

My final project report

A. 10 different html element types

On my layout page, I included, the root html elements which are

1. <html>
2. <head>
3. <meta>
4. <title>
5. <body>
6. <link> to connect to my css files, bootstrap cdn and cdn font awesome
7. <a> to link to more pages
8. <header> to indicate the header of my page
9. <div> to separate content of my webpage
10. <navbar> to create a navbar in my header

B. 10 different attribute name

On my layout page I included the following attributes

1. lang
2. charset
3. http-equiv
4. name
5. content
6. rel
7. href
8. class
9. id
10. type

C. At least one image with correct alt attribute

On my index page I included one image with the alt attribute saying "The logo image of ShopAmy"

D. 10 css rules

On my layout.css file I included the following rules

1. html { overflow-x: hidden;}
2. .firstinfo{ display: flex; flex-direction: row;}
3. .topheader > a{
font-family: 'Freehand', sans-serif;
font-size: 250%;
color: rgb(255, 0, 255);
text-decoration: none;}

4.

```
.topheader ul
{
  list-style: none;
  font-family: 'Patrick Hand';
  font-size: 1.5rem;
}
```
5.

```
.topheader li{
  position: relative;
  float: left;
}
```
6.

```
.mainnav li a{
  text-decoration: none;
  padding: 0.3rem;
  display: inline;
  color: black;
}
```
7.

```
.mainnav li a:hover{
  background: rgb(255, 0, 255);;
  color: white;
}
```
8.

```
.image1 img{
  flex:1 1 41em;
  width: 100%;
  height:550px;
}
```
9.

```
.image1-text {
  color: rgb(255, 0, 255);
  font-family: 'Shrikhand', cursive;
  padding-top: 10em;
}
```
10.

```
.topheader{
  position: sticky;
  align-items: center;
  display: flex;
  padding: 0 2%;
  justify-content: space-between;
  background: #fdeaf2;
  z-index: 1000;
}
```

E. CSS framework

I used bootstrap as my CSS framework

F. CSS flexbox or css grid

I used flexbox on my index page, we can verify this by checking the file layout.css as well as index.css or index.scss

G. Responsive design

I added responsive design to different sections of my website, on the layout.css we can see responsive CSS rules, I added max-width CSS properties, also using bootstrap elements on the cart.php, index.php, checkout.php.

H. Javascript event listener

I added some event listeners, one of them was in validation.js to verify if the inputs are empty, then in script2.js I added 2 event listeners to notify the user about the status of their actions, below the javascript class I added an event listener.

I. Javascript framework (e.g. jquery, vue, or react)

on the layout.php I added the CDN to use jquery, so on the folder js/script2.js file we can see the function that I call using jquery

J. Ajax to interact with an API

We can see the ajax trying to interact with the ajax.php

K. Optional

1. 5 different aria roles

We can see aria roles on my layout2.php within the container button, also in cart.php, checkout.php, and done.php

2. a CSS preprocessor (e.g. sass)

We can see different files using sass within my CSS folder

3. CSS grid or CSS flexbox (whichever one you didn't already use for #7 above)

In my file contact.scss and cart.scss I used a grid container, then In my about.scss in the image-container I used flexbox.

4. A javascript class

I added a javascript class in done.php, so the behavior of the class is to print the information that is displayed on the screen, so I did it using a javascript event listener calling the class.

5. Write your own simple backend (e.g. with express.js or even a different language)

I used PHP to create my backend. We can verify this, by checking that the extensions of the files end in .php, as well I used a MYSQL database to display data.