

# Code listings

HTML  
Processor  
Class  
Style

## Contents

<b>Index.php</b>	2
<b>DoLogin.php</b>	3
<b>Register.php</b>	5
<b>DoRegister.php</b>	6
<b>UserObject.php</b>	8
<b>Welcome.php</b>	8
<b>Basket.php</b>	10
<b>BasketObject.php</b>	11
<b>ProductObject.php</b>	13
<b>addToBasketProcess.php</b>	15
<b>removeFromBasket.php</b>	16
<b>myItems.php</b>	16
<b>addToMyItemsProcess.php</b>	17
<b>removeFromMyItems.php</b>	18
<b>Checkout.php</b>	18
<b>DoCheckout.php</b>	20
<b>LogOut.php</b>	22
<b>DiscountProcess.php</b>	22
<b>Utilities.php</b>	22
<b>NavigationBar.php</b>	23
<b>Admin.php</b>	24
<b>AccountsStyles.php</b>	25
<b>BasketStyles.php</b>	26

<b>CheckoutStyles.php</b>	28
<b>Header.php</b>	29
<b>WelcomeStyles.php</b>	31

## Index.php:

```

index.php
1 <html>
2 <head>
3     <title>Login - MyEcommerce</title>
4     <link rel="stylesheet" href="/MyEcommerce/fontawesome/css/all.css">
5     <?php
6         session_start();
7         require_once("/xampp/htdocs/MyEcommerce/Styles/AccountsStyles.php");
8         $myErrors = array();
9         if (isset($_SESSION["allerrors"])) {
10             $myErrors = $_SESSION["allerrors"];
11             unset($_SESSION["allerrors"]);
12         }
13     ?>
14     <head>
15         </body>
16         <?php include "NavigationBar.php"; ?>
17         <div id="LoginDiv">
18             <form id="loginForm" action="/MyEcommerce/Processors/DoLogin.php" method="post">
19                 <h1>Please Log in</h1>
20
21                 <p id="usernameLabel">Email Address</p>
22
23                 <p><input class="textInput" type="text" name="email" /></p>
24
25                 <p id="passwordLabel">Password</p>
26
27                 <p><input class="textInput" type="password" name="password" /></p>
28
29                 <?php
30                     //Check to see if an error is found in the email or password
31                     if (in_array("password", $myErrors) || in_array("email", $myErrors)) {
32
33                         echo "<p class='errorMessage'>Email or Password is invalid</p>";
34                     }
35                 ?>
36
37                 <p><input class="standardButton" type="submit" value="Log in" /></p>
38
39                 <a href="Register.php" target="_blank" id="regLink">Click here to Register</a>
40             </form>
41         </div>
42     </body>
43 </html>

```

## DoLogin.php:

```

DoLogin.php
1 <?php
2 require_once("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3 require_once("/xampp/htdocs/MyEcommerce/Utilities.php");
4 session_start();
5
6 //Using Regular Expressions to find a pattern in the data
7 $isUppercase = preg_match('@[A-Z]@', $_POST["password"]);
8 $isLowercase = preg_match('@[a-z]@', $_POST["password"]);
9 $atLeastOneNumber = preg_match('@[0-9]@', $_POST["password"]);
10
11 $errors = array();
12 $vals = array();
13
14 //Checks if the user inputted something into the email input in the login page
15 //using the attribute name of the input tag
16 //if no then add an error
17 //if yes then add it to the vals array
18 foreach (["email"] as $val) {
19     if (!isset($_POST[$val]) || strlen($_POST[$val]) == 0) {
20         $errors[] = $val;
21     }
22     $vals[$val] = $_POST[$val];
23 }
24
25 //Validate the pattern of the email
26 if (!filter_var($_POST["email"], FILTER_VALIDATE_EMAIL) || !isset($_POST["email"])) {
27     $errors[] = "email";
28 }
29
30 //Validate the password
31 if (!isset($_POST["password"]) || !$isUppercase || !$isLowercase
32     || !$atLeastOneNumber || strlen($_POST["password"]) < 8) {
33     $errors[] = "password";
34 }
35
36 if (sizeof($errors) == 0) {
37     //if there are no errors make a link with the database
38     $mysqli = new mysqli("localhost", "root", "", "e-commerce");
39     //retrieve the details of a user based on their email
40     $stmt = $mysqli->prepare("SELECT * FROM eUsers WHERE Email=?");
41     //using prepared statements to prevent SQL Injections
42     $stmt->bind_param("s", $_POST["email"]);

```

```
43     $stmt->execute();
44     $res = $stmt->get_result();
45     $stmt->close();
46     //check if any results were found
47     if ($res->num_rows == 1) {
48         $row = $res->fetch_assoc();
49         //Compare the inputted password to the password in the database
50         if (password_verify($_POST["password"], $row["Password"])) {
51             //Creating a user object
52             $myUser = new User($row["UserID"], $row["UserType"],
53                                $row["FirstName"], $row["LastName"], $row["Email"]);
54             //Uploading the user into the session
55             $_SESSION["myUser"] = $myUser;
56             header("Location: /MyEcommerce/Welcome.php");
57             exit;
58         } else {
59             $errors[] = "password";
60             $_SESSION["allerrors"] = $errors;
61             header("Location: /MyEcommerce/index.php?error=Failed Validation&" . $p);
62             exit;
63         }
64     } else {
65         $errors[] = "email";
66         $errors[] = "password";
67         $_SESSION["allerrors"] = $errors;
68         header("Location: /MyEcommerce/index.php");
69         exit;
70     }
71 } else {
72     $_SESSION["allerrors"] = $errors;
73     $p = http_build_query($vals);
74     header("Location: /MyEcommerce/index.php?error=Failed Validation&" . $p);
75     exit;
76 }
77 }
```

## Register.php:

```

1  <html>
2  <head>
3      <title>Register</title>
4      <link rel="stylesheet" href="/MyEcommerce/fontawesome/css/all.css">
5      <?php
6          session_start();
7          require_once("/xampp/htdocs/MyEcommerce/Styles/AccountsStyles.php");
8          $myErrors = array();
9          if (isset($_SESSION["allerrors"])) {
10              $myErrors = $_SESSION["allerrors"];
11              unset($_SESSION["allerrors"]);
12          }
13      ?>
14      <script src="jquery-3.6.0.min.js"></script>
15      <script>
16          //2 functions that use jquery to switch between two divs
17          //RegForm part 1 & part 2
18          function Next(hide, show) {
19              $("#" + hide).toggle();
20              $("#" + show).toggle();
21          }
22
23          function Previous(show, hide) {
24              $("#" + show).toggle();
25              $("#" + hide).toggle();
26          }
27      </script>
28  </head>
29  </body>
30  <?php include "NavigationBar.php"; ?>
31  <div id="RegDiv">
32      <form id="RegForm" action="/MyEcommerce/Processors/DoRegister.php" method="post">
33          <h1>Create an account</h1>
34          <div id="part1">
35              <p id="Label">First Name</p>
36              <p><input class="textInput" type="text" name="Fname" /></p>
37              <?php
38                  if (in_array("Fname", $myErrors)) {
39                      echo "<p class='errorMessage'>First Name is not valid</p>";
40                  }
41              ?>
42              <p id="Label">Last Name</p>
43              <p><input class="textInput" type="text" name="Lname" /></p>
44
45              <?php
46                  if (in_array("Lname", $myErrors)) {
47                      echo "<p class='errorMessage'>Last name is not valid</p>";
48                  }
49              ?>
50
51              <p id="Label">Email Address</p>
52              <p><input class="textInput" type="text" name="email" /></p>
53              <?php
54                  if (in_array("email", $myErrors)) {
55                      echo "<p class='errorMessage'>Email is not valid</p>";
56                  }
57              ?>
58
59              <p id="Label">Password</p>
60              <p><input class="textInput" type="password" name="password" /></p>
61              <?php
62                  if (in_array("password", $myErrors)) {
63                      echo "<p class='errorMessage'>Password is not valid</p>";
64                  }
65              ?>
66
67              <p id="Label">Password Confirmation</p>
68              <p><input class="textInput" type="password" name="passwordCon" /></p>
69              <?php
70                  if (in_array("passwordCon", $myErrors)) {
71                      echo "<p class='errorMessage'>Passwords does not match</p>";
72                  }
73              ?>
74
75              <p><input class="standardButton" type="button" value="Next" onclick="Next('part1','part2')"/></p>
76          </div>
77          <div id="part2" style="display: none;">
78              <p id="Label">Phone Number</p>
79              <p><input class="textInput" type="text" name="pNum" /></p>
80              <?php
81                  if (in_array("pNum", $myErrors)) {
82                      echo "<p class='errorMessage'>Phone Number is not valid</p>";
83                  }
84              ?>
85              <p id="Label">Country</p>
86              <p><input class="textInput" type="text" name="country" /></p>
87              <?php

```

```

87
88         if (in_array("country", $myErrors)) {
89             echo "<p class='errorMessage'>Country is not valid</p>";
90         }
91     ?>
92
93         <p id="Label">city</p>
94         <p><input class="textInput" type="text" name="city" /></p>
95     <?php
96         if (in_array("city", $myErrors)) {
97             echo "<p class='errorMessage'>City is not valid</p>";
98         }
99     ?>
100
101         <p id="Label">Address</p>
102         <p><input class="textInput" type="text" name="address" placeholder="E.g. Door Num + Street Name" /></p>
103     <?php
104         if (in_array("address", $myErrors)) {
105             echo "<p class='errorMessage'>Address is not valid</p>";
106         }
107     ?>
108
109         <p id="Label">Postcode</p>
110         <p><input class="textInput" type="text" name="postcode" /></p>
111     <?php
112         if (in_array("postcode", $myErrors)) {
113             echo "<p class='errorMessage'>Postcode invalid</p>";
114         }
115     ?>
116
117         <p><input class="standardButton" type="button" value="Previous" onclick="Previous('part1','part2')"></p>
118         <p><input class="standardButton" type="submit" value="submit"></p>
119     </div>
120 </div>
121 </body>
122 </html>

```

## DoRegister.php:

```

1 <?php
2 require_once("/xampp/htdocs/MyEcommerce/Utilities.php");
3 session_start();
4
5 //Retrieving the data of all the inputs from the register page
6 $fname = $_POST["Fname"];
7 $lname = $_POST["Lname"];
8 $email = $_POST["email"];
9 $password = $_POST["password"];
10 $passwordCon = $_POST["passwordCon"];
11 $pNum = $_POST["pNum"];
12 $country = $_POST["country"];
13 $city = $_POST["city"];
14 $address = $_POST["address"];
15 $postcode = $_POST["postcode"];
16
17 //Regular Expressions for validation
18 $isUppercase = preg_match('/@([A-Z]@', $password);
19 $isLowercase = preg_match('/@([a-z]@', $password);
20 $atLeastOneNumber = preg_match('/@([0-9]@', $password);
21 $pNumPattern = "/^(\+44\s?7\d{3}|\(07\d{3}\)\s?)\s?\d{3}\s?\d{3}$/";
22
23 $errors = array();
24 $vals = array();
25
26 //A check to see if the user has inputted something or left it blank
27 foreach (["Fname", "Lname", "email", "password", "passwordCon",
28         "pNum", "country", "city", "address", "postcode"] as $val) {
29     if (!isset($_POST[$val]) || strlen($_POST[$val]) == 0) {
30         $errors[] = $val;
31     }
32     $vals[$val] = $_POST[$val];
33 }
34
35 //Validating all the inputs
36 if (!preg_match("/^@[a-zA-Z]*$/", $fname)) {
37     $errors[] = "Fname";
38 }
39
40 if (!preg_match("/^@[a-zA-Z]*$/", $lname)) {
41     $errors[] = "Lname";
42 }

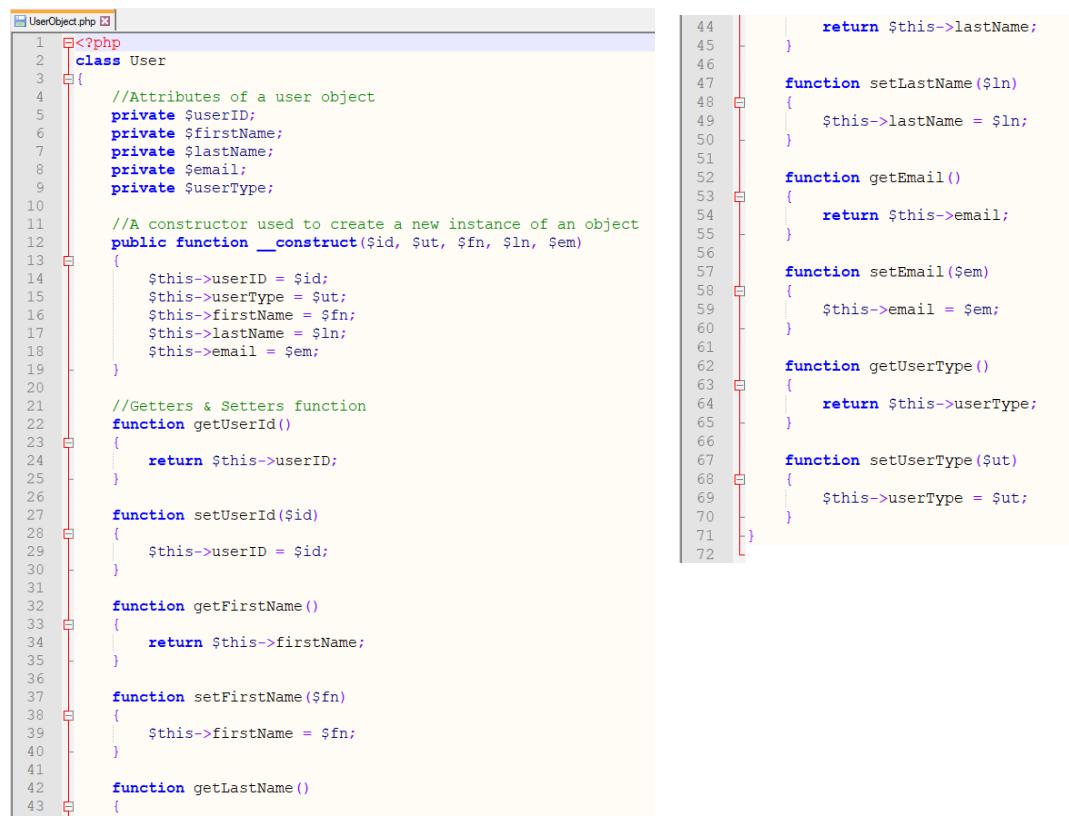
```

```

43
44 if (!filter_var($_POST["email"], FILTER_VALIDATE_EMAIL) || !isset($_POST["email"])) {
45     $errors[] = "email";
46 }
47
48 if (!isset($password) || !$isUppercase || !$isLowercase
49     || !$atLeastOneNumber || strlen($password) < 8) {
50     $errors[] = "password";
51 }
52
53 if ($password != $passwordCon) {
54     $errors[] = "passwordCon";
55 }
56
57 //Hashing the password using a built-in function
58 $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
59
60 if (!preg_match($pNumPattern, $_POST["pNum"])) {
61     $errors[] = "pNum";
62 }
63
64 if (!preg_match("/^[a-zA-Z]*$/", $_POST["country"])) {
65     $errors[] = "country";
66 }
67
68 if (!preg_match("/^[a-zA-Z]*$/", $_POST["city"])) {
69     $errors[] = "city";
70 }
71
72 if (!preg_match('/[A-Za-z0-9-\.\,]+/', $_POST["address"])) {
73     $errors[] = "address";
74 }
75
76 if (sizeof($errors) == 0) {
77     //if there are no errors make a link with the database
78     $mysqli = new mysqli("localhost", "root", "", "e-commerce");
79     //retrieve the details of the user from the users table based on their email
80     $stmt = $mysqli->prepare("SELECT * FROM `eusers` WHERE Email=?");
81     //using prepared statements to prevent SQL Injections
82     $stmt->bind_param("s", $email);
83     $stmt->execute();
84     $res = $stmt->get_result();
85
86     if ($res->num_rows > 0) {
87         $row = $res->fetch_assoc();
88         //Check to see if the email used to register already exists in the database
89         $errors[] = "email";
90         $_SESSION["allerrors"] = $errors;
91         $p = http_build_query($vals);
92         header("Location: /MyEcommerce/Register.php?error=Failed Validation&" . $p);
93         exit;
94     } else {
95         //Inserting the user's details to the database
96         $regQuery = "INSERT INTO `eusers`(`UserType`, `FirstName`, `LastName`, `Email`, `Password`,
97         `phoneNumber`, `Country`, `City`, `Address`, `Postcode`, `loginDate`)
98         VALUES ('1', ?, ?, ?, ?, ?, ?, ?, ?, ?, current_timestamp())";
99         $stmt1 = $mysqli->prepare($regQuery);
100        $stmt1->bind_param("ssssssss", $fname, $lname, $email, $hashedPassword,
101                           $pNum, $country, $city, $address, $postcode);
102        $stmt1->execute();
103        $stmt1->close();
104        header("Location: /MyEcommerce/Welcome.php");
105        exit;
106    }
107 } else {
108     $_SESSION["allerrors"] = $errors;
109     $p = http_build_query($vals);
110     header("Location: /MyEcommerce/Register.php?error=Failed Validation&" . $p);
111     exit;
112 }
113

```

## UserObject.php:

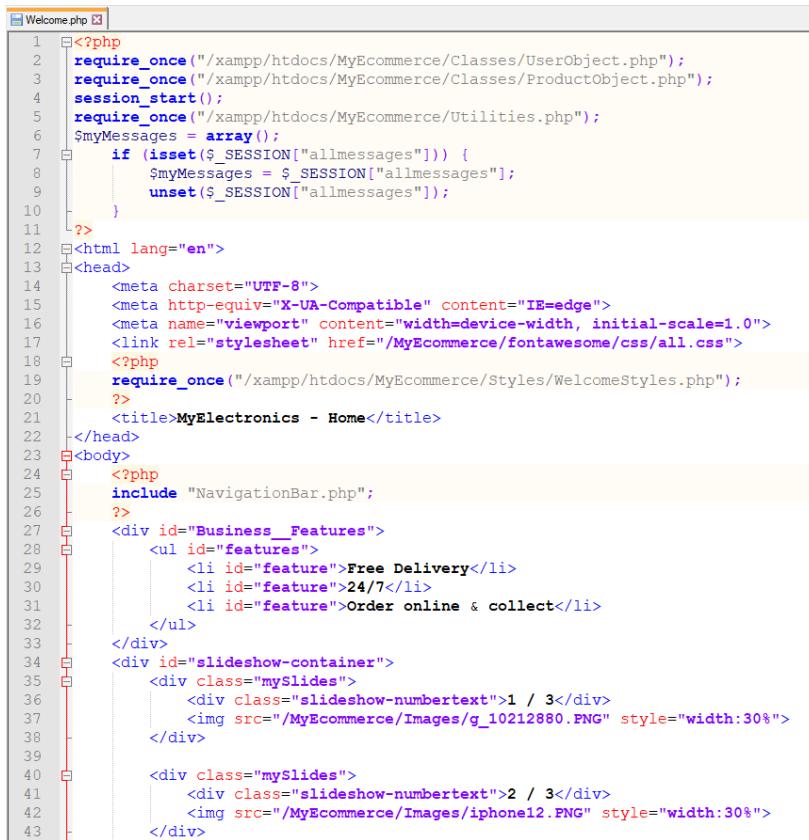


```

1  <?php
2  class User
3  {
4      //Attributes of a user object
5      private $userID;
6      private $firstName;
7      private $lastName;
8      private $email;
9      private $userType;
10
11     //A constructor used to create a new instance of an object
12     public function __construct($id, $ut, $fn, $ln, $em)
13     {
14         $this->userID = $id;
15         $this->userType = $ut;
16         $this->firstName = $fn;
17         $this->lastName = $ln;
18         $this->email = $em;
19     }
20
21     //Getters & Setters function
22     function getUserId()
23     {
24         return $this->userID;
25     }
26
27     function setUserId($id)
28     {
29         $this->userID = $id;
30     }
31
32     function getFirstName()
33     {
34         return $this->firstName;
35     }
36
37     function setFirstName($fn)
38     {
39         $this->firstName = $fn;
40     }
41
42     function getLastName()
43     {
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
    }

```

## Welcome.php:



```

1  <?php
2  require_once("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3  require_once("/xampp/htdocs/MyEcommerce/Classes/ProductObject.php");
4  session_start();
5  require_once("/xampp/htdocs/MyEcommerce/Utilities.php");
6  $myMessages = array();
7  if (isset($_SESSION["allmessages"])) {
8      $myMessages = $_SESSION["allmessages"];
9      unset($_SESSION["allmessages"]);
10 }
11 ?>
12 <html lang="en">
13 <head>
14     <meta charset="UTF-8">
15     <meta http-equiv="X-UA-Compatible" content="IE=edge">
16     <meta name="viewport" content="width=device-width, initial-scale=1.0">
17     <link rel="stylesheet" href="/MyEcommerce/fontawesome/css/all.css">
18     <?php
19     require_once("/xampp/htdocs/MyEcommerce/Styles/WelcomeStyles.php");
20     ?>
21     <title>MyElectronics - Home</title>
22 </head>
23 <body>
24     <?php
25     include "NavigationBar.php";
26     ?>
27     <div id="Business_Features">
28         <ul id="features">
29             <li id="feature">Free Delivery</li>
30             <li id="feature">24/7</li>
31             <li id="feature">Order online & collect</li>
32         </ul>
33     </div>
34     <div id="slideshow-container">
35         <div class="mySlides">
36             <div class="slideshow-numbertext">1 / 3</div>
37             
38         </div>
39
40         <div class="mySlides">
41             <div class="slideshow-numbertext">2 / 3</div>
42             
43         </div>

```

```

44
45     <div class="mySlides">
46         <div class="slideshow-numbertext">3 / 3</div>
47         
48     </div>
49     <br>
50     <div style="text-align:center">
51         <span class="dot"></span>
52         <span class="dot"></span>
53         <span class="dot"></span>
54     </div>
55
56
57     <div id="messages">
58         <?php
59             //Check to see which message to print after the user submits an order (after checkout)
60             if (in_array("card is already saved", $myMessages)) {
61                 echo "<p class='Messages'>This Card is already saved</p>";
62             } else if (in_array("card has been saved", $myMessages)) {
63                 echo "<p class='Messages'>Order has been submitted, and card has been saved</p>";
64             } else if (in_array("Order has been submitted", $myMessages)) {
65                 echo "<p class='Messages'>Order has been submitted</p>";
66             }
67         ?>
68     </div>
69
70     <div class="products">
71         <div class="best-selling">
72             <div id="title">
73                 <h1 style="text-align:center">Best Selling Products</h1>
74             </div>
75             <?php
76                 //load all the products from the database
77                 $theProducts = Product::loadAllProducts();
78                 for ($i = 0; $i < sizeof($theProducts); $i++) {
79                     //Loop through the array of products and check the status of each product
80                     if ($theProducts[$i]->getProductStatus() == "BestSelling")
81                     {
82                         //Print that product in this div
83                         echo $theProducts[$i]->printBestSellingProduct();
84                     }
85                 }
86                 if(isset($_REQUEST["StockError"])) {
87
88                     if(isset($_REQUEST["StockError"])) {
89                         echo "<p class='errorMessage'>Item Out Of Stock</p>";
90                     }
91                 }
92             </div>
93             //Creating a slideshow
94             var slideIndex = 0
95             showSlides()
96
97             function showSlides() {
98                 var i;
99                 var slides = document.getElementsByClassName("mySlides");
100                var dots = document.getElementsByClassName("dot");
101
102                //loop through the number of slides and display each slide
103                for (i = 0; i < slides.length; i++) {
104                    slides[i].style.display = "none";
105                }
106                slideIndex++;
107                //if all slides has been displayed, start again from the first slide
108                if (slideIndex > slides.length) slideIndex = 1
109
110                for (i = 0; i < dots.length; i++) {
111                    dots[i].className = dots[i].className.replace(" active", "");
112                }
113
114                slides[slideIndex - 1].style.display = "block";
115                dots[slideIndex - 1].className += " active";
116                setTimeout(showSlides, 2000);
117            }
118
119            //Returns the href (URL) of the current page and refresh the page,
120            //also send the product to the server (e.g. addToBasketProcess) to add it to the basket
121            function addToBasket(id) {
122                window.location.href = "Processors/addToBasketProcess.php?Product=" + id;
123            }
124
125            //Returns the href (URL) of the current page and refresh the page,
126            //also send the product to the server (e.g. addMyItemsProcess) to add it to the my items section
127            function addMyItems(id) {
128                window.location.href = "Processors/addMyItemsProcess.php?Product=" + id;
129            }
130
131            //It hides the messages printed after the user has submitted an order, after 3 seconds
132            var timeout = document.getElementById('messages');
133            setTimeout(hideElement, 3000); //milliseconds until timeout/
134            function hideElement() {
135                timeout.style.display = 'none';
136            }
137        </script>
138    </body>
139 </html>

```

## Basket.php:

```

1  <?php
2  require_once("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3  require_once("/xampp/htdocs/MyEcommerce/Utilities.php");
4  require_once("/xampp/htdocs/MyEcommerce/Classes/ProductObject.php");
5  require_once("/xampp/htdocs/MyEcommerce/Classes/BasketObject.php");
6  session_start();
7  isLoggedIn();
8  ?>
9  <html lang="en">
10 <head>
11   <meta charset="UTF-8">
12   <meta http-equiv="X-UA-Compatible" content="IE=edge">
13   <meta name="viewport" content="width=device-width, initial-scale=1.0">
14   <link rel="stylesheet" href="/MyEcommerce/fontawesome/css/all.css">
15   <?php
16   $myErrors = array();
17   //A check to see if there are any errors from the discount code input
18   if (isset($_SESSION["discountErrors"])) {
19     $myErrors = $_SESSION["discountErrors"];
20     unset($_SESSION["discountErrors"]);
21   }
22   ?>
23   <title>MyElectronics - Basket</title>
24   <script src="/MyEcommerce/jquery-3.6.0.min.js"></script>
25   <?php
26   require_once("/xampp/htdocs/MyEcommerce/Styles/BasketStyles.php");
27   ?>
28   <script>
29     //This function updates the total price in the basket based on the quantity input with the id amount
30     function updateTotal(id) {
31       var a = document.getElementById("amount" + id);
32       var url = "addToBasketProcess.php";
33       if (a.value < a.getAttribute("data-lastVal")) {
34         url = "removeFromBasket.php";
35       }
36
37       //Sending product data to the server (e.g. addToBasketProcess)
38       $.post("/MyEcommerce/Processors/" + url,
39             { Product: id, flag: true, quantity: a.value},
40             function( html ) {
41               $("#basketPage").html(html);
42             }
43       );
44     }
45     //Returns the href (URL) of the current page and refresh the page to remove a product from the the basket
46     function removeFromBasket(id) {
47       window.location.href = "Processors/removeFromBasket.php?Product=" + id;
48     }
49   </script>
50 </head>
51 <body>
52   <?php include "NavigationBar.php"; ?>
53   <div id="basketPage">
54   <?php
55     $basket = new Basket(array(), array());
56     if (isset($_SESSION["userBasket"])) {
57       $basket = $_SESSION["userBasket"];
58     }
59     //uses the function from the basket object to print the basket
60     echo $basket->printBasket();
61
62     if (in_array("No discount available", $myErrors)) {
63       echo "<p class='errorMessage'>Discount unavailable</p>";
64     }
65   ?>
66   </div>
67 </body>
68 </html>
69
70
71

```

## BasketObject.php:

```

BasketObject.php
1  <?php
2  require_once("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3  require_once("/xampp/htdocs/MyEcommerce/Classes/ProductObject.php");
4
5  class Basket {
6      //Basket Object Attributes
7      private $products;
8      private $discountCode;
9      private $discountPercentage;
10     private $discountAmount;
11
12     //A constructor used to create a new instance of a basket object
13     public function __construct($ps) {
14         $this->products = $ps;
15         $this->discountCode = null;
16         $this->discountPercentage = 0;
17         $this->discountAmount = 0;
18     }
19
20     //This function gets the product id from the product class
21     //Then checks if that product id exists in the products array in this class
22     //if yes it increases its quantity in the basket by one, if not then add it to the basket
23     function addToBasket($product) {
24         $productId = $product->getProductID();
25         if (array_key_exists($productId, $this->products)) {
26             $this->products[$productId] = [$product, $this->products[$productId][1] + 1];
27         } else {
28             $this->products[$productId] = [$product, 1];
29         }
30     }
31
32     //This works similar to add to basket,
33     //the only difference is that it unsets that product id from the products array
34     //which means deleting that item from the basket
35     function removeFromBasket($product) {
36         $productId = $product->getProductID();
37         if (array_key_exists($productId, $this->products)) {
38             unset($this->products[$productId]);
39             return true;
40         } else {
41             return false;
42         }
43     }
44
45     //Update quantity function is called in the removeFromBasket.php file
46     //To allow the user to decrease the quantity of an item
47     function updateQuantity($product, $quantity) {
48         if ($quantity == 0) {
49             $this->removeFromBasket($product);
50         } else {
51             $productId = $product->getProductID();
52             $this->products[$productId] = [$product, $quantity];
53         }
54     }
55
56     //This is a setter function that updates the values of the private attributes discountCode and discountPercentage
57     //This function is used in the discountProcess.php file
58     function addDiscountCode($code, $percentage) {
59         $this->discountCode = $code;
60         $this->discountPercentage = $percentage;
61     }
62
63     //Calculates the total price and returning the subtotal and VAT separately in an array
64     function getTotal() {
65         $total = 0;
66         $vat = 0;
67         $totalAmount = 0;
68         foreach($this->products as $id=>$details) {
69             $product = $details[0];
70             $total = $total + $product->getProductPrice() * $details[1];
71         }
72         $vat = $total * 0.2;
73         return [$total, $vat];
74     }
75
76     //Calculates the discount amount using percentage, total price and VAT
77     function getDiscountAmount() {
78         return $this->discountPercentage * ($this->getTotal()[0] + $this->getTotal()[1]);
79     }
80
81     //Calculates the sub total of an order
82     function getSubTotal() {
83         $subTotal = ($this->getTotal()[0] + $this->getTotal()[1]);
84         if ($this->discountCode != null) {

```

```

85     |     $sub_total = $sub_total - $this->getDiscountAmount();
86     |
87     |     return $sub_total;
88 }
89
90 //The print Basket function uses the HTML code that was hard-coded in the basket page
91 //The function prints the basket every time its called
92 function printBasket() {
93     //the values in the products array has been updated in the addToBasketProcess.php
94     //Therefore, now we set it to a variable so we can use it inside the print basket function
95     //and update the basket
96     $basket = $this->products;
97     $html = '<div class="basket">
98     <h1 id="title">Items in your Basket</h1>
99
100    if(isset($_REQUEST["error"])){
101        $html .= '<p>Unable to delete this Item</p>';
102    }
103
104    $html .= '<hr>';
105
106    //Using hardcoded HTML code to print each product with the below details such as image, description etc.
107    foreach($basket as $id=>$details) {
108        $product = $details[0];
109        $html .= '<div class="item" id="item'.$id.'>
110            <div id="remove">
111                <button id="removeButton" onclick="removeFromBasket('.$id.')">
112                    <i class="fas fa-trash-alt"></i></button>
113            </div>
114
115            <div class="image" id="image' . $id . '>
116                getProductName().'" style="width:50%">
117            </div>
118
119            <div class="description" id="description' . $id . '>
120                <span>' . $product->getProductDes() . '</span>
121            </div>
122
123            <div class="quantity" id="quantity' . $id . '>
124
125                <label for="quantity">Quantity</label>
126                <input data-lastVal="'.$details[1].'" onclick="updateTotal('.$id.')" id="amount' . $id . '" type="number" min="0" value="'.$details[1].'" style="width:50%">
127            </div>
128
129            <div id="itemPrice">
130                <input type="hidden" id="price' . $id . '" value="'.$product->getProductPrice().'">
131                <span>Price: £' . $product->getProductPrice() . '</span>
132            </div>
133
134            <div class="total" id="total' . $id . '>
135                <label for="total">Total:</label>
136                <span>£' . $product->getProductPrice() * $details[1] . '</span>
137            </div>
138        </div>';
139    }
140
141    $totals = $this->getTotal();
142    $total = $totals[0];
143    $vat = $totals[1];
144    $html .= '</div>
145
146    <form action="Processors/DiscountProcess.php" method="post">
147        <div class="totalprice">
148            <h1 id="title">Total</h1>
149            <span class="costs" >Sub-Total (ex VAT):<span style="margin-left:130px;"> £' . number_format($total, 2) . '</span></span>
150            <span class="costs">VAT (20%):<span style="margin-left:190px;"> £' . number_format($vat, 2) . '</span></span>';
151
152        if ($this->discountCode != null) { //if there is a valid discount code then apply it
153            //and output the discount amount
154            $html .= '<span class="costs">Discount Amount:<span style="margin-left:150px;"> £' . number_format($this->getDiscountAmount(), 2) . '</span></span>
155            <hr class="line"><span class="costs">Total:<span style="margin-left:150px;" name="totalAmount"> £' . number_format($this->getSubTotal(), 2) . '</span></span>';
156        } else { //otherwise, just output the sub-total
157            $html .= '<hr class="line"><span class="costs">Total:<span style="margin-left:150px;" name="totalAmount"> £' . number_format($this->getSubTotal(), 2) . '</span></span>';
158
159            <span class="costs">Discount Code: <input class="textInput" type="text" name="discountCode" />
160            <span style="margin-left:90px;">
161                <input type="submit" class="applyBtn" value="Apply"></input></span></span>';
162
163        }
164
165        //if there is no products in the basket and the user clicked on checkout, redirect them to the welcome page
166        //otherwise, redirect them to the checkout page
167
168    if (sizeof($this->products) == 0){
169        $html.= '<a href="Welcome.php" class="checkout">Check Out</a></div>
170        </form>';
171    } else {
172        $html.= '<a href="Checkout.php" class="checkout">Check Out</a></div>
173        </form>';
174    }
175
176    return $html;
177
178 }
179
180 ?>

```

## ProductObject.php:

```

ProductObject.php 3
1  <?php
2  class Product {
3
4      //Attributes of a product object
5      private $productID;
6      private $productName;
7      private $productStatus;
8      private $productPrice;
9      private $productImage;
10     private $productDescription;
11     private $productStock;
12
13     //A constructor used to create a new instance of a product object
14     public function __construct($id, $name, $status, $price, $image, $des, $st) {
15         $this->productID = $id;
16         $this->productName = $name;
17         $this->productStatus = $status;
18         $this->productPrice = $price;
19         $this->productImage = $image;
20         $this->productDescription = $des;
21         $this->productStock = $st;
22     }
23
24     //Getters & Setters functions
25     function getProductID()
26     {
27         return $this->productID;
28     }
29
30     function setProductID($id)
31     {
32         $this->productID = $id;
33     }
34
35     function getProductName()
36     {
37         return $this->productName;
38     }
39
40     function setProductName ($name)
41     {
42         $this->productName = $name;
43     }
44
45     function getProductStatus()
46     {
47         return $this->productStatus;
48     }
49
50     function setProductStatus($status)
51     {
52         $this->productStatus = $status;
53     }
54
55     function getProductPrice()
56     {
57         return $this->productPrice;
58     }
59
60     function setProductPrice ($price)
61     {
62         $this->productPrice = $price;
63     }
64
65     function getProductImage()
66     {
67         return $this->productImage;
68     }
69
70     function setProductImage ($image)
71     {
72         $this->productImage = $image;
73     }
74
75     function getProductDes()
76     {
77         return $this->productDescription;
78     }
79
80     function setProductDes ($des)
81     {
82         $this->productDescription = $des;
83     }
84

```

```

85     function getProductStock()
86     {
87         return $this->productStock;
88     }
89
90     function setProductStock($st)
91     {
92         $this->productStock = $st;
93     }
94
95     //This function use the hardcoded html code to print a product with all its details everytime its called
96     function printHTML() {
97         $html =
98             <div class="item" id="item' . $this->productID . '">
99                 <div id="remove">
100                     <button id="removeButton">Remove</button>
101                 </div>
102
103                 <div id="image' . $this->productID . '">
104                     
105                 </div>
106
107                 <div id="description' . $this->productID . '">
108                     <span>' . $this->productName . '</span>
109                     <span>' . $this->productDescription . '</span>
110                     <span>Silver</span>
111                 </div>
112
113                 <div id="quantity' . $this->productID . '">
114                     <label for="quantity">Quantity</label>
115                     <input onclick="updateTotal(1)" id="amount1" type="number" min="1" value="1">
116                 </div>
117
118                 <div id="itemPrice">
119                     <input type="hidden" id="price1" value="549">
120                     <span>Price: ' . $this->productPrice . '</span>
121                 </div>
122
123                 <div id="total1">
124                     <label for="total">Total:</label>
125                     <span>£549</span>
126                 </div>
127             </div>
128         ;
129         return $html;
130     }
131
132     //This function is used to print products in the welcome page as it holds the add to basket/my items buttons
133     //Even though its called print best selling products, it can be used for any product
134     function printBestSellingProduct() {
135         $html = '<div>
136             <div id="item1">
137
138                 <div id="image' . $this->productID . '">
139                     productName . '" style="width:40%">
140                 </div>
141
142                 <div id="name' . $this->productID . '">
143                     <span>' . $this->productName . '</span>
144                 </div>
145
146                 <div id="itemPrice">
147                     <span>Price: £' . $this->productPrice . '</span>
148                 </div>
149
150                 <div>
151                     <button id="basket-btn" onclick="addToBasket(' . $this->productID . ')">Add to Basket</button>
152                 </div>
153
154                 <div>
155                     <button id="myItems-btn" onclick="addMyItems(' . $this->productID . ')">
156                         <i class="far fa-heart"></i>
157                     </button>
158                 </div>
159             </div>';
160         ;
161         return $html;
162     }
163
164     //Retrieve the data of all the products in the database
165     public static function loadAllProducts() {
166         $mysqli = new mysqli("localhost", "root", "", "e-commerce");
167         $stmt = $mysqli->prepare("SELECT * FROM eProducts");
168         $stmt->execute();

```

```

169     $res = $stmt->get_result();
170     $stmt->close();
171     $products = [];
172     while ($row=$res->fetch_assoc()) {
173         //creating a product object
174         $myProduct = new Product($row["ProductID"], $row["ProductName"], $row["Status"],
175             $row["ProductPrice"], $row["ProductImage"], $row["ProductDescription"], $row["ProductStock"]);
176         $products[] = $myProduct;
177     }
178     return $products;
179 }
180
181 //Retrieve the data of a product with a given id
182 public static function loadProductById($id) {
183     $mysqli = new mysqli("localhost", "root", "", "e-commerce");
184     $stmt = $mysqli->prepare("SELECT * FROM eProducts WHERE ProductID = ?");
185     $stmt->bind_param("i", $id);
186     $stmt->execute();
187     $res = $stmt->get_result();
188     $stmt->close();
189     if ($res->num_rows == 1) {
190         $row = $res->fetch_assoc();
191         //creating a product object
192         $myProduct = new Product($row["ProductID"], $row["ProductName"], $row["Status"],
193             $row["ProductPrice"], $row["ProductImage"], $row["ProductDescription"], $row["ProductStock"]);
194         return $myProduct;
195     }
196     return null;
197 }
198
199 }
200 ?>
201

```

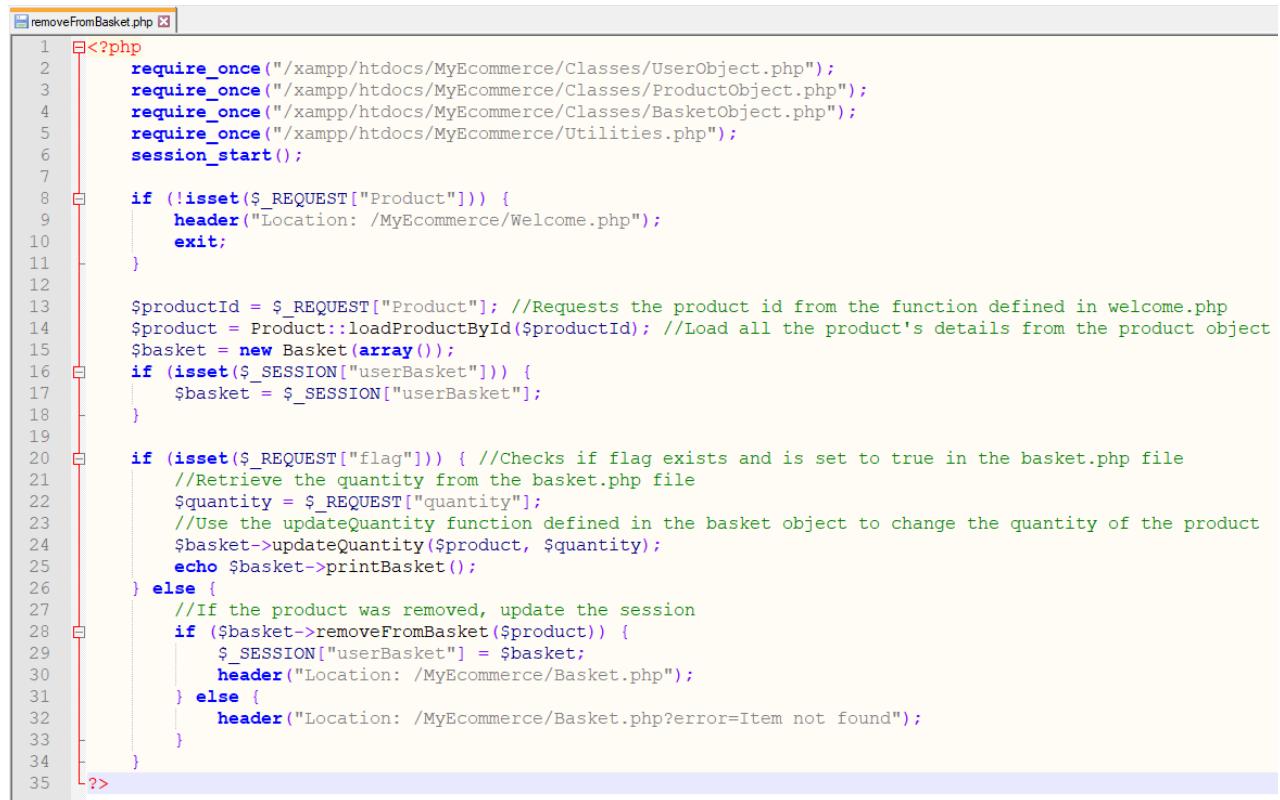
## addToBasketProcess.php:

```

addToBasketProcess.php
1 <?php
2     require_once("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3     require_once("/xampp/htdocs/MyEcommerce/Classes/ProductObject.php");
4     require_once("/xampp/htdocs/MyEcommerce/Classes/BasketObject.php");
5     require_once("/xampp/htdocs/MyEcommerce/Utilities.php");
6     session_start();
7
8     if (!isset($_REQUEST["Product"])) {
9         header("Location: /MyEcommerce/Welcome.php");
10        exit;
11    }
12
13     $productId = $_REQUEST["Product"]; //Requests the product id from the function defined in welcome.php
14     $product = Product::loadProductById($productId); //Load all the product's details from the product object
15     $basket = new Basket(array());
16     if (isset($_SESSION["userBasket"])) {
17         $basket = $_SESSION["userBasket"];
18     } //Checks if a basket exists in the session and if yes it adds it to the array
19
20     if ($product->getProductStock() > 0) {
21         $basket->addToBasket($product); //a function defined in produc object class
22         $_SESSION["userBasket"] = $basket;
23     } //if there is enough stock then add the product to the basket
24
25     if (isset($_REQUEST["flag"])) {
26         echo $basket->printBasket();
27     } //checks if the flag defined in basket.php is set to true and
28     //if yes print the basket and change any content e.g. quantity
29     else if ($product->getProductStock() > 0) {
30         header("Location: /MyEcommerce/Welcome.php");
31     } else {
32         header("Location: /MyEcommerce/Welcome.php?StockError=Item Out of Stock");
33     }
34
35 ?>

```

## removeFromBasket.php

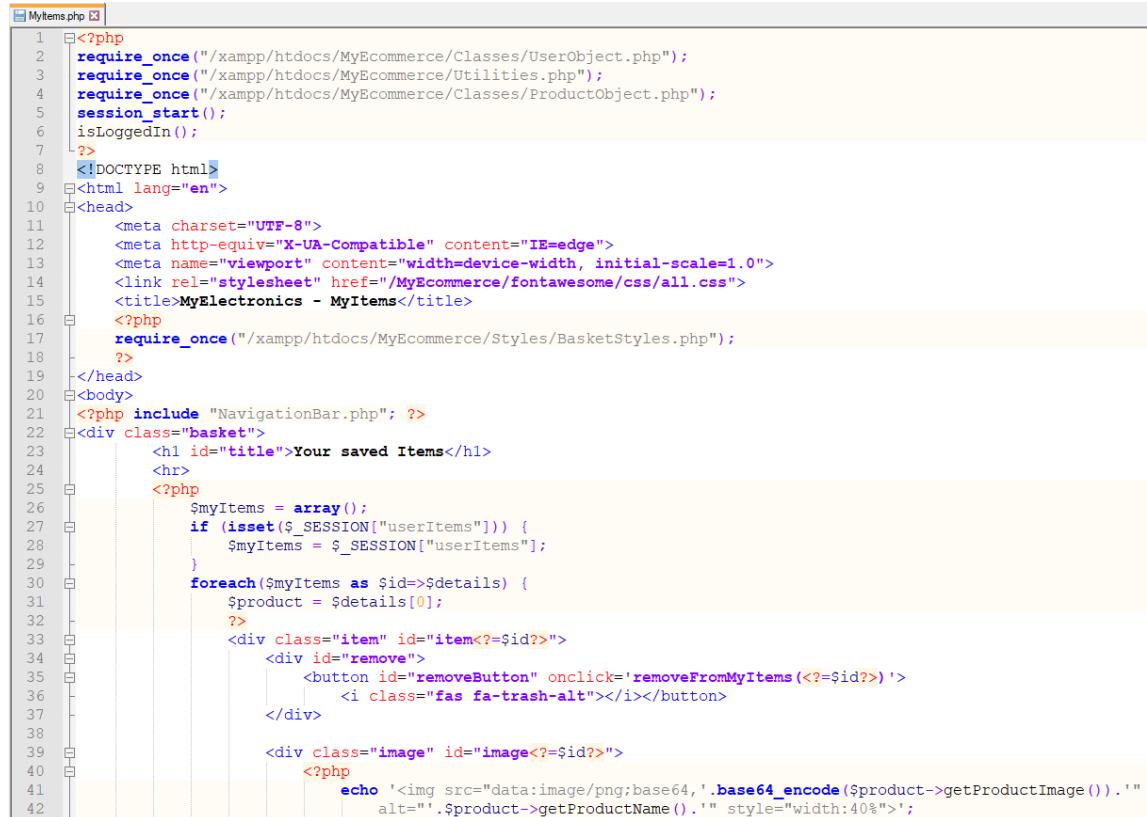


```

1  <?php
2  require_once("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3  require_once("/xampp/htdocs/MyEcommerce/Classes/ProductObject.php");
4  require_once("/xampp/htdocs/MyEcommerce/Classes/BasketObject.php");
5  require_once("/xampp/htdocs/MyEcommerce/Utilities.php");
6  session_start();
7
8  if (!isset($_REQUEST["Product"])) {
9      header("Location: /MyEcommerce/Welcome.php");
10     exit;
11 }
12
13 $productId = $_REQUEST["Product"]; //Requests the product id from the function defined in welcome.php
14 $product = Product::loadProductById($productId); //Load all the product's details from the product object
15 $basket = new Basket(array());
16 if (isset($_SESSION["userBasket"])) {
17     $basket = $_SESSION["userBasket"];
18 }
19
20 if (isset($_REQUEST["flag"])) { //Checks if flag exists and is set to true in the basket.php file
21     //Retrieve the quantity from the basket.php file
22     $quantity = $_REQUEST["quantity"];
23     //Use the updateQuantity function defined in the basket object to change the quantity of the product
24     $basket->updateQuantity($product, $quantity);
25     echo $basket->printBasket();
26 } else {
27     //If the product was removed, update the session
28     if ($basket->removeFromBasket($product)) {
29         $_SESSION["userBasket"] = $basket;
30         header("Location: /MyEcommerce/Basket.php");
31     } else {
32         header("Location: /MyEcommerce/Basket.php?error=Item not found");
33     }
34 }
35 ?>

```

## myItems.php:



```

1  <?php
2  require_once("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3  require_once("/xampp/htdocs/MyEcommerce/Utilities.php");
4  require_once("/xampp/htdocs/MyEcommerce/Classes/ProductObject.php");
5  session_start();
6  isLoggedIn();
7 ?>
8 <!DOCTYPE html>
9 <html lang="en">
10 <head>
11     <meta charset="UTF-8">
12     <meta http-equiv="X-UA-Compatible" content="IE=edge">
13     <meta name="viewport" content="width=device-width, initial-scale=1.0">
14     <link rel="stylesheet" href="/MyEcommerce/fontawesome/css/all.css">
15     <title>MyElectronics - MyItems</title>
16     <?php
17     require_once("/xampp/htdocs/MyEcommerce/Styles/BasketStyles.php");
18     ?>
19 </head>
20 <body>
21     <?php include "NavigationBar.php"; ?>
22     <div class="basket">
23         <h1 id="title">Your saved Items</h1>
24         <hr>
25         <?php
26             $myItems = array();
27             if (isset($_SESSION["userItems"])) {
28                 $myItems = $_SESSION["userItems"];
29             }
30             foreach($myItems as $id=>$details) {
31                 $product = $details[0];
32                 ?>
33                 <div class="item" id="item<?=$id?>">
34                     <div id="remove">
35                         <button id="removeButton" onclick='removeFromMyItems(<?=$id?>)'>
36                             <i class="fas fa-trash-alt"></i>
37                         </button>
38                     </div>
39                     <div class="image" id="image<?=$id?>">
40                         <?php
41                             echo 'getProductName() . '" style="width:40%">';
43                     </div>
44                 </div>
45             }
46         </div>
47     <?php
48     require_once("/xampp/htdocs/MyEcommerce/Styles/FooterStyles.php");
49     ?>
50 </body>
51 </html>

```

```

43
44
45
46    <?>
47    </div>
48
49    <div class="description" id="description<?=$id?>">
50        <span><?=$product->getProductDes() ?></span>
51    </div>
52
53    <div id="itemPrice">
54        <input type="hidden" id="price<?=$id?>" value="">
55        <span>Price: &lt;?=$product->getProductPrice() ?></span>
56    </div>
57
58    <div class="moveBtn">
59        <button id="moveButton" onclick="addToBasket(<?=$product->getProductID() ?>)">
60            <i class="fas fa-shopping-basket"> Move to Basket</i></button>
61    </div>
62
63    <?php
64
65        //Returns the href (URL) of the current page and refresh the page,
66        //also send the product to the server (e.g. addToBasketProcess) to add it to the basket
67        function addToBasket(id) {
68            window.location.href = "Processors/addToBasketProcess.php?Product=" + id;
69        }
70
71        //Returns the href (URL) of the current page and refresh the page and sends the product to the server
72        //to remove it from the the my items section
73        function removeFromMyItems(id) {
74            window.location.href = "Processors/removeFromMyItems.php?Product=" + id;
75        }
76    </script>
77</body>

```

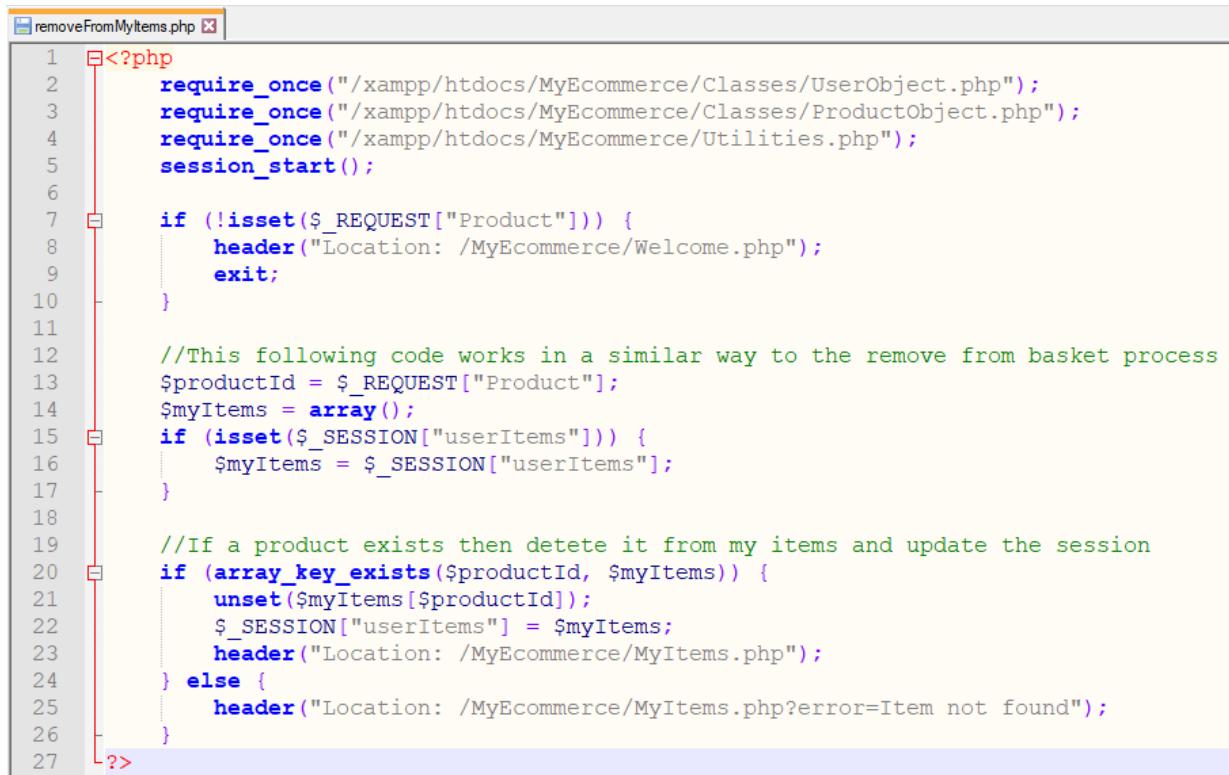
## addToMyItemsProcess.php:

```

addToMyItemsProcess.php
1 <?php
2     require_once("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3     require_once("/xampp/htdocs/MyEcommerce/Classes/ProductObject.php");
4     require_once("/xampp/htdocs/MyEcommerce/Utilities.php");
5     session_start();
6
7     if (!isset($_REQUEST["Product"])) {
8         header("Location: /MyEcommerce/Welcome.php");
9         exit;
10    }
11
12    $productId = $_REQUEST["Product"]; //Request the product id from the $_post function defined in welcome.php
13    $product = Product::loadProductById($productId); //Load all the product's details from the product object
14    $myItems = array();
15    if (isset($_SESSION["userItems"])) {
16        $myItems = $_SESSION["userItems"];
17    } //Check if a my items section exists in the session and if yes it adds it to the array
18
19    if (array_key_exists($productId, $myItems)) { //check if a specific product id exists in my items array
20        echo "Item already exists";
21    } else {
22        $myItems[$productId] = [$product]; //add the product id to the array
23    }
24
25    $_SESSION["userItems"] = $myItems; //store the array myItems in the session
26    header("Location: /MyEcommerce/Welcome.php");
27 ?>

```

## removeFromMyItems.php:

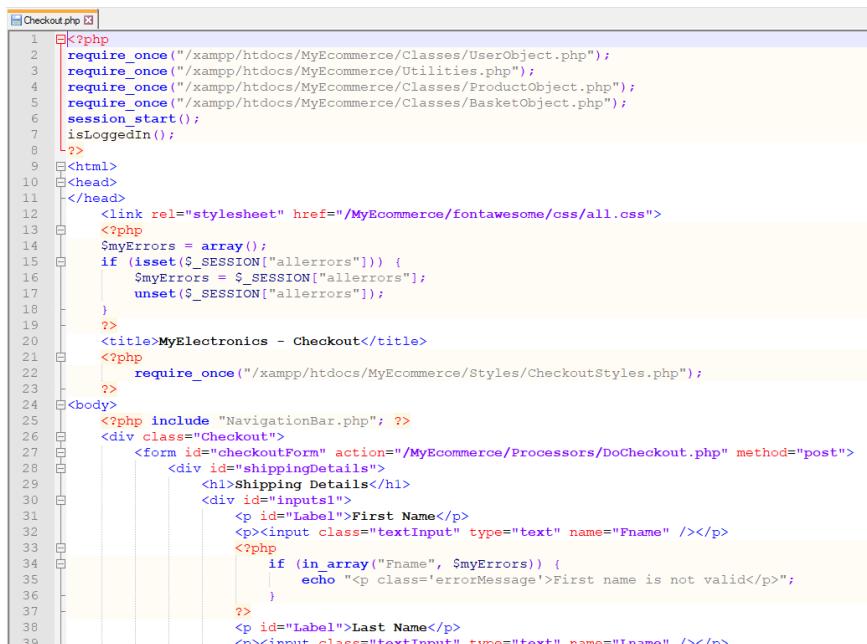


```

1 <?php
2     require_once ("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3     require_once ("/xampp/htdocs/MyEcommerce/Classes/ProductObject.php");
4     require_once ("/xampp/htdocs/MyEcommerce/Utilities.php");
5     session_start();
6
7     if (!isset($_REQUEST["Product"])) {
8         header("Location: /MyEcommerce/Welcome.php");
9         exit;
10    }
11
12    //This following code works in a similar way to the remove from basket process
13    $productId = $_REQUEST["Product"];
14    $myItems = array();
15    if (isset($_SESSION["userItems"])) {
16        $myItems = $_SESSION["userItems"];
17    }
18
19    //If a product exists then delete it from my items and update the session
20    if (array_key_exists($productId, $myItems)) {
21        unset($myItems[$productId]);
22        $_SESSION["userItems"] = $myItems;
23        header("Location: /MyEcommerce/MyItems.php");
24    } else {
25        header("Location: /MyEcommerce/MyItems.php?error=Item not found");
26    }
27
?>

```

## Checkout.php:



```

1 <?php
2     require_once ("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3     require_once ("/xampp/htdocs/MyEcommerce/Utilities.php");
4     require_once ("/xampp/htdocs/MyEcommerce/Classes/ProductObject.php");
5     require_once ("/xampp/htdocs/MyEcommerce/Classes/BasketObject.php");
6     session_start();
7     isLoggedIn();
8 ?>
9 <html>
10 <head>
11     <link rel="stylesheet" href="/MyEcommerce/fontawesome/css/all.css">
12     <?php
13     $myErrors = array();
14     if (isset($_SESSION["allerrors"])) {
15         $myErrors = $_SESSION["allerrors"];
16         unset($_SESSION["allerrors"]);
17     }
18     ?>
19     <title>MyElectronics - Checkout</title>
20     <?php
21         require_once ("/xampp/htdocs/MyEcommerce/Styles/CheckoutStyles.php");
22     ?>
23 <body>
24     <?php include "NavigationBar.php"; ?>
25     <div class="Checkout">
26         <form id="checkoutForm" action="/MyEcommerce/Processors/DoCheckout.php" method="post">
27             <div id="shippingDetails">
28                 <h1>Shipping Details</h1>
29                 <div id="inputs1">
30                     <p id="Label">First Name</p>
31                     <p><input class="textInput" type="text" name="Fname" /></p>
32                     <?php
33                         if (in_array("Fname", $myErrors)) {
34                             echo "<p class='errorMessage'>First name is not valid</p>";
35                         }
36                     ?>
37                     <p id="Label">Last Name</p>
38                     <p><input class="textInput" type="text" name="Lname" /></p>

```



## DoCheckout.php:

```

DoCheckout.php
1 <?php
2
3 require_once (" /xampp/htdocs/MyEcommerce/Utilities.php");
4 require_once (" /xampp/htdocs/MyEcommerce/Classes/UserObject.php");
5 require_once (" /xampp/htdocs/MyEcommerce/Classes/BasketObject.php");
6 session_start();
7
8 //Retrieving the data inputted by the user in the checkout page
9 $fname = $_POST["fname"];
10 $lname = $_POST["lname"];
11 $pnum = $_POST["pnum"];
12 $country = $_POST["country"];
13 $address = $_POST["address"];
14 $postcode = $_POST["postcode"];
15 $cardnum = $_POST["cardnum"];
16 $cvv = $_POST["cvv"];
17 $savecard = null;
18 if (isset($_POST["savecard"])) {
19     $savecard = $_POST["savecard"];
20 } else {
21     $savecard = null;
22 }
23
24 $exp = $_POST["exp"];
25 //Make the expiration date in the format month/year
26 $carddate = DateTime::createFromFormat('m/y', $exp);
27 $currentdate = new DateTime('now');
28
29 //A variable that stores the difference between the current date and the card date
30 $interval = $currentdate->diff($carddate);
31
32 $pnumpattern = "/^(\\+44\\s?7\\d{3})|(\\d{7}\\s?\\d{3}\\s?\\d{3})$/";
33
34 $messages = array();
35 $errors = array();
36 $vals = array();
37
38 foreach (["fname", "lname", "pnum", "country", "address", "postcode", "cardnum", "exp"] as $key) {
39     //A check to see if the user has inputted something into the fields or left it blank
40     if (!isset($_POST[$key]) || strlen($_POST[$key]) == 0) {
41         $errors[] = $key;
42     } else {
43         $vals[$key] = $_POST[$key];
44     }
45 }
46
47 //Validating all the inputs
48 if (!preg_match("/^a-zA-Z+$/i", $fname)) {
49     $errors[] = "Fname";
50 }
51
52 if (!preg_match("/^a-zA-Z+$/i", $lname)) {
53     $errors[] = "Lname";
54 }
55
56 if (!preg_match($pnumpattern, $POST["pnum"])) {
57     $errors[] = "pnum";
58 }
59
60 if (!preg_match("/^a-zA-Z+$/i", $POST["country"])) {
61     $errors[] = "country";
62 }
63
64 if (!preg_match('/[A-Z-a-z-0-9-\\.,]+/', $POST["address"])) {
65     $errors[] = "address";
66 }
67
68 //Validating bank card number
69 function validatecard($number)
70 {
71     global $type;
72
73     $cardtype = array(
74         "visa" => "/^4[0-9]{12}(?:[0-9]{3})?$/",
75         "mastercard" => "/^5[1-5][0-9]{14}$/",
76         "amex" => "/^3[47][0-9]{13}$/",
77         "discover" => "/^6(?:011|5[0-9]{2})[0-9]{12}$/",
78     );
79 }
80

```

```

79
80
81     if (preg_match($cardtype['visa'],$number))
82     {
83         $type= "visa";
84         return "visa";
85     }
86     else if (preg_match($cardtype['mastercard'],$number))
87     {
88         $type= "mastercard";
89         return "mastercard";
90     }
91     else if (preg_match($cardtype['amex'],$number))
92     {
93         $type= "amex";
94         return "amex";
95     }
96     else if (preg_match($cardtype['discover'],$number))
97     {
98         $type= "discover";
99         return "discover";
100    }
101   else
102   {
103     return false;
104   }
105 }
106 }
107
108 if (validatecard($cardNum) == false) {
109   $errors[] = "cardNum";
110 }
111
112 //Check to see if the interval is a negative value
113 // 1 means negative, 0 means otherwise
114 if ($interval->invert == 1) {
115   $errors[] = "exp";
116 }
117
118 if (!preg_match('/^([0-9]{3,4})$/', $_POST["cvv"])) {
119   $errors[] = "cvv";
120 }
121
122 //Retrieving the user id from the session using the getter from the user object
123 $userId = $_SESSION["myUser"]->getUserId();
124
125 if (sizeof($errors) == 0) {
126   $basket = new Basket(array());
127   if (isset($_SESSION["userBasket"])) {
128     $basket = $_SESSION["userBasket"];
129   }
130   $price = $basket->getSubTotal();
131   $mysql = new mysqli("localhost", "root", "", "e-commerce");
132   //Accessing card details in the payment methods table to check if it already exists
133   if (isset($saveCard)) {
134     $stmt = $mysql->prepare("SELECT * FROM `epayment_methods` WHERE CardNumber=?");
135     $stmt->bind_param("i", $cardNum);
136     $stmt->execute();
137     $res = $stmt->get_result();
138     $stmt->close();
139     if ($res->num_rows == 1) {
140       $row = $res->fetch_assoc();
141       $messages[] = "card is already saved";
142     } else {
143       //Saving the card into the database using SQL
144       $regQuery = "INSERT INTO `epayment_methods` ('CustomerID', 'CardNumber', 'Exp_Date', 'CVV') VALUES (?, ?, ?, ?)";
145       $stmt2 = $mysql->prepare($regQuery);
146       $stmt2->bind_param("iisi", $userId, $cardNum, $exp, $cvv);
147       $stmt2->execute();
148       $stmt2->close();
149       $messages[] = "card has been saved";
150     }
151   }
152   if (!isset($saveCard)){
153     $orderQuery = "INSERT INTO `orders` ('UserID', 'PaymentID', 'totalPrice') VALUES (?, ?, ?)";
154     $stmt1 = $mysql->prepare($orderQuery);
155     $stmt1->bind_param("iisi", $userId, $cardNum, $price);
156     $stmt1->execute();
157     $stmt1->close();
158   }
159   //Adding the order to the orders table
160   $orderQuery = "INSERT INTO `orders` ('UserID', 'PaymentID', 'totalPrice') VALUES (?, ?, ?)";
161   $stmt1 = $mysql->prepare($orderQuery);
162   $stmt1->bind_param("iisi", $userId, $row["CardID"], $price);
163   $stmt1->execute();
164   $stmt1->close();
165   $messages[] = "Order has been submitted";
166   $_SESSION["allmessages"] = $messages;
167   header("Location: /MyEcommerce/Welcome.php");
168   exit;
169 } else {
170   $_SESSION["allerrors"] = $errors;
171   $p = http_build_query($vals);
172   header("Location: /MyEcommerce/Checkout.php?error=Failed Validation" . $p);
173   exit;
174 }
175
?
```

## LogOut.php:

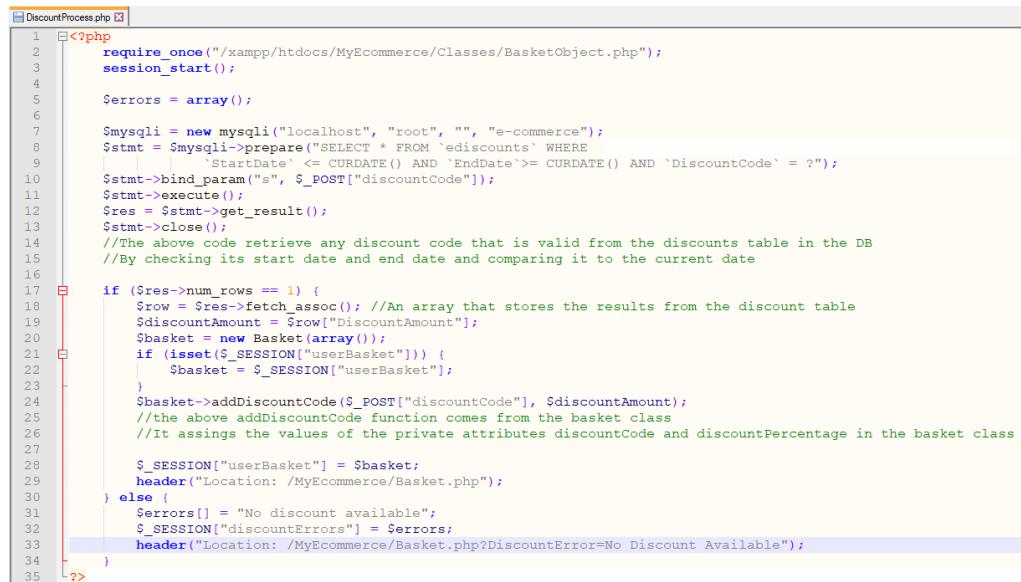


```

1 <?php
2 //Log the users out by removing them from the session
3 require_once("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
4 session_start();
5 session_unset();
6 session_destroy();
7
8 header("Location: /MyEcommerce/index.php");
9

```

## DiscountProcess.php:

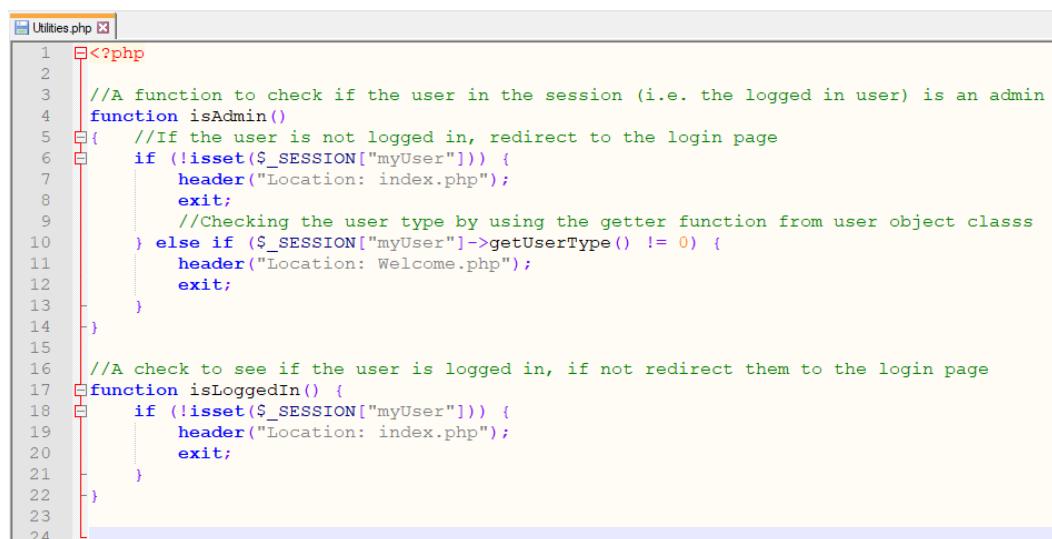


```

1 <?php
2 require_once("/xampp/htdocs/MyEcommerce/Classes/BasketObject.php");
3 session_start();
4
5 $errors = array();
6
7 $mysqli = new mysqli("localhost", "root", "", "e-commerce");
8 $stmt = $mysqli->prepare("SELECT * FROM `ediscounts` WHERE
9 `Startdate` <= CURDATE() AND `Enddate`>= CURDATE() AND `DiscountCode` = ?");
10 $stmt->bind_param("s", $_POST["discountCode"]);
11 $stmt->execute();
12 $res = $stmt->get_result();
13 $stmt->close();
14 //The above code retrieve any discount code that is valid from the discounts table in the DB
15 //By checking its start date and end date and comparing it to the current date
16
17 if ($res->num_rows == 1) {
18     $row = $res->fetch_assoc(); //An array that stores the results from the discount table
19     $discountAmount = $row["DiscountAmount"];
20     $basket = new Basket(array());
21     if (isset($_SESSION["userBasket"])) {
22         $basket = $_SESSION["userBasket"];
23     }
24     $basket->addDiscountCode($_POST["discountCode"], $discountAmount);
25     //the above addDiscountCode function comes from the basket class
26     //It assings the values of the private attributes discountCode and discountPercentage in the basket class
27
28     $_SESSION["userBasket"] = $basket;
29     header("Location: /MyEcommerce/Basket.php");
30 } else {
31     $errors[] = "No discount available";
32     $_SESSION["discountErrors"] = $errors;
33     header("Location: /MyEcommerce/Basket.php?DiscountError=No Discount Available");
34 }
35 ?>

```

## Utilities.php:



```

1 <?php
2
3 //A function to check if the user in the session (i.e. the logged in user) is an admin
4 function isAdmin()
5 {
6     //If the user is not logged in, redirect to the login page
7     if (!isset($_SESSION["myUser"])) {
8         header("Location: index.php");
9         exit;
10    //Checking the user type by using the getter function from user object classs
11    } else if ($_SESSION["myUser"]->getUserType() != 0) {
12        header("Location: Welcome.php");
13        exit;
14    }
15
16 //A check to see if the user is logged in, if not redirect them to the login page
17 function isLoggedIn() {
18     if (!isset($_SESSION["myUser"])) {
19         header("Location: index.php");
20         exit;
21     }
22 }
23

```

## NavigationBar.php:

```

1  <header id="navBar">
2    <div id="mySidenav" class="sidenav">
3      <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>
4      <div class="welcome">
5        <i class="fas fa-user-circle"> </i>
6        <span class="welcomeText">
7          Hello,
8          <?php
9            //Check to see if the user is in the session (i.e. logged in), then print their first name
10           //Otherwise, print a message saying Sign in
11           if(isset($_SESSION["myUser"])) {
12             echo $_SESSION["myUser"]->getFirstName();
13           } else {
14             echo "Sign in";
15           }
16         ?>
17       !
18     </span>
19   </div>
20   <hr>
21   <div class="main_sections">
22     <a href="#">Purchase History</a>
23     <a href="#">Bestsellers</a>
24     <a href="#">Deals</a>
25     <a href="#">New Releases</a>
26   </div>
27   <hr>
28   <div class="sidebar_headings">
29     <h1 class="welcomeText">Categories</h1>
30     <ul class="sidebar_links">
31       <li>
32         <a href="#">Laptops</a>
33       </li>
34       <li>
35         <a href="#">Smart Phones</a>
36       </li>
37       <li>
38         <a href="#">Tablets</a>
39       </li>
40       </ul>
41     </div>
42     <a href="#">TVs</a>
43   </li>
44   <li>
45     <a href="#">Electronic Accessories</a>
46   </li>
47 </ul>
48 </div>
49 <hr>
50 <div class="sidebar_headings">
51   <h1 class="welcomeText">Help & Settings</h1>
52   <ul class="sidebar_links">
53     <li>
54       <a href="#">Contact Us</a>
55     </li>
56     <li>
57       <a href="#">About Us</a>
58     </li>
59     <li>
60       <a href="#">Currency settings</a>
61     </li>
62     <li>
63       <a href="Processors/LogOut.php">
64         <?php
65           //if the user is in the session (i.e. logged in) make the function of the link to log out
66           if(isset($_SESSION["myUser"])) {
67             echo "Log Out";
68           }
69         ?>
70       </a>
71       <a href="index.php">
72         <?php
73           //if the user is in the session (i.e. logged in) make the function of the link to log in
74           if(!isset($_SESSION["myUser"])) {
75             echo "Log In";
76           }
77         ?>
78       </a>

```

```

79         </li>
80     </ul>
81   </div>
82 </div>
83 <span style="font-size:25px;cursor:pointer;color:white" onclick="openNav()">☰</span>
84 <a href="Welcome.php">h1 id="logo" MyElectronics <i class="fas fa-tag"></i> </h1></a>
85
86 <input type="text" placeholder="Search..." name="searchBar" id="search">
87 <nav>
88   <ul class="nav__links">
89     <li>
90       <div class="dropdown">
91         <i class="fas fa-user-circle"></i>
92         <button class="dropbtn">My Account</button>
93         <div class="dropdown-content">
94           <?php
95           if (isset($_SESSION["myUser"])) {
96             ?>
97             <a href="#">Purchase History</a>
98             <a href="Processors/LogOut.php">Log Out</a>
99           <?php } else { ?>
100             <a href="index.php">Log In</a>
101           <?php } ?>
102         </div>
103       </div>
104     </li>
105     <li>
106       <a href="MyItems.php"><i class="far fa-heart"></i> My Items</a>
107     </li>
108     <li><a href="Basket.php"><i class="fas fa-shopping-basket"></i> Basket</a></li>
109   <?php
110   //Check the user's user type, if its 0 then make the admin link appear in the header
111   if (isset($_SESSION["myUser"])) && $_SESSION["myUser"]->getUserType() == 0) {
112     ?>
113     <li><a href="Admin.php"><i class="fas fa-user-tie"></i> Admin</a></li>
114   <?php } ?>
115   </ul>
116 </nav>
117 </header>
118 <script>
119   //functions to open and collapse the sidebar navigation by chaning its width
120   function openNav() {
121     document.getElementById("mySidenav").style.width = "300px";
122   }
123
124   function closeNav() {
125     document.getElementById("mySidenav").style.width = "0";
126   }
127 </script>

```

## Admin.php:



```

1 <?php
2   require_once("/xampp/htdocs/MyEcommerce/Classes/UserObject.php");
3   session_start();
4   require_once("/xampp/htdocs/MyEcommerce/Utilities.php");
5   isAdmin();
6   ?>
7   <!DOCTYPE html>
8   <html lang="en">
9     <head>
10       <meta charset="UTF-8">
11       <meta http-equiv="X-UA-Compatible" content="IE=edge">
12       <meta name="viewport" content="width=device-width, initial-scale=1.0">
13       <link rel="stylesheet" href="/MyEcommerce/fontawesome/css/all.css">
14       <title>MyElectronics - Admin</title>
15     <?php
16       require_once("/xampp/htdocs/MyEcommerce/Styles/header.php");
17     ?>
18   </head>
19   <body>
20     <?php include "NavigationBar.php"; ?>
21   </body>
22 </html>

```

## AccountsStyles.php:

```

AccountsStyles.php
1 <style>
2   @import url("https://fonts.googleapis.com/css2?family=Blinker&display=swap");
3   @import url("https://fonts.googleapis.com/css2?family=Sunflower:wght@300&display=swap");
4
5   ***** FORM STYLES *****
6   #RegDiv {
7     margin: 100% auto;
8     margin-top: 75px;
9     margin-right: auto;
10    margin-left: auto;
11    width: 400px;
12    height: 475px;
13    border: 2px solid orange;
14    border-radius: 7px;
15    padding: 0% 2px;
16    background-color: #00224B;
17  }
18
19  #LoginDiv {
20   margin: 100% auto;
21   margin-top: 10px;
22   margin-bottom: 50px;
23   margin-right: auto;
24   margin-left: auto;
25   width: 400px;
26   height: 375px;
27   border: 2px solid orange;
28   border-radius: 7px;
29   padding: 0% 2px;
30   background-color: #00224B;
31 }
32
33  #loginForm {
34   width: 150px;
35   margin: 45 auto;
36   color: white;
37 }
38
39  #ReqForm {
40   width: 80px;
41   margin: 15 auto;
42   color: white;
43 }
44
45  .textInput {
46   border: 1px solid orange;
47   margin: 5% 5px;
48   padding: 2 30px;
49   border-radius: 5px;
50 }
51
52  .standardButton {
53   border: 1px solid black;
54   padding: 2 25px;
55   border-radius: 5px;
56   margin: 5 200px;
57 }
58
59  .standardButton:hover {
60   border: 1px solid black;
61   padding: 2 25px;
62   background: orange;
63 }
64
65  #regLink {
66   margin: 10 150px;
67   font-size: 20px;
68 }
69
70  #regLink:hover {
71   color: #ffc04c;
72 }
73
74  .errorMessage {
75   color: red;
76 }
77
78 </style>
<?php require once("./xampp/htdocs/MyEcommerce/Styles/header.php"); ?>

```

## BasketStyles.php:

```

BasketStyles.php
1 <style>
2   * {
3     box-sizing: border-box;
4   }
5
6   html,
7   body {
8     width: 100%;
9     height: 100%;
10    margin: 0;
11  }
12
13  #title {
14    padding: 10px 20px 30px 20px;
15    height: 60px;
16    border-bottom: 1px solid #ffc04c;
17    font-size: 36px;
18    font-weight: 400;
19  }
20
21  .basket {
22    width: 50%;
23    height: 50%;
24    margin: 80px 40px;
25    background: #FFFFFF;
26    box-shadow: 1px 2px 3px 0px rgba(0,0,0,0.10);
27    border-radius: 6px;
28    display: flex;
29    flex-direction: column;
30    float: left;
31  }
32
33  .item {
34    padding: 20px 30px;
35    height: 120px;
36    display: flex;
37    justify-content: space-between;
38  }
39
40  .item:nth-child(3) {
41    border-bottom: 5px solid #E1E8EE;
42  }
43
44  #remove {
45    position: relative;
46    padding-top: 5px;
47    margin-right: 30px;
48    align-self: center;
49  }
50
51  #removeButton {
52    display: inline-block;
53    cursor: pointer;
54    align-self: center;
55    width: 50px;
56    height: 50px;
57    background: none;
58    border-style: none;
59  }
60
61  .image {
62    margin-right: 10px;
63    margin-left: 10px;
64    width: 500px;
65    align-self: center
66  }
67
68  .description {
69    padding-top: 5px;
70    margin-right: 40px;
71    width: 250px;
72    align-self: center
73  }
74
75  .description span {
76    display: block;
77    font-size: 14px;
78    color: #43484D;

```

```

79   font-weight: 400;
80   align-self: center
81 }
82
83 .description span:first-child {
84   margin-bottom: 5px;
85   align-self: center
86 }
87 .description span:last-child {
88   font-weight: 300;
89   margin-top: 8px;
90   align-self: center
91 }
92
93 .quantity {
94   padding-top: 5px;
95   margin-right: 30px;
96   align-self: center
97 }
98
99 .quantity input {
100  -webkit-appearance: none;
101  border: 1px solid grey;
102  border-radius: 5px;
103  text-align: center;
104  width: 30px;
105  font-size: 16px;
106  color: #43484D;
107  font-weight: 200;
108  align-self: center
109 }
110
111 #itemPrice {
112   width: 83px;
113   padding-top: 5px;
114   margin-right: 12px;
115   text-align: center;
116   font-size: 16px;
117 }
118 color: #43484D;
119 font-weight: 300;
120 align-self: center;
121 }
122
123 .total {
124   width: 100px;
125   padding-top: 5px;
126   text-align: center;
127   font-size: 16px;
128   color: #43484D;
129   font-weight: 300;
130   align-self: center;
131 }
132
133 #moveButton {
134   padding-top: 20px;
135   display: inline-block;
136   cursor: pointer;
137   align-self: center;
138   width: 50px;
139   height: 50px;
140   background: none;
141   border-style: none;
142 }
143
144 .totalPrice {
145   width: 30%;
146   height: 35%;
147   margin: 80px 80px;
148   background: #FFFFFF;
149   box-shadow: 1px 2px 3px 0px rgba(0,0,0,0.10);
150   border-radius: 6px;
151   display: flex;
152   flex-direction: column;
153   float: right;
154 }
155
156 .costs {

```

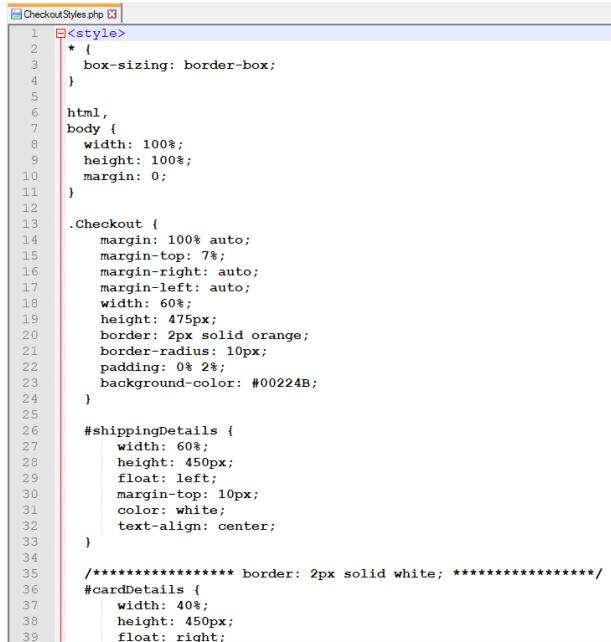
```

157 padding: 10px 20px 30px 20px;
158 height: 40px;
159 font-size: 16px;
160 font-weight: 400;
161 display: flex;
162 justify-content: space-between;
163 }
164
165 .checkout {
166 width: 50%;
167 color: white;
168 background-color: orange;
169 border-radius: 5px;
170 border: none;
171 margin-left: auto;
172 margin-right: auto;
173 margin-top: 10px;
174 text-align: center;
175 }
176
177 .applyBtn {
178 width: 150%;
179 color: white;
180 background-color: orange;
181 border-radius: 6px;
182 border: none;
183 margin-top: 5px;
184 }
185
186 .checkout:hover {
187 background-color: #e89805;
188 }
189
190 .applyBtn:hover {
191 background-color: #e89805;
192 }
193
194 .line {
195 width: 90%;

196 margin-left: auto;
197 margin-right: auto;
198 }
199
200 .textInput {
201 width: 120px;
202 height: 25px;
203 border: none;
204 border-bottom: 2px solid orange;
205 }
206
207 .errorMessage {
208 color: red;
209 margin-left: 500px;
210 }
211
212 </style>
213 <?php require_once("/xampp/htdocs/MyEcommerce/Styles/header.php"); ?>

```

## CheckoutStyles.php:



```

1 <style>
2 *
3 {
4   box-sizing: border-box;
5 }
6
7 html,
8 body {
9   width: 100%;
10  height: 100%;
11  margin: 0;
12 }
13 .Checkout {
14   margin: 100% auto;
15   margin-top: 7%;
16   margin-right: auto;
17   margin-left: auto;
18   width: 60%;
19   height: 475px;
20   border: 2px solid orange;
21   border-radius: 10px;
22   padding: 0 2px;
23   background-color: #00224B;
24 }
25
26 #shippingDetails {
27   width: 60%;
28   height: 450px;
29   float: left;
30   margin-top: 10px;
31   color: white;
32   text-align: center;
33 }
34
35 /****** border: 2px solid white; ******/
36 #cardDetails {
37   width: 40%;
38   height: 450px;
39   float: right;

```

```

40     margin-top: 10px;
41     color: white;
42     font-size: 14px;
43     text-align: center;
44   }
45
46   .textInput {
47     padding: 3 20px;
48     margin: 12px 20px 12px 20px;
49     border: none;
50     border-bottom: 2px solid orange;
51     text-align: center;
52   }
53
54   #inputs1 {
55     margin-top: 15px;
56   }
57
58   #cardNum {
59     margin-top: 22px;
60   }
61
62   .applyBtn {
63     width: 25%;
64     height: 5%;
65     color: white;
66     background-color: orange;
67     border-radius: 6px;
68     border: none;
69     margin-top: 70px;
70     margin-left: 70px;
71   }
72
73   .applyBtn:hover {
74     background-color: #e89805;
75   }
76
77   #saveCard {
78     margin-top: 5px;
79     margin-left: 5px;
80   }
81
82   .errorMessage {
83     color: red;
84   }
85
86
87   </style>
88   <?php require_once("/xampp/htdocs/MyEcommerce/Styles/header.php"); ?>

```

## Header.php:

```

1 <style>
2   @import url("https://fonts.googleapis.com/css2?family=Sunflower:wght@300&display=swap");
3
4   /****** HEADER STYLES *****/
5
6   *
7   * {
8     box-sizing: border-box;
9     margin: 0;
10    padding: 0;
11  }
12
13  html {
14    height: 100%;
15  }
16
17  body {
18    font-family: "Sunflower", sans-serif;
19    height: 100%;
20  }
21
22  li,
23  a {
24    font-family: "Sunflower", sans-serif;
25    font-weight: 500px;
26    font-size: 22px;
27    color: white;
28    text-decoration: none;
29  }
30
31  header {
32    display: flex;
33    justify-content: space-between;
34    align-items: center;
35    padding: 30px 3%;
36    background-color: #00224B;
37  }
38
39  .nav_links {
40    list-style: none;

```

```

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
}
.nav_links li {
    display: inline-block;
    padding: 0 5px 0 5px;
}
.nav_links li a {
    transition: all 0.3s ease 0s;
}
.nav_links li a:hover {
    color: #ff8c00;
}
***** My Account STYLES *****/
.droptbtn {
    cursor: pointer;
    background-color: transparent;
    border-color: transparent;
    font-family: "Sunflower", sans-serif;
    font-weight: 500px;
    font-size: 22px;
    color: white;
    text-decoration: none;
}
.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f1f1f1;
    min-width: 150px;
    box-shadow: 0px 8px 16px 0px rgba(0, 0, 0, 0.2);
    z-index: 1;
}
.dropdown-content a {
    color: black;
}
padding: 12px 16px;
text-decoration: none;
display: block;
}
.dropdown-content a:hover {
    background-color: #ddd;
}
.dropdown:hover .dropdown-content {
    display: block;
}
#search {
    padding: 10px 20px;
    border-radius: 10px;
    width: 30%;
}
#logo {
    color: #ffc04c;
    cursor: pointer;
}
i {
    color: #ffc04c;
    cursor: pointer;
}
i:hover {
    color: #ff8c00;
}
***** Sidebar STYLES *****/
.sidenav {
    height: 100%;
    width: 0;
    position: fixed;

```

```

118     z-index: 1;
119     top: 0;
120     left: 0;
121     background-color: #00224B;
122     overflow-x: hidden;
123     transition: 0.5s;
124     padding-top: 60px;
125   }
126   /*###00224B***/
127   /*###ffc04c***/
128   .sidenav a {
129     padding: 8px 8px 8px 32px;
130     text-decoration: none;
131     font-size: 25px;
132     color: #white;
133     display: block;
134     transition: 0.3s;
135   }
136   .sidenav a:hover {
137     color: #ffc04c;
138   }
139   .sidenav .closebtn {
140     position: absolute;
141     top: 0;
142     right: 10px;
143     font-size: 36px;
144     margin-left: 50px;
145   }
146   .welcome {
147     background-color: #00224B;
148   }
149   .welcome i {
150     margin-top: 5px;
151     margin-bottom: 20px;
152     padding-top: 10px;
153     padding-left: 30px;
154     font-size: 24px;
155     color: #ffc04c;
156   }
157   .welcomeText {
158     color: white;
159     font-size: 24px;
160   }
161   .main_sections {
162     margin-top: 20px;
163     margin-bottom: 20px;
164   }
165   .sidebar_headings {
166     margin-top: 20px;
167     margin-bottom: 20px;
168     margin-left: 18px;
169   }
170   .sidebar_links {
171     margin-left: 18px;
172   }
173   .sidebar_links a {
174     font-size: 20px;
175   }
176   
```

184

185 </style>

## WelcomeStyles.php:

```

1 <style>
2   @import url("https://fonts.googleapis.com/css2?family=Blinker&display=swap");
3   @import url("https://fonts.googleapis.com/css2?family=Sunflower:wght@300&display=swap");
4
5   #Business_Features {
6     width: 100%;
7     border-bottom: 2px solid orange;
8     background-color: #004599;
9     padding: 5px 10px 5px 10px;
10    color: white;
11    display: flex;
12    justify-content: space-between;
13    align-items: center;
14  }
15
16  #feature {
17    color: white;
18  }
19
20  #features li {
21    display: inline-block;
22    padding: 0 225px 0 225px;
23  }
24
25  #features {
26    text-align: center;
27  }
28
29  .slideshow-container {
30    max-width: 100%;
31    position: relative;
32    margin: auto;
33  }
34
35  .myslides {
36    display: none;
37    padding-left: 40%;
38    border: 2px solid #bbb;
39    background-color: #e8ebe9;

```

```

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139

```

```

        }
    }
    .slideshow-numbertext {
        color: black;
        font-size: 12px;
        padding: 8px 12px;
        position: absolute;
        top: 140;
    }
    .dot {
        cursor: pointer;
        height: 15px;
        width: 15px;
        margin: 0 2px;
        background-color: #bbb;
        border-radius: 50%;
        display: inline-block;
        transition: background-color 0.6s ease;
    }
    .active {
        background-color: #717171;
    }
    .products {
        width: 100%;
        height: 100%;
        margin: 80px auto;
        margin-top: 100px;
        background: #FFFFFF;
        box-shadow: 1px 2px 3px 0px rgba(0,0,0,0.10);
        border-radius: 6px;
        border: 2px solid orange;
        display: flex;
        flex-direction: row;
    }
    .best-selling {
        width: 100%;
        padding: 30px 50px;
        height: 220px;
        display: flex;
        justify-content: space-between;
        border-bottom: 2px solid grey;
    }
    #item1 {
        display: inline-block;
        width: 200px;
        margin-bottom: 2px;
    }
    #basket-btn {
        border: 2px solid orange;
        background-color: orange;
        border-radius: 10px;
        font-family: "Sunflower", sans-serif;
        color: white;
        font-size: 14px;
        padding: 2px 8px 2px 8px;
        cursor: pointer;
        float: left;
    }
    #myItems-btn {
        background-color: transparent;
        border: 0px;
        float: right;
        margin-right: 70px;
        margin-top: 6px;
    }
    .errorMessage {
        color: red;
        margin-left: 30px;
        margin-top: 10px;
    }
    #messages {
        border: 2px solid orange;
        background-color: #00224B;
        width: 25%;
        margin-left: auto;
        margin-right: auto;
    }
    #messages p {
        font-size: 24px;
        margin-left: auto;
        margin-right: auto;
        text-align: center;
    }
    .Messages {
        color: white;
        margin-top: 25px;
    }
</style>
<?php require_once ("xampp/htdocs/MyEcommerce/Styles/header.php"); ?>

```