

# **WEB DEVELOPMENT**

Summer Internship Report

submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

in

Computer Science & Engineering

By

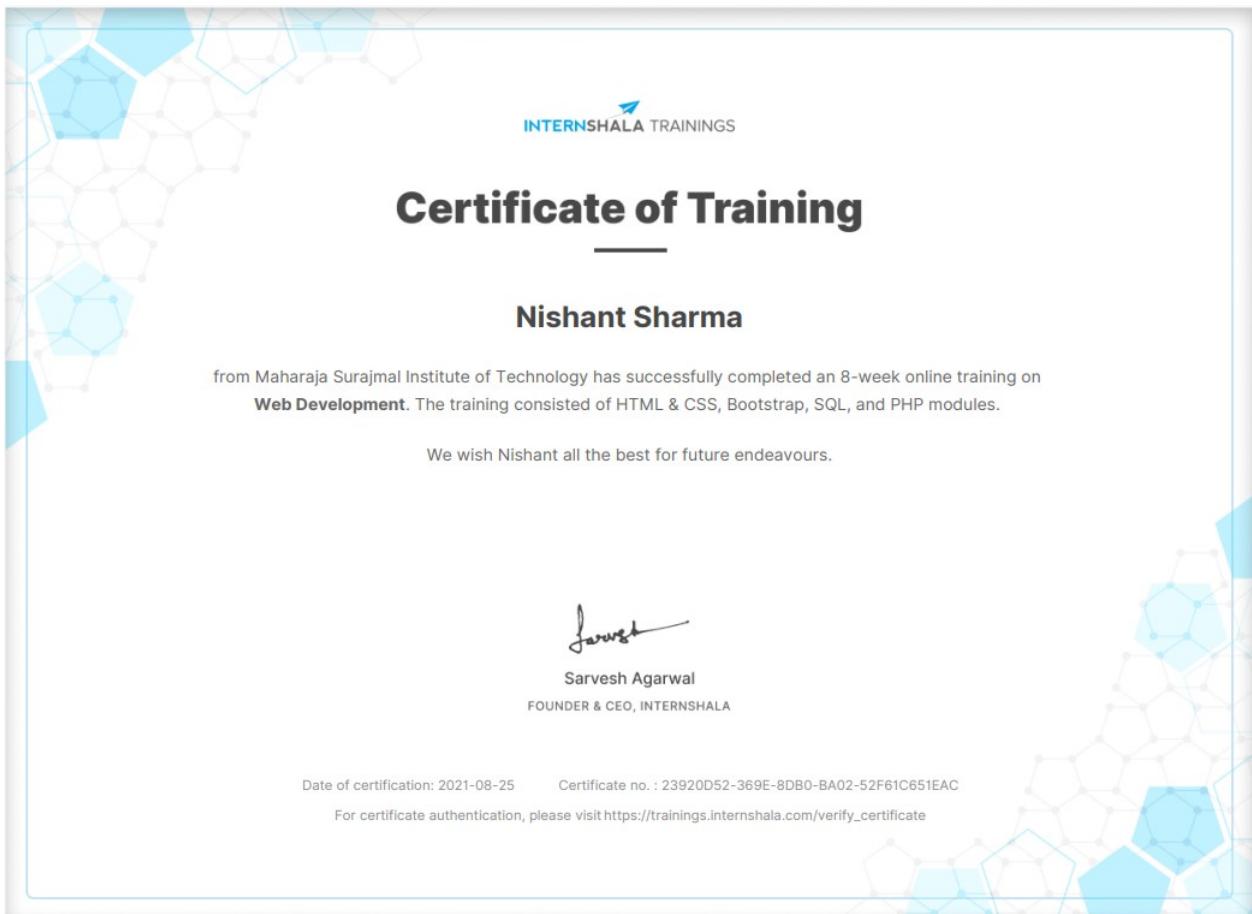
**Nishant Sharma– 02696302719**



Maharaja Surajmal Institute of Technology  
(Affiliated to Guru Gobind Singh Indraprastha University)  
Janakpuri, New Delhi-58

October 2021

## Certificate



### DECLARATION BY THE CANDIDATE

I, NISHANT SHARMA, Enrollment No. - 02696302719, Btech (Semester - 5th ) of Maharaja Surajmal Institute of Technology, New Delhi hereby declare that the project report is based upon my work carried out during the course of the summer training by Internshala. This report has not been submitted to any other institute for the award of any other degree/diploma/certificate, further certify that I have followed the guidelines provided by the college in writing the report.

**Nishant Sharma  
Enrollment No.- 02696302719  
(CSE - Evening)**

## Acknowledgement

The training opportunity I had with Internshala was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me through this internship period.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to Sarvesh Agrawal who is the Founder and CEO of Internshala and who in spite of being extraordinarily busy with her/his duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to Dr.Adeel Hashmi, HOD of CSE Department, Ms. Gunjan Beniwal, Professor for their careful and precious guidance which were extremely valuable for my study both theoretically and practically.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future,

Sincerely,

Nishant Sharma

## Table of content

I. Title Page	I
II. Certificate	II
III. Declaration	III
IV. Acknowledgement	IV
V. Table of Content	V
VI. Abstract of the Project	6
VII. Chapter 1 – Company Profile	7
VIII. Chapter 2- Technology tools studied during training	8
1. HTML	8
2. CSS	8
3. BOOTSTRAP	8
4. PHP	8
IX. Chapter 3- Demonstration of technology through project	10
1. Index.php	10
2. about.php	14
3. cart.php	17
4. header.php	21
5. footer.php	22
6. CreateDb.php	22
7. component.php	24
8. checkout.php	26
9. payscript.php	29
10. script.css	31
X. Chapter 4- Screenshot of the project / DFD / sequence diagram	32
XI. Chapter 5- Conclusion	39
XII. Bibliography or list of references	40
XIII. Appendices	41

## Abstract of Project

The project is based upon the technology related to HTML & CSS(for Front end) Bootstrap Framework (for better designing) PHP & SQL (for Back end). The project name is Lifestyle Store and as the name suggest it is basically a product buying website like Amazon, Flipkart with Payment Gateway inclusion.

The Lifestyle store consist of Five web pages texted below:-

### **1. Home Page:-**

One can buy Multiple items from the main Home Page and then by clicking on Add to cart Button the user got redirected to second page of the website which is cart page. This web page contains different virtual products that website provides.

### **2. Cart page:-**

The cart page contains every product that user added to its cart . It also has a feature of the total price counter at the top right side of the page. Below that we have a checkout button that redirects user to the checkout web page.

### **3. About:-**

The about page consists of the overview of the complete website. Moreover it provides the information regarding the usage of the website. It includes information about the developer also.

### **4. Checkout:-**

When the user buys its preferred products and presses the checkout button then the user redirects to the payment gateway section.

### **5. Payment Gateway:-**

It first take the basic information about the customer Like email, phone, name, address and when the user completely fill up the form it is ready to checkout and get the customer payment done and notifies the customer about successful receive of the payment.

## Chapter 1 – Company Profile

### INTERNSHALA:-

Internshala is an internship and online training platform, based in Gurgaon, India. Founded by Sarvesh Aggarwal an IIT Madras alumnus, in 2011, the website helps students find internships with organization in India.

It is a technology company on a mission to equip students with relevant skills & practical exposure through internships and online training. Imagine a world full of freedom and possibilities.

### Its vision:-

A world where you can discover your passion and turn it into your career. A world where your practical skills matter more than your university degree. A world where you do not have to wait till 21 to taste your first work experience (and get a rude shock that it is nothing like you had imagine it to be). A world where you graduate fully assured, fully confident, and fully prepared to stake claim on your place in the world.



Fig 1.1 Logo of the company

## Chapter 2 – Technology tools studied during training

### 1. HTML

HTML was first created by Tim Berners-Lee, Robert Cailliau, and others starting in 1989. It stands for Hyper Text Markup Language.

Hypertext means that the document contains links that allow the reader to jump to other places in the document or to another document altogether. The latest version is known as HTML5.

A Markup Language is a way that computers speak to each other to control how text is processed and presented. To do this HTML uses two things: tags and attributes.

### 2. BOOTSTRAP

Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website. It is absolutely free to download and use. It is a front-end framework used for easier and faster web development.

### 3. CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces.



Fig.2.3.1 CSS Logo

#### 4. PHP

PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP is well suited for web development. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.).



Fig.2.1 Logo for php

#### 5. SQL

SQL is a short-form of the structured query language, and it is pronounced as S-Q-L or sometimes as See-Quell.

This database language is mainly designed for maintaining the data in relational database management systems. It is a special tool used by data professionals for handling structured data (data which is stored in the form of tables).It is also designed for stream processing in RDBMS.

You can easily create and manipulate the database, access and modify the table rows and columns, etc. This query language became the standard of ANSI in the year of 1986 and ISO in the year of 1987.



Fig.2.5.2 Php My admin



Fig.2.5.3 SQL

## Chapter 3 – Demonstration of technology through project

HTML, CSS & BOOTSTRAP is used in every page of this project.

And PHP is used in order to create these web pages in the back-end.

The web pages are as follows:-

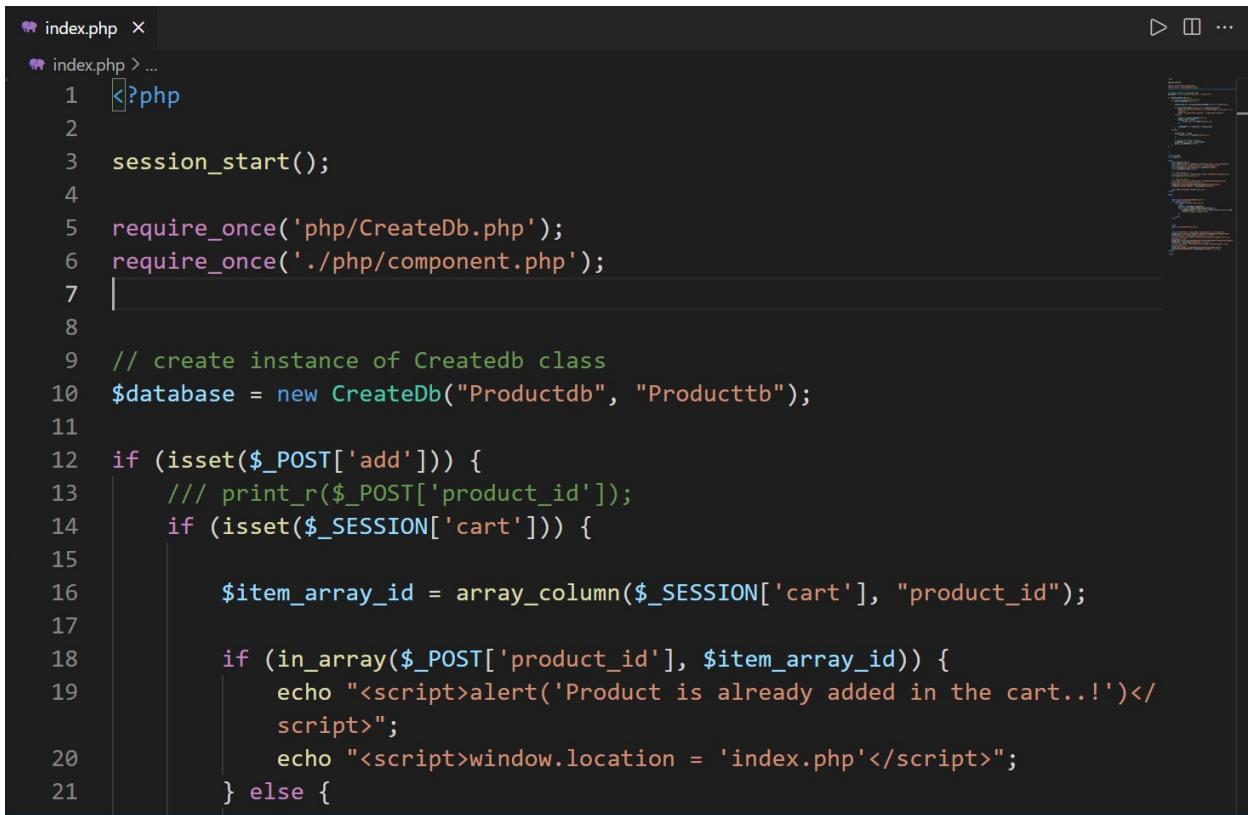
- Index.php
- about.php
- cart.php
- header.php
- footer.php
- CreateDb.php
- component.php
- checkout.php
- payscript.php
- style.css

### **1) Index.php:-**

This file contains the front-end part of the website with php script in order fetch data from the database. First I link the web page with my style.css & bootstrap and also with font awesome website which provide awesome icons that is used as a logo of this Lifestyle store project.

Then I used php script to bring header of the website. Further more php script to fetch products information from the database which is created using myphpadmin and sql.

## CODE



The screenshot shows a code editor window with the file 'index.php' open. The code is written in PHP and performs several tasks:

- It starts with the PHP opening tag `<?php`.
- It initializes a session with `session\_start();`.
- It includes 'CreateDb.php' with `require\_once('php/CreateDb.php');` and 'component.php' with `require\_once('./php/component.php');`.
- It creates an instance of the 'Createdb' class with `\$database = new CreateDb("Productdb", "Producttb");`.
- It checks if the 'add' button was clicked via `\$\_POST['add']`. If true, it prints the product ID from the POST data with `print\_r(\$\_POST['product\_id']);` and checks if the product is already in the user's cart session.
- If the product is already in the cart, it displays a JavaScript alert message: `echo "<script>alert('Product is already added in the cart..!')</script>";` and reloads the page with `echo "<script>window.location = 'index.php'</script>";`.
- If the product is not in the cart, it adds it to the session array.

Fig. 3.1.1 Code for index page.

Fig.3.1.1 image shows php code which first check for createdb.php and component.php in order to use them in next part of the code. Createdb.php helps to create a database with name Productdb and also creates a table with name Producttb.

Then the if statement after creation of database check for the click by the user on add to cart button then code fetches the values of the product from the product table in Productdb Database

If the same product is added twice then the user will get pop message that Product is already in the cart using some javascript.

Component.php helps to create different products using single function by getting information from the database.

```
index.php X
index.php > ...
21     } else {
22
23         $count = count($_SESSION['cart']);
24         $item_array = array(
25             'product_id' => $_POST['product_id']
26         );
27
28         $_SESSION['cart'][$count] = $item_array;
29     }
30 } else {
31
32     $item_array = array(
33         'product_id' => $_POST['product_id']
34     );
35
36     // Create new session variable
37     $_SESSION['cart'][0] = $item_array;
38     print_r($_SESSION['cart']);
39 }
40 }
41
42
```

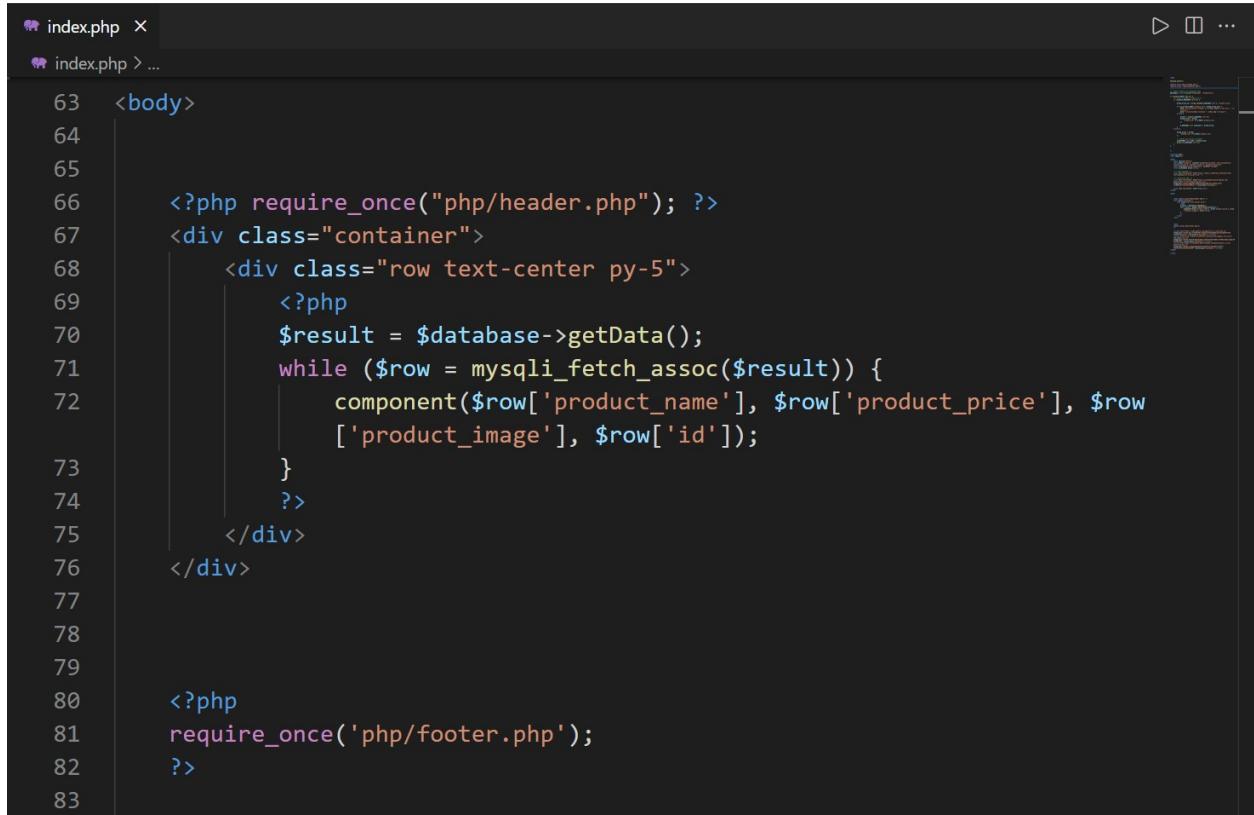
Fig.3.1.2 counting items in cart

Here count variable is created in order to get number of items inside the cart and item\_array variable is an array which is used to store product elements using it's ID.

```
index.php X
index.php > ...
45 <!doctype html>
46 <html lang="en">
47
48 <head>
49     <meta charset="UTF-8">
50     <meta name="viewport" content="width=device-width, user-scalable=no,
51     initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
52     <meta http-equiv="X-UA-Compatible" content="ie=edge">
53     <title>Lifestyle Store</title>
54
55     <!-- Font Awesome -->
56     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/
57     font-awesome/5.8.2/css/all.css" />
58
59     <!-- Bootstrap CDN -->
60     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
61     bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-gg0yR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQU0hcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
62
63     <link rel="stylesheet" href="style.css">
64 </head>
```

Fig.3.1.3 HTML code inclusion of bootstrap and font-awesome

Fig 3.1.3 shows the inclusion of Bootstrap style-sheet link and font-awesome style-sheet link inside the head of the HTML.



The screenshot shows a code editor window with a dark theme. The file is named 'index.php'. The code is a PHP script. It includes a header.php file at line 66. A container div is defined at line 67. Inside the container, a row div with text-center and py-5 classes is shown at line 68. A PHP block starts at line 69, which includes a database call to get data. A while loop begins at line 70 to fetch rows from the database. Inside the loop, component functions are called with parameters from the \$row array. The loop ends at line 73. The PHP block ends at line 74. The container div ends at line 75. The PHP block starts again at line 80, requiring footer.php. The PHP block ends at line 82. The entire code is numbered from 63 to 83.

```
63 <body>
64
65
66     <?php require_once("php/header.php"); ?>
67     <div class="container">
68         <div class="row text-center py-5">
69             <?php
70                 $result = $database->getData();
71                 while ($row = mysqli_fetch_assoc($result)) {
72                     component($row['product_name'], $row['product_price'], $row
73                         ['product_image'], $row['id']);
74                 }
75             </div>
76         </div>
77
78
79
80     <?php
81         require_once('php/footer.php');
82     ?>
83
```

Fig. 3.1.4 Fetching Data from Database

Fig. 3.1.4 show the code where header.php is included as it is used to get the navigation bar. And result variable stores the data from the database. Then there is a while loop to get the row from the product table inside the product database. That row contains product name , product price and product image and product id. After that there is inclusion of footer.php which provides footer of the webpage.

Fig. 3.1.5 This is the end part of the index.php page includes the jquery link and stackpath link which is necessary component of the bootstrap . These links helps bootstrap to work properly.

```
index.php > ...
81     require_once('php/footer.php');
82     ?>
83
84     <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
85             integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH
86             +8abtTE1Pi6jizo" crossorigin="anonymous"></script>
87     <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/
88             umd/popper.min.js"
89             integrity="sha384-sh3gD7c9hZUdjXuost9xDCoOmxWe江山
90             6dIHNDz0W1" crossorigin="anonymous"></script>
91     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/
92             bootstrap.min.js"
93             integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/
94             nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
95
96     </body>
97
98     </html>
```

Fig. 3.1.5 Inclusion of jQuery and stackpath For Bootstrap

## 2) About.php:-

The about page consists of the overview of the complete website. Moreover it provides the information regarding the usage of the website. It includes information about the developer also.

## CODE

```
about.php > ...
1  <!doctype html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport"
6          content="width=device-width, user-scalable=no, initial-scale=1.0,
7          maximum-scale=1.0, minimum-scale=1.0">
8      <meta http-equiv="X-UA-Compatible" content="ie=edge">
9      <title>About</title>
10
11      <!-- Font Awesome -->
12      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/
13          font-awesome/5.8.2/css/all.css" />
14
15      <!-- Bootstrap CDN -->
16      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.
17          3.1/css/bootstrap.min.css" integrity="sha384-gg0yR0iXCbMQv3Xipma34MD+dH/
18          1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
19
20      <link rel="stylesheet" href="style.css">
21  </head>
22  <body>
```

Fig.3.2.1 HTML code

Fig. 3.2.1 shows the html code having about as the title of web page. And includes font-awesome style-sheet link and bootstrap style-sheet link in the head of the web page.

It also contains the meta tags:-

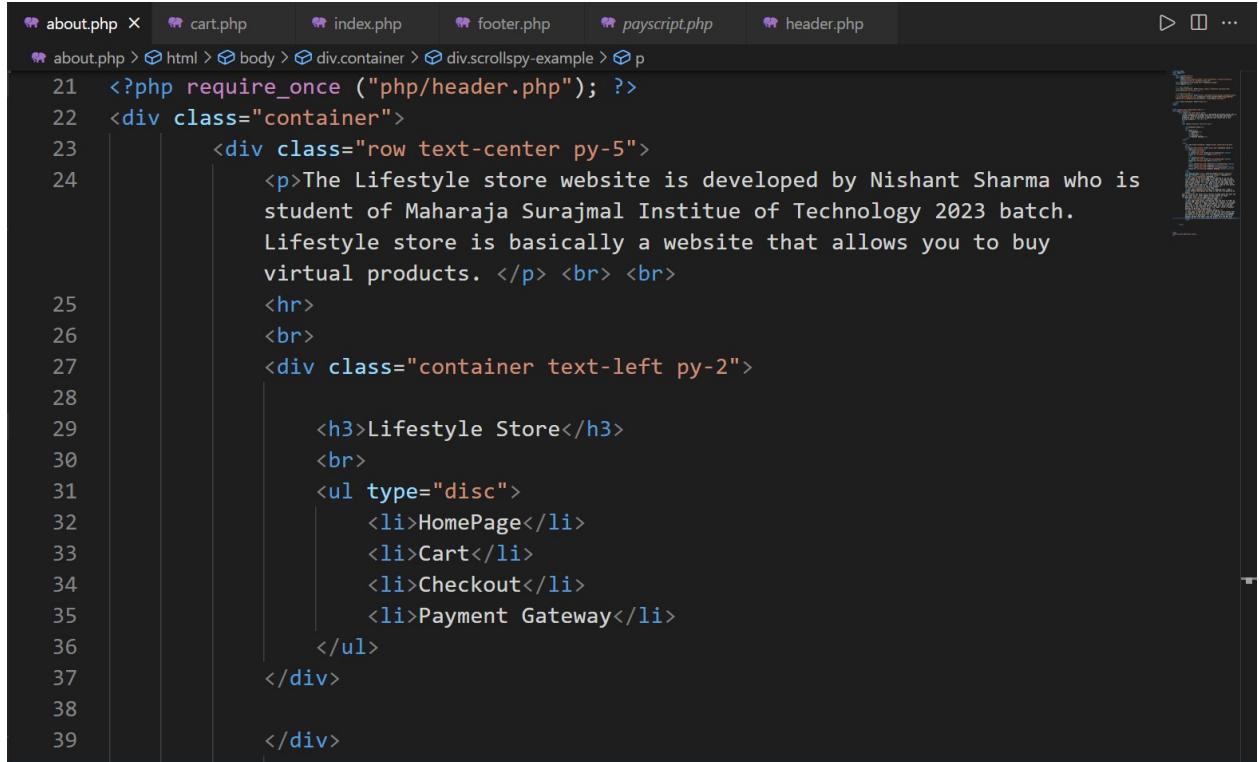
## META TAGS:-

The <meta> tag defines metadata about an HTML document. Metadata is data (information) about data.<meta>

<meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and view port settings.

Metadata will not be displayed on the page, but is machine parsable.

Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

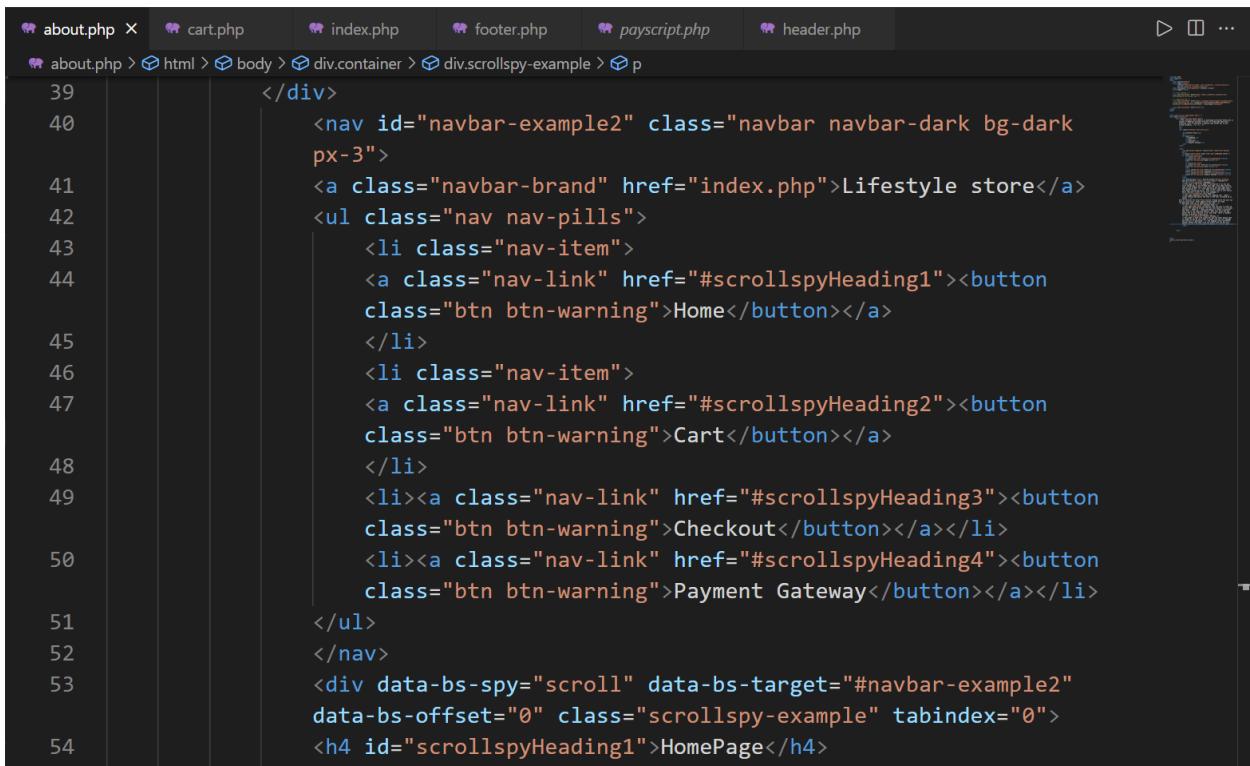


The screenshot shows a code editor with the file 'about.php' open. The code is as follows:

```
about.php X cart.php index.php footer.php payscript.php header.php
about.php > html > body > div.container > div.scrollspy-example > p
21 <?php require_once ("php/header.php"); ?>
22 <div class="container">
23   <div class="row text-center py-5">
24     <p>The Lifestyle store website is developed by Nishant Sharma who is
        student of Maharaja Surajmal Institute of Technology 2023 batch.
        Lifestyle store is basically a website that allows you to buy
        virtual products. </p> <br> <br>
25     <hr>
26     <br>
27     <div class="container text-left py-2">
28
29       <h3>Lifestyle Store</h3>
30       <br>
31       <ul type="disc">
32         <li>HomePage</li>
33         <li>Cart</li>
34         <li>Checkout</li>
35         <li>Payment Gateway</li>
36       </ul>
37     </div>
38
39   </div>
```

Fig. 3.2.2 Includes header with a new container

Fig.3.2.2 shows the php script to include header from header.php in order to get the navigation bar and then there is a bootstrap container inside of which there is a paragraph tag in order to write some information about the developer and then another container which handles a list of web pages names and then div tag closes.



The screenshot shows a code editor with several tabs at the top: about.php, cart.php, index.php, footer.php, payscript.php, and header.php. The active tab is about.php. The code editor displays the following HTML and CSS code:

```
about.php X cart.php index.php footer.php payscript.php header.php
about.php > html > body > div.container > div.scrollspy-example > p
39     
```

Fig.3.2.3 Navbar2 to create a scroll spy using Bootstrap

### The Scroll Spy:-

Automatically update Bootstrap navigation or list group components based on scroll position to indicate which link is currently active in the view port.

When successfully implemented, your navigation or list group will update accordingly, moving the .active class from one item to the next based on their associated targets.

Fig.3.2.3 shows the inclusion of bootstrap scroll spy which helps to create more dynamic pages and increase the overall experience of the about web page.

The scroll spy contains different scroll spy headings in order to make user to scroll some amount of distance when the user clicks on that button.

It must be used on a Bootstrap nav component or list group.

- Scroll spy requires position; relative; on the element you're spying on, usually the <body>.

- Anchors (<a>) are required and must point to an element with that id.

```
about.php X cart.php index.php footer.php payscript.php header.php
about.php > html > body > div.container > div.scrollspy-example > p
55      <p>This page is the first page of the website or we can say
first impression is one through this page. This page contains
logo of the company and a cart button on the top right part of
the navigation bar. <br> This page contains some dummy items
with their rating and price . Also it has a add to cart button
which redorects client to the cart webpage</p>
56      <h4 id="scrollspyHeading2">Cart Page</h4>
57      <p>This page implements the logic of shopping cart . When a
client presses add option the item in the cart increased by an
amount . <br>
58      Also it contains the Total Price section through which the user can
see its final price after adding certain amount of items.
<br>NOTE:Each item can be added only one time. </p>
59      <h4 id="scrollspyHeading3">Checkout Page</h4>
60      <p>This page takes basic information from the user in order to
receive the required product from the this website. It includes
the Email , Phone no. , Name and Address in order to deliever
that item to the specified address. When user press Pay Now
button the it get redirect to our next page which is payment
gateway by Razorpay organisation. </p>
61      <h4 id="scrollspyHeading4">Payment Gateway</h4>
62      <p>This page is made from javascript and the total amount that
```

Fig.3.2.4 Including Different parts of Scroll Spy

Fig.3.2.4 this figure shows different scroll spy headings like checkout page and Payment gateway page with their description in paragraph tag.

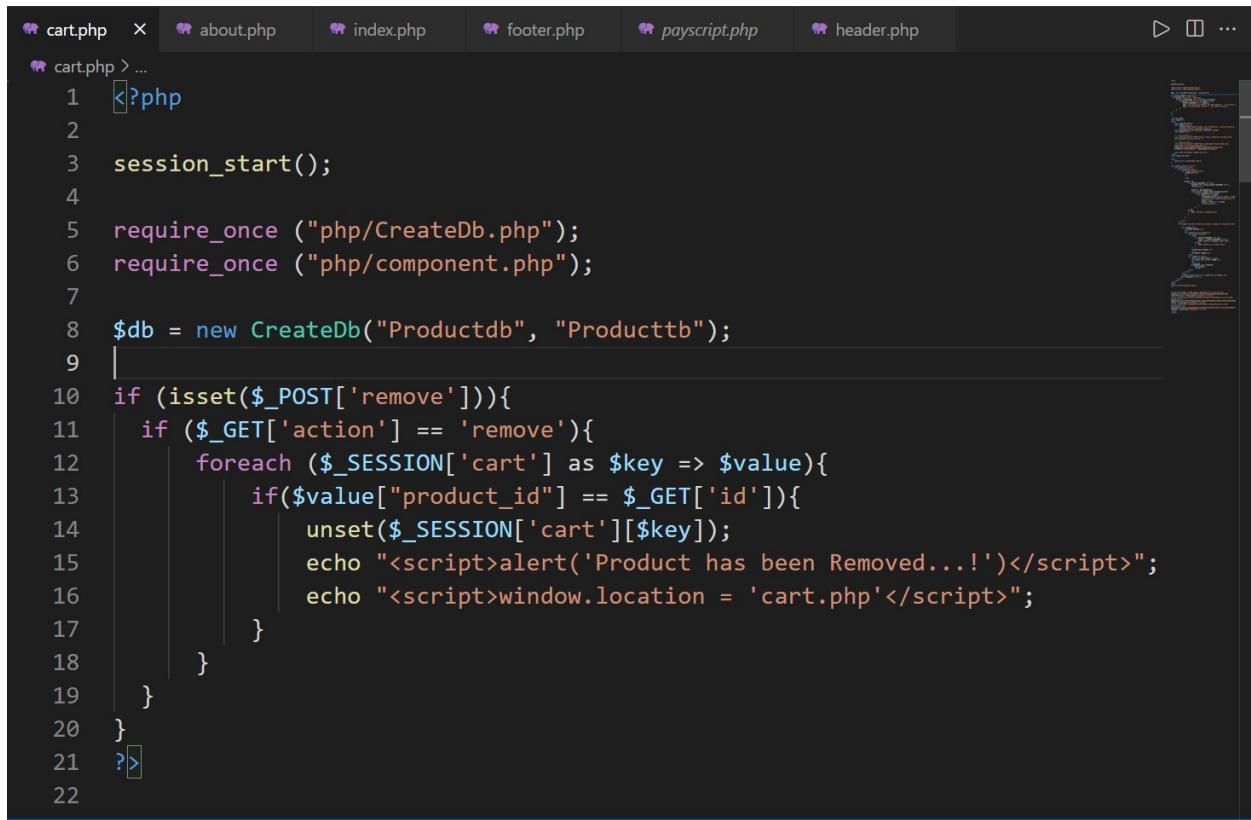
<h4> tag is used to increase the size of the main title of the scroll spy headings which makes it more stylish and presentable.

### 3) Cart.php:-

This page implements the logic of shopping cart. When a client presses add option the item in the cart increased by an amount.

Also it contains the Total Price section through which the user can see its final price after adding certain amount of items.

## CODE



```
cart.php > ...
1 <?php
2
3 session_start();
4
5 require_once ("php/CreateDb.php");
6 require_once ("php/component.php");
7
8 $db = new CreateDb("Productdb", "Producttb");
9
10 if (isset($_POST['remove'])){
11     if ($_GET['action'] == 'remove'){
12         foreach ($_SESSION['cart'] as $key => $value){
13             if($value["product_id"] == $_GET['id']){
14                 unset($_SESSION['cart'][$key]);
15                 echo "<script>alert('Product has been Removed...!')</script>";
16                 echo "<script>window.location = 'cart.php'</script>";
17             }
18         }
19     }
20 }
21 ?>
22
```

Fig.3.3.1.1 Remove button code

In Fig.3.3.1, the remove button inside the cart web page is implemented using php script. First we check for the user has clicked on remove option or not by isset function and then if the remove button is clicked the code between the if statement starts executing. Foreach loop is used in order to get different id from the product table of the productdb database and each id is compared with the get action id and when it is matched the unset function is executed by passing the cart and the id of the product in it,

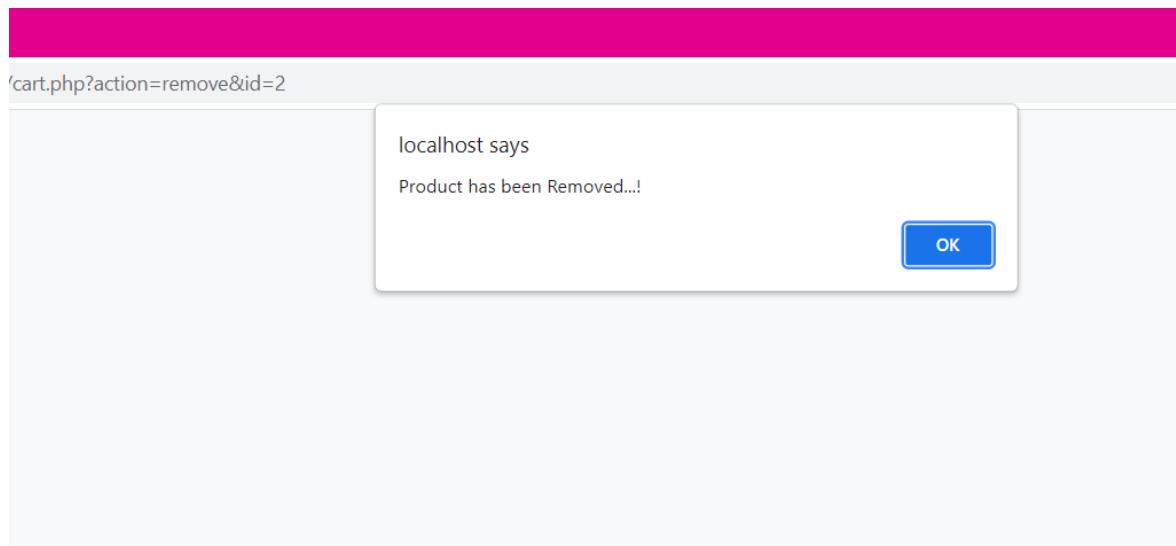


Fig3.3.1.2 Removing Element

```
cart.php < cart.php > ...
23  <!doctype html>
24  <html lang="en">
25  <head>
26      <meta charset="UTF-8">
27      <meta name="viewport"
28          |     content="width=device-width, user-scalable=no, initial-scale=1.0,
29          |     maximum-scale=1.0, minimum-scale=1.0">
30      <meta http-equiv="X-UA-Compatible" content="ie=edge">
31      <title>Cart</title>
32
33      <!-- Font Awesome -->
34      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/
35          |     font-awesome/5.8.2/css/all.css" />
36
37      <!-- Bootstrap CDN -->
38      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
39          |     bootstrap/4.3.1/css/bootstrap.min.css"
          |     integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
          |     iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
40
41      <link rel="stylesheet" href="style.css">
42  </head>
```

Fig.3.3.2 HTML code with Bootstrap and font-awesome links

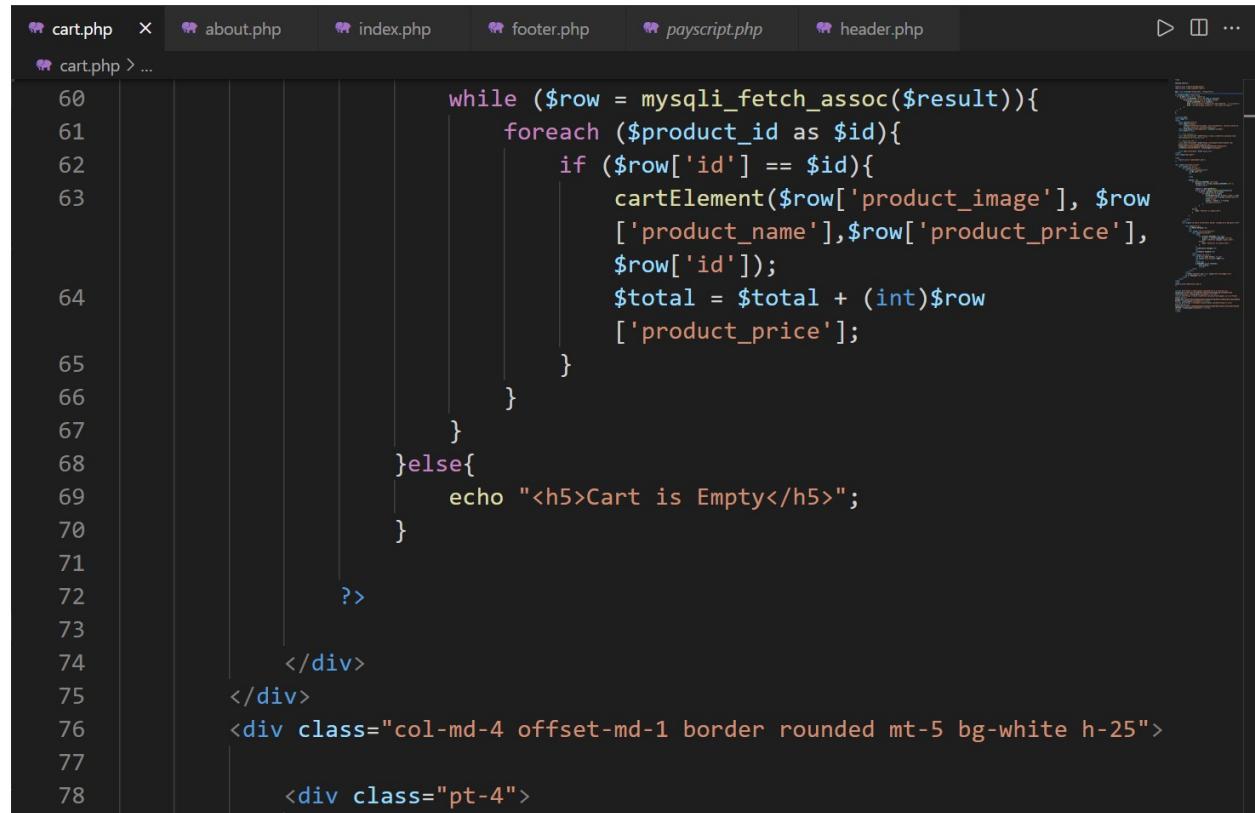
The basic HTML and bootstrap links are included in the code along with font-awesome style-sheet link as shown in Fig.3.3.2.

```
cart.php < cart.php > ...
40  <body class="bg-light">
41
42  <?php
43  |     require_once ('php/header.php');
44  ?>
45
46  <div class="container-fluid">
47      <div class="row px-5">
48          <div class="col-md-7">
49              <div class="shopping-cart">
50                  <h6>My Cart</h6>
51                  <hr>
52
53                  <?php
54
55                      $total = 0;
56                      if (isset($_SESSION['cart'])){
57                          $product_id = array_column($_SESSION['cart'],
58                          'product_id');
59
60                          $result = $db->getData();
61                          while ($row = mysqli_fetch_assoc($result)){
```

Fig.3.3.3 Total price calculation

The total price is calculated in Fig.3.3.3 by checking price from the database using getData() function and is

stored inside result variable.

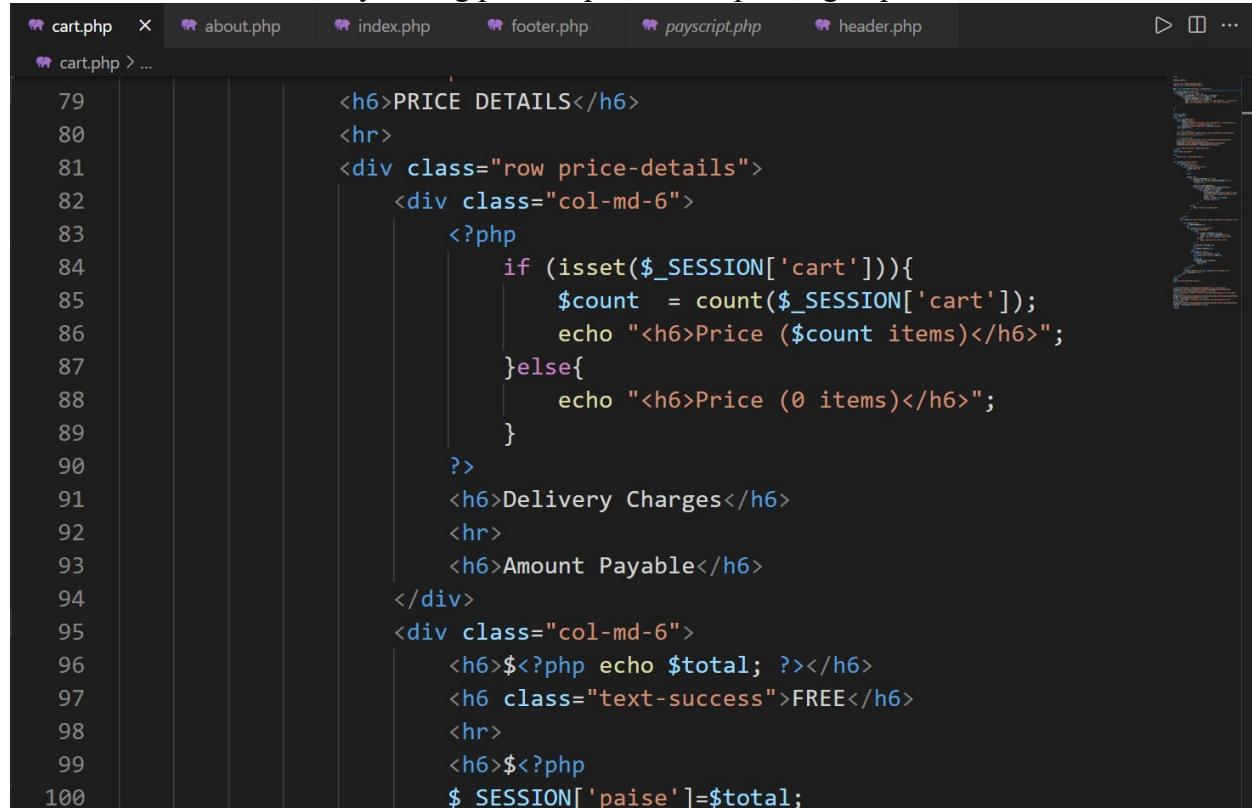


```
cart.php X about.php index.php footer.php payscript.php header.php
cart.php > ...
60     while ($row = mysqli_fetch_assoc($result)){
61         foreach ($product_id as $id){
62             if ($row['id'] == $id){
63                 cartElement($row['product_image'], $row
64                     ['product_name'],$row['product_price'],
65                     $row['id']);
66                 $total = $total + (int)$row
67                     ['product_price'];
68             }
69         }else{
70             echo "<h5>Cart is Empty</h5>";
71         }
72     ?>
73     </div>
74     </div>
75     <div class="col-md-4 offset-md-1 border rounded mt-5 bg-white h-25">
76         <div class="pt-4">
```

Fig.3.3.4 While loop is used to calculate total price

The row variable inside the while loop stores the entire row from the `mysqli_fetch_assoc` function.

Then the row id is compared with each and every product id present in the database and if the ID's matches then the total is calculated by adding product price corresponding to product id.

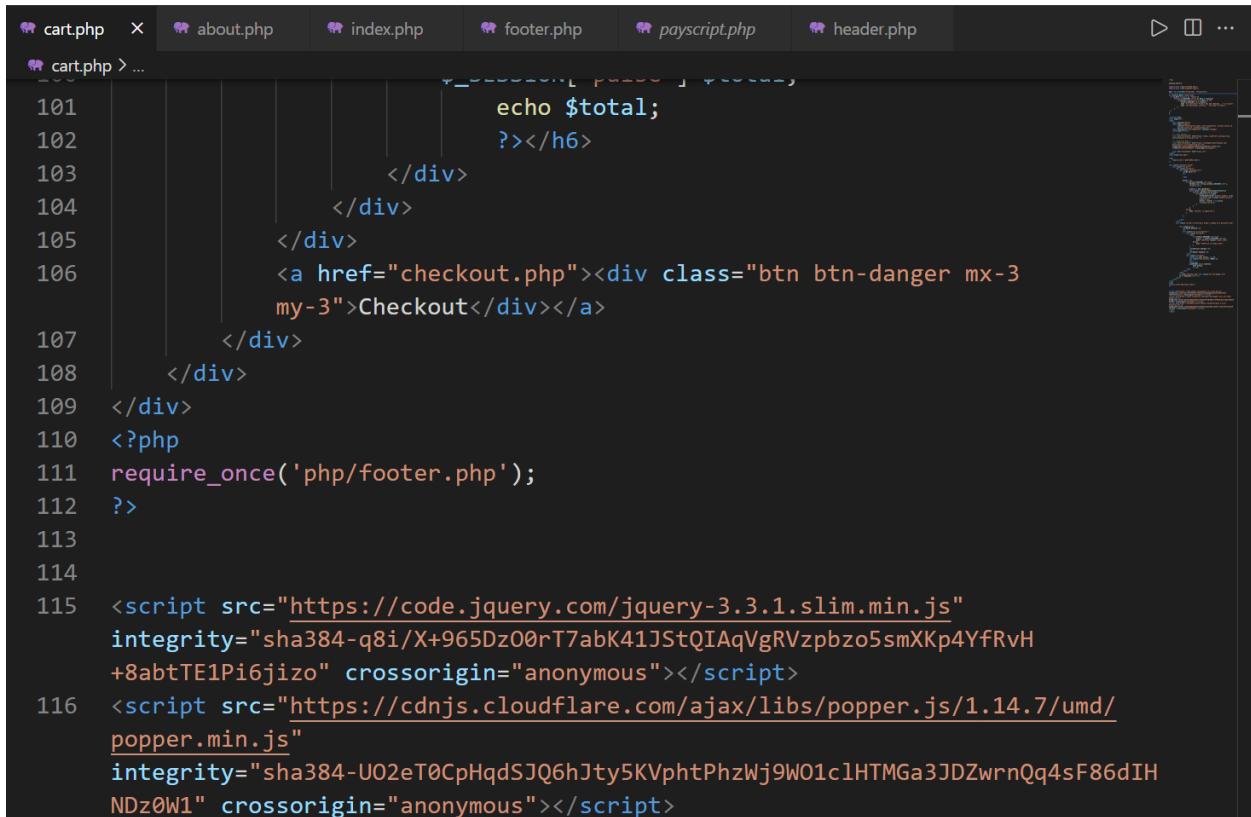


```
cart.php X about.php index.php footer.php payscript.php header.php
cart.php > ...
79     <h6>PRICE DETAILS</h6>
80     <hr>
81     <div class="row price-details">
82         <div class="col-md-6">
83             <?php
84                 if (isset($_SESSION['cart'])){
85                     $count = count($_SESSION['cart']);
86                     echo "<h6>Price ($count items)</h6>";
87                 }else{
88                     echo "<h6>Price (0 items)</h6>";
89                 }
90             ?>
91             <h6>Delivery Charges</h6>
92             <hr>
93             <h6>Amount Payable</h6>
94         </div>
95         <div class="col-md-6">
96             <h6>$<?php echo $total; ?></h6>
97             <h6 class="text-success">FREE</h6>
98             <hr>
99             <h6>$<?php
100                $_SESSION['paise']=$total;
```

Fig.3.3.5 counting the number of items inside the cart

The total number of items present inside the cart is counted and stored inside the count variable.

And total variable is displayed inside the column created using bootstrap (col-md-6), the delivery charges is made free for the customers. A new variable paise is created in order to store total price so it can be transfer to payment gateway where the client needs to make his/her payment.



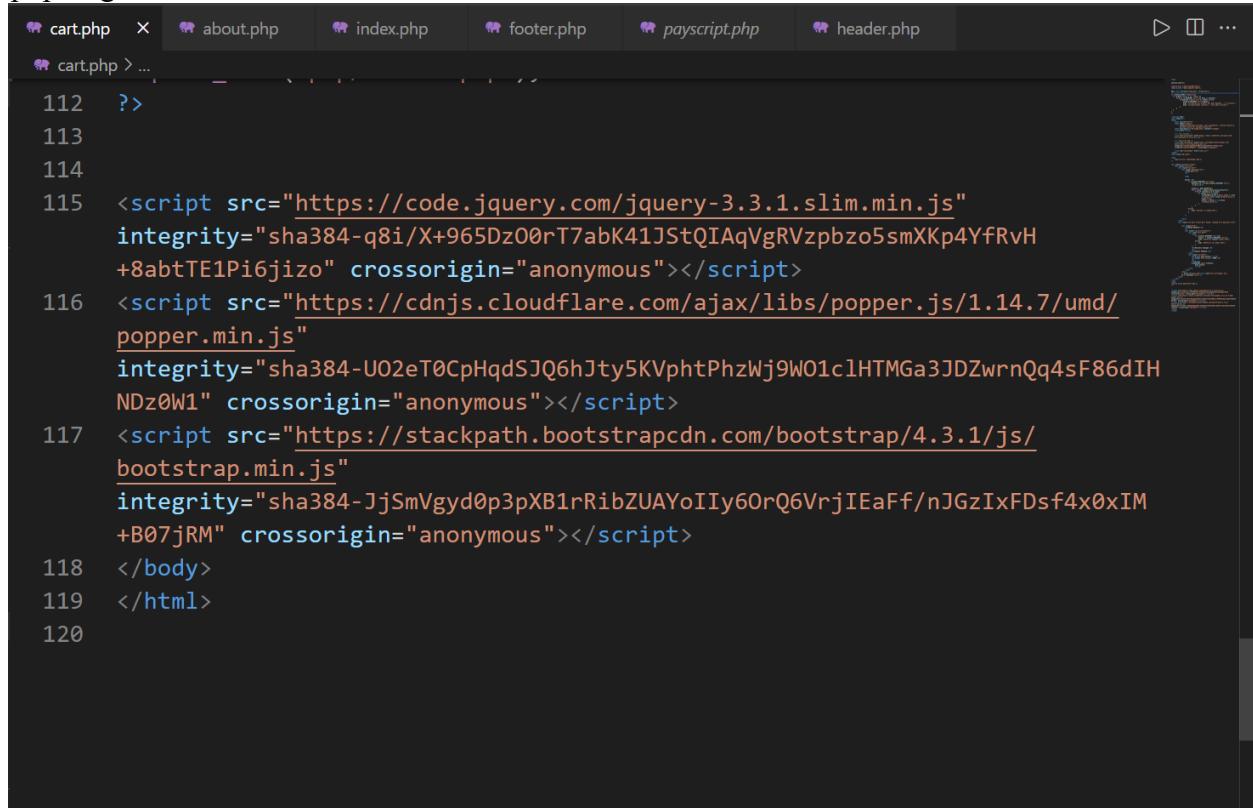
```

101     echo $total;
102     ?></h6>
103     </div>
104     </div>
105     <a href="checkout.php"><div class="btn btn-danger mx-3
106 my-3">Checkout</div></a>
107     </div>
108   </div>
109 </div>
110 <?php
111 require_once('php/footer.php');
112 ?>
113
114
115 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
116 integrity="sha384-q8i/X+965Dz0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH
+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
117 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/
popper.min.js"
118 integrity="sha384-U02eT0CpHqdSJQ6hJty5KVphtPhzWj9W01c1HTMGa3JDZwrnQq4sF86dIH
NDz0W1" crossorigin="anonymous"></script>

```

Fig.3.3.6 Checkout Button inside cart

In Fig.3.3.6 the checkout button is there in order to redirect the client to checkout page. Then there is footer.php to get footer for the website.



```

112 ?>
113
114
115 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
116 integrity="sha384-q8i/X+965Dz0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH
+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
117 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/
popper.min.js"
118 integrity="sha384-U02eT0CpHqdSJQ6hJty5KVphtPhzWj9W01c1HTMGa3JDZwrnQq4sF86dIH
NDz0W1" crossorigin="anonymous"></script>
119 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/
bootstrap.min.js"
120 integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFF/nJGzIxFDsf4x0xIM
+B07jRM" crossorigin="anonymous"></script>
121 </body>
122 </html>

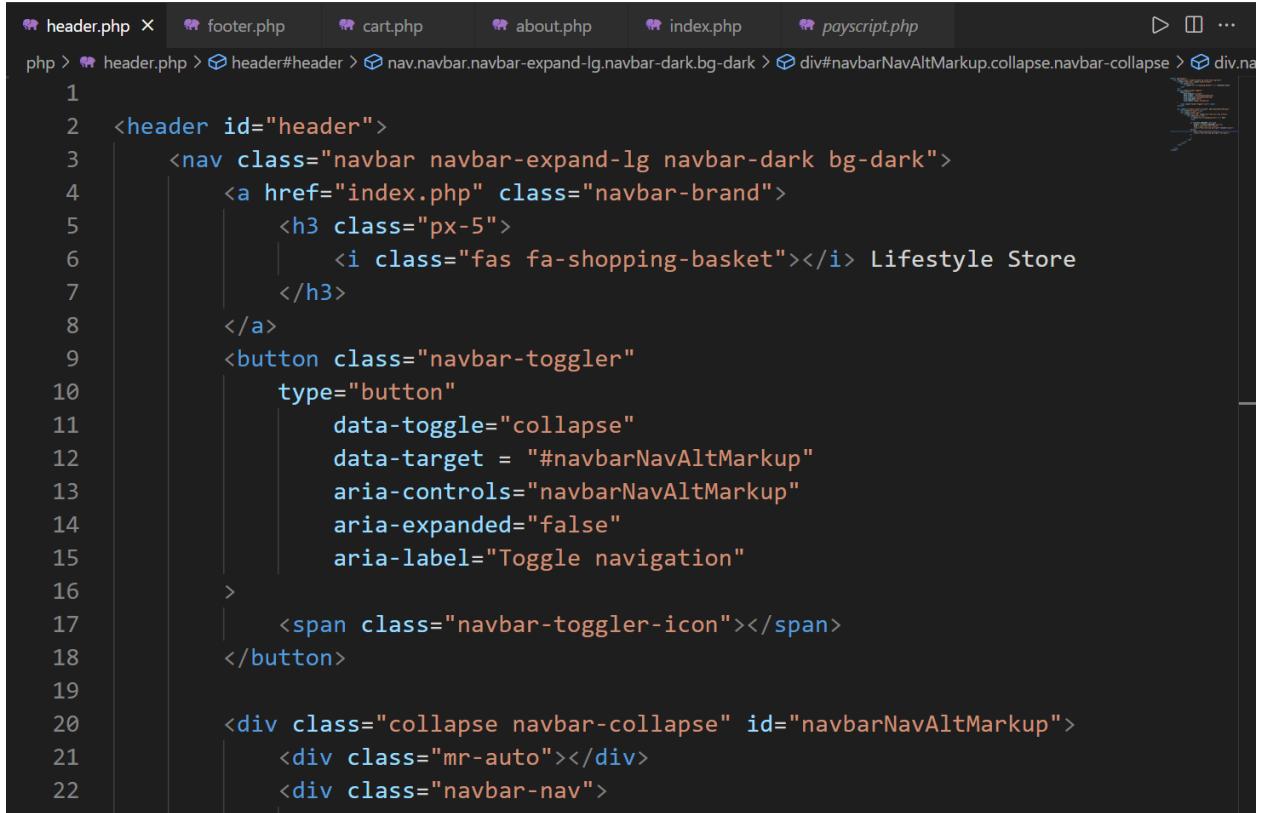
```

Fig.3.3.7 jQuery ,js & stackpath links for bootstrap

Script for jQuery and javascript is included for the proper working of Bootstrap.

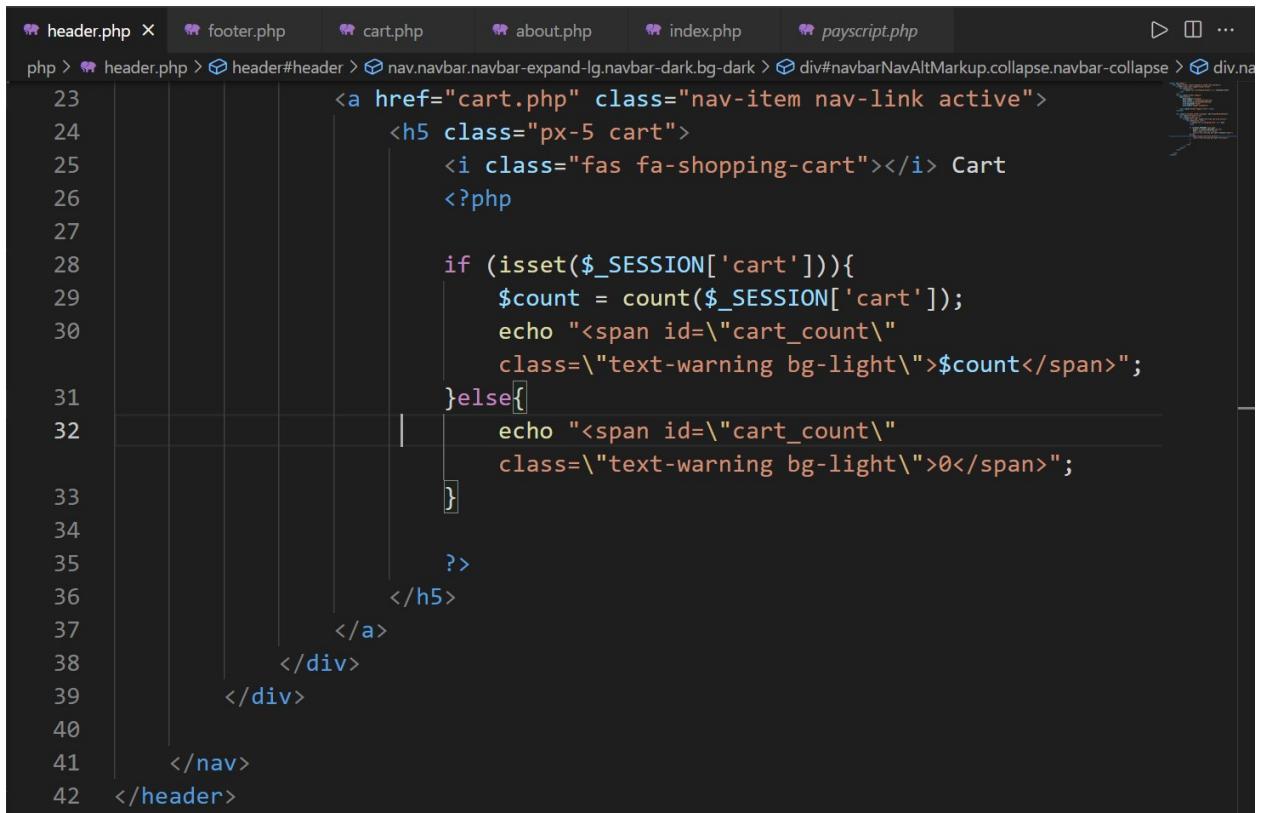
#### 4) header.php:-

This page is used in order to get header of the website inside every web page which include it.



```
header.php X footer.php cart.php about.php index.php payscript.php ...  
php > header.php > header#header > nav.navbar.navbar-expand-lg.navbar-dark.bg-dark > div#navbarNavAltMarkup.collapse.navbar-collapse > div.navbar-nav  
1 <header id="header">  
2     <nav class="navbar navbar-expand-lg navbar-dark bg-dark">  
3         <a href="index.php" class="navbar-brand">  
4             <h3 class="px-5">  
5                 <i class="fas fa-shopping-basket"></i> Lifestyle Store  
6             </h3>  
7         </a>  
8         <button class="navbar-toggler" type="button"  
9             data-toggle="collapse"  
10            data-target = "#navbarNavAltMarkup"  
11            aria-controls="navbarNavAltMarkup"  
12            aria-expanded="false"  
13            aria-label="Toggle navigation"  
14         >  
15             <span class="navbar-toggler-icon"></span>  
16         </button>  
17  
18         <div class="collapse navbar-collapse" id="navbarNavAltMarkup">  
19             <div class="mr-auto"></div>  
20             <div class="navbar-nav">
```

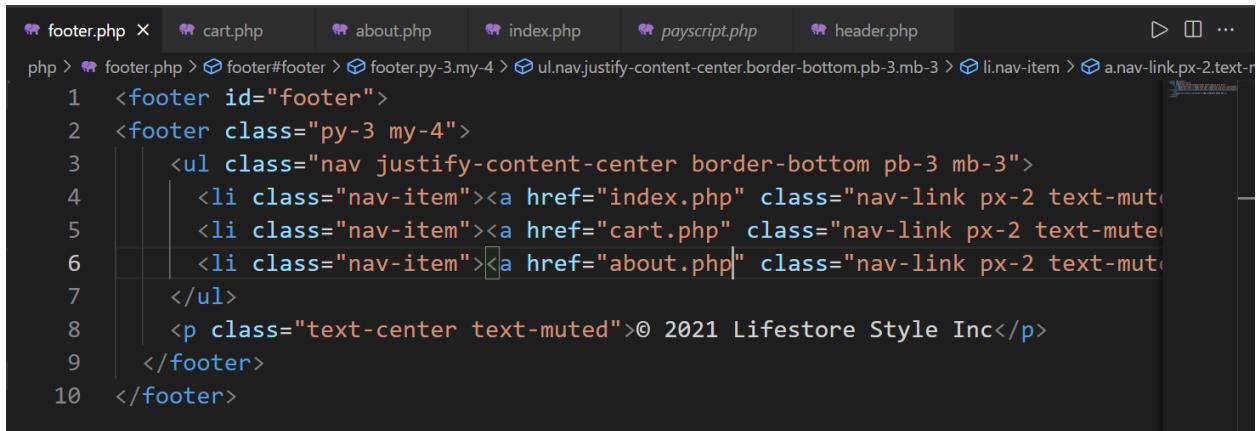
Fig.3.4.1 Header.php included navigation bar



```
header.php X footer.php cart.php about.php index.php payscript.php ...  
php > header.php > header#header > nav.navbar.navbar-expand-lg.navbar-dark.bg-dark > div#navbarNavAltMarkup.collapse.navbar-collapse > div.navbar-nav  
23     <a href="cart.php" class="nav-item nav-link active">  
24         <h5 class="px-5 cart">  
25             <i class="fas fa-shopping-cart"></i> Cart  
26             <?php  
27  
28             if (isset($_SESSION['cart'])){  
29                 $count = count($_SESSION['cart']);  
30                 echo "<span id=\"cart_count\" class=\"text-warning bg-light\">$count</span>";  
31             }else{  
32                 echo "<span id=\"cart_count\" class=\"text-warning bg-light\">0</span>";  
33             }  
34  
35             ?>  
36         </h5>  
37     </a>  
38 </div>  
39 </div>  
40 </nav>  
41 </header>
```

Fig.3.4.2 Code for displaying number of items inside cart

## 5) footer.php:-



```
1 <footer id="footer">
2 <footer class="py-3 my-4">
3   <ul class="nav justify-content-center border-bottom pb-3 mb-3">
4     <li class="nav-item"><a href="index.php" class="nav-link px-2 text-muted">Home</a>
5     <li class="nav-item"><a href="cart.php" class="nav-link px-2 text-muted">Cart</a>
6     <li class="nav-item"><a href="about.php" class="nav-link px-2 text-muted">About</a>
7   </ul>
8   <p class="text-center text-muted">© 2021 Lifestore Style Inc</p>
9 </footer>
10 </footer>
```

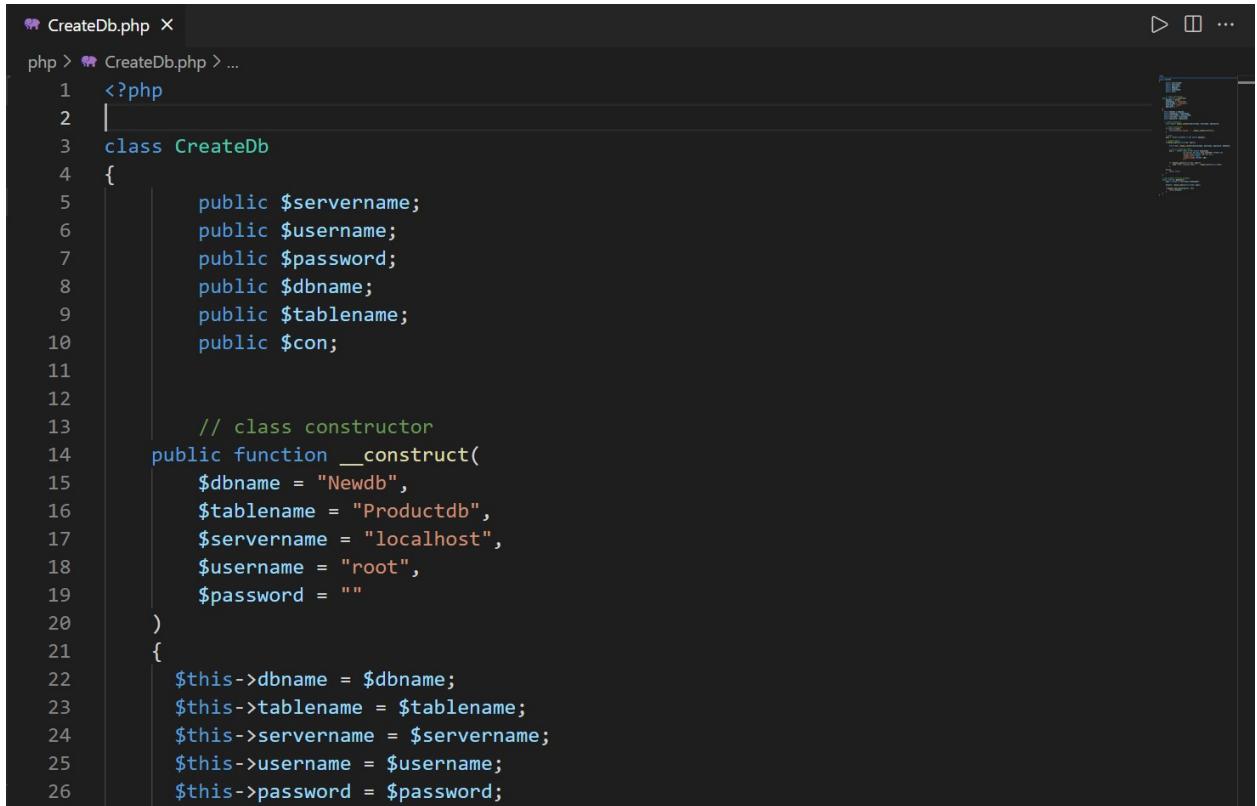
Fig. 3.5.1 Footer.php complete code

Footer code as shown in Fig. 3.5.1 includes hyperlink to every part of the website including cart ,homepage and about section of the Lifestyle store project.

## 6) createDb.php:-

The createDb is coded in order to create products database along with product table.

Also it ensures about the connectivity between database. \$con variable is used to check the connection of the database.

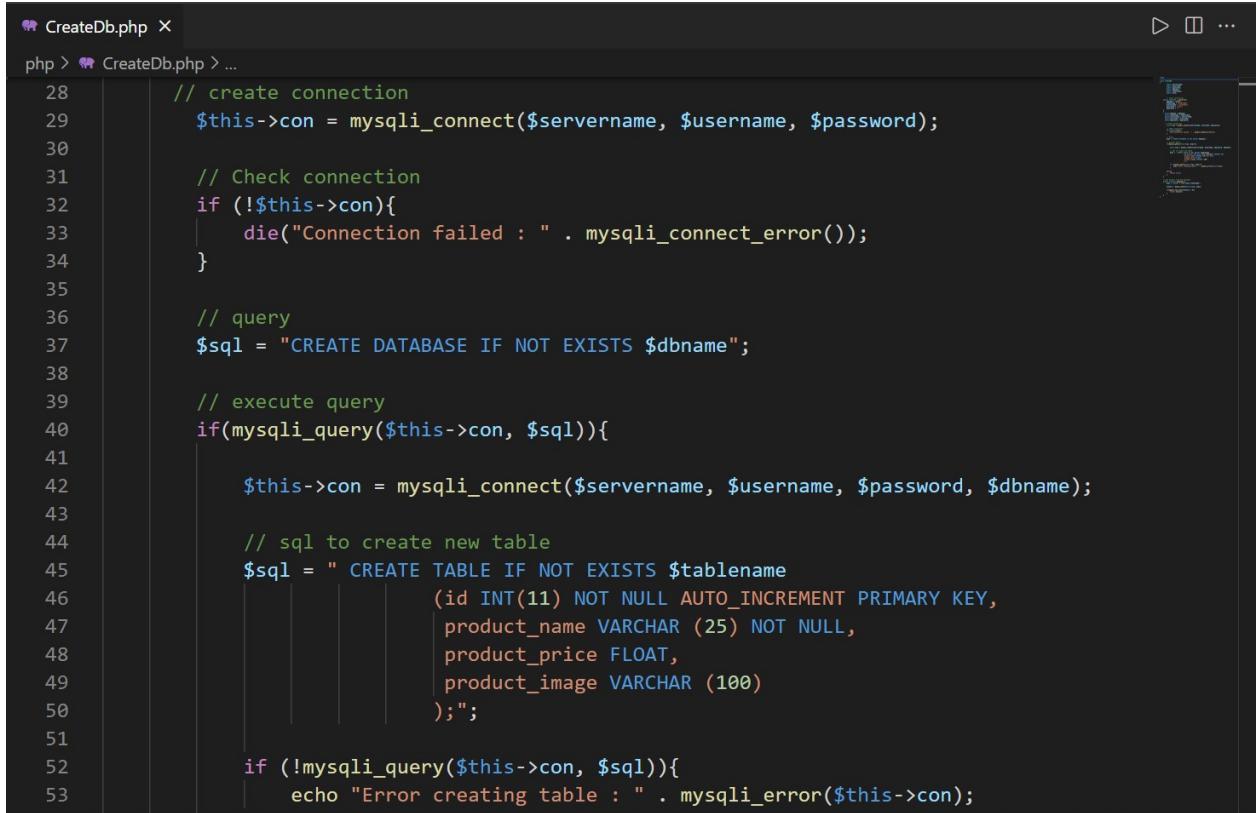


```
1 <?php
2 |
3 class CreateDb
4 {
5   public $servername;
6   public $username;
7   public $password;
8   public $dbname;
9   public $tablename;
10  public $con;
11
12
13  // class constructor
14  public function __construct(
15    $dbname = "Newdb",
16    $tablename = "Producttbl",
17    $servername = "localhost",
18    $username = "root",
19    $password = ""
20  )
21  {
22    $this->dbname = $dbname;
23    $this->tablename = $tablename;
24    $this->servername = $servername;
25    $this->username = $username;
26    $this->password = $password;
```

Fig.3.6.1 Declaration of different variables

As shown in Fig.3.6.1 , there are different variables like username , password, dbname(to store the name of the database) , tablename, con.

These variables are then initialized with different values like products database name here is Newdb, tablename is Productdb,etc.



```
⑥ CreateDb.php ×
php > ⑥ CreateDb.php > ...
28     // create connection
29     $this->con = mysqli_connect($servername, $username, $password);
30
31     // Check connection
32     if (!$this->con){
33         die("Connection failed : " . mysqli_connect_error());
34     }
35
36     // query
37     $sql = "CREATE DATABASE IF NOT EXISTS $dbname";
38
39     // execute query
40     if(mysqli_query($this->con, $sql)){
41
42         $this->con = mysqli_connect($servername, $username, $password, $dbname);
43
44         // sql to create new table
45         $sql = " CREATE TABLE IF NOT EXISTS $tablename
46                         (id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
47                          product_name VARCHAR (25) NOT NULL,
48                          product_price FLOAT,
49                          product_image VARCHAR (100)
50                     );";
51
52         if (!mysqli_query($this->con, $sql)){
53             echo "Error creating table : " . mysqli_error($this->con);
54         }
55     }
56 }
```

Fig.3.6.2 Creating Table & checking for connection

The connection checking is done through con variable.

The sql variable is used to store the query that is needed to create the table inside the database. And if the sql query is not able to execute properly the if statement is given in order to print the type of error query is interrupting.

mysql\_query() sends a unique query (multiple queries are not supported) to the currently active database on the server that's associated with the specified link\_identifier.

## Parameters

a) Query - (An SQL query)

The query string should not end with a semicolon. Data inside the query should be properly escaped.

b) Link\_identifier

The MySQL connection. If the link identifier is not specified, the last link opened by mysql\_connect() is assumed. If no such link is found, it will try to create one as if mysql\_connect() had been called with no arguments. If no connection is found or established, an E\_Warning level error is generated.

```

46     (id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
47      product_name VARCHAR (25) NOT NULL,
48      product_price FLOAT,
49      product_image VARCHAR (100)
50      );";
51
52     if (!mysqli_query($this->con, $sql)){
53         echo "Error creating table : " . mysqli_error($this->con);
54     }
55
56 }else{
57     return false;
58 }
59 }
60
61 // get product from the database
62 public function getData(){
63     $sql = "SELECT * FROM $this->tablename";
64
65     $result = mysqli_query($this->con, $sql);
66
67     if(mysqli_num_rows($result) > 0){
68         return $result;
69     }
70 }
71 }
72

```

Fig. 3.6.3 getData function for receiving data from the table

Here getData function declaration and implemented is done in order to use same sql query to retrieve data form the table. The code of which is provided in Fig. 3.6.3.

## 7) component.php:-

```

1  <?php
2
3  function component($productname, $productprice, $productimg, $productid){
4      $element = "
5
6      <div class=\"col-md-3 col-sm-6 my-3 my-md-0\">
7          <form action=\"index.php\" method=\"post\">
8              <div class=\"card shadow\">
9                  <div>
10                     <img src=\"$productimg\" alt=\"Image1\" class=\"img-fluid
11                         card-img-top\">
12                 </div>
13                 <div class=\"card-body\">
14                     <h5 class=\"card-title\">$productname</h5>
15                     <h6>
16                         <i class=\"fas fa-star\"></i>
17                         <i class=\"fas fa-star\"></i>
18                         <i class=\"fas fa-star\"></i>
19                         <i class=\"fas fa-star\"></i>
20                         <i class=\"far fa-star\"></i>
21                     </h6>
22                     <p class=\"card-text\">
23                         Some quick example text to build on the card.
24                     </p>
25                     <h5>

```

Fig.3.7.1 component function creation

The component.php is used as it's name suggest as to create different component of the products.

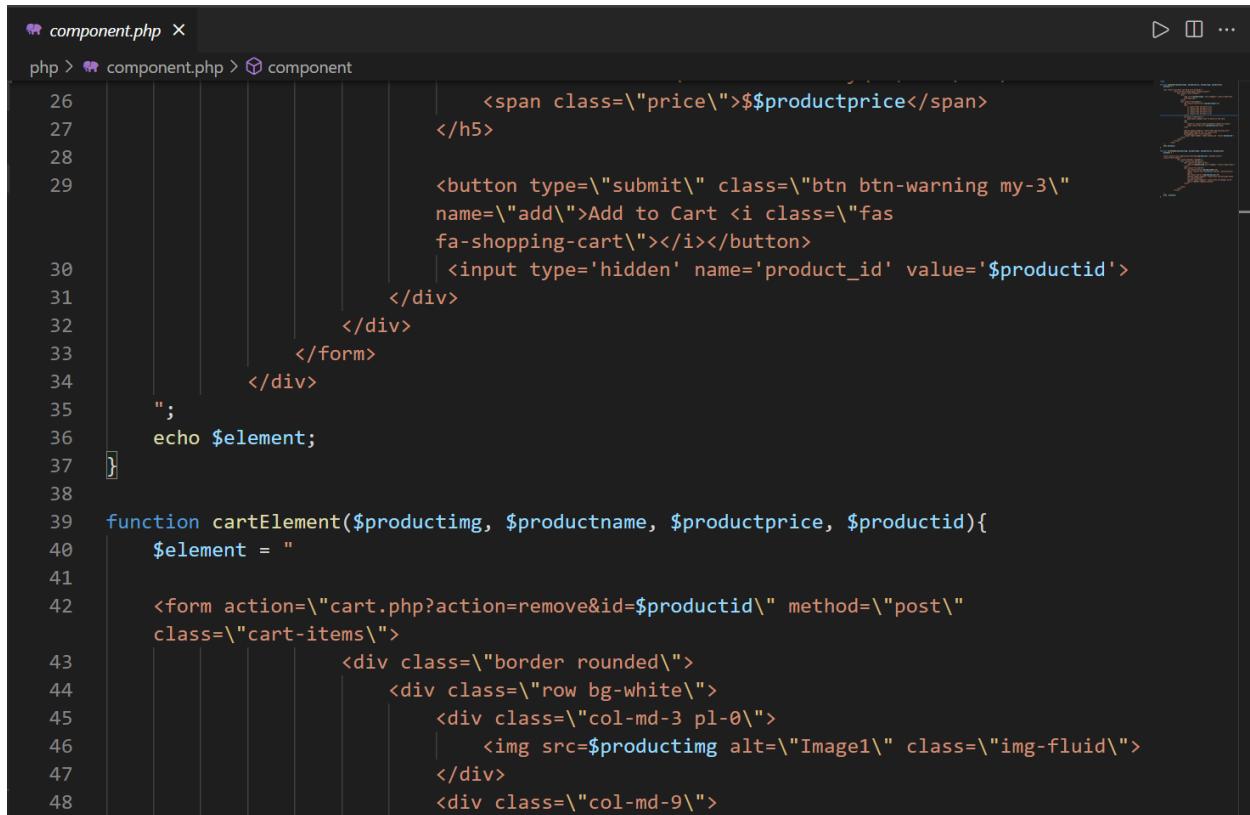
In the Fig.3.7.1 the component function is declared and its arguments are product name, product price, product price, product image, product\_id. The function just returns a single element.

It contains bootstrap cards in which the values like product image is used and product name.

Below there is fa-star which represent the code for a full star rating icon that comes from Bootstrap. Then the sample text to describe that element.

Then there is <small> tag which is used to create a small text than a normal text, also <s> tag is used to create strike through effect, text secondary class of bootstrap defines the color of the text.

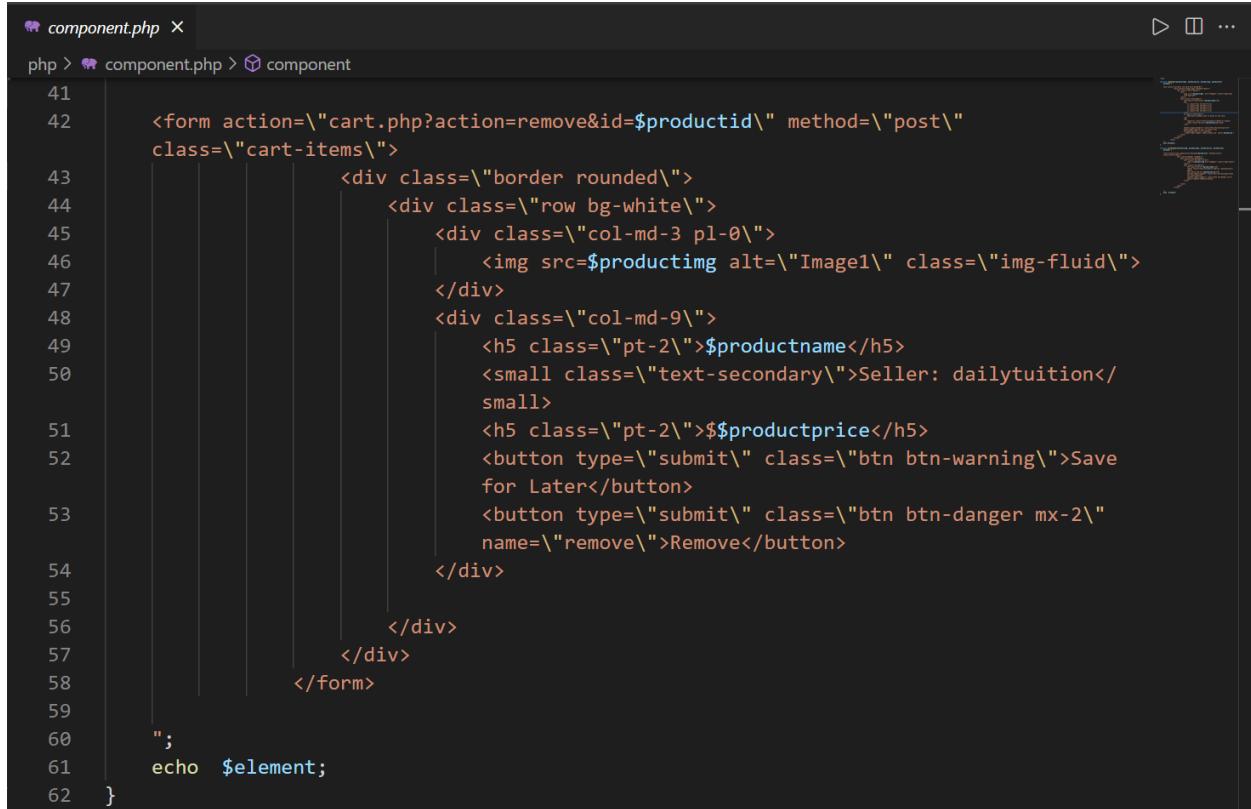
Card title is assigned through the variable name \$productname and image is assigned through \$productimg. Both of which is coming from the products database.



```
component.php
php > component.php > component
26     <span class="price">$productprice</span>
27     </h5>
28
29     <button type="submit" class="btn btn-warning my-3" name="add">Add to Cart <i class="fas fa-shopping-cart"></i></button>
30     <input type='hidden' name='product_id' value='$productid'>
31   </div>
32 </div>
33 </form>
34 </div>
35 ";
36 echo $element;
37 }
38
39 function cartElement($productimg, $productname, $productprice, $productid){
40   $element = "
41
42   <form action=\"cart.php?action=remove&id=$productid\" method=\"post\" class=\"cart-items\">
43     <div class=\"border rounded\">
44       <div class=\"row bg-white\">
45         <div class=\"col-md-3 pl-0\">
46           <img src=$productimg alt=\"Image1\" class=\"img-fluid\">
47         </div>
48         <div class=\"col-md-9\">
```

Fig.3.7.2 cart element function

The component function is then terminated by echo statement which just outputs the element at the end of the function. In Fig.3.7.2 & 3.7.3 the cart element function is used in order to define the element which is present in cart web page. The number of items present inside the cart is shown by this function by passing the product image , product name , product price, product id.



```

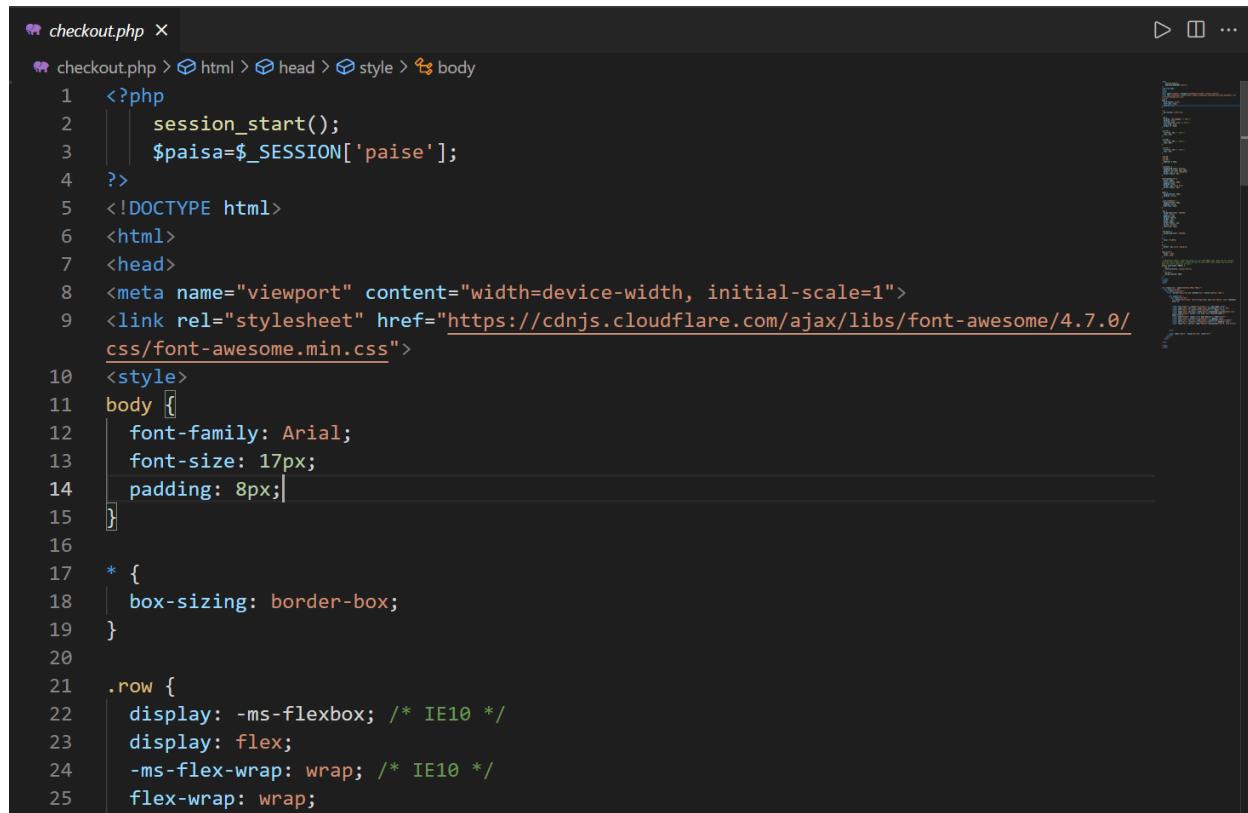
component.php X
php > component.php > component
41
42     <form action=\"cart.php?action=remove&id=$productid\" method=\"post\" class=\"cart-items\">
43         <div class=\"border rounded\">
44             <div class=\"row bg-white\">
45                 <div class=\"col-md-3 pl-0\">
46                     <img src=$productimg alt=\"Image1\" class=\"img-fluid\">
47                 </div>
48                 <div class=\"col-md-9\">
49                     <h5 class=\"pt-2\">$productname</h5>
50                     <small class=\"text-secondary\">Seller: dailytuition</small>
51                     <h5 class=\"pt-2\">$productprice</h5>
52                     <button type=\"submit\" class=\"btn btn-warning\">Save for Later</button>
53                     <button type=\"submit\" class=\"btn btn-danger mx-2\" name=\"remove\">Remove</button>
54                 </div>
55             </div>
56         </div>
57     </form>
58
59     ";
60
61     echo $element;
62 }

```

Fig.3.7.3 Cart Element function body

## 8) checkout.php:-

The checkout.php is used as a form that accepts input from the user after pressing checkout button in the cart web page.



```

checkout.php X
checkout.php > html > head > style > body
1  <?php
2      session_start();
3      $paisa=$_SESSION['paise'];
4 ?>
5  <!DOCTYPE html>
6  <html>
7      <head>
8          <meta name="viewport" content="width=device-width, initial-scale=1">
9          <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
10     <style>
11     body {
12         font-family: Arial;
13         font-size: 17px;
14         padding: 8px;
15     }
16
17     * {
18         box-sizing: border-box;
19     }
20
21     .row {
22         display: -ms-flexbox; /* IE10 */
23         display: flex;
24         -ms-flex-wrap: wrap; /* IE10 */
25         flex-wrap: wrap;

```

Fig.3.8.1 CSS for checkout page

```

checkout.php x
checkout.php > html > head > style > {} @media (max-width: 800px)

96 hr {
97   border: 1px solid lightgrey;
98 }
99
100 span.price {
101   float: right;
102   color: grey;
103 }
104
105 /* Responsive layout - when the screen is less than 800px wide, make the two columns
106 stack on top of each other instead of next to each other (also change the direction -
107 make the "cart" column go on top) */
108 @media (max-width: 800px) {
109   .row {
110     flex-direction: column-reverse;
111   }
112   .col-25 {
113     margin-bottom: 20px;
114   }
115 }
116 
```

Fig.3.8.2 CSS to make the form responsive

## Responsive layout -

when the screen is less than 800px wide, make the two columns stack on top of each other instead of next to each other (also change the direction-make the "cart" column go on top)

In Fig.3.8.1 & Fig.3.8.2 shows the code for CSS that is used in order to style the form in such a way such that the page when shown in screen whose size is less than 800px still able to see the form in more presentable and convenient way.

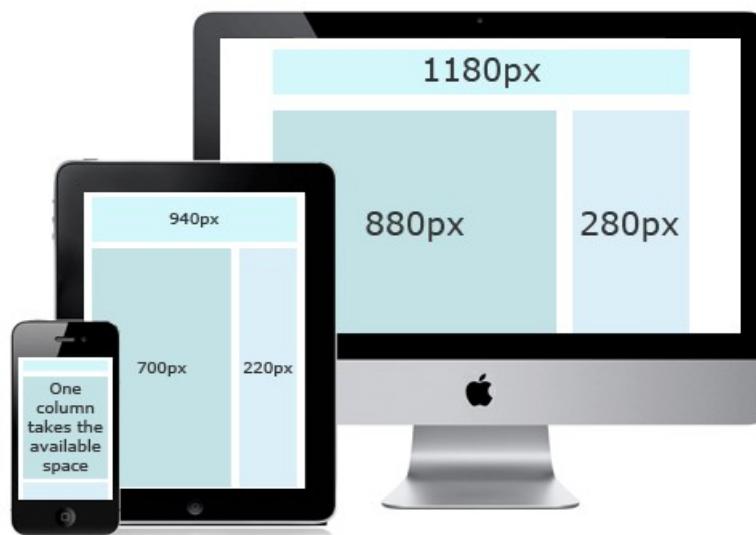
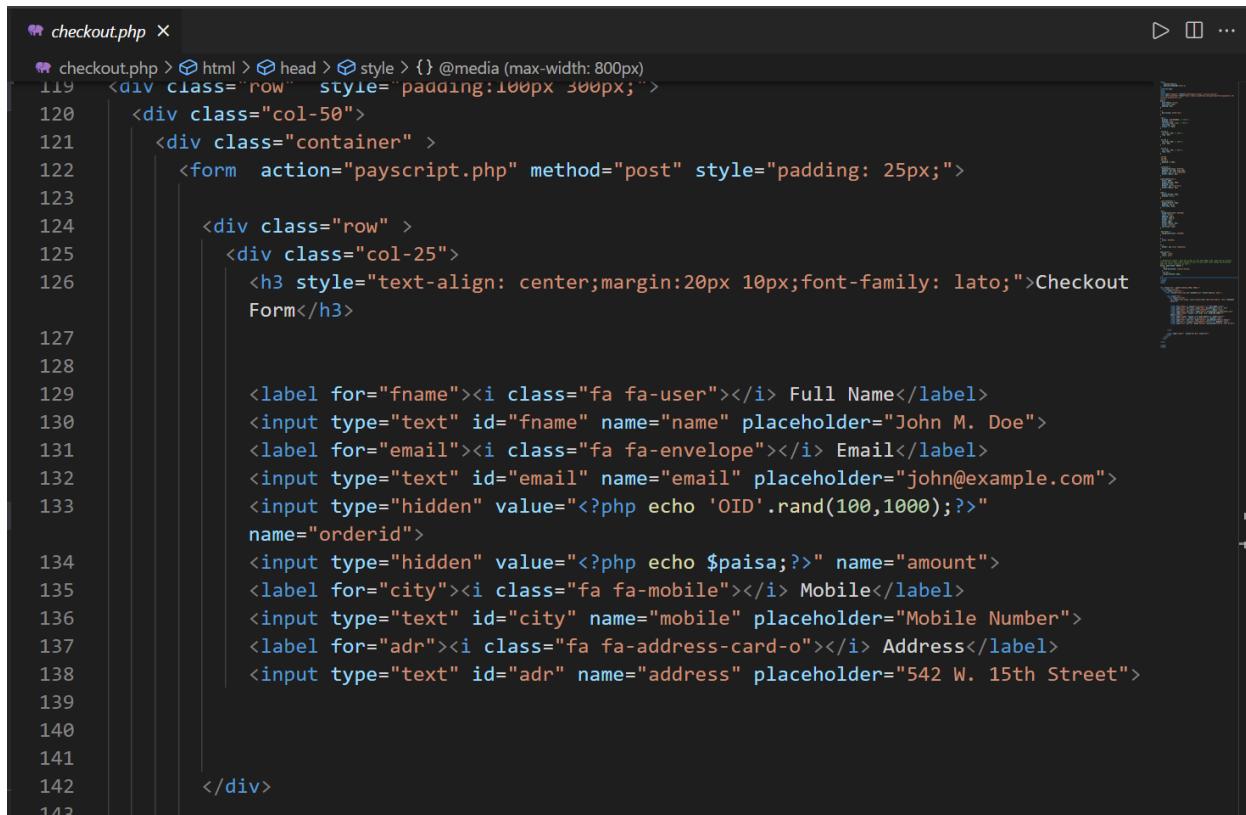


Fig.3.8.3.1 Responsive Layouts



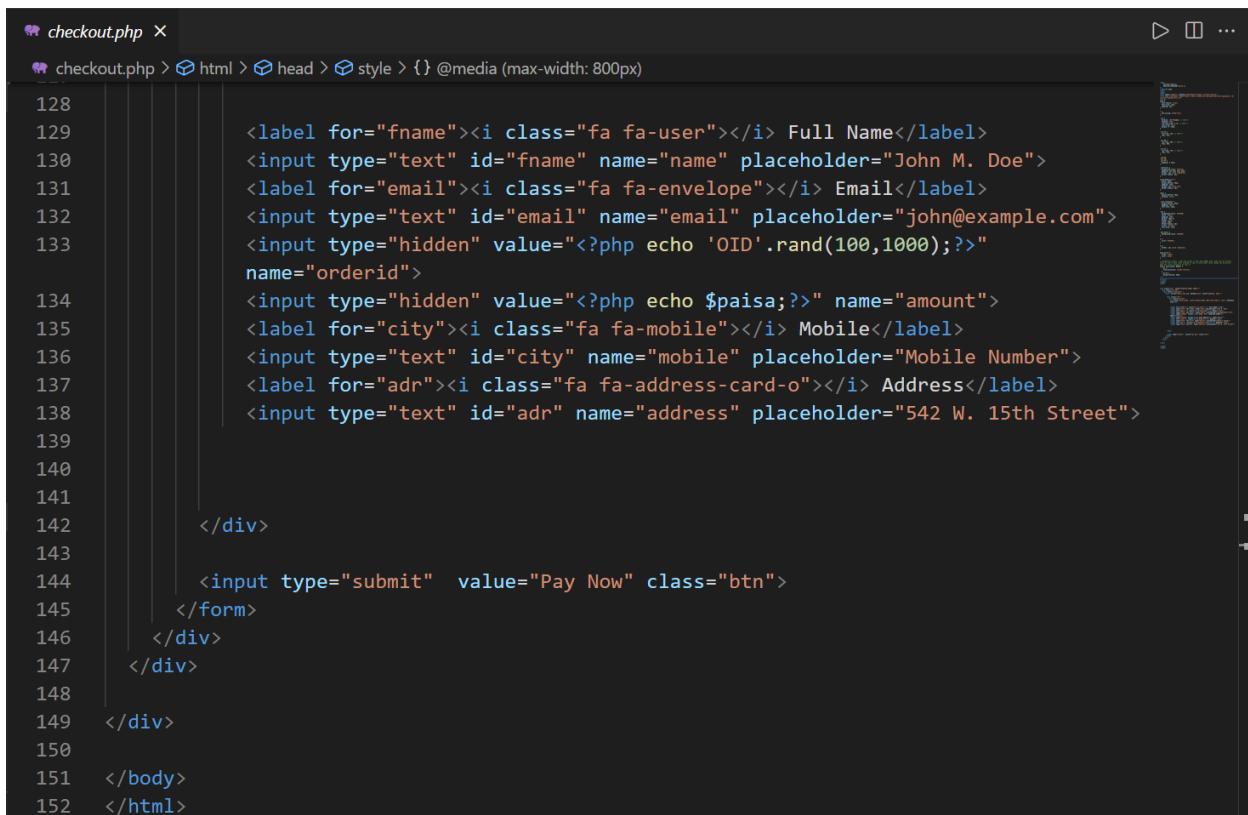
```

checkout.php ×
checkout.php > html > head > style > {} @media (max-width: 800px)
119  <div class="row" style="padding:100px 300px;">
120    <div class="col-50">
121      <div class="container" >
122        <form action="payscript.php" method="post" style="padding: 25px;">
123
124          <div class="row" >
125            <div class="col-25">
126              <h3 style="text-align: center; margin:20px 10px; font-family: lato;">Checkout
127                Form</h3>
128
129              <label for="fname"><i class="fa fa-user"></i> Full Name</label>
130              <input type="text" id="fname" name="name" placeholder="John M. Doe">
131              <label for="email"><i class="fa fa-envelope"></i> Email</label>
132              <input type="text" id="email" name="email" placeholder="john@example.com">
133              <input type="hidden" value="<?php echo 'OID'.rand(100,1000);?>" name="orderid">
134              <input type="hidden" value="<?php echo $paisa;?>" name="amount">
135              <label for="city"><i class="fa fa-mobile"></i> Mobile</label>
136              <input type="text" id="city" name="mobile" placeholder="Mobile Number">
137              <label for="adr"><i class="fa fa-address-card-o"></i> Address</label>
138              <input type="text" id="adr" name="address" placeholder="542 W. 15th Street">
139
140
141
142        </div>
143
144        <input type="submit" value="Pay Now" class="btn">
145      </form>
146    </div>
147  </div>
148
149 </div>
150
151 </body>
152 </html>

```

Fig.3.8.3 Main form code inside the checkout.php

The HTML code for form is shown in Fig.3.8.3 to take input from the user and also the total price for which the user needed to pay for the bought products from the lifestyle store is stored under a variable named as paisa which is in type hidden.



```

checkout.php ×
checkout.php > html > head > style > {} @media (max-width: 800px)
128
129          <label for="fname"><i class="fa fa-user"></i> Full Name</label>
130          <input type="text" id="fname" name="name" placeholder="John M. Doe">
131          <label for="email"><i class="fa fa-envelope"></i> Email</label>
132          <input type="text" id="email" name="email" placeholder="john@example.com">
133          <input type="hidden" value="<?php echo 'OID'.rand(100,1000);?>" name="orderid">
134          <input type="hidden" value="<?php echo $paisa;?>" name="amount">
135          <label for="city"><i class="fa fa-mobile"></i> Mobile</label>
136          <input type="text" id="city" name="mobile" placeholder="Mobile Number">
137          <label for="adr"><i class="fa fa-address-card-o"></i> Address</label>
138          <input type="text" id="adr" name="address" placeholder="542 W. 15th Street">
139
140
141
142        </div>
143
144        <input type="submit" value="Pay Now" class="btn">
145      </form>
146    </div>
147  </div>
148
149 </div>
150
151 </body>
152 </html>

```

Fig.3.8.4 Pay now button

The checkout page ends in the Fig.3.8.4 when the user is done after filling the mandatory input fields, the user presses the Pay Now button which redirect the user to Razorpay website where the user get different option to pay the amount like Debit Card/ Credit Card and UPI is also available (right now it's not available as the project is in Test mode, i.e, it sells virtual products).

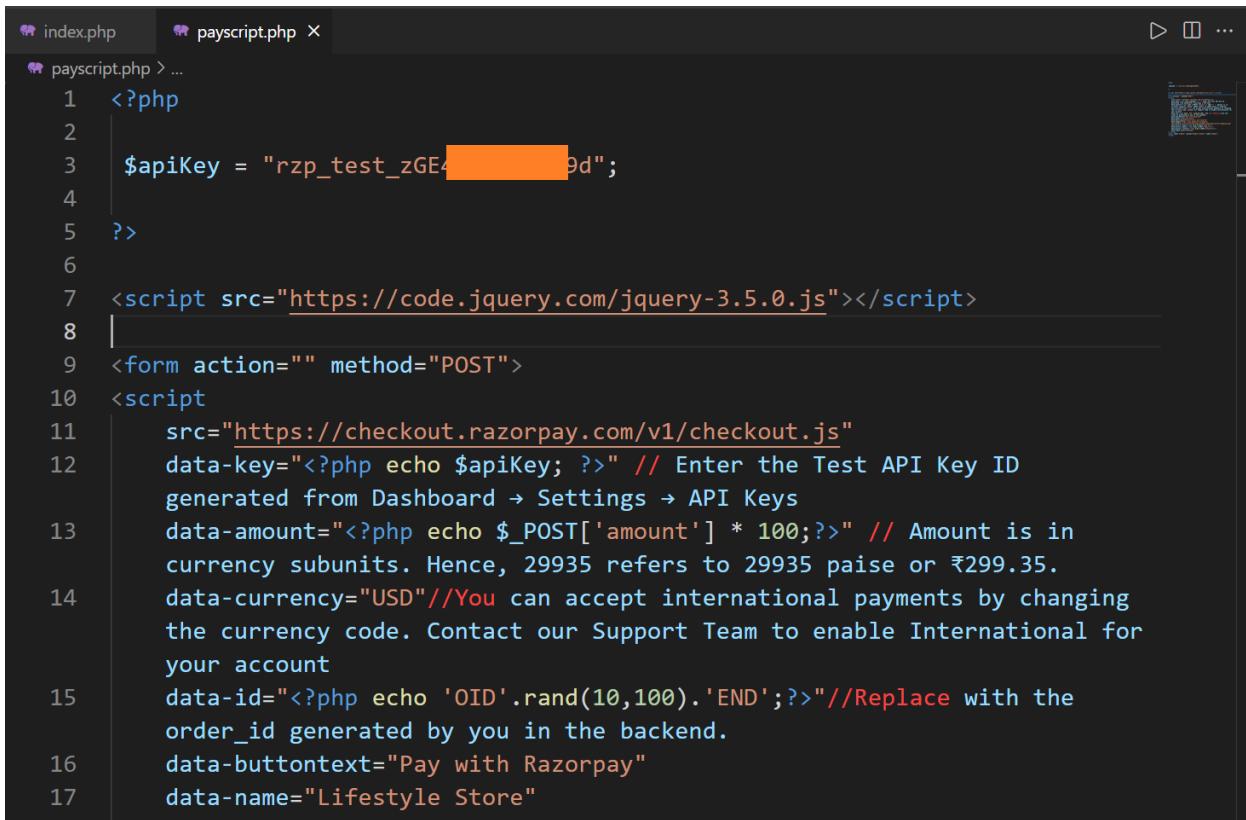
## 9) Payscript.php:-

Pay-script file includes the code which is written in javascript.

Although I hadn't told about javascript in this summer training.

But still this code is completely understandable.

This page includes an api key for the transaction in order to receive payment from the customer.



```
index.php payscript.php < ...  
payscript.php > ...  
1 <?php  
2  
3 $apiKey = "rzp_test_zGE4...9d";  
4  
5 ?>  
6  
7 <script src="https://code.jquery.com/jquery-3.5.0.js"></script>  
8 |  
9 <form action="" method="POST">  
10 <script  
11     src="https://checkout.razorpay.com/v1/checkout.js"  
12     data-key="<?php echo $apiKey; ?>" // Enter the Test API Key ID  
13     generated from Dashboard → Settings → API Keys  
14     data-amount="<?php echo $_POST['amount'] * 100;?>" // Amount is in  
15     currency subunits. Hence, 29935 refers to 29935 paise or ₹299.35.  
16     data-currency="USD"//You can accept international payments by changing  
17     the currency code. Contact our Support Team to enable International for  
     your account  
     data-id="<?php echo 'OID'.rand(10,100).'END';?>"//Replace with the  
     order_id generated by you in the backend.  
     data-buttontext="Pay with Razorpay"  
     data-name="Lifestyle Store"
```

Fig3.9.1 Pay-script code for transaction

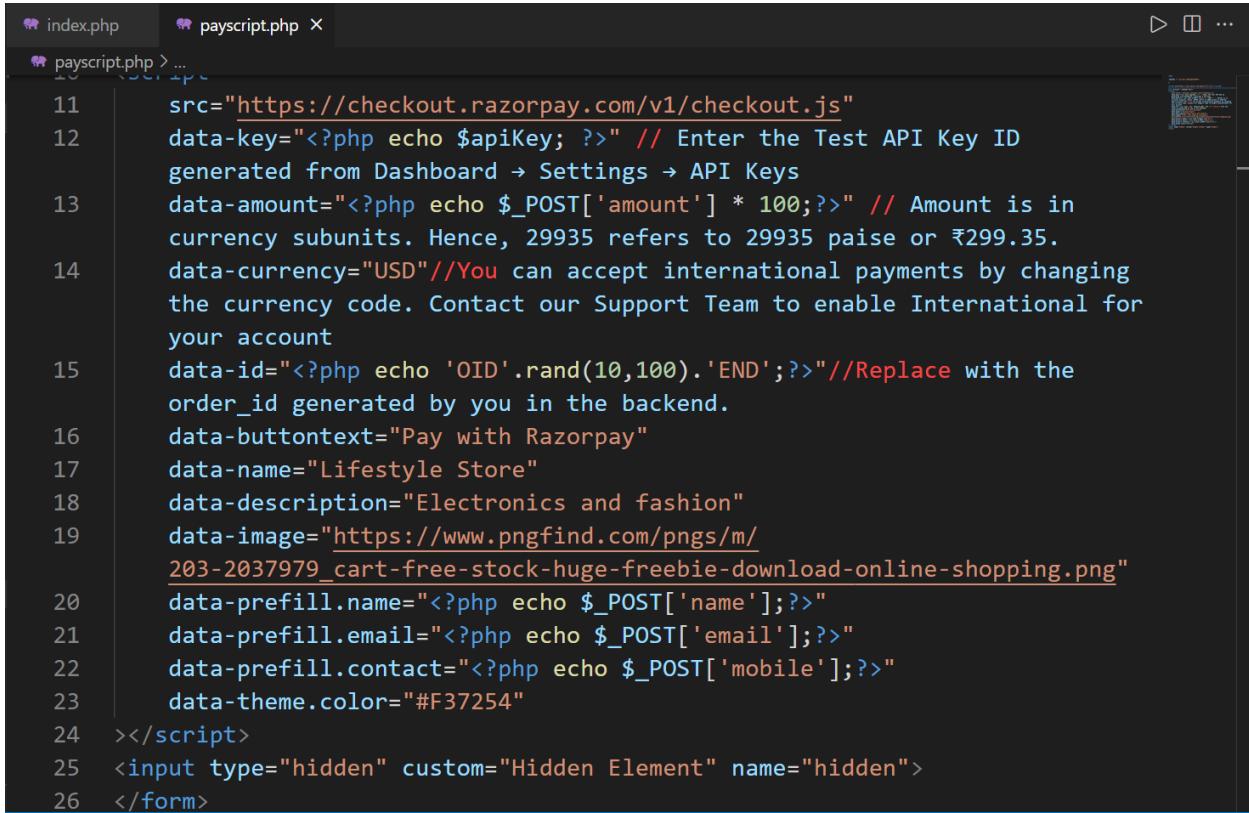
## API key:-

(Application programming interface key)

An application programming interface key is a unique identifier used to authenticate a user, developer, or calling program to an API. However, they are typically used to authenticate a project with the API rather than a human user. Different platforms may implement and use API keys in different ways.

API key is a combination of the key\_id and key\_secret and is required to make any API request to Razorpay. You also have to implement the API key in your code as part of your integration process.

There is php script inside every field so that the script is able to get information from the previous web pages. The type of currency is also defined here (in our case is '\$' USD). Then there is a data-id field in which the order id is generated using random function.



```

index.php payscript.php
payscript.php > ...
11     src="https://checkout.razorpay.com/v1/checkout.js"
12     data-key="<?php echo $apiKey; ?>" // Enter the Test API Key ID
13     generated from Dashboard → Settings → API Keys
14     data-amount="<?php echo $_POST['amount'] * 100;?>" // Amount is in
15     currency subunits. Hence, 29935 refers to 29935 paise or ₹299.35.
16     data-currency="USD"//You can accept international payments by changing
17     the currency code. Contact our Support Team to enable International for
18     your account
19     data-id="<?php echo 'OID'.rand(10,100).'END';?>"//Replace with the
20     order_id generated by you in the backend.
21     data-buttontext="Pay with Razorpay"
22     data-name="Lifestyle Store"
23     data-description="Electronics and fashion"
24     data-image="https://www.pngfind.com/pngs/m/
25     203-2037979_cart-free-stock-huge-freebie-download-online-shopping.png"
26     data-prefill.name="<?php echo $_POST['name'];?>"
27     data-prefill.email="<?php echo $_POST['email'];?>"
28     data-prefill.contact="<?php echo $_POST['mobile'];?>"
29     data-theme.color="#F37254"
30   ></script>
31   <input type="hidden" custom="Hidden Element" name="hidden">
32 </form>

```

Fig3.9.2 Inserting information from the checkout form by post method

In Fig.3.9.2 the information that is received from the checkout.php page. The information is prefilled inside the script using the previous information from the checkout page. All this information is in hidden format and everything is done in Back-end to implement security and privacy of the customer.

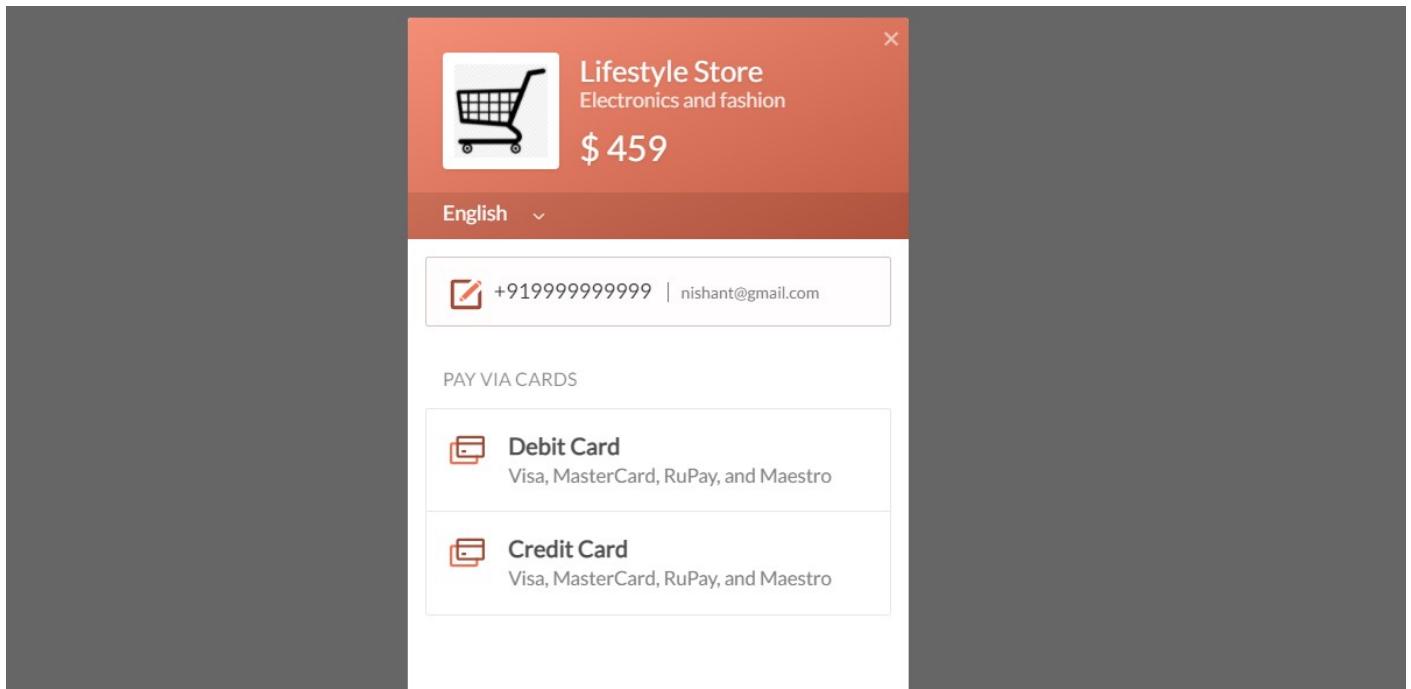
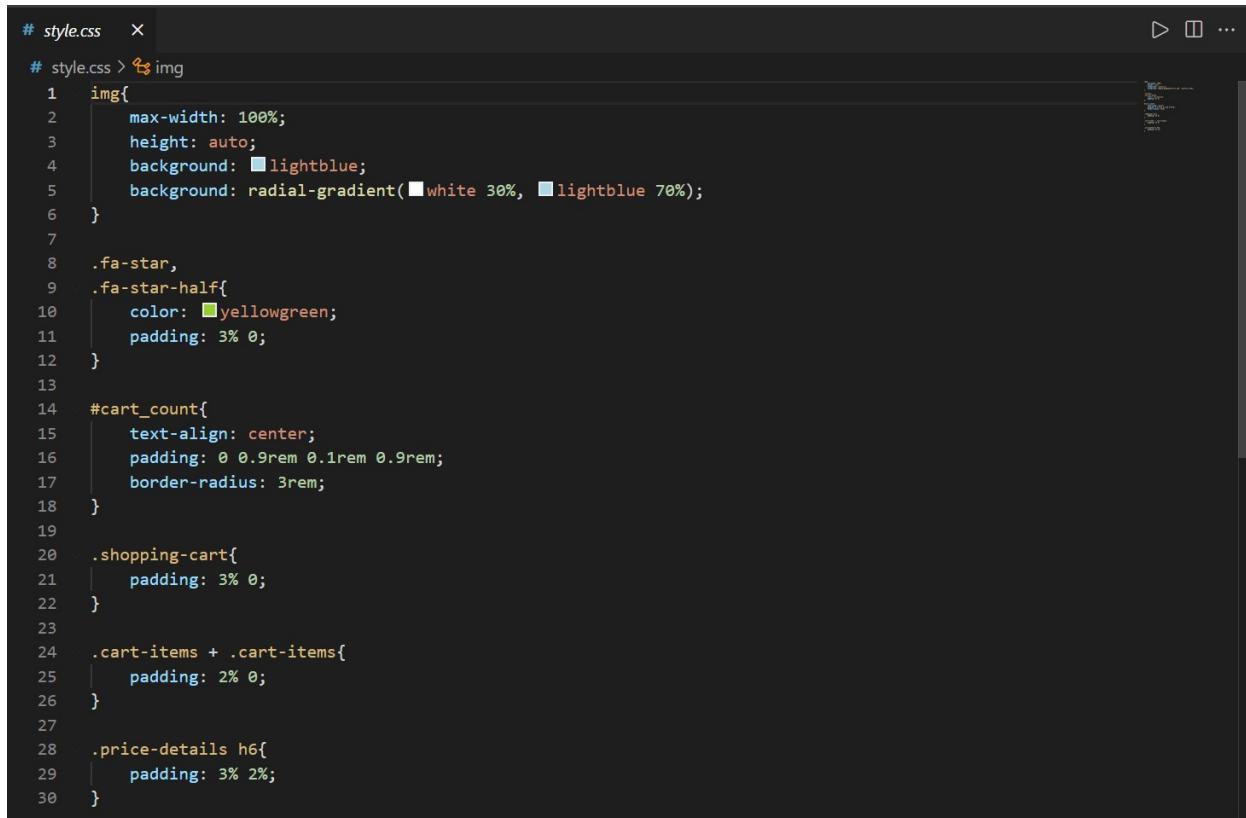


Fig3.9.3 Pay-script

## 10) style.css:-

This style.css is a style-sheet which is included in some parts of the web pages to increase the overall design of the web page.

### Code:-



```
# style.css  x
# style.css > ↗ img
1  img{
2    max-width: 100%;
3    height: auto;
4    background: lightblue;
5    background: radial-gradient(white 30%, lightblue 70%);
6  }
7
8  .fa-star,
9  .fa-star-half{
10   color: yellowgreen;
11   padding: 3% 0;
12 }
13
14 #cart_count{
15   text-align: center;
16   padding: 0 0.9rem 0.1rem 0.9rem;
17   border-radius: 3rem;
18 }
19
20 .shopping-cart{
21   padding: 3% 0;
22 }
23
24 .cart-items + .cart-items{
25   padding: 2% 0;
26 }
27
28 .price-details h6{
29   padding: 3% 2%;
30 }
```

Fig3.10.1 style.css code

Style.css affects the design of the web page in these particular areas as given below:-

- style.css is implemented to change the color of star icon used to indicate ratings.
- Cart count which is present at top right side corner is also designed by this CSS.
- Shopping cart is designed by giving it some padding 3%.
- Images of the products is also designed as the radial blue effect is caused by this code using radial gradient of white and blue color to a range of 30% to 70% in value respectively.
- Price Details <h6> tag which is present in cart.php have its padding by including this style-sheet.

## Chapter 4- Screenshot of the project / DFD

### Screenshots:-

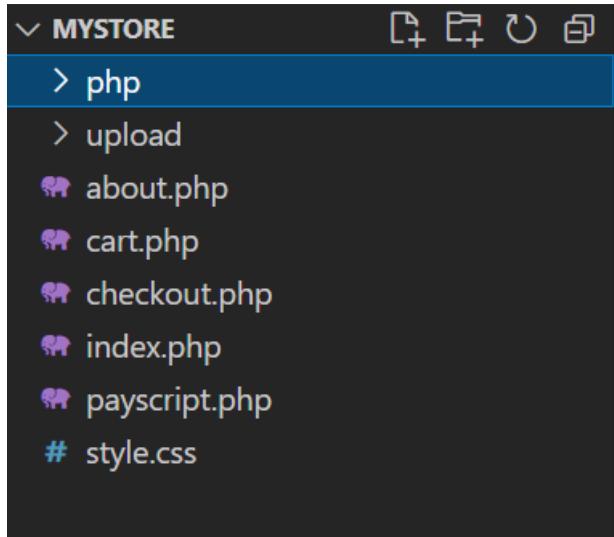


Fig.4.1.1

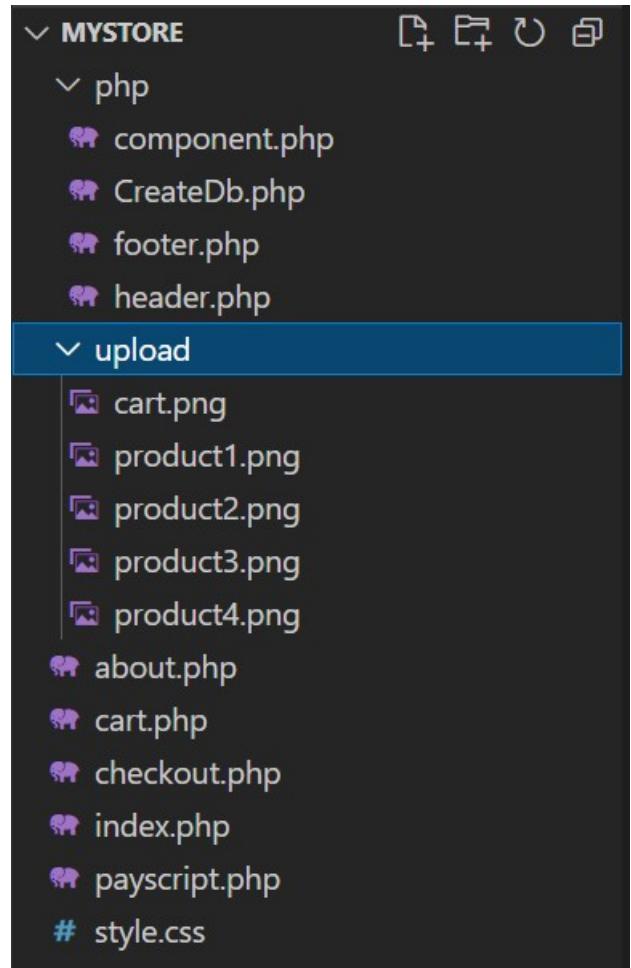


Fig.4.1.2

The Basic Structure of the different pages inside the Mystore directory is represented in Fig.4.1.1 & Fig.4.1.2

## DFD (DATA FLOW DIAGRAM):-

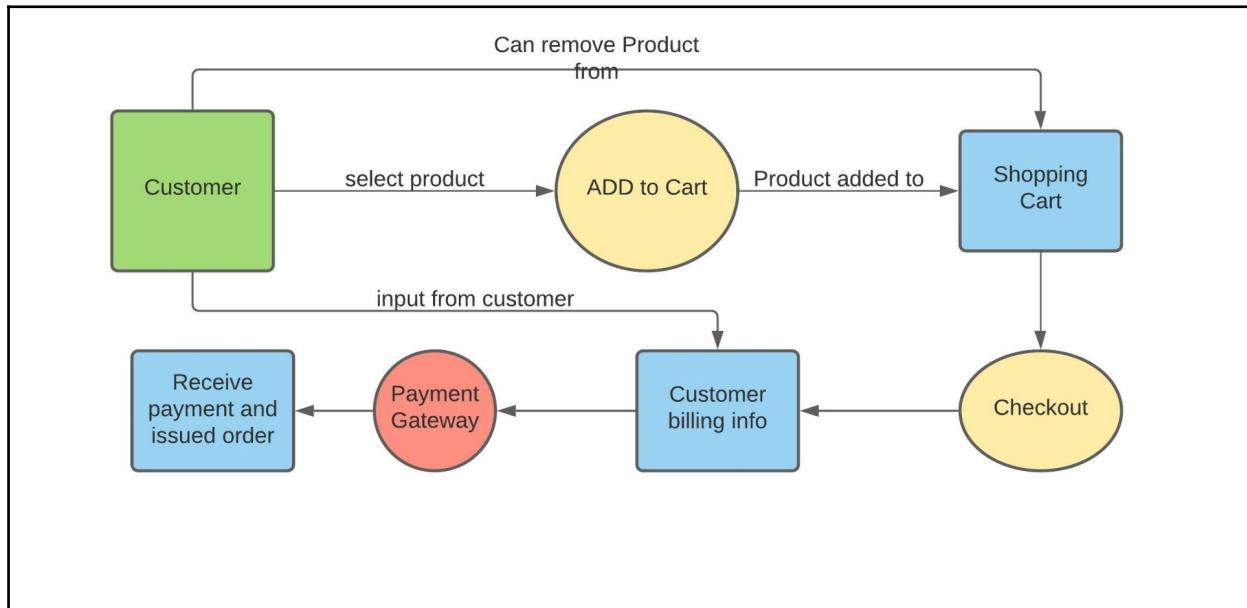


Fig.4.2.1 Data Flow Diagram for Lifestyle Store

## Lifestyle Store (Complete web pages):-

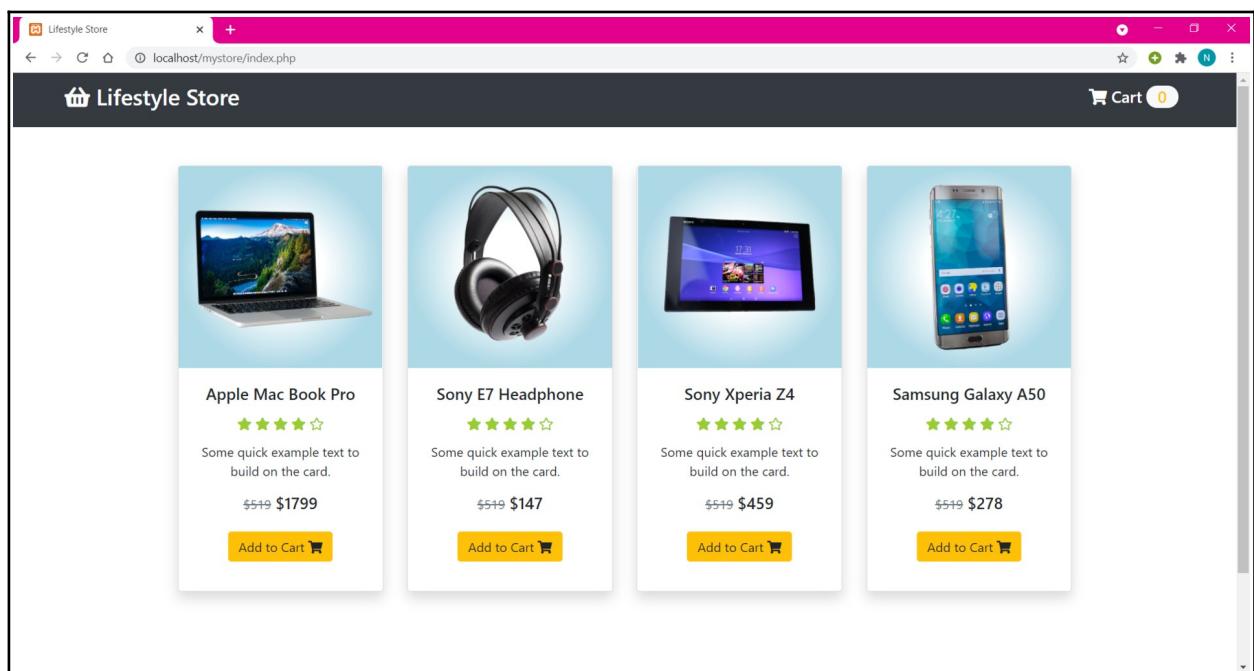


Fig.4.2.1 index.php

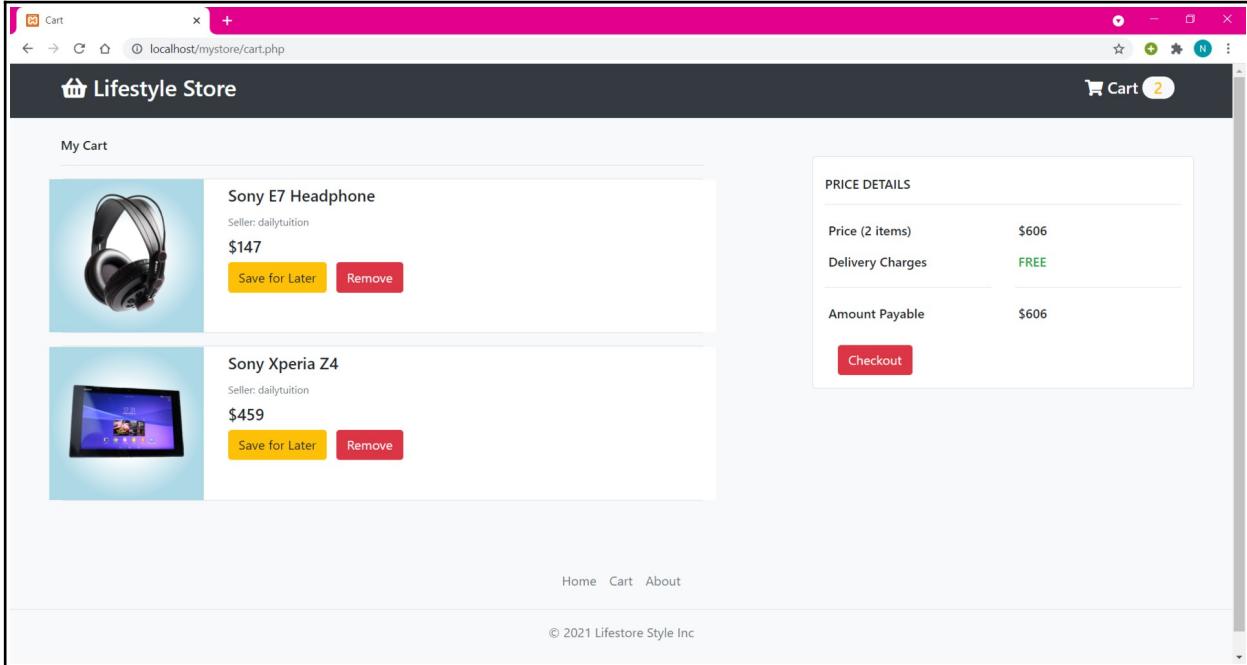


Fig.4.2.2 Cart.php

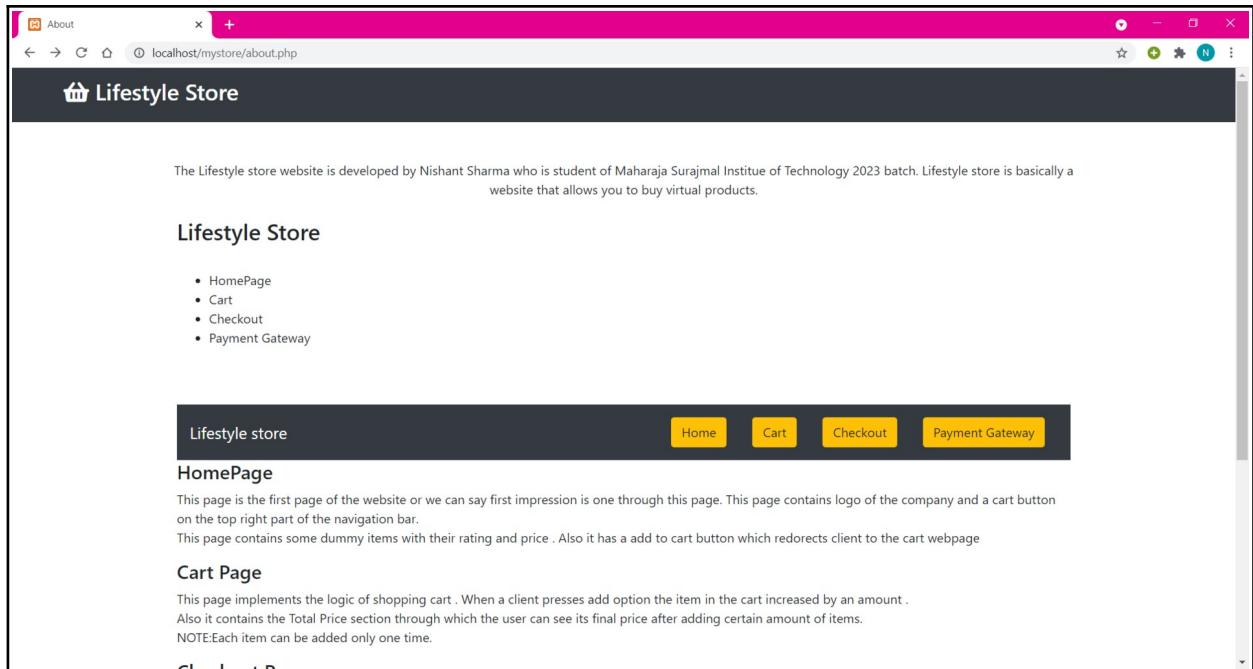


Fig.4.2.3 about.php

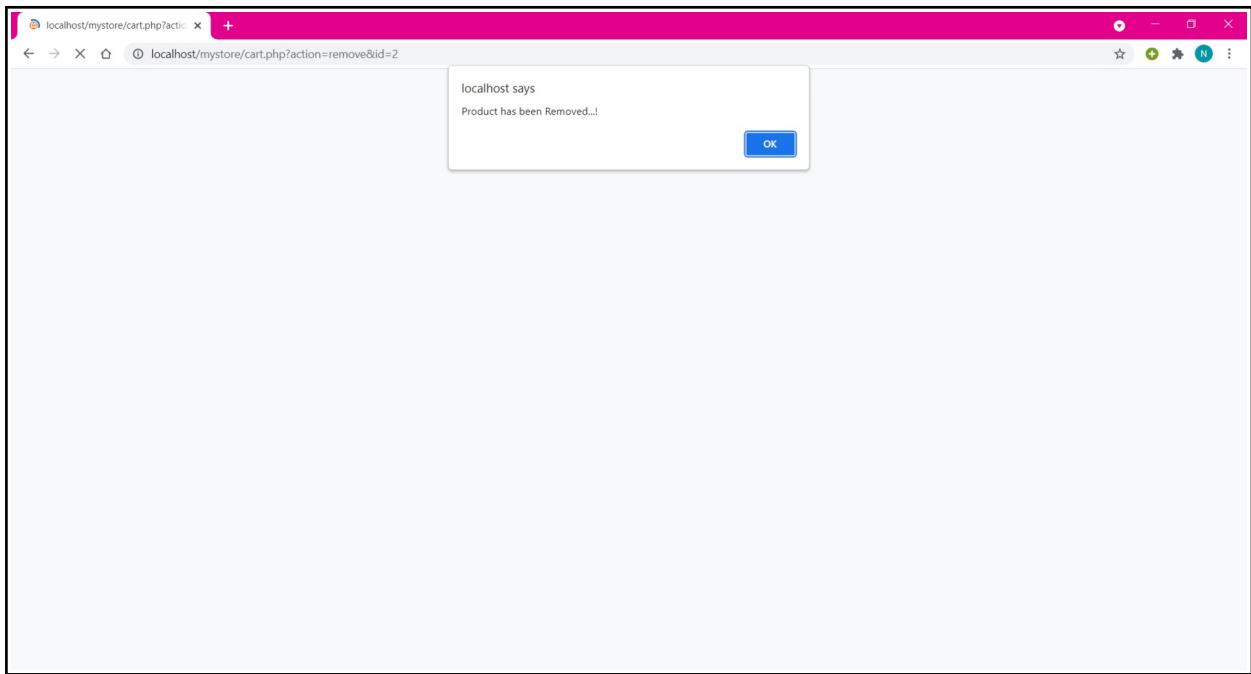


Fig.4.2.4 Remove button implemented

A screenshot of a web browser window titled "localhost/mystore/checkout.php". The page displays a "Checkout Form" with the following fields:

- Full Name: nishant
- Email: nishant@gmail.com
- Mobile: 9999999999
- Address: Janakpuri, New Delhi-58

A large green button at the bottom is labeled "Pay Now".

Fig.4.2.5 checkout.php

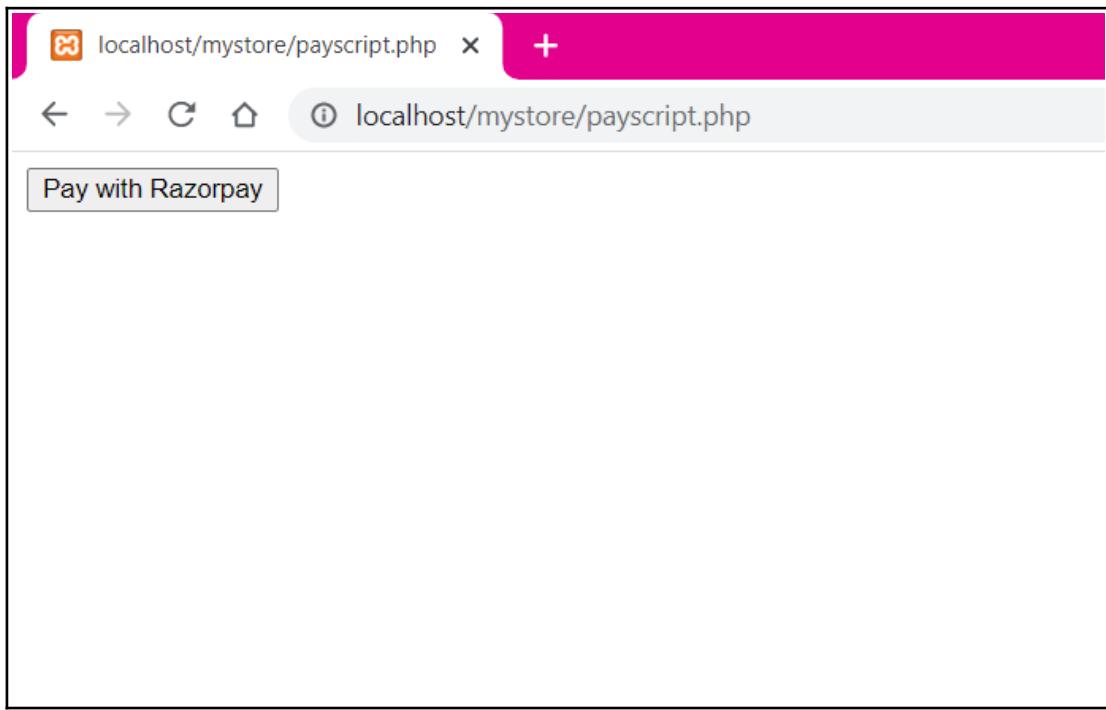


Fig.4.2.6 Pay-script of Razorpay

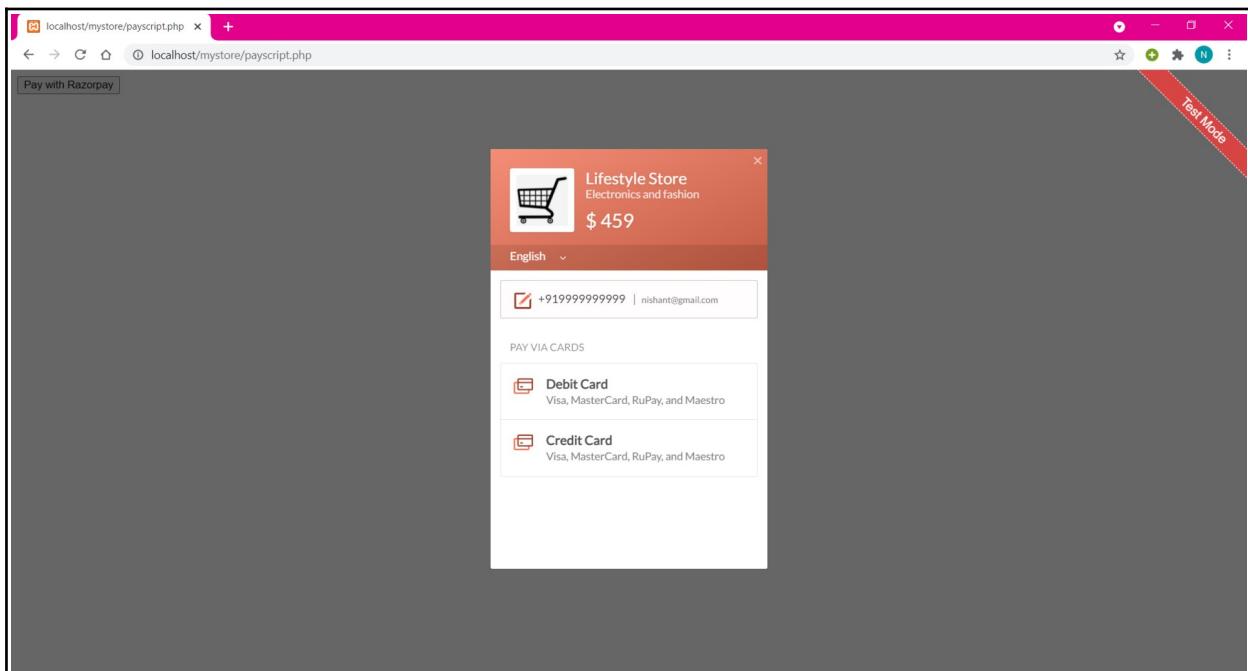


Fig.4.2.7 Demonstration of Payment Gateway

## Database Screenshot For Project



Fig.4.3.1 Database Structure

A screenshot of the phpMyAdmin interface. The top navigation bar shows 'Server: 127.0.0.1 &gt; Database: productdb'. The 'Structure' tab is selected. In the main area, there's a table titled 'producttb' with 4 rows and 4 columns. Below the table, there's a 'Create table' form with fields for 'Name:' (empty) and 'Number of columns:' (set to 4). A 'Go' button is at the bottom of the form.

Fig.4.3.2 Database containing product table

A screenshot of the phpMyAdmin interface. The top navigation bar shows 'Server: 127.0.0.1 &gt; Database: productdb &gt; Table: producttb'. The 'Structure' tab is selected. The table structure is shown with 4 columns: 'id' (int(11)), 'product\_name' (varchar(25)), 'product\_price' (float), and 'product\_image' (varchar(100)). Below the table, there's an 'Indexes' section with a single entry: 'id' (PRIMARY, BTREE, Yes, No, id, 4, A, No).

Fig.4.3.3 Product Table Structure

## Product Table with different Product Records

The screenshot shows the phpMyAdmin interface for a MySQL database named 'productdb'. The current table is 'producttb'. The table structure includes columns: id, product\_name, product\_price, and product\_image. There are four records displayed:

	id	product_name	product_price	product_image
<input type="checkbox"/>	1	Apple Mac Book Pro	1799	./upload/product1.png
<input type="checkbox"/>	2	Sony E7 Headphone	147	./upload/product2.png
<input type="checkbox"/>	3	Sony Xperia Z4	459	./upload/product3.png
<input type="checkbox"/>	4	Samsung Galaxy A50	278	./upload/product4.png

Below the table, there are buttons for 'Edit', 'Copy', and 'Delete' for each row. At the bottom of the page, there are additional buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

Fig.4.3.1 Product Table with different Products

## CONCLUSION

- The Lifestyle store project is completely responsive. And the technology that I have used is working efficiently for the purpose they are included in the project.
- This website have the ability to make client to add some products in its cart using add to cart button and cart page is used to get information about the products that user has in its cart.
- If the client wants to remove certain products from its cart then this features is also enabled in lifestyle store project using a remove button inside the product container of the cart web page
- The client also can see the total price he/she has to pay to the company using the lifestyle store cart page where the total price section is also made available using php.
- The cart page also contains a checkout button which redirect user to checkout page where it asks for information about the client and address where the products need to be delivered.
- There are some functions inside component.php which helps to generate more products by entering product name and product price. The navigation bar is coded under header.php in order to reuse the same navigation bar more times in different pages in which php comes in handy.
- HTML is used to build the skeleton of every page and CSS is used to increase some sort of styling and the predefined cards, crouseles, layouts of the Bootstrap completely helps to reduce few styling lines.
- Then comes the font-awesome website that helps to include awesome icons in my website.
- My php admin is used in order to create a database and sql commands comes handy in order to manipulate, add, delete data in the database.
- The checkout.php includes its own styling css in it and a form which works fabulous in order to receive information from the client and transfer it to the payment gateway of Razorpay.org
- payscript.php has a javascript code which helps to transfer data of the client to the Razorpay website which generates a bill of the client according to the amount of purchase the client has made from the website.

## List of References

1. Intenshala summer training videos

link:-<https://internshala.com/>

2. Getbootstrap.com

link:- <https://getbootstrap.com/>

3. font awesome

link:- <https://fontawesome.com/>

4. Php manual

link:- <https://www.php.net/manual/en/index.php>

5. w3schools

link:- <https://www.w3schools.com/>

6. Razorpay

link:- <https://razorpay.com/>

7. Youtube videos

link:- <https://www.youtube.com/c/CodeWithHarry>

## Appendices

- HTML:-

The Hypertext Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser.

- CSS:-

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

- BOOTSTRAP:-

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.

- SQL:-

SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system

- PHP:-

PHP is a general-purpose scripting language geared towards web development.

- API:-

An application programming interface is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software.

- DATABASE:-

A database is an organized collection of data stored and accessed electronically from a computer system.

- MARGIN:-

Margins are used to create space around elements, outside of any defined borders.

- QUERY:-

A query is a request for data or information from a database table or combination of tables. This data may be generated as results returned by Structured Query Language (SQL) or as pictorials, graphs or complex results

- PADDING:-

Padding is white space immediately surrounding an element or another object on a web page.