention and encourage them to browse further.

Using HTML for structuring the content and CSS for styling is a great approach to building an e-commerce website. This allows for greater flexibility in terms of design and layout, and can help create a more engaging user experience. Additionally, incorporating features such as product recommendations, customer reviews, and a streamlined checkout process can all contribute to a successful e-commerce website.

Ultimately, the success of an e-commerce website relies on providing users with a seamless and enjoyable shopping experience. By focusing on user experience, incorporating effective design principles, and utilizing HTML and CSS effectively, businesses can create an e-commerce website that not only looks great but also drives conversions and grows their customer base.

OBJECTIVE:

The objective of building an e-commerce website using HTML and CSS is to create a visually appealing, user-friendly, and functional online platform for selling products or services. The website should provide an engaging and seamless shopping experience to users, encouraging them to browse, select, and purchase items with ease.

HERE ARE THE KEY OBJECTIVES:

1. **Aesthetically Pleasing Design:** To make a strong first impression, design an attractive and visually appealing layout using HTML and CSS. Use appropriate color schemes, typography, and imagery to enhance the overall look and feel of the website.
2. **Intuitive Navigation:** Enable users to easily browse through product categories, locate specific items, and access important sections of the website by implementing intuitive navigation menus, breadcrumbs, and search functionality.
3. **Mobile Responsiveness:** Ensure that the website is mobile-friendly and responsive to different screen sizes and devices, providing a seamless user experience across desktops, tablets, and mobile phones.
4. **Clear and Organized Product Display:** Clearly display product images, titles, descriptions, prices, and other relevant information in a clear and organized manner. Use CSS to create visually appealing product grids, carousel sliders, or other elements to effectively showcase products.

FEATURES:

1. **Adaptable Design:** Develop a responsive layout that adjusts to various screen sizes and devices, ensuring optimal viewing and usability on desktops, tablets, and mobile devices.
2. **Navigational Menu:** Create a navigation menu that allows users to easily browse between different sections or pages of the website, providing a clear and intuitive browsing experience.
3. **Styling and Visual Enhancements:** Utilize CSS to enhance the website's visual appeal by consistently applying colors, fonts, backgrounds, borders, and other styling elements across pages.
4. **CSS Layout Techniques:** Use CSS to design various layout structures such as grid layouts, flexbox layouts, or responsive frameworks like Bootstrap, for effective organization and positioning of content.
5. **Typography Management:** Apply CSS styles to control font families, sizes, weights, and other typographic properties, ensuring readability and consistency throughout the website.
6. **Interactive Image Displays:** Build image galleries or sliders using HTML and CSS to showcase multiple images in an engaging and visually appealing manner.

7. Interactive Effects and Animations: Apply CSS hover effects and animations to increase interactivity and provide visual feedback to users when they interact with elements such as buttons, links, or images.

8. Adaptive Images: Utilize CSS techniques like media queries or the srcset attribute to provide appropriately sized and optimized images based on the user's device and viewport size.

9. CSS Transitions and Transformations: Implement CSS transitions and transformations to create smooth and visually appealing effects like fading, scaling, rotating, or sliding elements on the page.

By integrating these functionalities into an HTML and CSS project, I have created a visually appealing, interactive, and user-friendly website that effectively conveys information and engages visitors.

HTML and CSS Implementation:

For this project, I utilized HTML and CSS to develop a visually appealing and dynamic website. The website is designed for a fictional clothing store, showcasing various clothing products and providing users with a seamless browsing and shopping experience.

HTML Structure:

To begin, I created the website's overall structure using HTML. This involved defining the main sections, including the header, navigation menu, content area, sidebar, and footer. I made use of HTML tags to create semantic structure, ensuring accessibility and clarity.

CSS Styling:

Next, I used CSS to style the elements defined in the HTML structure. I applied selectors to target specific elements and added styles such as colors, fonts, backgrounds, borders, and spacing. To maintain consistency and efficiency in styling, I considered cascading and inheritance principles.

SOURCE CODE

1. HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="C:\Users\battula suhas\OneDrive\Desktop\suhas adp
project\style.css">
</head>
<body class="">
  <nav>
    <div class="logo"></div>
    </div>
  </nav>
  <div class="container">
    <header>
      <h1 style="margin-left: 50px;">THE HOODIES SHOP</h1>
      <div class="shopping">
        
        <span class="quantity">0</span>
      </div>
    </header>
    <br>
```


<div class="list">

</div>

</div>

<div class="card">

<h1>Cart</h1>

<ul class="listCard">

<div class="checkOut">

<a

href="file:///C:/Users/battula%20suhas/OneDrive/Desktop/suhas%20adp%20project/
popup.html" target="_blank"><button class="total" type="submit">0</button>

<button class="closeShopping" onclick="close()">close</button>

</div>

</div>

</div>

<section id="footer">

<h4>DEVELOPED BY SUHAS</h4>

<h2 style="color: black;">THANKS FOR SHOPPING!</h2>

<h4>©THE HOODIES SHOP</h4>

</section>

```
<script src="app.js"></script>
</body>
</html>
```

2. HTML CODE FOR POPUP WEBSITE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="popup.css">
  <title>Document</title>
</head>
<body>
  <div class="center">
    <h1>THANKS FOR VISITING!</h1>
    <h2>your order is placed!</h2>
    <h2>&copy;THE HOODIES SHOP</h2>
  </div>
</body>
</html>
```

3. CSS:

```
body{
  background-color: #f8b9b9;
  font-family: system-ui;
}
.container{
  width: 1000px;
  margin: auto;
  transition: 0.5s;
}
header{
  display: flex;
  justify-content: space-between;
  align-items: center;
  background: rgb(167, 103, 103);
  color: white;
  padding: 15px;
  border-radius: 150px;
  align-items:center;
  border-bottom: 2px hotpink;
  width:950px;
  box-shadow:0px 10px 80px #ff0000;
}
header{
  display: grid;
  grid-template-columns: 1fr 50px;
```

```
margin-top: 40px;

}

header .shopping{
    position: relative;
    text-align: right;
}

header .shopping img{
    width: 30px;
}

header .shopping span{
    background: red;
    border-radius: 50%;
    display: flex;
    justify-content: center;
    align-items: center;
    color: #fff;
    position: absolute;
    top: -5px;
    left: 80%;
    padding: 3px 15px;
}

.list{
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    column-gap: 20px;
    row-gap: 20px;
    margin-top: 50px;
```

```
}  
.list .item{  
    text-align: center;  
    background-color: #DCE0E1;  
    padding: 20px;  
    box-shadow:0px 10px 80px #ff0000;  
    letter-spacing: 1px;  
    border-radius: 60px;  
}  
.list .item img{  
    width: 90%;  
}  
.list .item .title{  
    font-weight: 600;  
}  
.list .item .price{  
    margin: 10px;  
}  
.list .item button{  
    background-color: hsl(12, 59%, 57%);  
    color: #fff;  
    width: 100px;  
    padding: 12px;  
    border-radius: 25px;  
}  
.card{  
    position: fixed;  
    border-radius: 50px;
```

```
top:40px;
left: 100%;
width: 500px;
background:rgb(167, 103, 103);
box-shadow: 0px 10px 80px #000000;
height: 90vh;
transition: 0.8s;
}
.active .card{
    left: calc(100% - 500px);
}
.active .container{
    transform: translateX(-200px);
}
.card h1{
    color: #ffd900;
    font-weight: 100;
    margin: 0;
    padding: 0 20px;
    height: 80px;
    display: flex;
    align-items: center;
}
.card .checkOut{
    position: absolute;
    bottom: 0;
    width: 100%;
    display: grid;
```

```
    grid-template-columns: repeat(2, 1fr);

}

.card .checkOut div{
    background-color: #E8BC0E;
    width: 100%;
    height: 70px;
    display: flex;
    justify-content: center;
    align-items: center;
    font-weight: bold;
    cursor: pointer;
}

.card .checkOut div:nth-child(2){
    background-color: #000000;
    color: #000000;
;
}

.listCard li{
    display: grid;
    grid-template-columns: 100px repeat(3, 1fr);
    color: #fff;
    row-gap: 10px;
}

.listCard li div{
    display: flex;
    justify-content: center;
    align-items: center;
```

```
}  
.listCard li img{  
    width: 90%;  
}  
.listCard li button{  
    background-color: #fff5;  
    border: none;  
}  
.listCard .count{  
    margin: 0 10px;  
}  
#footer{  
    display: flex;  
    justify-content: space-evenly;  
    align-items: center;  
    background: rgb(167, 103, 103);  
    color: white;  
    padding: 5px;  
    border-radius: 40px;  
    border-bottom: 2px hotpink;  
    box-shadow: 0px 10px 80px #ff0000;  
}  
.total{  
    background-color: #E8BC0E;  
    align-items: center;  
    padding: 20px;  
    border-radius: 20px;  
    text-decoration: none;
```



```
}  
.closeShopping{  
    background-color: rgb(255, 255, 255);  
    padding: 20px;  
    border-radius: 20px;  
}
```

4. CSS CODE FOR POPUP WEBSITE:

```
body{  
    background-color: #d56565;  
    font-family: system-ui;  
}  
.center {  
    text-align: center;  
    padding: 250px;  
    border: 5px solid rgb(255, 255, 255);  
    border-radius: 50px;  
    margin-top: 20px;  
}
```

5. JAVASCRIPT:

```
let openShopping = document.querySelector('.shopping');
let closeShopping = document.querySelector('.closeShopping');
let list = document.querySelector('.list');
let listCard = document.querySelector('.listCard');
let body = document.querySelector('body');
let total = document.querySelector('.total');
let quantity = document.querySelector('.quantity');
openShopping.addEventListener('click', ()=>{
    body.classList.add('active');
})
closeShopping.addEventListener('click', ()=>{
    body.classList.remove('active');
})
let products = [
    {
        id: 1,
        name: 'Imagine Dragons Hoodie',
        image: '1.PNG',
        price: 699
    },
    {
        id: 2,
        name: 'Orange Hoodie',
        image: '2.PNG',
        price: 699
    }
]
```

```
    },  
    {  
      id: 3,  
      name: 'Biege ghost Hoodie',  
      image: '3.PNG',  
      price: 799  
    },  
    {  
      id: 4,  
      name: 'Grey Savage Hoodie',  
      image: '4.PNG',  
      price: 999  
    },  
    {  
      id: 5,  
      name: 'White Water Hoodie',  
      image: '5.PNG',  
      price: 599  
    },  
    {  
      id: 6,  
      name: 'Black Hoodie',  
      image: '6.PNG',  
      price: 899  
    }  
  ];  
  let listCards = [];  
  function initApp(){
```

```

products.forEach((value, key) =>{
  let newDiv = document.createElement('div');
  newDiv.classList.add('item');
  newDiv.innerHTML = `
    
    <div class="title">${value.name}</div>
    <div class="price">${value.price.toLocaleString()}</div>
    <button onclick="addToCard(${key})">Add To Cart</button>`;
  list.appendChild(newDiv);
})
}
initApp();
function addToCard(key){
  if(listCards[key] == null){
    // copy product form list to list card
    listCards[key] = JSON.parse(JSON.stringify(products[key]));
    listCards[key].quantity = 1;
  }
  reloadCard();
}
function reloadCard(){
  listCard.innerHTML = "";
  let count = 0;
  let totalPrice = 0;
  listCards.forEach((value, key)=>{
    totalPrice = totalPrice + value.price;
    count = count + value.quantity;
    if(value != null){

```

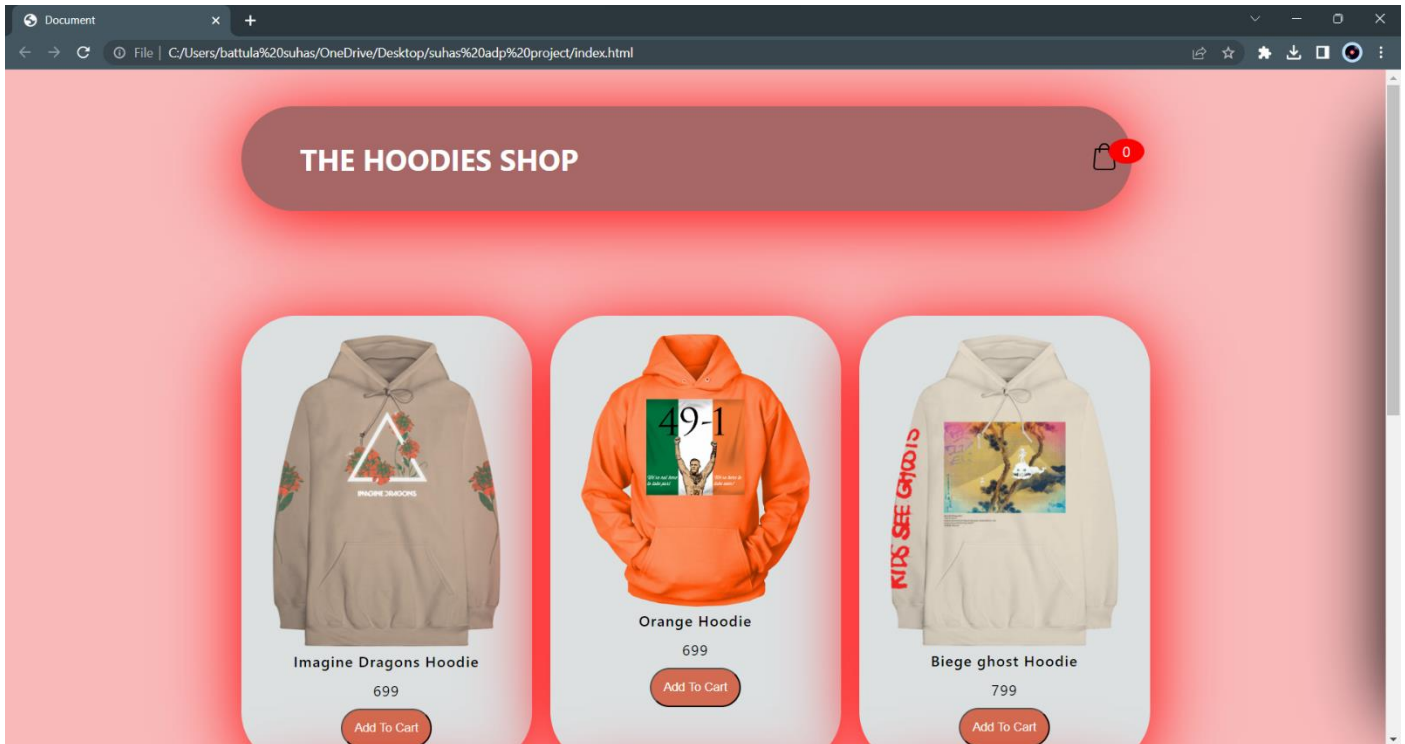
```

let newDiv = document.createElement('li');
newDiv.innerHTML = `
    <div></div>
    <div>${value.name}</div>
    <div>${value.price.toLocaleString()}</div>
    <div>
        <button onclick="changeQuantity(${key}, ${value.quantity - 1})">-
</button>
        <div class="count">${value.quantity}</div>
        <button onclick="changeQuantity(${key}, ${value.quantity +
1})">+</button>
    </div>`;
    listCard.appendChild(newDiv);
}
})
total.innerText = totalPrice.toLocaleString();
quantity.innerText = count;
}
function changeQuantity(key, quantity){
    if(quantity == 0){
        delete listCards[key];
    }else{
        listCards[key].quantity = quantity;
        listCards[key].price = quantity * products[key].price;
    }
    reloadCard();
}

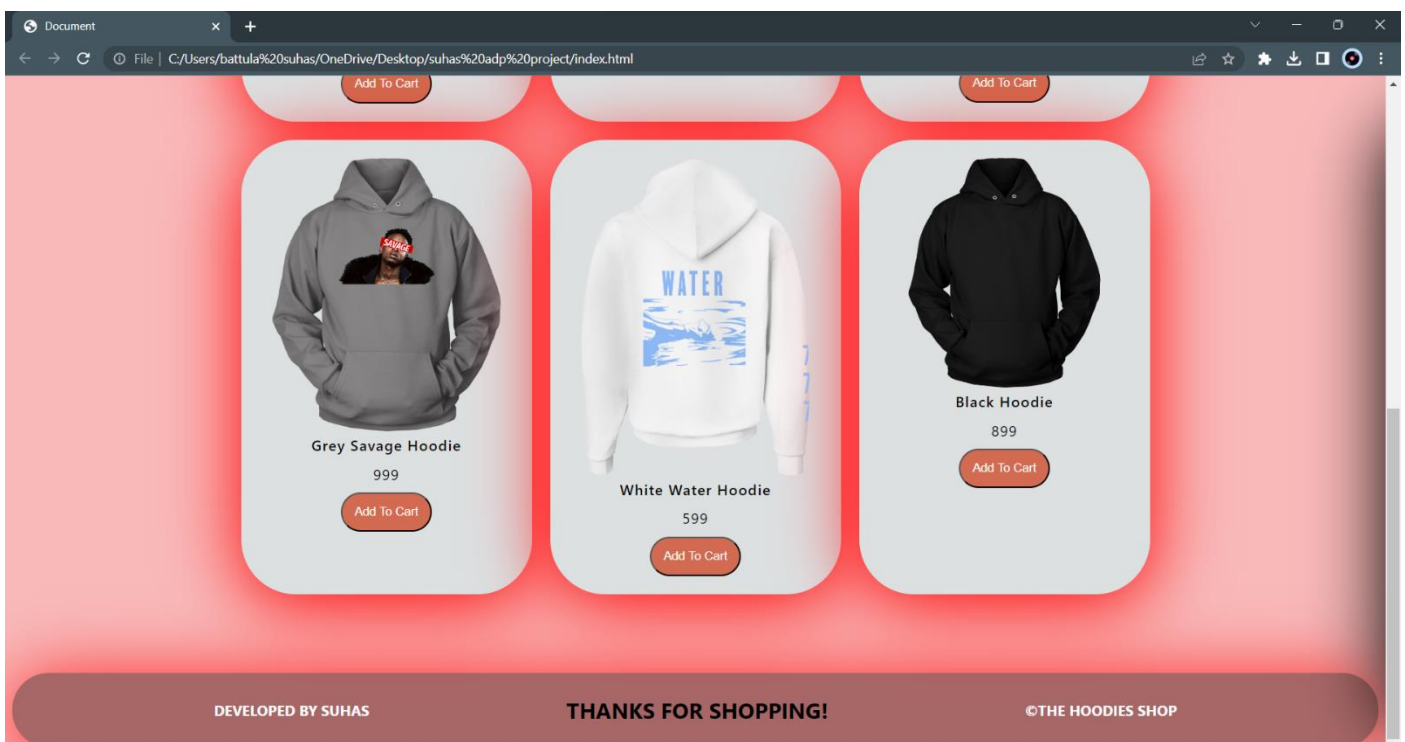
```

OUTPUT

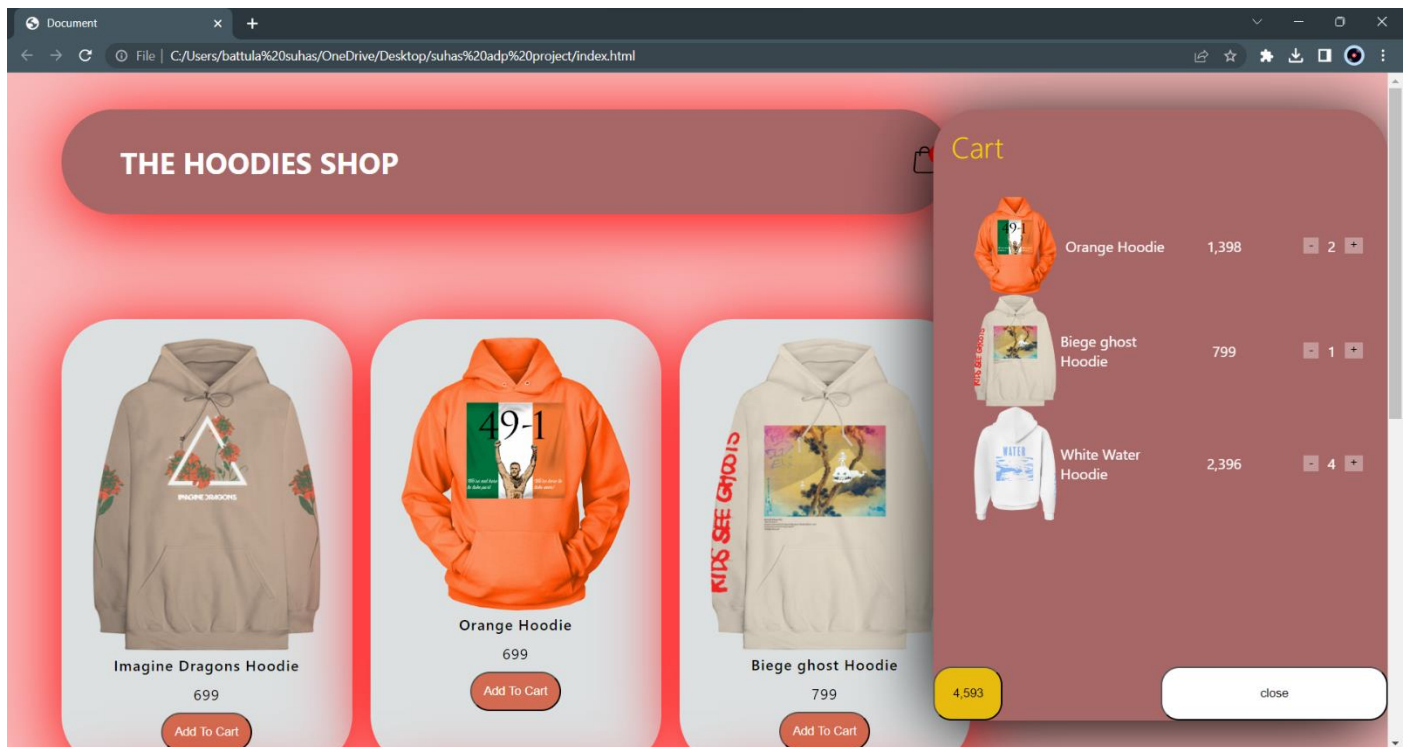
HEADER SECTION WITH BODY OF THE WEBPAGE:



FOOTER SECTION WITH BODY OF THE WEBPAGE:



CART SECTION:



AFTER CHECKOUT:

