

Online Shopping System

Project report (CA3) submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Shivam Nath

12018682

SUBJECT

INT220 - SERVER-SIDE SCRIPTING



School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

November 2023

Table of Contents

S.No.	Topics	Page No.
1.	Introduction	3
2.	Technologies Used	4
3.	Snapshots	7
4.	Key Features	10
5.	Database	13
6.	Project Link	16
7.	Bibliograph	17

INTRODUCTION

The digital age has ushered in a transformation in the way we approach commerce and consumerism. With the advent of the internet, the retail landscape has undergone a seismic shift, giving birth to the Online Shopping System - a virtual marketplace that has reshaped the very essence of retailing. In a world where convenience, accessibility, and efficiency are paramount, online shopping has emerged as a game-changer, impacting the lives of consumers and businesses alike.

This report delves into the multifaceted world of the Online Shopping System, seeking to explore its evolution, impact, and the dynamics that make it an integral part of modern life. We will examine the historical backdrop, technological underpinnings, key players, and the implications for consumers and businesses as we navigate this intricate web of e-commerce.

As we embark on this journey, it is essential to recognize that the online shopping ecosystem encompasses a vast array of platforms, from massive marketplaces like Amazon and Alibaba to niche boutiques specializing in artisanal goods. It caters to the diverse needs of consumers, from the tech-savvy millennials who crave instant gratification to the more traditional shoppers who seek the comfort of familiar retail experiences.

TECHNOLOGIES USED

This section encapsulates the array of technologies that have been harnessed in the development and construction of the Online Shopping System. Within this section, you can detail the various technologies—PHP, PhpMyAdmin, HTML, CSS, and Bootstrap—explaining their individual roles and collective contributions to the system's functionality and user interface.

1. PHP (Hypertext Preprocessor):

At the core of the Heritage Online Shopping System is PHP, a server-side scripting language renowned for its versatility and efficiency. PHP facilitates dynamic content generation, enabling seamless interaction between the user interface and the server. Its usage in this project ensures the smooth execution of server-side operations, making real-time data processing, such as booking validations and user authentication, possible.

2. MySQL:

MySQL is a popular open-source relational database management system, widely recognized for its reliability and performance. In the Online Shopping System, MySQL is used as the database backend to store and manage critical data such as Cart Information, product details, Customers details. Key aspects of MySQL's contribution include:

Data Storage: MySQL efficiently stores and organizes all the essential data related to Customers, Products, Prices, Number of Product, admin and Users along with Users Type in structured tables.

Data Retrieval: It provides powerful SQL querying capabilities, enabling the system to retrieve and present data to users as needed.

Data Integrity: MySQL ensures data integrity and security, enforcing constraints and permissions to prevent unauthorized access or data corruption.

Scalability: MySQL's scalability supports the growth of the Student Project Management system, making it capable of handling elections of varying sizes.

3. Bootstrap:

Employing Bootstrap, a front-end framework, has notably elevated the visual aesthetics and responsiveness of the ACI Online Shopping System. By leveraging Bootstrap's ready-to-use templates and adaptable grid system, crafting a user interface that is both visually appealing and mobile-friendly has become effortlessly achievable. This results in a uniform and immersive user experience, accessible across a spectrum of devices, thereby enhancing the application's overall usability.

4. PhpMyAdmin

PhpMyAdmin functions as the go-to database management tool for this project, providing an intuitive interface to interact with the MySQL database. This tool significantly simplifies the process of administering the database, enabling the smooth handling of user and reservation data storage, retrieval, and organization. Its role in maintaining data security, integrity, and facilitating seamless communication between the application and the database is pivotal to the project's success.

5. XAMPP:

XAMPP, standing for cross-platform, Apache, MySQL, PHP, and Perl, stands as an open-source solution for web server management, playing a pivotal role in the development of the Online Voting System through:

Unified Local Environment: XAMPP integrates essential components like the Apache server, MySQL database, PHP, and other vital tools into a singular package. This amalgamation facilitates developers in establishing and evaluating the system on their personal machines.

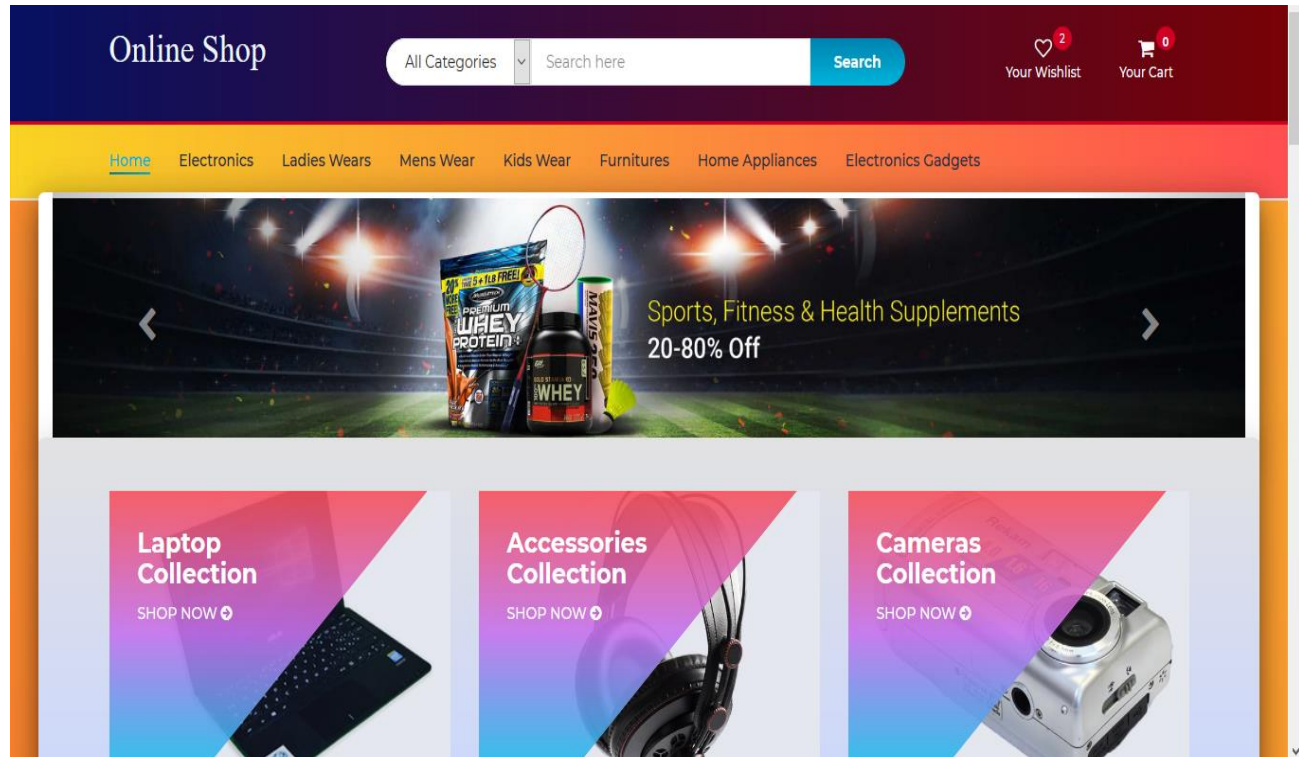
Testing and Debugging: XAMPP serves as a supportive platform for system evaluation, enabling thorough testing and debugging procedures. Its function ensures that the Online Shopping system operates seamlessly before it is deployed onto a live server.

Seamless Migration: The convenience of transitioning the system from the local XAMPP environment to a production server is a noteworthy advantage. This simplifies the process of making the system live once it's thoroughly tested and ready for deployment.

6. JavaScript:

JavaScript, a flexible scripting language, is utilized to boost the dynamism and user responsiveness within the Online Shopping system. Its implementation enables real-time content updates, eliminating the need for entire page reloads and creating a smoother, more immersive user journey. Furthermore, it plays a crucial role in validating user inputs directly on the client side, ensuring data precision and bolstering the overall trustworthiness of information entered by users.

SNAPSHOTS



Login

Email

shivannath

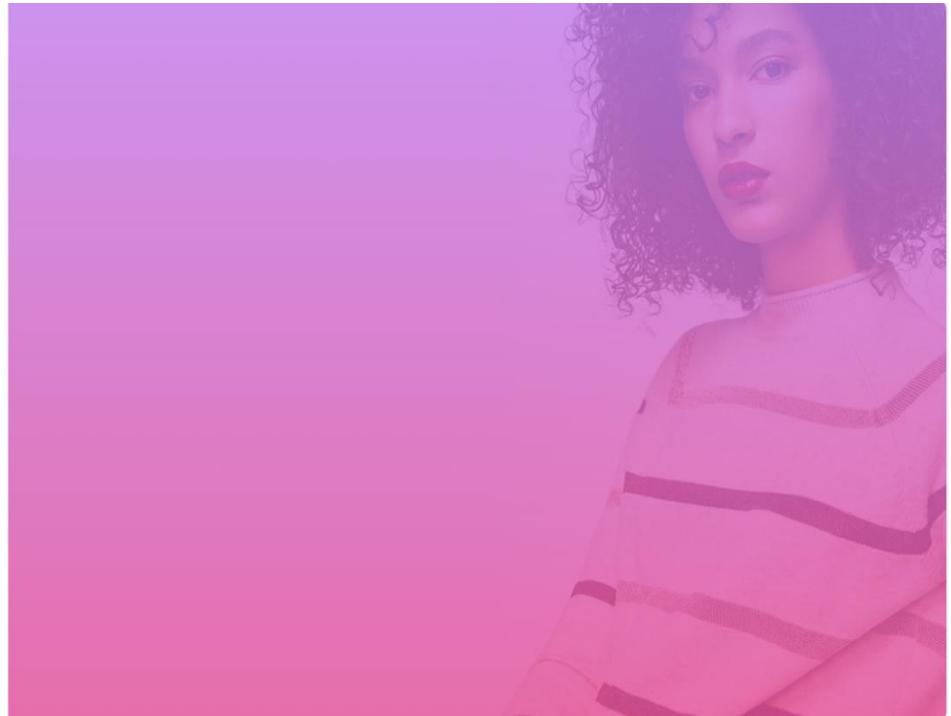
Password

☐ I agree to the [Terms of User](#)

Sign in

Sign up →

☐ Skip SignIn →



Online Shop

All Categories
Search here
Search










2

Your Wishlist

3

Your Cart


[Home](#)
[Electronics](#)
[Ladies Wears](#)
[Mens Wear](#)
[Kids Wear](#)
[Furnitures](#)
[Home Appliances](#)
[Electronics Gadgets](#)

Product	Price	Quantity	Subtotal	
 T shirt Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum dolor sit amet.	800	1	800	 
 boys shirts Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum dolor sit amet.	299	1	299	 
 boys Jeans Pant Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum dolor sit amet.	360	1	360	 

Continue Shopping


Total : \$ 1459





Ready to Checkout



Sign Up for the **NEWSLETTER**

Enter Your Email


Subscribe

Key Features:

1. **User-Friendly Interface:** The system should have an intuitive and user-friendly interface, making it easy for customers to browse products, add them to the cart, and complete transactions.
2. **Search and Navigation:** Efficient search functionality and intuitive navigation menus help users find products quickly, with filters and sorting options to refine search results.
3. **Product Listings:** Comprehensive product listings with high-quality images, detailed descriptions, pricing, and customer reviews provide valuable information for users.
4. **Shopping Cart:** A well-designed shopping cart allows users to review and modify their selected items, estimate shipping costs, and proceed to checkout.
5. **Secure Payment Processing:** Secure payment gateways and multiple payment options (credit cards, digital wallets, etc.) instill trust in users and ensure the safety of their financial information.
6. **Order Tracking:** Order tracking and real-time updates on shipping and delivery status keep customers informed about their purchases.
7. **Wishlist and Favorites:** Users can save items to Wishlist or favorites for future reference, facilitating repeat purchases and reducing cart abandonment.

Main Layout

```
login.php x
login.php > ...
1 <?php
2 include "db.php";
3
4 session_start();
5
6 #Login script is begin here
7 #If user given credential matches successfully with the data available in database then we will echo string login_success
8 #login_success string will go back to called Anonymous funtion $("#login").click()
9
10 if(isset($_POST["email"]) && isset($_POST["password"])){
11     $email = mysqli_real_escape_string($con,$_POST["email"]);
12     $password = $_POST["password"];
13     $sql = "SELECT * FROM user_info WHERE email = '$email' AND password = '$password'";
14     $run_query = mysqli_query($con,$sql);
15     $count = mysqli_num_rows($run_query);
16     $row = mysqli_fetch_array($run_query);
17
18     //we have created a cookie in login_form.php page so if that cookie is available means user is not login
19
20     //if user record is available in database then $count will be equal to 1
21     if($count == 1){
22
23         if (isset($_COOKIE["product_list"])) {
24
25             $p_list = stripslashes($_COOKIE["product_list"]);
26             //here we are decoding stored json product list cookie to normal array
27             $product_list = json_decode($p_list,true);
28             for ($i=0; $i < count($product_list); $i++) {
29                 //After getting user id from database here we are checking user cart item if there is already product is listed or not
30                 $verify_cart = "SELECT id FROM cart WHERE user_id = $_SESSION[uid] AND p_id = ".$product_list[$i];
31                 $result = mysqli_query($con,$verify_cart);
32                 if(mysqli_num_rows($result) < 1){
33                     //If user is adding first time product into cart we will update user_id into database table with valid id
34                     $update_cart = "UPDATE cart SET user_id = '$_SESSION[uid]' WHERE ip_add = '$ip_add' AND user_id = -1";
35                     mysqli_query($con,$update_cart);
36                 }else{
37                     //if already that product is available into database table we will delete that
38                     $delete_existing_product = "DELETE FROM cart WHERE user_id = -1 AND ip_add = '$ip_add'";
39                     mysqli_query($con,$delete_existing_product);
40                 }
41             }
42         }
43     }
44 }
```

Checkout Layout

```
checkout_process.php x
checkout_process.php > ...
1 <?php
2 session_start();
3 include "db.php";
4 if (isset($_SESSION["uid"])) {
5
6     $f_name = $_POST["firstname"];
7     $email = $_POST["email"];
8     $address = $_POST["address"];
9     $city = $_POST["city"];
10    $state = $_POST["state"];
11    $zip = $_POST["zip"];
12    $cardname = $_POST["cardname"];
13    $cardnumber = $_POST["cardNumber"];
14    $expdate = $_POST["expdate"];
15    $cvv = $_POST["cvv"];
16    $user_id = $_SESSION["uid"];
17    $cardnumberstr = (string)$cardnumber;
18    $total_count = $_POST["total_count"];
19    $prod_total = $_POST["total_price"];
20
21
22    $sql0 = "SELECT order_id from `orders_info`";
23    $runquery = mysqli_query($con,$sql0);
24    if (mysqli_num_rows($runquery) == 0) {
25        echo( mysqli_error($con));
26        $order_id = 1;
27    }else if (mysqli_num_rows($runquery) > 0) {
28        $sql2 = "SELECT MAX(order_id) AS max_val from `orders_info`";
29        $runquery1 = mysqli_query($con,$sql2);
30        $row = mysqli_fetch_array($runquery1);
31        $order_id = $row["max_val"];
32        $order_id = $order_id + 1;
33        echo( mysqli_error($con));
34    }
35
36    $sql = "INSERT INTO `orders_info`
37    (`order_id`,`user_id`,`f_name`,`email`,`address`,`city`,`state`,`zip`,`cardname`,`cardnumber`,`expdate`,`prod_count`,`total_amt`,`cvv`)
38    VALUES ($order_id, '$user_id','$f_name','$email',
39    '$address','$city','$state','$zip','$cardname','$cardnumberstr','$expdate','$total_count','$prod_total','$cvv')";
40    $runquery = mysqli_query($con,$sql);
41    if ($runquery) {
42        echo "Order placed successfully";
43    }else{
44        echo "Order not placed";
45    }
46 }
```

Config Layout

```
config.php > ...
1  <?php
2  session_start();
3
4  // initializing variables
5  $username = "";
6  $email = "";
7  $errors = array();
8
9
10 // connect to the database
11 define('DB_SERVER', 'localhost');
12 define('DB_USERNAME', 'supportReddy');
13 define('DB_PASSWORD', '');
14 define('DB_DATABASE', 'ecommerce');
15 $db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
16 if (!$db) {
17     die("Connection failed: " . mysqli_connect_error());
18 }
19
20 // REGISTER USER
21 if (isset($_POST['reg_user'])) {
22     // receive all input values from the form
23     $username = mysqli_real_escape_string($db, $_POST['username']);
24     $email = mysqli_real_escape_string($db, $_POST['email']);
25     $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
26     $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);
27
28     // form validation: ensure that the form is correctly filled ...
29     // by adding (array_push()) corresponding error unto $errors array
30     if (empty($username)) { array_push($errors, "Username is required"); }
31     if (empty($email)) { array_push($errors, "Email is required"); }
32     if (empty($password_1)) { array_push($errors, "Password is required"); }
33     if ($password_1 != $password_2) {
34         array_push($errors, "The two passwords do not match");
35     }
36
37     // first check the database to make sure
38     // a user does not already exist with the same username and/or email
39     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
40     $result = mysqli_query($db, $user_check_query);
41     $user = mysqli_fetch_assoc($result);
42     if ($user) { // user exists
43         array_push($errors, "User already exists");
44     }
45
46     // insert into database
47     if (count($errors) == 0) {
48         $password = password_hash($password_1, PASSWORD_DEFAULT);
49         $query = "INSERT INTO register (Name, email, password) VALUES ('$username', '$email', '$password')";
50         $result = mysqli_query($db, $query);
51         if ($result) {
52             // success
53             $username = "";
54             $email = "";
55             $password_1 = "";
56             $password_2 = "";
57             $errors = array();
58             header("Location: login.php");
59             exit();
60         } else {
61             // failed
62             array_push($errors, "Registration failed");
63         }
64     }
65 }
66
67 // LOGIN USER
68 if (isset($_POST['login_user'])) {
69     $username = mysqli_real_escape_string($db, $_POST['username']);
70     $password = mysqli_real_escape_string($db, $_POST['password']);
71
72     if (empty($username) || empty($password)) {
73         array_push($errors, "Username and password are required");
74     }
75
76     $login_query = mysqli_query($db, "SELECT * FROM register WHERE Name='$username' AND password='$password'");
77     $login_row = mysqli_fetch_assoc($login_query);
78     if ($login_row) {
79         // success
80         session_start();
81         $_SESSION['username'] = $username;
82         $_SESSION['email'] = $login_row['email'];
83         header("Location: index.php");
84         exit();
85     } else {
86         // failed
87         array_push($errors, "Invalid username or password");
88     }
89 }
90
91 // Logout
92 if (isset($_POST['logout'])) {
93     session_destroy();
94     header("Location: login.php");
95     exit();
96 }
97
98 // Update Profile
99 if (isset($_POST['update_profile'])) {
100     $username = mysqli_real_escape_string($db, $_POST['username']);
101     $email = mysqli_real_escape_string($db, $_POST['email']);
102     $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
103     $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);
104
105     // form validation: ensure that the form is correctly filled ...
106     // by adding (array_push()) corresponding error unto $errors array
107     if (empty($username)) { array_push($errors, "Username is required"); }
108     if (empty($email)) { array_push($errors, "Email is required"); }
109     if (empty($password_1)) { array_push($errors, "Password is required"); }
110     if ($password_1 != $password_2) {
111         array_push($errors, "The two passwords do not match");
112     }
113
114     // first check the database to make sure
115     // a user does not already exist with the same username and/or email
116     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
117     $result = mysqli_query($db, $user_check_query);
118     $user = mysqli_fetch_assoc($result);
119     if ($user) { // user exists
120         array_push($errors, "User already exists");
121     }
122
123     // insert into database
124     if (count($errors) == 0) {
125         $password = password_hash($password_1, PASSWORD_DEFAULT);
126         $query = "UPDATE register SET Name='$username', email='$email', password='$password' WHERE Name='$username' OR email='$email'";
127         $result = mysqli_query($db, $query);
128         if ($result) {
129             // success
130             $username = "";
131             $email = "";
132             $password_1 = "";
133             $password_2 = "";
134             $errors = array();
135             header("Location: index.php");
136             exit();
137         } else {
138             // failed
139             array_push($errors, "Update failed");
140         }
141     }
142 }
143
144 // Change Password
145 if (isset($_POST['change_password'])) {
146     $current_password = mysqli_real_escape_string($db, $_POST['current_password']);
147     $new_password_1 = mysqli_real_escape_string($db, $_POST['new_password_1']);
148     $new_password_2 = mysqli_real_escape_string($db, $_POST['new_password_2']);
149
150     // form validation: ensure that the form is correctly filled ...
151     // by adding (array_push()) corresponding error unto $errors array
152     if (empty($current_password)) { array_push($errors, "Current password is required"); }
153     if (empty($new_password_1)) { array_push($errors, "New password is required"); }
154     if (empty($new_password_2)) { array_push($errors, "New password is required"); }
155     if ($new_password_1 != $new_password_2) {
156         array_push($errors, "The two new passwords do not match");
157     }
158
159     // first check the database to make sure
160     // a user does not already exist with the same username and/or email
161     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
162     $result = mysqli_query($db, $user_check_query);
163     $user = mysqli_fetch_assoc($result);
164     if ($user) { // user exists
165         array_push($errors, "User already exists");
166     }
167
168     // insert into database
169     if (count($errors) == 0) {
170         $password = password_hash($new_password_1, PASSWORD_DEFAULT);
171         $query = "UPDATE register SET password='$password' WHERE Name='$username' OR email='$email'";
172         $result = mysqli_query($db, $query);
173         if ($result) {
174             // success
175             $current_password = "";
176             $new_password_1 = "";
177             $new_password_2 = "";
178             $errors = array();
179             header("Location: index.php");
180             exit();
181         } else {
182             // failed
183             array_push($errors, "Change password failed");
184         }
185     }
186 }
187
188 // Forgot Password
189 if (isset($_POST['forgot_password'])) {
190     $email = mysqli_real_escape_string($db, $_POST['email']);
191
192     // form validation: ensure that the form is correctly filled ...
193     // by adding (array_push()) corresponding error unto $errors array
194     if (empty($email)) { array_push($errors, "Email is required"); }
195
196     $forgot_query = "SELECT * FROM register WHERE email='$email'";
197     $forgot_row = mysqli_fetch_assoc($forgot_query);
198     if ($forgot_row) {
199         // success
200         $token = bin2hex(random_bytes(32));
201         $query = "UPDATE register SET token='$token' WHERE email='$email'";
202         $result = mysqli_query($db, $query);
203         if ($result) {
204             // success
205             $email = "";
206             $errors = array();
207             header("Location: forgot_password.php");
208             exit();
209         } else {
210             // failed
211             array_push($errors, "Forgot password failed");
212         }
213     } else {
214         // failed
215         array_push($errors, "Email not found");
216     }
217 }
218
219 // Reset Password
220 if (isset($_POST['reset_password'])) {
221     $token = mysqli_real_escape_string($db, $_POST['token']);
222     $new_password_1 = mysqli_real_escape_string($db, $_POST['new_password_1']);
223     $new_password_2 = mysqli_real_escape_string($db, $_POST['new_password_2']);
224
225     // form validation: ensure that the form is correctly filled ...
226     // by adding (array_push()) corresponding error unto $errors array
227     if (empty($token)) { array_push($errors, "Token is required"); }
228     if (empty($new_password_1)) { array_push($errors, "New password is required"); }
229     if (empty($new_password_2)) { array_push($errors, "New password is required"); }
230     if ($new_password_1 != $new_password_2) {
231         array_push($errors, "The two new passwords do not match");
232     }
233
234     // first check the database to make sure
235     // a user does not already exist with the same username and/or email
236     $user_check_query = "SELECT * FROM register WHERE token='$token' LIMIT 1";
237     $result = mysqli_query($db, $user_check_query);
238     $user = mysqli_fetch_assoc($result);
239     if ($user) { // user exists
240         array_push($errors, "User already exists");
241     }
242
243     // insert into database
244     if (count($errors) == 0) {
245         $password = password_hash($new_password_1, PASSWORD_DEFAULT);
246         $query = "UPDATE register SET password='$password' WHERE token='$token'";
247         $result = mysqli_query($db, $query);
248         if ($result) {
249             // success
250             $token = "";
251             $new_password_1 = "";
252             $new_password_2 = "";
253             $errors = array();
254             header("Location: index.php");
255             exit();
256         } else {
257             // failed
258             array_push($errors, "Reset password failed");
259         }
260     }
261 }
262
263 // Add to Cart
264 if (isset($_POST['add_to_cart'])) {
265     $product_id = mysqli_real_escape_string($db, $_POST['product_id']);
266     $qty = mysqli_real_escape_string($db, $_POST['qty']);
267
268     // form validation: ensure that the form is correctly filled ...
269     // by adding (array_push()) corresponding error unto $errors array
270     if (empty($product_id)) { array_push($errors, "Product ID is required"); }
271     if (empty($qty)) { array_push($errors, "Qty is required"); }
272
273     // first check the database to make sure
274     // a product does not already exist with the same product_id
275     $product_check_query = "SELECT * FROM products WHERE product_id='$product_id' LIMIT 1";
276     $product_result = mysqli_query($db, $product_check_query);
277     $product = mysqli_fetch_assoc($product_result);
278     if ($product) { // product exists
279         // insert into database
280         if (count($errors) == 0) {
281             $query = "INSERT INTO cart (product_id, qty) VALUES ('$product_id', '$qty')";
282             $result = mysqli_query($db, $query);
283             if ($result) {
284                 // success
285                 $product_id = "";
286                 $qty = "";
287                 $errors = array();
288                 header("Location: index.php");
289                 exit();
290             } else {
291                 // failed
292                 array_push($errors, "Add to cart failed");
293             }
294         }
295     } else {
296         // failed
297         array_push($errors, "Product not found");
298     }
299 }
300
301 // Remove from Cart
302 if (isset($_POST['remove_from_cart'])) {
303     $product_id = mysqli_real_escape_string($db, $_POST['product_id']);
304
305     // form validation: ensure that the form is correctly filled ...
306     // by adding (array_push()) corresponding error unto $errors array
307     if (empty($product_id)) { array_push($errors, "Product ID is required"); }
308
309     // first check the database to make sure
310     // a product does not already exist with the same product_id
311     $product_check_query = "SELECT * FROM cart WHERE product_id='$product_id' LIMIT 1";
312     $product_result = mysqli_query($db, $product_check_query);
313     $product = mysqli_fetch_assoc($product_result);
314     if ($product) { // product exists
315         // insert into database
316         if (count($errors) == 0) {
317             $query = "DELETE FROM cart WHERE product_id='$product_id'";
318             $result = mysqli_query($db, $query);
319             if ($result) {
320                 // success
321                 $product_id = "";
322                 $errors = array();
323                 header("Location: index.php");
324                 exit();
325             } else {
326                 // failed
327                 array_push($errors, "Remove from cart failed");
328             }
329         }
330     } else {
331         // failed
332         array_push($errors, "Product not found");
333     }
334 }
335
336 // Update Cart
337 if (isset($_POST['update_cart'])) {
338     $product_id = mysqli_real_escape_string($db, $_POST['product_id']);
339     $qty = mysqli_real_escape_string($db, $_POST['qty']);
340
341     // form validation: ensure that the form is correctly filled ...
342     // by adding (array_push()) corresponding error unto $errors array
343     if (empty($product_id)) { array_push($errors, "Product ID is required"); }
344     if (empty($qty)) { array_push($errors, "Qty is required"); }
345
346     // first check the database to make sure
347     // a product does not already exist with the same product_id
348     $product_check_query = "SELECT * FROM cart WHERE product_id='$product_id' LIMIT 1";
349     $product_result = mysqli_query($db, $product_check_query);
350     $product = mysqli_fetch_assoc($product_result);
351     if ($product) { // product exists
352         // insert into database
353         if (count($errors) == 0) {
354             $query = "UPDATE cart SET qty='$qty' WHERE product_id='$product_id'";
355             $result = mysqli_query($db, $query);
356             if ($result) {
357                 // success
358                 $product_id = "";
359                 $qty = "";
360                 $errors = array();
361                 header("Location: index.php");
362                 exit();
363             } else {
364                 // failed
365                 array_push($errors, "Update cart failed");
366             }
367         }
368     } else {
369         // failed
370         array_push($errors, "Product not found");
371     }
372 }
373
374 // Checkout
375 if (isset($_POST['checkout'])) {
376     $username = mysqli_real_escape_string($db, $_POST['username']);
377     $email = mysqli_real_escape_string($db, $_POST['email']);
378     $password = mysqli_real_escape_string($db, $_POST['password']);
379
380     // form validation: ensure that the form is correctly filled ...
381     // by adding (array_push()) corresponding error unto $errors array
382     if (empty($username)) { array_push($errors, "Username is required"); }
383     if (empty($email)) { array_push($errors, "Email is required"); }
384     if (empty($password)) { array_push($errors, "Password is required"); }
385
386     // first check the database to make sure
387     // a user does not already exist with the same username and/or email
388     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
389     $result = mysqli_query($db, $user_check_query);
390     $user = mysqli_fetch_assoc($result);
391     if ($user) { // user exists
392         array_push($errors, "User already exists");
393     }
394
395     // insert into database
396     if (count($errors) == 0) {
397         $password = password_hash($password, PASSWORD_DEFAULT);
398         $query = "INSERT INTO orders (Name, email, password) VALUES ('$username', '$email', '$password')";
399         $result = mysqli_query($db, $query);
400         if ($result) {
401             // success
402             $username = "";
403             $email = "";
404             $password = "";
405             $errors = array();
406             header("Location: index.php");
407             exit();
408         } else {
409             // failed
410             array_push($errors, "Checkout failed");
411         }
412     }
413 }
414
415 // View Order
416 if (isset($_POST['view_order'])) {
417     $order_id = mysqli_real_escape_string($db, $_POST['order_id']);
418
419     // form validation: ensure that the form is correctly filled ...
420     // by adding (array_push()) corresponding error unto $errors array
421     if (empty($order_id)) { array_push($errors, "Order ID is required"); }
422
423     // first check the database to make sure
424     // a order does not already exist with the same order_id
425     $order_check_query = "SELECT * FROM orders WHERE order_id='$order_id' LIMIT 1";
426     $order_result = mysqli_query($db, $order_check_query);
427     $order = mysqli_fetch_assoc($order_result);
428     if ($order) { // order exists
429         // insert into database
430         if (count($errors) == 0) {
431             $query = "SELECT * FROM cart WHERE product_id IN (SELECT product_id FROM orders WHERE order_id='$order_id')";
432             $result = mysqli_query($db, $query);
433             if ($result) {
434                 // success
435                 $order_id = "";
436                 $errors = array();
437                 header("Location: index.php");
438                 exit();
439             } else {
440                 // failed
441                 array_push($errors, "View order failed");
442             }
443         }
444     } else {
445         // failed
446         array_push($errors, "Order not found");
447     }
448 }
449
450 // Delete Order
451 if (isset($_POST['delete_order'])) {
452     $order_id = mysqli_real_escape_string($db, $_POST['order_id']);
453
454     // form validation: ensure that the form is correctly filled ...
455     // by adding (array_push()) corresponding error unto $errors array
456     if (empty($order_id)) { array_push($errors, "Order ID is required"); }
457
458     // first check the database to make sure
459     // a order does not already exist with the same order_id
460     $order_check_query = "SELECT * FROM orders WHERE order_id='$order_id' LIMIT 1";
461     $order_result = mysqli_query($db, $order_check_query);
462     $order = mysqli_fetch_assoc($order_result);
463     if ($order) { // order exists
464         // insert into database
465         if (count($errors) == 0) {
466             $query = "DELETE FROM orders WHERE order_id='$order_id'";
467             $result = mysqli_query($db, $query);
468             if ($result) {
469                 // success
470                 $order_id = "";
471                 $errors = array();
472                 header("Location: index.php");
473                 exit();
474             } else {
475                 // failed
476                 array_push($errors, "Delete order failed");
477             }
478         }
479     } else {
480         // failed
481         array_push($errors, "Order not found");
482     }
483 }
484
485 // Admin Login
486 if (isset($_POST['admin_login'])) {
487     $username = mysqli_real_escape_string($db, $_POST['username']);
488     $password = mysqli_real_escape_string($db, $_POST['password']);
489
490     // form validation: ensure that the form is correctly filled ...
491     // by adding (array_push()) corresponding error unto $errors array
492     if (empty($username)) { array_push($errors, "Username is required"); }
493     if (empty($password)) { array_push($errors, "Password is required"); }
494
495     // first check the database to make sure
496     // a user does not already exist with the same username and/or email
497     $user_check_query = "SELECT * FROM admin WHERE Name='$username' AND password='$password'";
498     $result = mysqli_query($db, $user_check_query);
499     $user = mysqli_fetch_assoc($result);
500     if ($user) { // user exists
501         // success
502         session_start();
503         $_SESSION['username'] = $username;
504         $_SESSION['email'] = $user['email'];
505         header("Location: admin.php");
506         exit();
507     } else {
508         // failed
509         array_push($errors, "Invalid username or password");
510     }
511 }
512
513 // Admin Logout
514 if (isset($_POST['admin_logout'])) {
515     session_destroy();
516     header("Location: admin.php");
517     exit();
518 }
519
520 // Admin Add Product
521 if (isset($_POST['admin_add_product'])) {
522     $product_title = mysqli_real_escape_string($db, $_POST['product_title']);
523     $product_price = mysqli_real_escape_string($db, $_POST['product_price']);
524     $product_image = mysqli_real_escape_string($db, $_POST['product_image']);
525     $product_qty = mysqli_real_escape_string($db, $_POST['product_qty']);
526
527     // form validation: ensure that the form is correctly filled ...
528     // by adding (array_push()) corresponding error unto $errors array
529     if (empty($product_title)) { array_push($errors, "Product title is required"); }
530     if (empty($product_price)) { array_push($errors, "Product price is required"); }
531     if (empty($product_image)) { array_push($errors, "Product image is required"); }
532     if (empty($product_qty)) { array_push($errors, "Product qty is required"); }
533
534     // first check the database to make sure
535     // a product does not already exist with the same product_title
536     $product_check_query = "SELECT * FROM products WHERE product_title='$product_title' LIMIT 1";
537     $product_result = mysqli_query($db, $product_check_query);
538     $product = mysqli_fetch_assoc($product_result);
539     if ($product) { // product exists
540         array_push($errors, "Product already exists");
541     }
542
543     // insert into database
544     if (count($errors) == 0) {
545         $query = "INSERT INTO products (product_title, product_price, product_image, product_qty) VALUES ('$product_title', '$product_price', '$product_image', '$product_qty')";
546         $result = mysqli_query($db, $query);
547         if ($result) {
548             // success
549             $product_title = "";
550             $product_price = "";
551             $product_image = "";
552             $product_qty = "";
553             $errors = array();
554             header("Location: admin.php");
555             exit();
556         } else {
557             // failed
558             array_push($errors, "Add product failed");
559         }
560     }
561 }
562
563 // Admin Edit Product
564 if (isset($_POST['admin_edit_product'])) {
565     $product_id = mysqli_real_escape_string($db, $_POST['product_id']);
566     $product_title = mysqli_real_escape_string($db, $_POST['product_title']);
567     $product_price = mysqli_real_escape_string($db, $_POST['product_price']);
568     $product_image = mysqli_real_escape_string($db, $_POST['product_image']);
569     $product_qty = mysqli_real_escape_string($db, $_POST['product_qty']);
570
571     // form validation: ensure that the form is correctly filled ...
572     // by adding (array_push()) corresponding error unto $errors array
573     if (empty($product_id)) { array_push($errors, "Product ID is required"); }
574     if (empty($product_title)) { array_push($errors, "Product title is required"); }
575     if (empty($product_price)) { array_push($errors, "Product price is required"); }
576     if (empty($product_image)) { array_push($errors, "Product image is required"); }
577     if (empty($product_qty)) { array_push($errors, "Product qty is required"); }
578
579     // first check the database to make sure
580     // a product does not already exist with the same product_id
581     $product_check_query = "SELECT * FROM products WHERE product_id='$product_id' LIMIT 1";
582     $product_result = mysqli_query($db, $product_check_query);
583     $product = mysqli_fetch_assoc($product_result);
584     if ($product) { // product exists
585         // insert into database
586         if (count($errors) == 0) {
587             $query = "UPDATE products SET product_title='$product_title', product_price='$product_price', product_image='$product_image', product_qty='$product_qty' WHERE product_id='$product_id'";
588             $result = mysqli_query($db, $query);
589             if ($result) {
590                 // success
591                 $product_id = "";
592                 $product_title = "";
593                 $product_price = "";
594                 $product_image = "";
595                 $product_qty = "";
596                 $errors = array();
597                 header("Location: admin.php");
598                 exit();
599             } else {
600                 // failed
601                 array_push($errors, "Edit product failed");
602             }
603         }
604     } else {
605         // failed
606         array_push($errors, "Product not found");
607     }
608 }
609
610 // Admin Delete Product
611 if (isset($_POST['admin_delete_product'])) {
612     $product_id = mysqli_real_escape_string($db, $_POST['product_id']);
613
614     // form validation: ensure that the form is correctly filled ...
615     // by adding (array_push()) corresponding error unto $errors array
616     if (empty($product_id)) { array_push($errors, "Product ID is required"); }
617
618     // first check the database to make sure
619     // a product does not already exist with the same product_id
620     $product_check_query = "SELECT * FROM products WHERE product_id='$product_id' LIMIT 1";
621     $product_result = mysqli_query($db, $product_check_query);
622     $product = mysqli_fetch_assoc($product_result);
623     if ($product) { // product exists
624         // insert into database
625         if (count($errors) == 0) {
626             $query = "DELETE FROM products WHERE product_id='$product_id'";
627             $result = mysqli_query($db, $query);
628             if ($result) {
629                 // success
630                 $product_id = "";
631                 $errors = array();
632                 header("Location: admin.php");
633                 exit();
634             } else {
635                 // failed
636                 array_push($errors, "Delete product failed");
637             }
638         }
639     } else {
640         // failed
631         array_push($errors, "Product not found");
632     }
633 }
634
635 // Admin View Order
636 if (isset($_POST['admin_view_order'])) {
637     $order_id = mysqli_real_escape_string($db, $_POST['order_id']);
638
639     // form validation: ensure that the form is correctly filled ...
640     // by adding (array_push()) corresponding error unto $errors array
641     if (empty($order_id)) { array_push($errors, "Order ID is required"); }
642
643     // first check the database to make sure
644     // a order does not already exist with the same order_id
645     $order_check_query = "SELECT * FROM orders WHERE order_id='$order_id' LIMIT 1";
646     $order_result = mysqli_query($db, $order_check_query);
647     $order = mysqli_fetch_assoc($order_result);
648     if ($order) { // order exists
649         // insert into database
650         if (count($errors) == 0) {
651             $query = "SELECT * FROM cart WHERE product_id IN (SELECT product_id FROM orders WHERE order_id='$order_id')";
652             $result = mysqli_query($db, $query);
653             if ($result) {
654                 // success
655                 $order_id = "";
656                 $errors = array();
657                 header("Location: admin.php");
658                 exit();
659             } else {
660                 // failed
661                 array_push($errors, "View order failed");
662             }
663         }
664     } else {
665         // failed
666         array_push($errors, "Order not found");
667     }
668 }
669
670 // Admin Delete Order
671 if (isset($_POST['admin_delete_order'])) {
672     $order_id = mysqli_real_escape_string($db, $_POST['order_id']);
673
674     // form validation: ensure that the form is correctly filled ...
675     // by adding (array_push()) corresponding error unto $errors array
676     if (empty($order_id)) { array_push($errors, "Order ID is required"); }
677
678     // first check the database to make sure
679     // a order does not already exist with the same order_id
680     $order_check_query = "SELECT * FROM orders WHERE order_id='$order_id' LIMIT 1";
681     $order_result = mysqli_query($db, $order_check_query);
682     $order = mysqli_fetch_assoc($order_result);
683     if ($order) { // order exists
684         // insert into database
685         if (count($errors) == 0) {
686             $query = "DELETE FROM orders WHERE order_id='$order_id'";
687             $result = mysqli_query($db, $query);
688             if ($result) {
689                 // success
690                 $order_id = "";
691                 $errors = array();
692                 header("Location: admin.php");
693                 exit();
694             } else {
695                 // failed
696                 array_push($errors, "Delete order failed");
697             }
698         }
699     } else {
700         // failed
701         array_push($errors, "Order not found");
702     }
703 }
704
705 // Admin Add User
706 if (isset($_POST['admin_add_user'])) {
707     $username = mysqli_real_escape_string($db, $_POST['username']);
708     $email = mysqli_real_escape_string($db, $_POST['email']);
709     $password = mysqli_real_escape_string($db, $_POST['password']);
710
711     // form validation: ensure that the form is correctly filled ...
712     // by adding (array_push()) corresponding error unto $errors array
713     if (empty($username)) { array_push($errors, "Username is required"); }
714     if (empty($email)) { array_push($errors, "Email is required"); }
715     if (empty($password)) { array_push($errors, "Password is required"); }
716
717     // first check the database to make sure
718     // a user does not already exist with the same username and/or email
719     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
720     $result = mysqli_query($db, $user_check_query);
721     $user = mysqli_fetch_assoc($result);
722     if ($user) { // user exists
723         array_push($errors, "User already exists");
724     }
725
726     // insert into database
727     if (count($errors) == 0) {
728         $password = password_hash($password, PASSWORD_DEFAULT);
729         $query = "INSERT INTO register (Name, email, password) VALUES ('$username', '$email', '$password')";
730         $result = mysqli_query($db, $query);
731         if ($result) {
732             // success
733             $username = "";
734             $email = "";
735             $password = "";
736             $errors = array();
737             header("Location: admin.php");
738             exit();
739         } else {
740             // failed
741             array_push($errors, "Add user failed");
742         }
743     }
744 }
745
746 // Admin Edit User
747 if (isset($_POST['admin_edit_user'])) {
748     $username = mysqli_real_escape_string($db, $_POST['username']);
749     $email = mysqli_real_escape_string($db, $_POST['email']);
750     $password = mysqli_real_escape_string($db, $_POST['password']);
751
752     // form validation: ensure that the form is correctly filled ...
753     // by adding (array_push()) corresponding error unto $errors array
754     if (empty($username)) { array_push($errors, "Username is required"); }
755     if (empty($email)) { array_push($errors, "Email is required"); }
756     if (empty($password)) { array_push($errors, "Password is required"); }
757
758     // first check the database to make sure
759     // a user does not already exist with the same username and/or email
760     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
761     $result = mysqli_query($db, $user_check_query);
762     $user = mysqli_fetch_assoc($result);
763     if ($user) { // user exists
764         // insert into database
765         if (count($errors) == 0) {
766             $password = password_hash($password, PASSWORD_DEFAULT);
767             $query = "UPDATE register SET Name='$username', email='$email', password='$password' WHERE Name='$username' OR email='$email'";
768             $result = mysqli_query($db, $query);
769             if ($result) {
770                 // success
771                 $username = "";
772                 $email = "";
773                 $password = "";
774                 $errors = array();
775                 header("Location: admin.php");
776                 exit();
777             } else {
778                 // failed
779                 array_push($errors, "Edit user failed");
780             }
781         }
782     } else {
783         // failed
784         array_push($errors, "User not found");
785     }
786 }
787
788 // Admin Delete User
789 if (isset($_POST['admin_delete_user'])) {
790     $username = mysqli_real_escape_string($db, $_POST['username']);
791
792     // form validation: ensure that the form is correctly filled ...
793     // by adding (array_push()) corresponding error unto $errors array
794     if (empty($username)) { array_push($errors, "Username is required"); }
795
796     // first check the database to make sure
797     // a user does not already exist with the same username and/or email
798     $user_check_query = "SELECT * FROM register WHERE Name='$username' LIMIT 1";
799     $result = mysqli_query($db, $user_check_query);
800     $user = mysqli_fetch_assoc($result);
801     if ($user) { // user exists
802         // insert into database
803         if (count($errors) == 0) {
804             $query = "DELETE FROM register WHERE Name='$username'";
805             $result = mysqli_query($db, $query);
806             if ($result) {
807                 // success
808                 $username = "";
809                 $errors = array();
810                 header("Location: admin.php");
811                 exit();
812             } else {
813                 // failed
814                 array_push($errors, "Delete user failed");
815             }
816         }
817     } else {
818         // failed
819         array_push($errors, "User not found");
820     }
821 }
822
823 // Admin View Cart
824 if (isset($_POST['admin_view_cart'])) {
825     // first check the database to make sure
826     // a user does not already exist with the same username and/or email
827     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
828     $result = mysqli_query($db, $user_check_query);
829     $user = mysqli_fetch_assoc($result);
830     if ($user) { // user exists
831         // insert into database
832         if (count($errors) == 0) {
833             $query = "SELECT * FROM cart";
834             $result = mysqli_query($db, $query);
835             if ($result) {
836                 // success
837                 $errors = array();
838                 header("Location: admin.php");
839                 exit();
840             } else {
841                 // failed
842                 array_push($errors, "View cart failed");
843             }
844         }
845     } else {
846         // failed
847         array_push($errors, "User not found");
848     }
849 }
850
851 // Admin Delete Cart
852 if (isset($_POST['admin_delete_cart'])) {
853     // first check the database to make sure
854     // a user does not already exist with the same username and/or email
855     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
856     $result = mysqli_query($db, $user_check_query);
857     $user = mysqli_fetch_assoc($result);
858     if ($user) { // user exists
859         // insert into database
860         if (count($errors) == 0) {
861             $query = "DELETE FROM cart";
862             $result = mysqli_query($db, $query);
863             if ($result) {
864                 // success
865                 $errors = array();
866                 header("Location: admin.php");
867                 exit();
868             } else {
869                 // failed
870                 array_push($errors, "Delete cart failed");
871             }
872         }
873     } else {
874         // failed
875         array_push($errors, "User not found");
876     }
877 }
878
879 // Admin View Orders
880 if (isset($_POST['admin_view_orders'])) {
881     // first check the database to make sure
882     // a user does not already exist with the same username and/or email
883     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
884     $result = mysqli_query($db, $user_check_query);
885     $user = mysqli_fetch_assoc($result);
886     if ($user) { // user exists
887         // insert into database
888         if (count($errors) == 0) {
889             $query = "SELECT * FROM orders";
890             $result = mysqli_query($db, $query);
891             if ($result) {
892                 // success
893                 $errors = array();
894                 header("Location: admin.php");
895                 exit();
896             } else {
897                 // failed
898                 array_push($errors, "View orders failed");
899             }
900         }
901     } else {
902         // failed
903         array_push($errors, "User not found");
904     }
905 }
906
907 // Admin Delete Orders
908 if (isset($_POST['admin_delete_orders'])) {
909     // first check the database to make sure
910     // a user does not already exist with the same username and/or email
911     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
912     $result = mysqli_query($db, $user_check_query);
913     $user = mysqli_fetch_assoc($result);
914     if ($user) { // user exists
915         // insert into database
916         if (count($errors) == 0) {
917             $query = "DELETE FROM orders";
918             $result = mysqli_query($db, $query);
919             if ($result) {
920                 // success
921                 $errors = array();
922                 header("Location: admin.php");
923                 exit();
924             } else {
925                 // failed
926                 array_push($errors, "Delete orders failed");
927             }
928         }
929     } else {
930         // failed
931         array_push($errors, "User not found");
932     }
933 }
934
935 // Admin View Users
936 if (isset($_POST['admin_view_users'])) {
937     // first check the database to make sure
938     // a user does not already exist with the same username and/or email
939     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
940     $result = mysqli_query($db, $user_check_query);
941     $user = mysqli_fetch_assoc($result);
942     if ($user) { // user exists
943         // insert into database
944         if (count($errors) == 0) {
945             $query = "SELECT * FROM register";
946             $result = mysqli_query($db, $query);
947             if ($result) {
948                 // success
949                 $errors = array();
950                 header("Location: admin.php");
951                 exit();
952             } else {
953                 // failed
954                 array_push($errors, "View users failed");
955             }
956         }
957     } else {
958         // failed
959         array_push($errors, "User not found");
960     }
961 }
962
963 // Admin Delete Users
964 if (isset($_POST['admin_delete_users'])) {
965     // first check the database to make sure
966     // a user does not already exist with the same username and/or email
967     $user_check_query = "SELECT * FROM register WHERE Name='$username' OR email='$email' LIMIT 1";
968     $result = mysqli_query($db, $user_check_query);
969     $user = mysqli_fetch_assoc($result);
970     if ($user) { // user exists
971         // insert into database
972         if (count($errors) == 0) {
973             $query = "DELETE FROM register";
974             $result = mysqli_query($db, $query);
975             if ($result) {
976                 // success
977                 $errors = array();
978                 header("Location: admin.php");
979                 exit();
980             } else {
981                 // failed
982                 array_push($errors, "Delete users failed");
983             }
984         }
985     } else {
986         // failed
98
```

DATABASE

A well-structured and efficient database is essential to store and manage various types of data, including product information, user accounts, orders, and more. You can design a database using a relational database management system (RDBMS) like MySQL, PostgreSQL, or Microsoft SQL Server. Below, I'll provide an online shopping system:

1. User Information Table:

- User ID (Primary Key)
- Username
- Email
- Password (Hashed)
- Name
- Address
- Phone Number
- Payment Information (encrypted)

2. Product Catalog Table:

- Product ID (Primary Key)
- Product Name
- Description
- Category
- Price
- Stock Quantity
- Manufacturer
- Product Image URLs

phpMyAdmin

Server: 127.0.0.1 > Database: onlineshop

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer Central columns

Recent Favorites

New

- information_schema
- mysql
- onlineshop
 - Procedures
 - Tables
 - New
 - admin_info
 - brands
 - cart
 - categories
 - email_info
 - logs
 - orders
 - orders_info
 - order_products
 - products
 - reviews
 - user_info
 - user_info_backup
 - wishlist
- performance_schema
- phpmyadmin
- test

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> admin_info		1	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> brands		6	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> cart		4	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> categories		7	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> email_info		3	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> logs		0	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> orders		2	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> orders_info		1	InnoDB	latin1_swedish_ci	32.0 KIB	-
<input type="checkbox"/> order_products		3	InnoDB	latin1_swedish_ci	48.0 KIB	-
<input type="checkbox"/> products		71	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> reviews		0	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> user_info		1	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> user_info_backup		1	InnoDB	latin1_swedish_ci	16.0 KIB	-
<input type="checkbox"/> wishlist		5	InnoDB	latin1_swedish_ci	16.0 KIB	-
14 tables	Sum	105	InnoDB	utf8mb4_general_ci	272.0 KIB	0 B

☐ Check all With selected:

Print Data dictionary

Create new table

Table name: Number of columns:

Console

PROJECT LINK

GitHub Link:

https://github.com/shivamnath2001/12018682_PHP.git

BIBLIOGRAPHY

1. PHP Documentation:

URL: [PHP Official Documentation] (<https://www.php.net/manual/en/>)

2. Bootstrap Documentation:

URL: [Bootstrap Documentation] (<https://getbootstrap.com/docs/4.0/>)

3. MySQL Documentation:

URL: [MySQL Documentation] (<https://dev.mysql.com/doc/>)

4. PhpMyAdmin Documentation:

URL: [PhpMyAdmin Documentation] (<https://docs.phpmyadmin.net/>)

5. JavaScript Documentation:

URL: [MDN JavaScript Documentation] (<https://developer.mozilla.org/en-US/docs/Web/JavaScript>)

Thank You