
A Micro Project report on

Full Stack Web Development for Records Management

Submitted to the CMR Institute of Technology in partial fulfillment of the
requirement of the

FULL STACK WEB DEVELOPMENT of III-B.Tech.

II Semester in

Computer Science and Engineering Department

Submitted by

B. Ajaykumar	:	19R01A05DO
B. Sagar	:	19R01A05D2
G. Saikiran	:	19R01A05G2
M. Bhanuprasad	:	19R01A05G7
G. Shravan kumar	:	19R01A05G9
T. Naveen	:	19R01A05H6

Under the Guidance Of

T.Nagaraju

(Assistant Professor, Computer Science and Engineering Dept)



(UGC AUTONOMUS)

(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad **20212022**

INDEX

Sl.No	Particulars	Page No.
1.	Introduction	1
2.	Requirements	2

3. Source Code	3
4. Output	15
5. Conclusion	16
6. Reference	17

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMUS)

(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad.

Department of Computer Science and Engineering



CERTIFICATE

This is to certify that a Micro Project entitled with: “Develop a shopping cart application by using HTML5, CSS, JQUERY in GITHUB” is being submitted by:

B. Ajaykumar	:	19R01A05D0
B. Sagar	:	19R01A05D2
M. Saikiran	:	19R01A05G2
M. Bhanuprasad	:	19R01A05G7
G. Shravan kumar	:	19R01A05G9
T. Naveen	:	19R01A05H6

In partial fulfillment of the requirement for completion of the Full Stack Web Development of III-B.Tech II- Semester is a record of a bonafide work carried out under guidance and supervision.

Signature of Faculty

Signature of HOD

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. M. Janga Reddy, Director, Dr. B. Satyanarayana, Principal** and **Mr. Prakash, Head of Department**, Dept of Computer Science and Engineering, CMR Institute of Technology for their inspiration and valuable guidance during entire duration.

We are extremely thankful to our **T.Nagaraju, Assistant Professor**, Computer Science and Engineering department, CMR Institute of Technology for his constant guidance, encouragement and moral support throughout the project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this micro project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for successful completion of the project.

B. Ajaykumar	:	19R01A05D0
B. Sagar	:	19R01A05D2
M. saikiran	:	19R05A05G2
M. Bhanuprasad	:	19R05A05G7
G. Shravan kumar	:	19R05A05G9
T. Naveen	:	19R05A05H6

1.INTRODUCTION

An electronic shopping cart is a software resource that functions as an interface for users that buy items from a business online. E-commerce websites use electronic shopping carts to facilitate purchasing by Web users. A broad range of customized designs with a variety of style and utility features are available for electronic shopping carts.

Think of the shopping cart page as this moment. Just before you check out. If you have a shopping list, now is the time to check it to make sure you have all the items you want to buy. You do not have a credit card. But you know that if everything goes according to plan, your next stop will be at the checkout.

The implementation is done using HTML,CSS, JQuery. The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript. Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License. As of May 2019, jQuery is used by 73% of the 10 million most popular websites.

Today we will be building an app that lets us view the shopping cart and total amount for each item on the right side along with images and description on right side. We will also implement it for multiple items and it is shared on GitHub.

2.Requirements

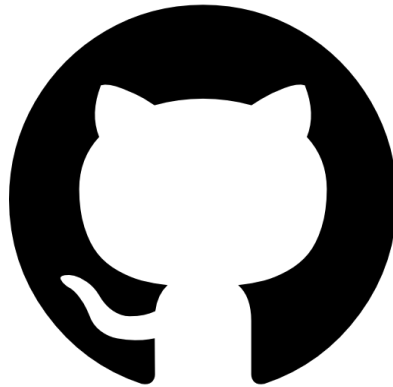
This website requires following tools:

-
- HTML - With HTML you can create your own Website
 - CSS - CSS is the language we use to style an HTML document.
 - JQuery - jQuery is a fast, small, and feature-rich JavaScript library.
 - GitHub - it provider of Internet hosting for software development and version control.
 - Browser - it is used to display webpage.

Hardware Requitemnets:

- Computer/Laptop
 - Internet Connection
- #### Software Requirements:
- Visual Studio Code
 - Node.js
 - Any working browser

HTML



Visual Studio Code



3.SOURCE CODE

Step 1: Let's create our HTML structure.

First, we need a container div, which we'll call ".shopping-cart".

Inside the container, we will have a title and three items which will include:

- two buttons — delete button and favorite button

- product image product name and description

- buttons that will adjust quantity of products

- total price

```
1 <div class="shopping-cart">
2   <!-- Title -->
3   <div class="title">
4     Shopping Bag
5   </div>
6
7   <!-- Product #1 -->
8   <div class="item">
9     <div class="buttons">
10      <span class="delete-btn"></span>
11      <span class="like-btn"></span>
12    </div>
13
14    <div class="image">
15      
16    </div>
17
18    <div class="description">
19      <span>Common Projects</span>
20      <span>Bball High</span>
21      <span>White</span>
22    </div>
23
24    <div class="quantity">
25      <button class="plus-btn" type="button" name="button">
26        
27      </button>
28      <input type="text" name="name" value="1">
29      <button class="minus-btn" type="button" name="button">
30        
31      </button>
32    </div>
33
34    <div class="total-price">$549</div>
```



```
35 </div>
36
37 <!-- Product #2 -->
38 <div class="item">
39   <div class="buttons">
40     <span class="delete-btn"></span>
41     <span class="like-btn"></span>
42   </div>
43
44   <div class="image">
45     
46   </div>
47
48   <div class="description">
49     <span>Maison Margiela</span>
50     <span>Future Sneakers</span>
51     <span>White</span>
52   </div>
53
54   <div class="quantity">
55     <button class="plus-btn" type="button" name="button">
56       
57     </button>
58     <input type="text" name="name" value="1">
59     <button class="minus-btn" type="button" name="button">
60       
61     </button>
62   </div>
63
64   <div class="total-price">$870</div>
65 </div>
66
67 <!-- Product #3 -->
68 <div class="item">
```

```

68 <div class="item">
69   <div class="buttons">
70     <span class="delete-btn"></span>
71     <span class="like-btn"></span>
72   </div>
73
74   <div class="image">
75     
76   </div>
77
78   <div class="description">
79     <span>Our Legacy</span>
80     <span>Brushed Scarf</span>
81     <span>Brown</span>
82   </div>
83
84   <div class="quantity">
85     <button class="plus-btn" type="button" name="button">
86       
87     </button>
88     <input type="text" name="name" value="1">
89     <button class="minus-btn" type="button" name="button">
90       
91     </button>
92   </div>
93
94   <div class="total-price">$349</div>
95 </div>
96 </div>

```

CSS

Make sure you include the font we use for this tutorial.

Now, let's add some style to our body with the following lines:

```

1  * {
2    box-sizing: border-box;
3  }
4
5  html,
6  body {
7    width: 100%;
8    height: 100%;
9    margin: 0;
10   background-color: #7EC855;
11   font-family: 'Roboto', sans-serif;
12 }

```

Great, now let's make our Shopping Cart to be 750×423 and style it to look nice. Notice that we are using flexbox, so we set it to display flex and make the flex direction column, because by default the flex direction is set as row.

```
1 .shopping-cart {
2   width: 750px;
3   height: 423px;
4   margin: 80px auto;
5   background: #FFFFFF;
6   box-shadow: 1px 2px 3px 0px rgba(0,0,0,0.10);
7   border-radius: 6px;
8
9   display: flex;
10  flex-direction: column;
11 }
```

Next, let's style the first item, which is the title, by changing the height to 60px and giving it some basic styling, and the next three items which are the shopping cart products, will make them 120px height each and set them to display flex.

```
.title {
  height: 60px;
  border-bottom: 1px solid #E1E8EE;
  padding: 20px 30px;
  color: #5E6977;
  font-size: 18px;
  font-weight: 400;
}

.item {
  padding: 20px 30px;
  height: 120px;
  display: flex;
}

.item:nth-child(3) {
  border-top: 1px solid #E1E8EE;
  border-bottom: 1px solid #E1E8EE;
}
```

Now we've styled the basic structure of our shopping cart.
Let's style our products in order.

The first elements are the delete and favorite buttons.

I've always loved Twitter's heart button animation, I think it would look great on our Shopping Cart.

```
.buttons {
  position: relative;
  padding-top: 30px;
  margin-right: 60px;
}
.delete-btn,
.like-btn {
  display: inline-block;
  cursor: pointer;
}
.delete-btn {
  width: 18px;
  height: 17px;
  background: url(&quot;delete-icn.svg&quot;) no-repeat center;
}

.like-btn {
  position: absolute;
  top: 9px;
  left: 15px;
  background: url('twitter-heart.png');
  width: 60px;
  height: 60px;
  background-size: 2900%;
  background-repeat: no-repeat;
}
```

We set class "is-active" for when we click the button to animate it using jQuery, but this is for the next session

```
.is-active {
  animation-name: animate;
  animation-duration: .8s;
  animation-iteration-count: 1;
  animation-timing-function: steps(28);
  animation-fill-mode: forwards;
}

@keyframes animate {
  0% { background-position: left; }
  50% { background-position: right; }
  100% { background-position: right; }
}
```

Next, is the product image which needs a 50px right margin.

```
.image {  
  margin-right: 50px;  
}
```

Let's add some basic style to product name and description.

```
.description {  
  padding-top: 10px;  
  margin-right: 60px;  
  width: 115px;  
}
```

```
.description span {  
  display: block;  
  font-size: 14px;  
  color: #43484D;  
  font-weight: 400;  
}
```

```
.description span:first-child {  
  margin-bottom: 5px;  
}
```

```
.description span:last-child {  
  font-weight: 300;  
  margin-top: 8px;  
  color: #86939E;  
}
```

Then we need to add a quantity element, where we have two buttons for adding or removing product quantity. First, make the CSS and then we'll make it work by adding some JavaScript.

```
.quantity {
  padding-top: 20px;
  margin-right: 60px;
}

.quantity input {
  -webkit-appearance: none;
  border: none;
  text-align: center;
  width: 32px;
  font-size: 16px;
  color: #43484D;
  font-weight: 300;
}

button[class*=btn] {
  width: 30px;
  height: 30px;
  background-color: #E1E8EE;
  border-radius: 6px;
  border: none;
  cursor: pointer;
}

.minus-btn img {
  margin-bottom: 3px;
}

.plus-btn img {
  margin-top: 2px;
}

button:focus,
input:focus {
  outline: 0;
}
```

And last, which is the total price of the product.

```
.total-price {  
  width: 83px;  
  padding-top: 27px;  
  text-align: center;  
  font-size: 16px;  
  color: #43484D;  
  font-weight: 300;  
}
```

Let's also make it responsive by adding the following lines of code:

```
@media (max-width: 800px) {  
  .shopping-cart {  
    width: 100%;  
    height: auto;  
    overflow: hidden;  
  }  
  .item {  
    height: auto;  
    flex-wrap: wrap;  
    justify-content: center;  
  }  
  .image img {  
    width: 50%;  
  }  
  .image,  
  .quantity,  
  .description {  
    width: 100%;  
    text-align: center;  
    margin: 6px 0;  
  }  
  .buttons {  
    margin-right: 20px;  
  }  
}
```

Java Script:-

Let's make the heart animate when we click on it by adding the following code:

```
$('.like-btn').on('click', function() {  
  $(this).toggleClass('is-active');  
});
```

here we are using toggleClass() method.

The toggleClass() method toggles between adding and removing one or more class names from the selected elements.

Now let's make the quantity buttons work.

```
$('.minus-btn').on('click', function(e) {
    e.preventDefault();
    var $this = $(this);
    var $input = $this.closest('div').find('input');
    var value = parseInt($input.val());

    if (value &gt; 1) {
        value = value - 1;
    } else {
        value = 0;
    }

    $input.val(value);
});

$('.plus-btn').on('click', function(e) {
    e.preventDefault();
    var $this = $(this);
    var $input = $this.closest('div').find('input');
    var value = parseInt($input.val());

    if (value &lt; 100) {
        value = value + 1;
    } else {
        value = 100;
    }

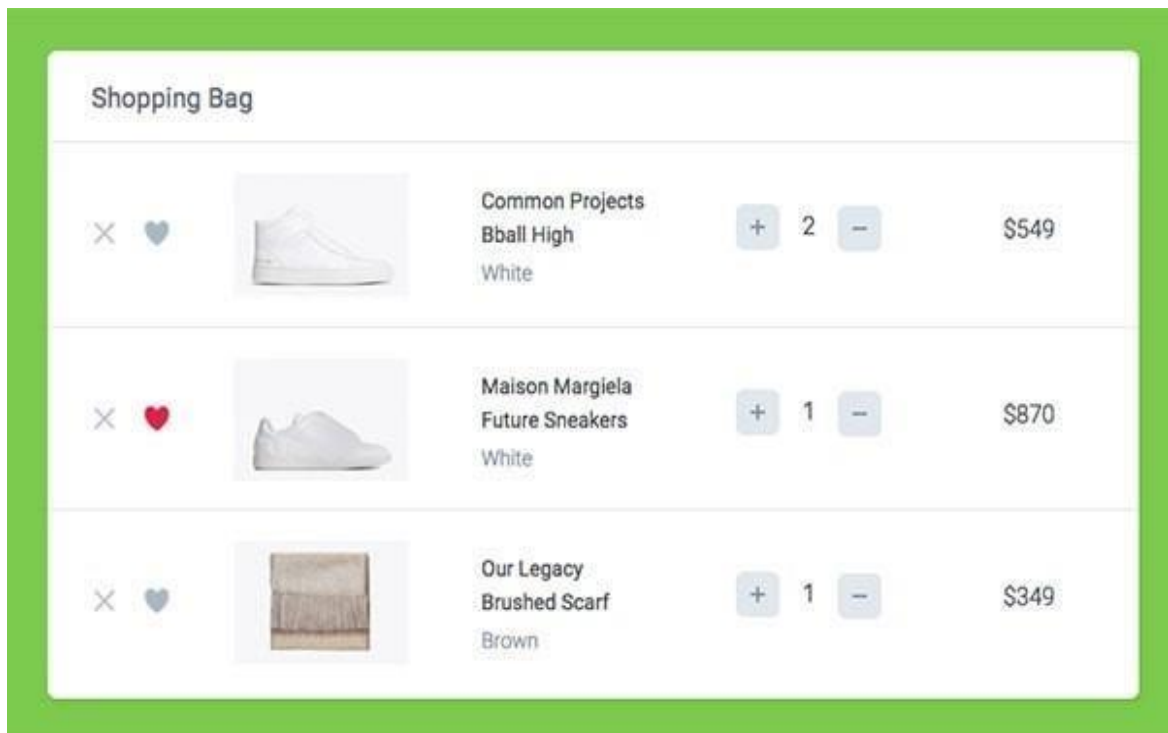
    $input.val(value);
});
```

We are adding minus and plus buttons.

When ever the plus button is pressed on the particular item the rate will be added to the cart(value will increase). It means an item is selected buy. We are providing the minus button also. You can also remove the items using minus symbol.

Then the item will be removed from the cart(value will decreased).

4.OUTPUT



5.CONCLUSION

congratulations! We have a working e shopping cart that we can add items and check the total amount. This can also be shared with the help of GitHub pages. You can add as many items you want and if you like a product you can also give rating to it.

You can view the source code here:

<https://github.com/naveenadg/microproject>

6.REFERENCES

This website was made using following resources:

- <https://designmodo.com/shopping-cart-ui/#download-the-shopping-cart-ui>
- <https://docs.github.com/en/repositories/working-with-files/managing-files/adding-a-file-to-a-repository>
- https://www.w3schools.com/jquery/jquery_get_started.asp
- <https://www.w3schools.com/html/default.asp>