

Web Dev Assignment

Jason Gaynor

Student Number: C23409212

Register.php

```
register.php > ...
1      <?php
2
3      // database connection settings
4      $servername = "localhost";
5      $username = "root";
6      $password = "";
7      $database = "website";
8
9      // create a connection to the database
10     $conn = new mysqli(hostname: $servername, username: $username, password: $password, database: $database);
11
12     // check the connection
13     if ($conn->connect_error) {
14         die("Connection failed: " . $conn->connect_error);
15     }
16
17     // error variable empty initially
18     $error = "";
19
20     if ($_SERVER["REQUEST_METHOD"] == "POST") {
21         // get user inputs from the form
22         $username = $conn->real_escape_string(string: trim(string: $_POST['username']));
23         $password = $conn->real_escape_string(string: trim(string: $_POST['password']));
24         $confirm_password = $conn->real_escape_string(string: trim(string: $_POST['confirm_password']));
25         $first_name = $conn->real_escape_string(string: trim(string: $_POST['first_name']));
26         $last_name = $conn->real_escape_string(string: trim(string: $_POST['last_name']));
27         $address_line1 = $conn->real_escape_string(string: trim(string: $_POST['address_line1']));
28         $address_line2 = $conn->real_escape_string(string: trim(string: $_POST['address_line2']));
29         $city = $conn->real_escape_string(string: trim(string: $_POST['city']));
30         $telephone = $conn->real_escape_string(string: trim(string: $_POST['telephone']));
31         $mobile = $conn->real_escape_string(string: trim(string: $_POST['mobile']));
32
33         // validate inputs to meet criteria using multiple ifs
34         if (empty($username) || empty($password) || empty($confirm_password) || empty($first_name) ||
35             empty($last_name) || empty($address_line1) || empty($city) || empty($telephone) || empty($mobile))
36         {
37             $error = "All fields must be entered.";
38         }
39         //check if password confirmation = password
40         elseif ($password != $confirm_password)
41         {
42             $error = "Passwords do not match!";
43         }
44         //check mobile number is numeric
45         elseif (!is_numeric($mobile))
46         {
47             $error = "Mobile number must be numeric ";
48         }
49         //check mobile number is 10 digits
50         elseif (strlen($mobile) != 10)
51         {
52             $error = "Mobile Number must be 10 digits";
53         }
54         //check password is at least 6 characters
55         elseif (strlen($password) < 6)
56         {
57             $error = "Password must be at least 6 characters long.";
58         }
59
60         // check for duplicate username
61         else
62         {
63             $check_user_sql = "SELECT * FROM users WHERE username = '$username'";
64             $result = $conn->query(query: $check_user_sql);
65
66             if ($result->num_rows > 0)
67             {
68                 $error = "Username already exists!";
69             }
70             else
71             {
72                 // prepare & execute sql query
73                 $sql = "INSERT INTO users (username, password, first_name, last_name, address_line1, address_line2, city, telephone, mobile)
74                     VALUES ('$username', '$password', '$first_name', '$last_name', '$address_line1', '$address_line2', '$city', '$telephone', '$mobile')";
75                 //display success with inline style
76                 if ($conn->query(query: $sql) === TRUE)
77                 {
78                     echo "<p style='color:green;'>New user registered successfully!</p>";
79                 }
80                 else
81                 {
82                     $error = "Error: " . $conn->error;
83                 }
84             }
85         }
86     }
87 }
```

```

81         $error = "Error: " . $conn->error;
82     }
83 }
84 }
85 }
86
87 // Close database connection
88 $conn->close();
89 }
90
91 <!DOCTYPE html>
92 <html lang="en">
93 <head>
94     <meta charset="UTF-8">
95     <meta name="viewport" content="width=device-width, initial-scale=1.0">
96     <title>User Registration</title>
97     <link rel="stylesheet" href="style.css">
98 </head>
99 <body class="register_body">
100     <!-- header section with nav links -->
101     <header>
102         <nav>
103             <ul>
104                 <li><a href="login.php">Login</a></li>
105                 <li><a href="register.php">Register</a></li>
106                 <li><a href="search.php">Search Books</a></li>
107                 <li><a href="reserved.php">Reserved Books</a></li>
108             </ul>
109         </nav>
110     </header>
111     <!-- user registration form -->
112     <h2>User Registration</h2>
113     <php if (!empty($error)) { echo "<p style='color:red;'>$error</p>"; } >
114     <form action="register.php" method="post">
115         <label for="username">Username:</label>
116         <input type="text" id="username" name="username" ><br><br>
117
118         <label for="password">Password:</label>
119         <input type="password" id="password" name="password" ><br><br>
120
121         <label for="confirm_password">Confirm Password:</label>
122         <input type="password" id="confirm_password" name="confirm_password" ><br><br>
123
124         <label for="first_name">First Name:</label>
125         <input type="text" id="first_name" name="first_name" ><br><br>
126
127         <label for="last_name">Last Name:</label>
128         <input type="text" id="last_name" name="last_name" ><br><br>
129
130         <label for="address1">Address Line 1:</label>
131         <input type="text" id="address1" name="address_line1" ><br><br>
132
133         <label for="address2">Address Line 2:</label>
134         <input type="text" id="address2" name="address_line2" ><br><br>
135
136         <label for="city">City:</label>
137         <input type="text" id="city" name="city" ><br><br>
138
139         <label for="telephone">Telephone:</label>
140         <input type="text" id="telephone" name="telephone" ><br><br>
141
142         <label for="mobile">Mobile:</label>
143         <input type="text" id="mobile" name="mobile" ><br><br>
144
145         <input type="submit" value="Register">
146     </form>
147     <p>Already have an account? <br><a href="login.php">Login here</a></p>
148     <br><br>
149
150     <!-- common footer -->
151     <div class = "footer">
152         <footer>Jason Gaynor's Library Website</footer>
153     </div>
154
155 </body>
156 </html>
157
158

```

This is the registration page. It allows users to create a new account on the website by filling out a form with their details.

The page has many validation checks using If statements in php to meet the criteria required to create a User.

If a user doesn't meet the criteria their submission will result in the corresponding error message.

For example if a user has a password of less then length 6, the \$error variable will be updated to "Password must be at least 6 letters long"(line 57) and will be displayed to the user (on line 113) in which there is an if statement that checks if the \$error variable is empty and if it isn't, it will echo the contents of \$error in red using inline styling.

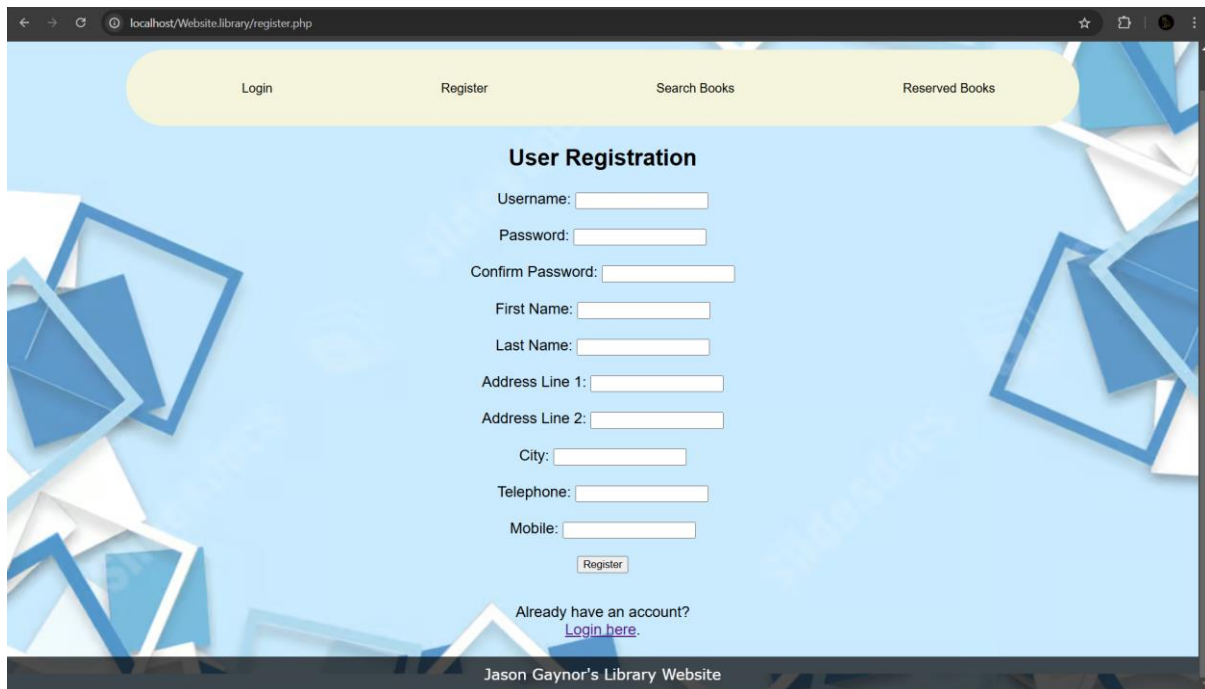
If the input from the user passes all validations, it will then query the database to check if there already exists a user with the same Username as this is a primary key and cant be duplicated. (line 69)

Once the username is unique and all details meet criteria, it will then show the message "New User registered successfully!" (seen on line 78).

It will then enter the users details into the User table in the mysql database storing their details for future sign in

Other components of the page include the common header and footer that are found across all pages along with an

option to sign in if the user already has an account.
Here is the Register page as viewed in the browser



Here you can see the common header at the top and the common footer across all pages along with the form's fields for the required information

Once all info is submitted via the register button it will update the page to display "New User registered successfully" in which the user can then Login to the system with their details

Login.php

```
login.php > ...
1  <?php
2  session_start();
3
4
5  // database connection details
6  $servername = "localhost";
7  $dbusername = "root";
8  $dbpassword = "";
9  $dbname = "website";
10
11 // Connect to database
12 $conn = new mysqli($servername, $dbusername, $dbpassword, $dbname);
13
14 // check connection
15 if ($conn->connect_error) {
16     die("Connection failed: " . $conn->connect_error);
17 }
18
19 $error = ""; // initialize error message variable
20
21 // handle form submission
22 if ($_SERVER["REQUEST_METHOD"] == "POST") {
23     // check if username and password fields are set
24     if (isset($_POST['username'], $_POST['password'])) {
25         $username = trim(string: $_POST['username']);
26         $password = trim(string: $_POST['password']);
27         // Prepare sql query
28         $stmt = $conn->prepare(query: "SELECT username FROM users WHERE username = ? AND password = ?");
29         //bind $username and $password to ??, ss indicating they're strings
30         $stmt->bind_param(types: "ss", vars: $username, vars: $password);
31         $stmt->execute(); // execute statement
32         $result = $stmt->get_result();
33
34         //check if result matches 1 row ie only one user record matches from database
35         if ($result->num_rows == 1) {
36             $_SESSION['username'] = $username; // store username in session
37             header(header: "Location: registered.php");
38             exit();
39         } else {
40
41         }
42     }
43 }
```

```
login.php > ...
42     } else {
43         $error = "Invalid username or password."; // set $error variable initialised earlier
44     }
45     $stmt->close();
46 }
47 }
48 $conn->close();
49 }>
50
51 <!DOCTYPE html>
52 <html lang="en">
53 <head>
54     <meta charset="UTF-8">
55     <meta name="viewport" content="width=device-width, initial-scale=1.0">
56     <title>Login</title>
57     <link rel="stylesheet" href="style.css">
58 </head>
59 <body class="login_body">
60     <!-- common header section with nav links -->
61     <header>
62         <nav>
63             <ul> <!-- nav links -->
64                 <li><a href="login.php">Login</a></li>
65                 <li><a href="register.php">Register</a></li>
66                 <li><a href="search.php">Search Books</a></li>
67                 <li><a href="reserved.php">Reserved Books</a></li>
68             </ul>
69         </nav>
70     </header>
71
72
73
74     <h2>Login</h2>
75     <!-- checks if form was submitted and that the error variable isnt empty -->
76     <?php if ($_SERVER["REQUEST_METHOD"] == "POST" && !empty($error)): ?>
77         <div class="error"><?php echo htmlspecialchars(string: $error); ?></div>
78     <?php endif; ?>
79     <!-- create form -->
80     <form method="POST" action="login.php">
81
82         <div class="form-group">
83             <label for="username">Username:</label>
```

```

84             <input type="text" id="username" name="username" required>
85         </div>
86         <div class="form-group">
87             <label for="password">Password:</label>
88             <input type="password" id="password" name="password" required>
89         </div>
90         <button type="submit" class="btn">Login</button>
91     </form>
92     <?php if (!empty($error)): ?>
93         <div class="error"><?php echo htmlspecialchars(string: $error); ?></div>
94     <?php endif; ?>
95     <!-- common footer -->
96     <div class="footer">
97         <p>Jason Gaynor's Library Website</p>
98     </div>
99 </body>
100
101 </html>
102
103 </body>
104 </html>
105
```

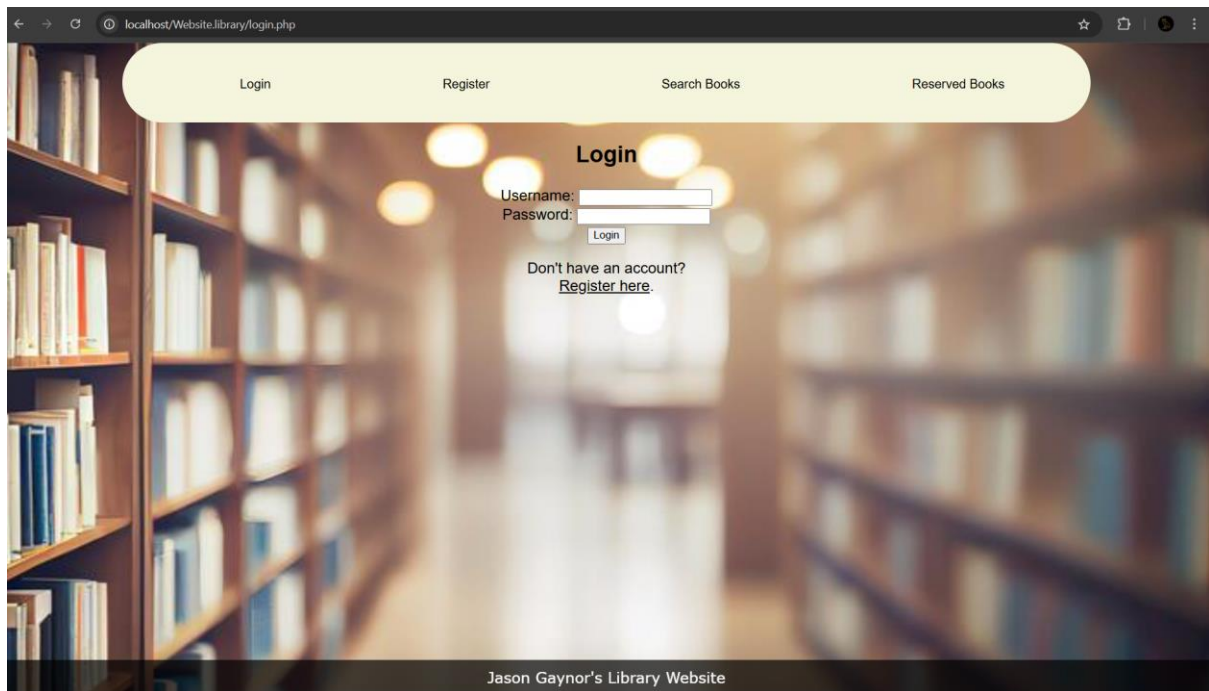
This is the page for user login.

The core of this page is the Login form, - it requires a username and password to be entered and once entered the user presses the 'Login' button. The form uses the Post method to send the username and password to the server. (this can be seen on line 80)

The error message – if a user submits the form with an incorrect username and password that isn't in the users table in the database, the error variable will be set to "Invalid username or password" (seen on line 43)

The sql statement (line 28) will query the database checking if the username or password matches in the user table. If it matches, it will start a session for that user and bring them to registered.php (line 38) which will then display options to search for a book, view reserved books or sign out.

Here is the Login page as viewed in the browser



Here you can see again the common header&footer along with the login form for username and password. There is also the option to register if the user does not already have an account

Registered.php

```

1  <?php
2  session_start();
3
4  // check if user is logged in
5  if (!isset($_SESSION['username']) || empty($_SESSION['username'])) {
6      header(header: "Location: login.php"); // back to login if not logged in
7      exit(); //
8  }
9  //logout
10 if (isset($_GET['action']) && $_GET['action'] == 'logout') {
11     session_destroy();
12     header(header: "Location: login.php"); // redirect to login page
13     exit();
14 }
15
16 // database connection settings
17 $servername = "localhost";
18 $username = "root";
19 $password = "";
20 $database = "website";
21
22 // create connection to the database
23 $conn = new mysqli(hostname: $servername, username: $username, password: $password, database: $database);
24
25 // check connection
26 if ($conn->connect_error) {
27     die("Connection failed: " . $conn->connect_error);
28 }
29
30 ?>
31
32 <!DOCTYPE html>
33 <html lang="en">
34 <head>
35     <meta charset="UTF-8">
36     <meta name="viewport" content="width=device-width, initial-scale=1.0">
37     <title>User Registration</title>
38     <link rel="stylesheet" href="style.css">
39 </head>
40 <body class="registered_body">
41     <!-- header section with nav links -->
42     <header>

```

```

39 </head>
40 <body class="registered_body">
41 <!-- header section with nav links -->
42 <header>
43 <nav>
44 <ul>
45 <li><a href="login.php">Login</a></li>
46 <li><a href="register.php">Register</a></li>
47 <li><a href="search.php">Search Books</a></li>
48 <li><a href="reserved.php">Reserved Books</a></li>
49 <li><a href="?action=logout" class="btn">Sign Out</a></li> <!-- sign out link -->
50 </ul>
51 </nav>
52 </header>
53
54 <h2>Welcome, <?php echo htmlspecialchars(string: $_SESSION['username']); ?>!<br><br> Choose an option:</h2>
55
56 <div class="registered_links">
57 <a href="search.php" class="btn">Search Books</a>
58 <a href="reserved.php" class="btn">See Your Reserved Books</a>
59 <br><br>
60 <a href="?action=logout" class="btn">Sign Out</a>
61 </div>
62
63 <!-- Footer -->
64 <div class="footer">
65 <footer>Jason Gaynor</footer>
66 </div>
67 </body>
68 </html>
69

```

This is registered.php, the page the user is taken to after logging in.

It is minimalistic and only displays 3 options -> Search Books, See reserved books the user has and option to sign out.

it starts with checking if the user is logged in (line 5). If the user isn't logged in, it will send them back to login.php through the header (on line 6)

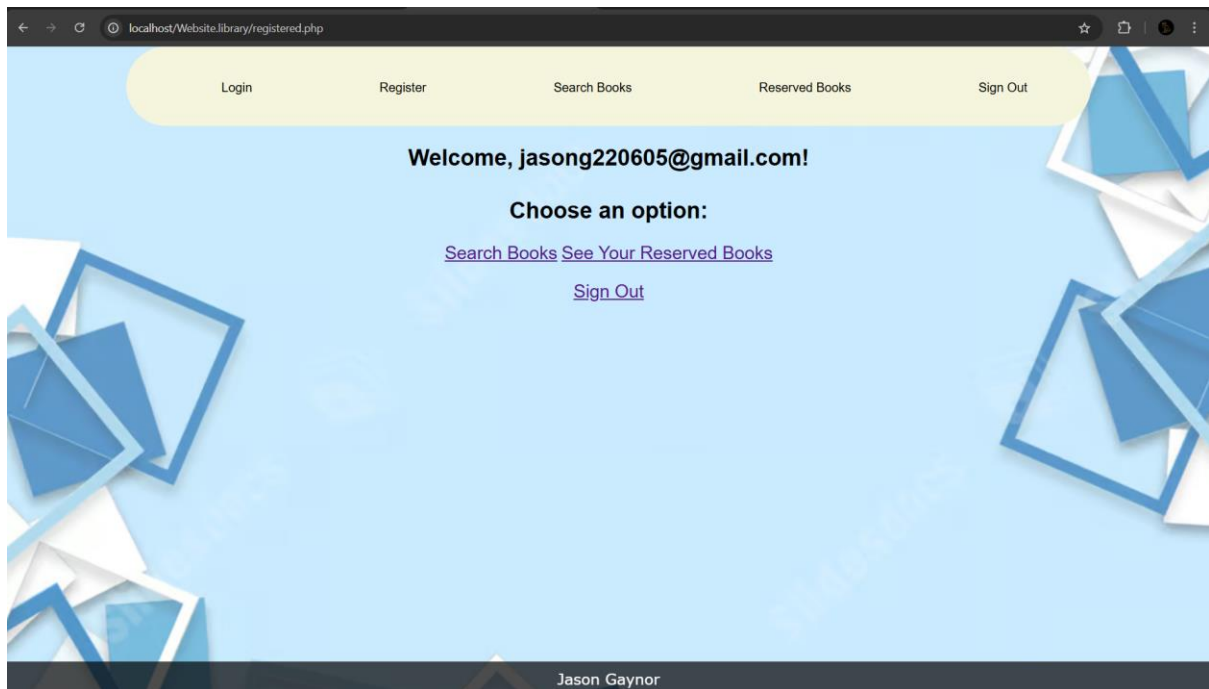
this prevents users who aren't signed in from accessing data that should only be visible by signed in users

After this we see on line 10 the code for handling the logout process. `$GET['action']` checks if the url contains a parameter named action and it ensures the value of the action parameter is logout " (&& `$_GET['action']` == 'logout') " This refers to line 60 where I added the logout option to the header. This log out option is only available on pages that require a login to access.

In the html section of the code, it welcomes the logged in user by their username (line 54) using `htmlspecialchars` for security and and displays the username that's logged using `$_SESSION`.

Along with this is the 3 links mentioned earlier Search Books, See reserved books, and sign out

Here is registered.php as seen in the browser



Not much to talk about here but you can see now that the sign out option has been added to the header and the Welcome (username)

Search.php

```

1  <?php
2  session_start();
3
4  // check if user is logged in
5  if (!isset($_SESSION['username']) || empty($_SESSION['username'])) {
6      header("Location: login.php"); // back to login if not logged in
7      exit();
8  }
9  //logout
10 if (isset($_GET['action']) && $_GET['action'] == 'logout') {
11     session_destroy();
12     header("Location: login.php"); // redirect to login page
13     exit();
14 }
15
16 // database connection details
17 $servername = "localhost";
18 $dbusername = "root";
19 $dbpassword = "";
20 $dbname = "website";
21
22 // create a connection to database
23 $conn = new mysqli($servername, $dbusername, $dbpassword, $dbname);
24
25 // check connection
26 if ($conn->connect_error) {
27     die("connection failed: " . $conn->connect_error);
28 }
29
30 // fetch categories
31 $categoryQuery = "SELECT Category_desc, CategoryID FROM categories";
32 $categoryResult = $conn->query($categoryQuery);
33 $categories = [];
34 while ($row = $categoryResult->fetch_assoc()) {
35     $categories[] = $row;
36 }
37
38 // Initialize search parameters
39 $title = $author = $category = "";
40 $books = [];
41
42 // initialize search parameters for book search

```

First part of search.php is mainly the common pieces of code seen across each page ie check login, logout process, database connection etc

We see on line 31 the query to fetch the categories to display them in a dropdown menu.

This query is stored in \$categoryQuery, which is then stored in \$CategoryResult which executes the sql query.

I then initialized an empty array called \$categories to store the categories.

Then in the while loop (line 34), it retrieves each row as an associative array using fetch_assoc() and adds each row to the categories array


```

43 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
44     // Capture form data for search
45     $title = $_POST['title'] ?? '';
46     $author = $_POST['author'] ?? '';
47     $category = $_POST['category'] ?? '';
48
49     // build SQL query based on search criteria
50     $sql = "SELECT books.ISBN, books.BookTitle, books.Author, books.Edition, books.Year, categories.Category_desc, books.Rese
51         FROM books
52         JOIN categories ON books.Category = categories.CategoryID
53         WHERE 1=1";
54
55     // add conditions based on user input allowing for similar search
56     if (empty($title)) {
57         $sql .= " AND books.BookTitle LIKE '%" . $conn->real_escape_string(string: $title) . "%'";
58     }
59     if (empty($author)) {
60         $sql .= " AND books.Author LIKE '%" . $conn->real_escape_string(string: $author) . "%'";
61     }
62     if (empty($category)) {
63         $sql .= " AND books.Category = " . intval(value: $category);
64     }
65
66     // execute query
67     $result = $conn->query(query: $sql);
68
69     // fetch books
70     while ($row = $result->fetch_assoc()) {
71         $books[] = $row;
72     }
73 }
74
75 // Close the connection
76 $conn->close();
77 }
78
79 <!DOCTYPE html>
80 <html lang="en">
81 <head>
82     <meta charset="UTF-8">
83     <meta name="viewport" content="width=device-width, initial-scale=1.0">
84     <title>Search Books</title>

```

line 43 – checks if form was submitted using POST

inside the if statement is retrieving the user entries for the title, author and category

line 50 – This is the sql query for retrieving the fields from the books and categories table.

it uses a simple join statement to display the category description as there is only the category id in the books table.

After this are conditions to allow for similar search by the user eg The user can enter “computer” and the book

“computers in business” will show. This is done using the Like ‘%’

line 63 - intval(\$category) ensures the category input is an integer

line 70 – displays all the books that match the search criteria

```

85 <link rel="stylesheet" href="style.css">
86 </head>
87 <body class="search_body">
88
89 <header>
90     <nav>
91         <ul> <!-- nav links -->
92             <li><a href="login.php">login</a></li>
93             <li><a href="register.php">Register</a></li>
94             <li><a href="search.php">Search Books</a></li>
95             <li><a href="reserved.php">Reserved Books</a></li>
96             <li><a href="action=logout" class="btn">Sign out</a></li>
97         </ul>
98     </nav>
99 </header>
100
101 <h2>Search for a Book</h2>
102
103 <!-- search form -->
104 <form method="POST" action="search.php">
105     <div class="form-group">
106         <label for="title">Book Title:</label>
107         <input type="text" id="title" name="title" value="{?php echo htmlspecialchars(string: $title); ?}" placeholder="Search by title">
108     </div>
109     <div class="form-group">
110         <label for="author">Author:</label>
111         <input type="text" id="author" name="author" value="{?php echo htmlspecialchars(string: $author); ?}" placeholder="Search by author">
112     </div>
113     <div class="form-group">
114         <label for="category">Category:</label>
115         <select name="category" id="category">
116             <option value="">Select a Category</option>
117             <?php foreach ($categories as $cat): ?>
118                 <option value="{?php echo $cat['CategoryID']; ?}" <?php echo $cat['CategoryID'] == $category ? 'selected' : '' ?>
119                     <?php echo $cat['Category_desc']; ?>
120                 </option>
121             </?php endforeach: ?>
122         </select>
123     </div>

```

Moving onto the html

we have the common header with nav links again including the logout feature now.

Below this is the search form that is the main display for the page, it holds input fields that retains the the user input and stores them in corresponding variables eg \$title , \$author

there is also added placeholders such as “Search by title” for added aesthetic

Below this we have the category dropdown (line 116) where the foreach loop generates each option from the categories array. It also ensures that the selected category remains selected after the form submission (line120)

```
123         <?php endforeach; ?>
124     </select>
125 </div>
126 <br>
127 <button type="submit" class="btn">Search</button>
128 </form>
129
130 <!-- display error message if reservation failed -->
131 <?php if (isset($errorMessage)): ?>
132     <div class="error-message"><?php echo htmlspecialchars(string: $errorMessage); ?></div>
133 <?php endif; ?>
134
135 <h3>Search Results</h3>
136
137 <?php if (empty($books)): ?>
138     <div class="book-results">
139         <!-- display books from array -->
140         <?php foreach ($books as $book): ?>
141             <div class="book-item">
142                 <b><?php echo htmlspecialchars(string: $book['BookTitle']); ?></b><br>
143                 Author: <?php echo htmlspecialchars(string: $book['Author']); ?><br>
144                 Category: <?php echo htmlspecialchars(string: $book['Category_desc']); ?><br>
145
146                 <!-- checks if book reserved and if so disables reserve button -->
147                 <?php if ($book['Rese'] == 'Y'): ?>
148                     <button class="reserved-btn" disabled>Reserved</button>
149                 <?php else: ?>
150
151                     <!-- create a form for reservation -->
152                     <form action="after_reserved.php" method="POST">
153                         <input type="hidden" name="isbn" value="<?php echo $book['ISBN']; ?>">
154                         <button type="submit" class="btn">Reserve</button>
155                     </form>
156                 <br><br>
157                 <?php endif; ?>
158             </div>
159         <?php endforeach; ?>
160     </div>
161 <?php elseif ($_SERVER['REQUEST_METHOD'] == 'POST'): ?>
162     <p>No books found matching your search criteria.</p>
163 <?php endif; ?>
164
165 <!-- common footer -->
166 <div class = "footer">
167     <footer>Jason Gaynor's Library Website</footer>
168 </div>
169
170
171 </body>
172 </html>
173
```

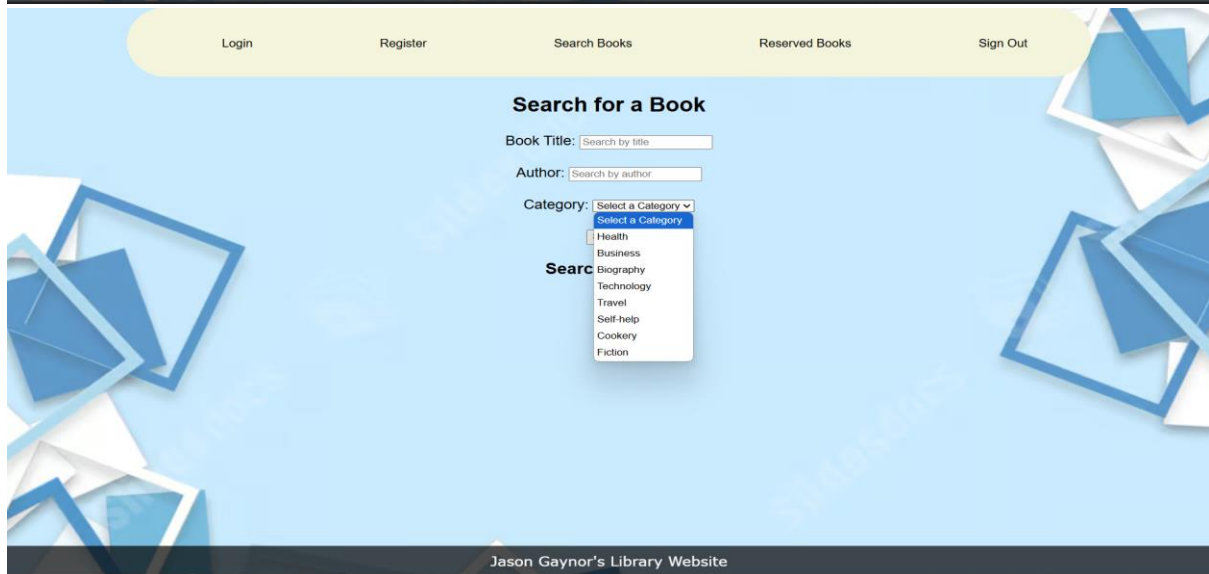
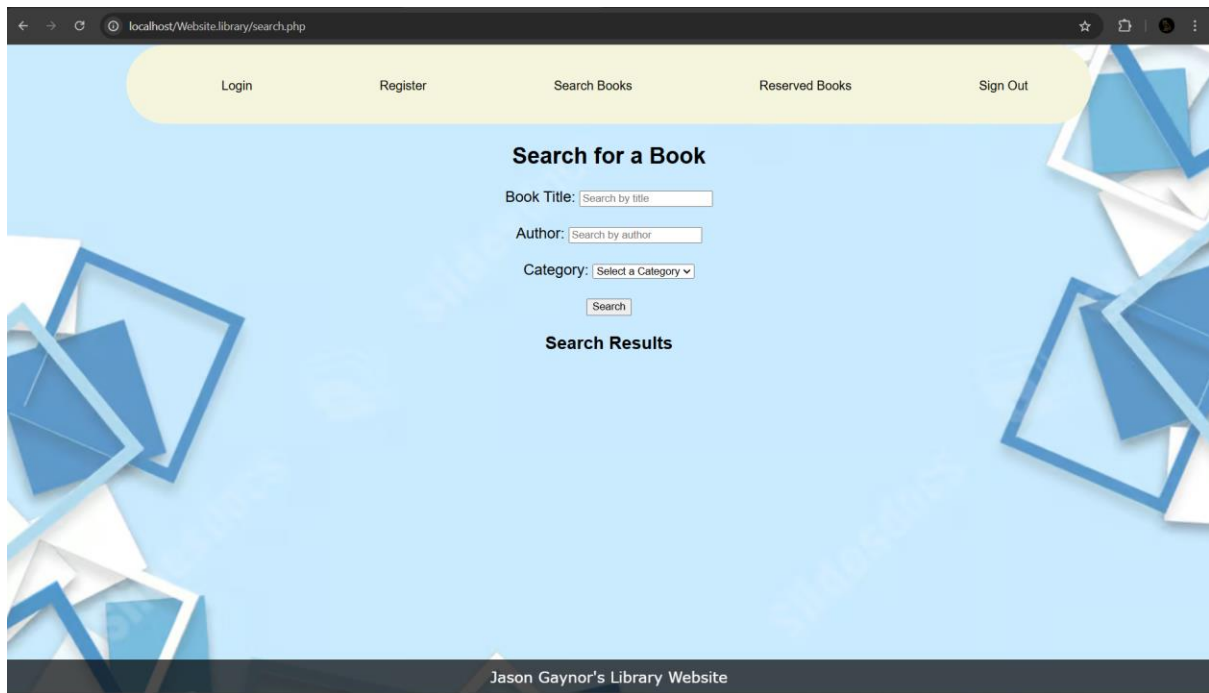
line 140 – displays each book by iterating through each book in the \$books array

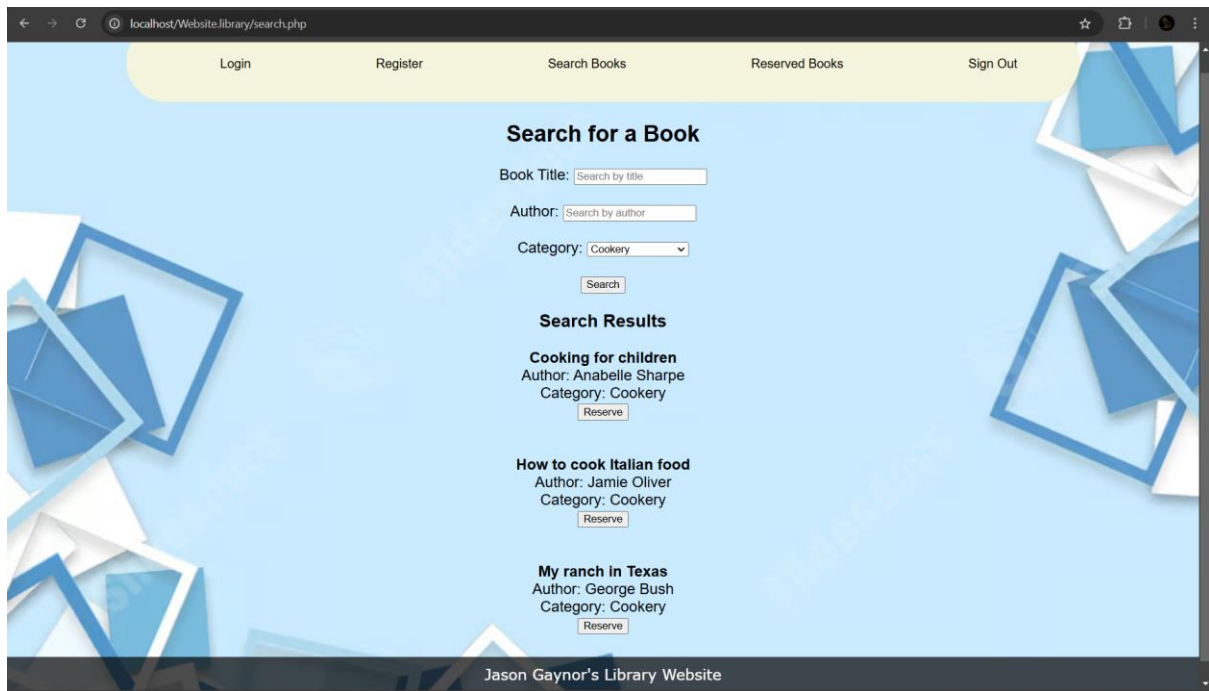
Line 147 – checks If a book is reserved (‘Y’ for yes). If a book is reserved, it disables the reserve button to stop a user reserving a book that is already reserved

line 152 – Sends the book ISBN to after_reserved.php using POST so it can process the reservation.

line 162 – handles the event that no books matched the search criteria after form submission

Here is search.php as it is displayed in the browser





Here you can see the dropdown menu and the corresponding results for the example category search 'Cookery' once Reserve is pressed on a book, the user is directed to after_reserved.php where the reservation is processed

After_reserved.php

```

1  <?php
2  session_start();
3
4  // check if user is logged in
5  if (!isset($_SESSION['username']) || empty($_SESSION['username'])) {
6      header("Location: login.php"); // back to login if not logged in
7      exit(); //
8  }
9  //logout
10 if (isset($_GET['action']) && $_GET['action'] == 'logout') {
11     session_destroy();
12     header("Location: login.php"); // redirect to login page
13     exit();
14 }
15
16 // database connection details
17 $servername = "localhost";
18 $dbUsername = "root";
19 $dbPassword = "";
20 $dbName = "website";
21
22 // Create a connection to database
23 $conn = new mysqli(hostname: $servername, username: $dbUsername, password: $dbPassword, database: $dbName);
24
25 // Check the connection
26 if ($conn->connect_error) {
27     die("Connection failed: " . $conn->connect_error);
28 }
29
30 // Get the ISBN from the POST request
31 $isbn = $_POST['isbn'] ?? '';
32
33 // If ISBN is not provided, redirect back to search page
34 if (empty($isbn)) {
35     header("Location: search.php");
36     exit();
37 }
38
39 // Check if the book is already reserved
40 $sql = "SELECT * FROM reservedbooks WHERE ISBN = ?";
41 $stmt = $conn->prepare($sql);
42 $stmt->bind_param("s", $isbn);

```

Here on after_reserved.php the book reservation is processed with minimal html on line 31 I'm retrieving the isbn from the post request that was sent from search.php with an if statement below that redirects user back to search provided there was no isbn. then we check if the book has been reserved (line 40) using sql query to select from reserved books where isbn is the isbn of the book that the user chose to reserve

```

43 $stmt->execute();
44 $result = $stmt->get_result();
45
46 // If the book is already reserved, show an error message
47 if ($result->num_rows > 0) {
48     $errorMessage = "This book has already been reserved.";
49 } else {
50     // Reserve the book for the user
51     $username = $_SESSION['username'];
52     $dateOfReservation = date(format: 'Y-m-d H:i:s'); // Current date and time
53
54     $insertSql = "INSERT INTO reservedbooks (username, ISBN, DateOfReservation) VALUES (?, ?, ?)";
55     $insertStmt = $conn->prepare(query: $insertSql);
56     $insertStmt->bind_param(types: "sss", var: &$username, vars: &$isbn, $dateOfReservation);
57
58     if ($insertStmt->execute()) {
59         $updateSql = "UPDATE books SET Rese = 'Y' WHERE ISBN = ?";
60         $updateStmt = $conn->prepare(query: $updateSql);
61         $updateStmt->bind_param(types: "s", var: &$isbn);
62         $updateStmt->execute(); // Execute the update query
63
64         $successMessage = "Book reserved successfully!";
65     } else {
66         $errorMessage = "Failed to reserve the book. Please try again later.";
67     }
68 }
69
70 // Close the connection
71 $conn->close();
72 }
73
74 <DOCTYPE html>
75 <html lang="en">
76 <head>
77     <meta charset="UTF-8">
78     <meta name="viewport" content="width=device-width, initial-scale=1.0">
79     <title>After Reservation</title>
80     <link rel="stylesheet" href="style.css">
81 </head>
82 <body class="after_reserved_body">
83
84

```

line 47 displays if the book has already been reserved by checking if there is already exists a reservation for that books isbn

then in the else block is for reserving the book for the user.

It stores the current users username in the session and gets the current date and time using the built in date function.

Then it prepares the sql query (line 54) for inserting the book into reserved books

on line 58, it updates the Rese collum in the books table to Y if the user reserves the book to indicate that the book is now reserved

```

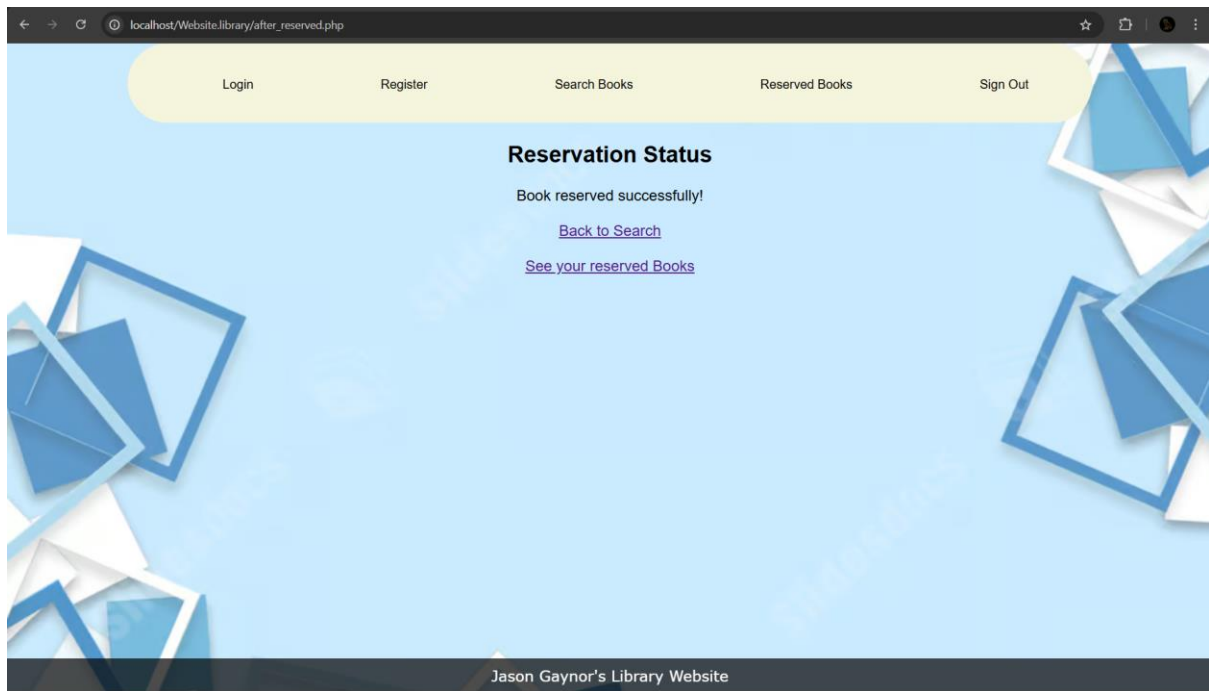
85 <header>
86     <nav>
87         <ul>
88             <li><a href="login.php">Login</a></li>
89             <li><a href="register.php">Register</a></li>
90             <li><a href="search.php">Search Books</a></li>
91             <li><a href="reserved.php">Reserved Books</a></li>
92             <li><a href="?action=logout" class="btn">Sign Out</a></li>
93         </ul>
94     </nav>
95 </header>
96
97 <h2>Reservation Status</h2>
98
99 <?php if (isset($successMessage)): ?>
100     <div class="success-message"><?php echo htmlspecialchars(string: $successMessage); ?></div>
101 <?php elseif (isset($errorMessage)): ?>
102     <div class="error-message"><?php echo htmlspecialchars(string: $errorMessage); ?></div>
103 <?php endif; ?>
104
105 <br>
106 <a href="search.php" class="btn">Back to Search</a>
107 <br>
108 <a href="reserved.php">See your reserved Books</a>
109
110 <!--common footer -->
111 <div class = "footer">
112     <footer>Jason Gaynor's Library Website</footer>
113 </div>
114
115
116 </body>
117 </html>
118

```

In the html there is not much, it contains two links Back to search and see reserved books.

It also displays the corresponding message for success or failure in reserving the book

here is after_reserved.php as it is displayed in the browser



Here you can see success message and the two links

reserved.php

```

1  <?php
2  session_start();
3
4  // check if user is logged in
5  if (!isset($_SESSION['username']) || empty($_SESSION['username'])) {
6      header(header: "location: login.php"); // back to login if not logged in
7      exit(); //
8  }
9  //logout
10 if (isset($_GET['action']) && $_GET['action'] == 'logout') {
11     session_destroy();
12     header(header: "location: login.php"); // redirect to login page
13     exit();
14 }
15
16 // database connection details
17 $servername = "localhost";
18 $dbUsername = "root";
19 $dbPassword = "";
20 $dbName = "website";
21
22 // create connection to database
23 $conn = new mysqli(hostname: $servername, username: $dbUsername, password: $dbPassword, database: $dbName);
24
25 // check connection
26 if ($conn->connect_error) {
27     die("connection failed: " . $conn->connect_error);
28 }
29
30 // initialize reserved books array
31 $reservedBooks = [];
32
33 // get reserved books for current user
34 $username = $_SESSION['username'];
35 $reservedQuery = "SELECT books.ISBN, books.BookTitle, books.Author, books.Edition, books.Year, categories.Category_desc
36                  FROM reservedbooks
37                  JOIN books ON reservedbooks.ISBN = books.ISBN
38                  JOIN categories ON books.Category = categories.CategoryID
39                  WHERE reservedbooks.username = ?";
40
41 $stmt = $conn->prepare(query: $reservedQuery);
42 $stmt->bind_param(types: "s", var: &$username);

```

line 34 we retrieve the username from the session, we then store the sql query to get the users reserved books in \$reservedQuery

```

43 $stmt->execute();
44 $result = $stmt->get_result();
45 //fetches each row and adds to $reservedbooks array
46 while ($row = $result->fetch_assoc()) {
47     $reservedBooks[] = $row;
48 }
49
50 // handle removal of a reservation
51 if (isset($_POST['remove']) && !empty($_POST['remove'])) {
52     $isbnToRemove = $_POST['remove'];
53
54     // delete the reservation from reservedbooks table
55     $removeQuery = "DELETE FROM reservedbooks WHERE username = ? AND ISBN = ?";
56     $removeStat = $conn->prepare(query: $removeQuery);
57     $removeStat->bind_param(types: "ss", var: &$username, vars: &$isbnToRemove);
58
59     if ($removeStat->execute()) {
60         // update the Rese field to N in books table
61         $updateSql = "UPDATE books SET Rese = 'N' WHERE ISBN = ?";
62         $updateStat = $conn->prepare(query: $updateSql);
63         $updateStat->bind_param(types: "s", var: &$isbnToRemove);
64         $updateStat->execute();
65
66         // after successful removal, redirect with success message
67         header(header: "Location: reserved.php?success=Reservation removed successfully");
68         exit();
69     } else {
70         $errorMessage = "Failed to remove reservation";
71     }
72 }
73
74 }
75
76 // Close the connection
77 $conn->close();
78 ?>
79
80 <!DOCTYPE html>
81 <html lang="en">
82 <head>
83     <meta charset="UTF-8">
84     <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

line 46 uses while loop to iterate through each row in the results then stores each book in the \$reservedbooks array then on line 51, it handles the removal of a reservation by checking if a remove request has been sent in POST. \$isbnToRemove gets the isbn of the book to be removed from the form submission

On line 60, it checks whether the deletion was successful and if it was, it uses an sql query to update the rese column in books table to 'N' indicating that the book is no longer reserved and displays a message indicating successful removal. If the removal was unsuccessful the else clause is processed and \$errorMessage variable is set to "failed to remove reservation"

```

81 <html lang="en">
82 <head>
83     <title>Your Reserved Books</title>
84     <link rel="stylesheet" href="style.css">
85 </head>
86 <body class="reserved_body">
87
88 <header> <!-- nav links -->
89 <nav>
90     <ul>
91         <li><a href="login.php">Login</a></li>
92         <li><a href="register.php">Register</a></li>
93         <li><a href="search.php">Search Books</a></li>
94         <li><a href="reserved.php">Reserved Books</a></li>
95         <li><a href="?action=logout" class="btn">Sign Out</a></li>
96     </ul>
97 </nav>
98 </header>
99
100 <h2>Your Reserved Books</h2>
101
102 <!-- Display success message if reservation removed -->
103 <?php if (isset($_GET['success'])): ?>
104     <div class="success-message"><?php echo htmlspecialchars(string: $_GET['success']); ?></div>
105 <?php endif; ?>
106
107 <!-- Display error message if reservation removal failed -->
108 <?php if (isset($errorMessage)): ?>
109     <div class="error-message"><?php echo htmlspecialchars(string: $errorMessage); ?></div>
110 <?php endif; ?>
111
112 <?php if (empty($reservedBooks)): ?>
113     <div class="reserved-books-list">
114         <br><br>
115         <?php foreach ($reservedBooks as $book): ?>
116             <div class="book-item">
117                 <b><?php echo htmlspecialchars(string: $book['BookTitle']); ?><br>
118                 Author: <?php echo htmlspecialchars(string: $book['Author']); ?><br>
119                 Edition: <?php echo htmlspecialchars(string: $book['Edition']); ?><br>
120                 Year: <?php echo htmlspecialchars(string: $book['Year']); ?><br>
121                 Category: <?php echo htmlspecialchars(string: $book['Category_desc']); ?><br>
122             </div>
123         </foreach>
124     </div>

```

line 105 contains if statement that checks for 'success' in url and displays success message, if not successful it will display the error message

line 114 checks if the \$reservedbooks array isn't empty, if not it will display the users reserved books

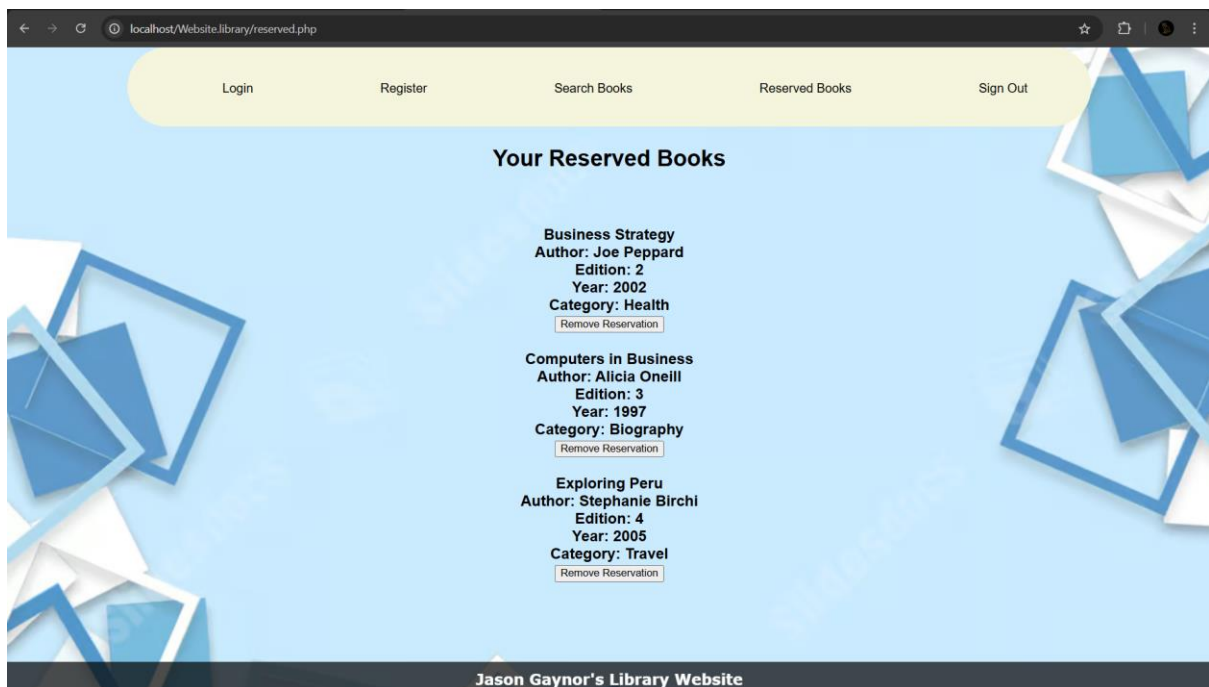

```

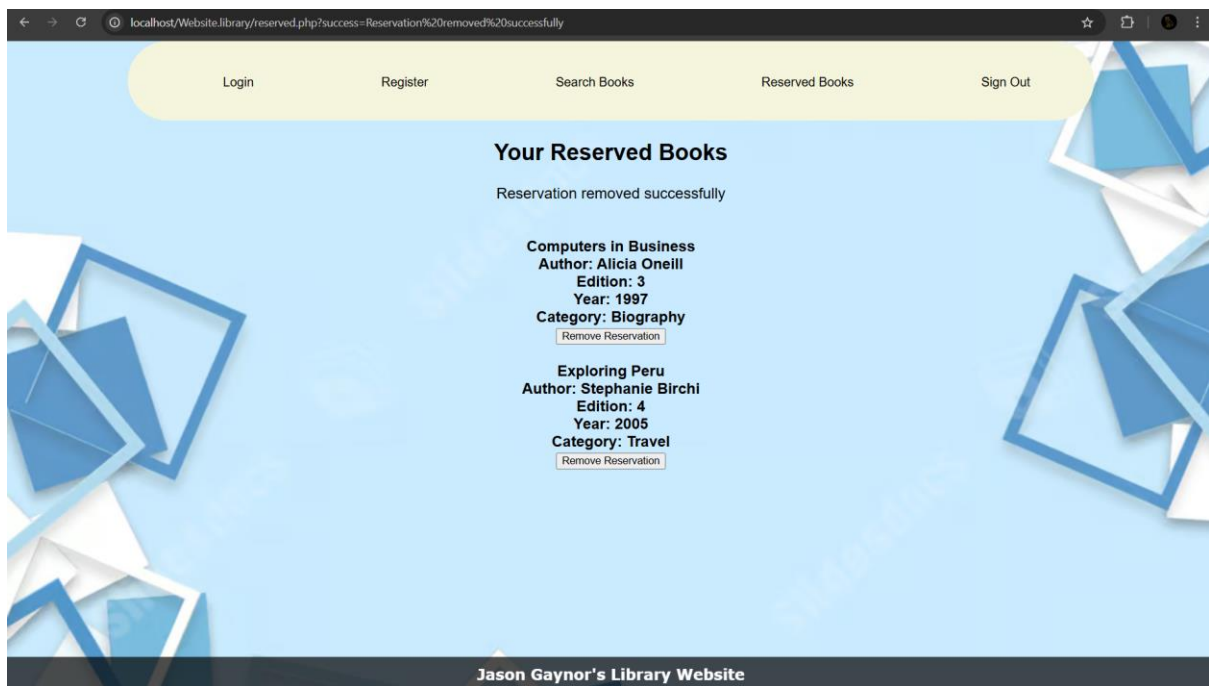
122     Year: <?php echo htmlspecialchars(string: $book['Year']); ?><br>
123     Category: <?php echo htmlspecialchars(string: $book['Category_desc']); ?><br>
124
125     <!-- Form to remove reservation -->
126     <form action="reserved.php" method="POST" style="display:inline;">
127         <input type="hidden" name="remove" value="<?php echo $book['ISBN']; ?>">
128         <button type="submit" class="btn">Remove Reservation</button>
129     </form>
130     <br><br>
131 </div>
132 <?php endforeach; ?>
133 </div>
134 <?php else: ?>
135     <p>You have no reserved books at the moment.</p>
136     <a href="search.php">Back to search</a>
137 <?php endif; ?>
138
139 <!--common footer -->
140 <div class = "footer">
141     <footer>Jason Gaynor's Library Website</footer>
142 </div>
143
144
145 </body>
146 </html>
147
148

```

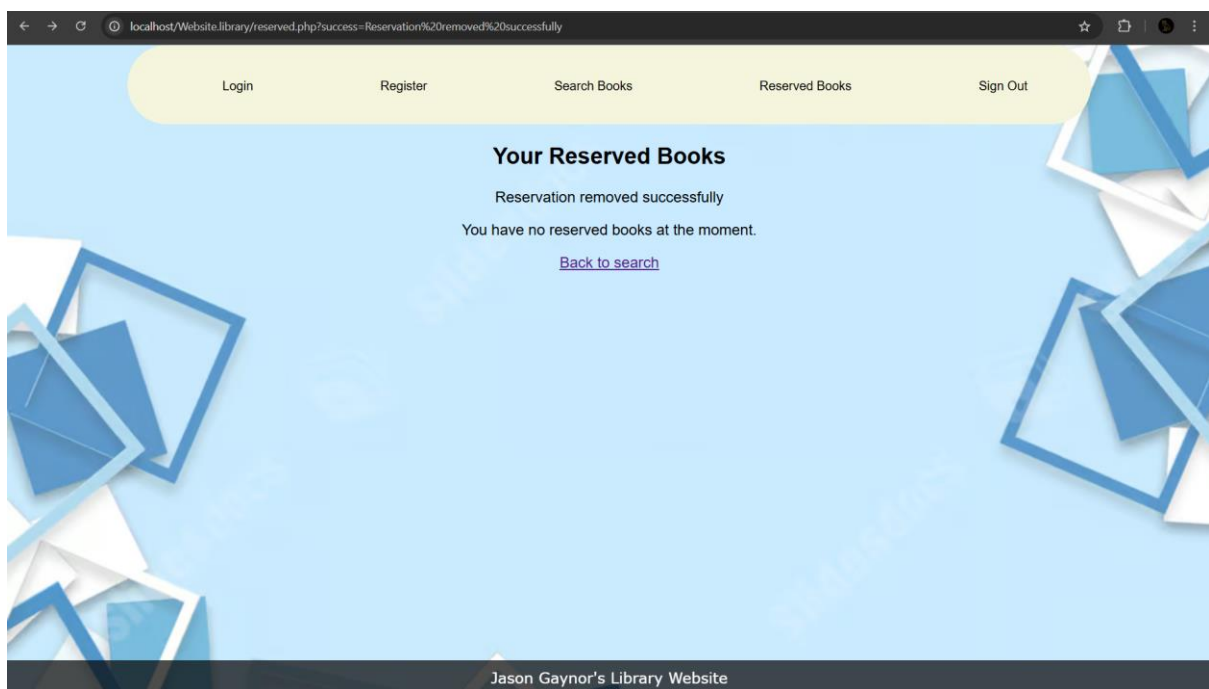
line 126 contains the simple form to remove a reservation with a button

Here is reserved.php as viewed in the browser





after removing a reservation



after removing all

Css style sheet

```
# style.css > ...
1  @import url('https://fonts.googleapis.com/css2?family=Autowide&display=swap'); /*imported font*/
2
3  /* header styling accross all pages */
4  header {
5      width: 80%; /* header width to 80% of page */
6      background-color: #f5f5dc; /* beige background color for the header */
7      padding: 20px 0; /*make header stretch longer vertically*/
8      margin: 0 auto; /* centers the header */
9      text-align: center; /* center text inside the header */
10     border-radius: 100px; /*rounded corners*/
11 }
12 /*next few styling is for links in header*/
13 nav ul {
14
15     display: flex; /* display links in a row */
16     justify-content: space-around; /* distribute links evenly across header */
17 }
18
19 nav ul li {
20     display: inline; /* display list items horizontally */
21 }
22
23
24 nav ul li a {
25     text-decoration: none; /* remove underline from links */
26     color: black; /* black text color for links */
27     font-size: 16px; /* set font size */
28     padding: 10px 15px; /* add padding to links to space them out */
29 }
30 /*makes links turn green when hovering over*/
31 nav ul li a:hover {
32     color: green;
33 }
34
35
36
37 /*Body styling*/
38 /*common accross all*/
39 html, body {
40     height: 100%;
41     margin: 0;
42     display: flex;
43 }
44
45
46
47
48 .login_body {
49     text-align: center;
50     font-family: 'Autowide', sans-serif; /*from imported font */
51     background-image: url("../Website.library/CSS/abstract-blurred-public-library-interior-space-blurry-room-with-bookshelves-by-defocused-effect-use-for-background-or-background-image-9c874dd0f4_960_540.avif"); /*copyright free image*/
52     background-size: cover; /*makes background image not repeat itself*/
53     color: black;
54     font-size: larger;
55 }
56 /*make links black */
57 .login_body a {
58     color: black;
59 }
60 /*these following declarations are common for each pages body*/
61 .register_body {
62     text-align: center;
63     font-family: 'Autowide', sans-serif; /*imported google font*/
64     font-size: larger;
65     background-image: url("../Website.library/CSS/line-professional-frame-blue-square-shape-business-powerpoint-background-9c874dd0f4_960_540.avif"); /*copyright free image*/
66     background-size: cover;
67     padding-bottom: 50px;
68 }
69 .registered_body {
70     text-align: center;
71     font-family: 'Autowide', sans-serif;
72     font-size: larger;
73     background-image: url("../Website.library/CSS/line-professional-frame-blue-square-shape-business-powerpoint-background-9c874dd0f4_960_540.avif");
74     background-size: cover;
75 }
76 .registered_links a {
77     text-align: center;
78     font-size: larger;
79 }
80
```

~

```
81
82 .search_body{
83     text-align: center;
84     font-family:'Autowide',sans-serif;
85     font-size: larger;
86     background-image: url("../Website.library/CSS/line-professional-frame-blue-square-shape-business-powerpoint-background_9c874dd0fa_960_540.avif");
87     background-size: cover;
88     padding-bottom: 50px; /*prevent footer from covering page*/
89 }
90 .after_reserved_body{
91     text-align: center;
92     font-family:'Autowide',sans-serif;
93     font-size: larger;
94     background-image: url("../Website.library/CSS/line-professional-frame-blue-square-shape-business-powerpoint-background_9c874dd0fa_960_540.avif");
95     background-size: cover;
96 }
97
98 .reserved_body{
99     text-align: center;
100     font-family:'Autowide',sans-serif;
101     font-size: larger;
102     background-image: url("../Website.library/CSS/line-professional-frame-blue-square-shape-business-powerpoint-background_9c874dd0fa_960_540.avif");
103     background-size: cover;
104 }
105
106 /*Common footer styling accross all pages*/
107 .footer {
108     background-color: rgba(0, 0, 0, 0.7); /* Black with 70% transparency */
109     color: white; /*text color*/
110     text-align: center;
111     padding: 10px 0; /*positioning of text*/
112     width: 100%; /*make it stretch accross the screen */
113     font-family:Verdana, Geneva, Tahoma, sans-serif;
114
115     /* to make it always stay at the bottom */
116     bottom: 0;
117     position: fixed;
118     margin-top: auto;
119 }
120
121
```