

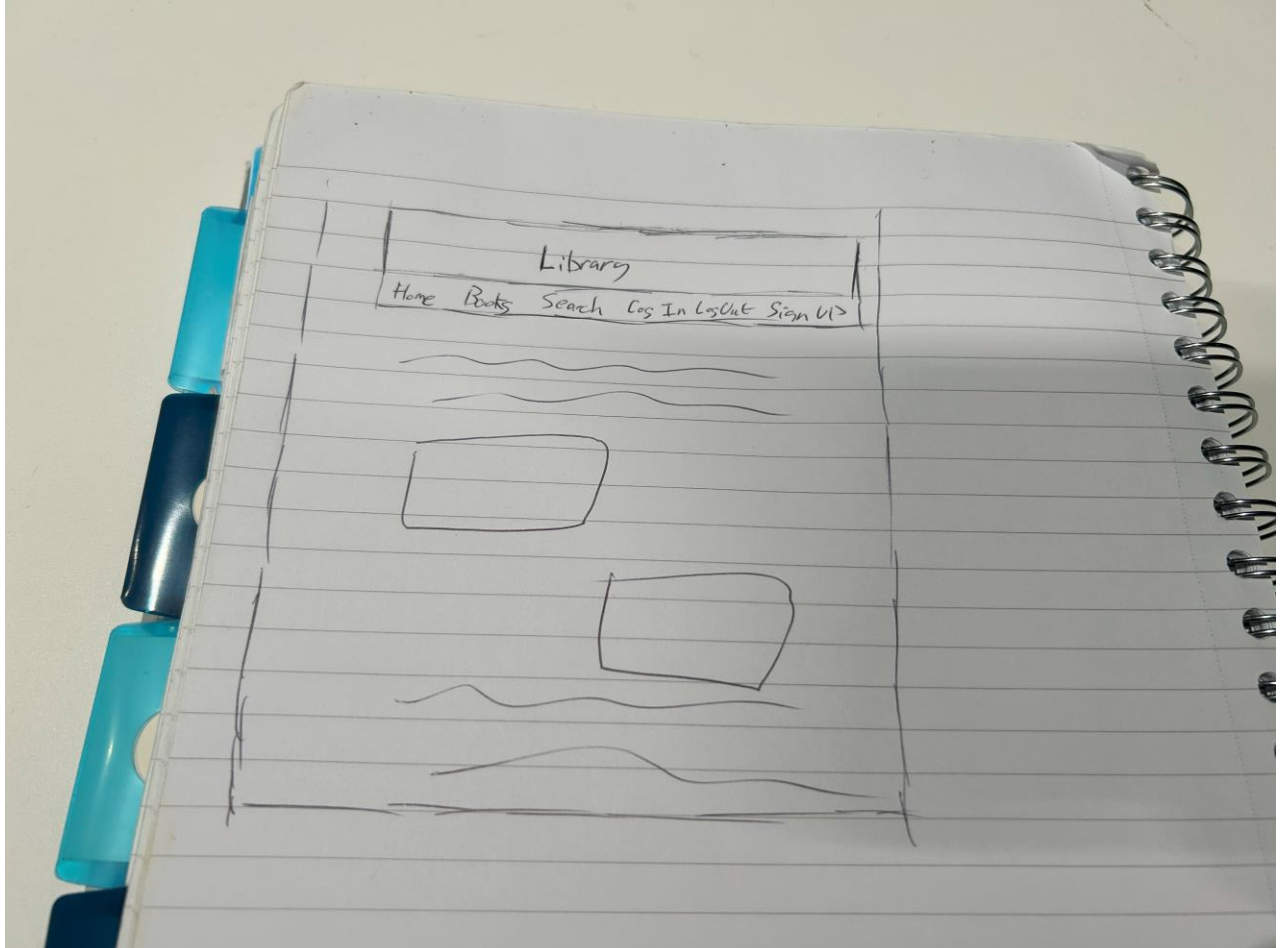
Web Development 2 Project

Fionn Brien

Design Document

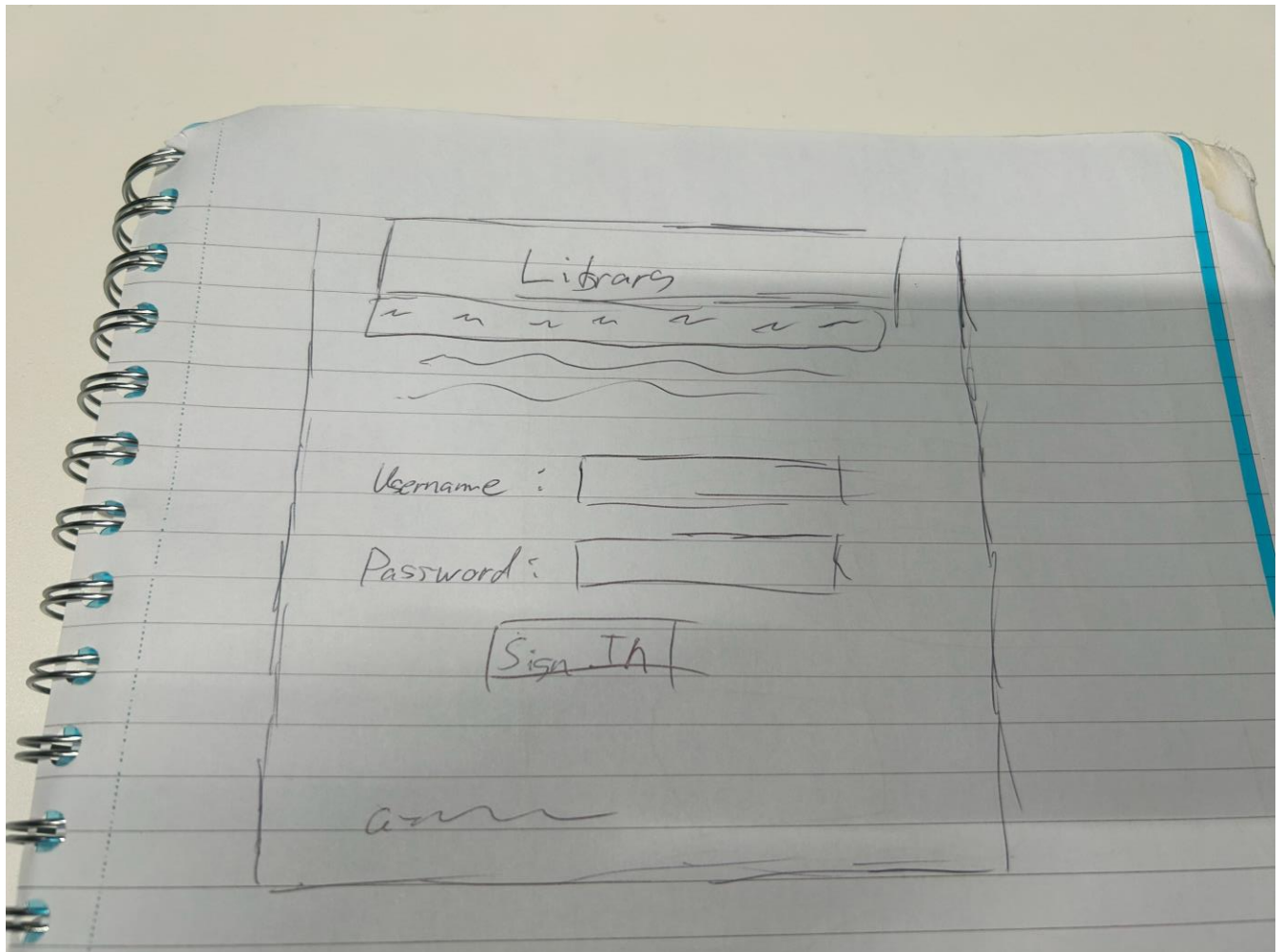
-Here are the designs I came up with for how each page should roughly look on pen and paper.

-Index



-The Home page. Should have links to all the other pages

-Log In



-This page should allow the user to enter a password and username. The page should connect to the database and log the user in if the password and username match.

-Sign Up

Library

Sign Up:

Username:

Password:

Password again:

-This page should add a row entered by the user to the database, but only if it passes error checks so the wrong kind of data isn't entered into the database.

-Search

The sketch shows a library search interface. At the top is a box labeled "Library" with a wavy line below it. Below that is a search input field labeled "Search by Name/Author:". To the right of this is a dropdown menu labeled "Category:" with a list of categories: "Health", "Fiction", and two empty boxes. Below the search field is a table with 6 columns: "ISBN", "Title", "Author", and three empty boxes, followed by a "Res" column. The table contains 5 rows of data. At the bottom of the page are four small boxes containing the numbers 1, 7, 3, and 4.

ISBN	Title	Author				Res
123	Dune	Frank				Res
432	LotR	Tolkien				
999	Harry	John				Res

1 7 3 4

-The user should be able to enter a string, and the page should search the database for all matching rows with partial search. Should be able to do that for author and Book Title. Must have a dropdown menu to search by the categories. The user should be able to reserve available books here too.

-Your Books

Library

~~~~~

Your Books:

| ISBN | Title  | Author | Ect | Cancel |
|------|--------|--------|-----|--------|
| 1    | Book   | John   | ~   | Cancel |
| 2    | Saving | Mary   | ~   | Cancel |
| 3    | Cobble | Sue    | ~   | Cancel |
| 4    | One    | Dave   | ~   | Cancel |
| 5    | Man    | Mike   | ~   | Cancel |

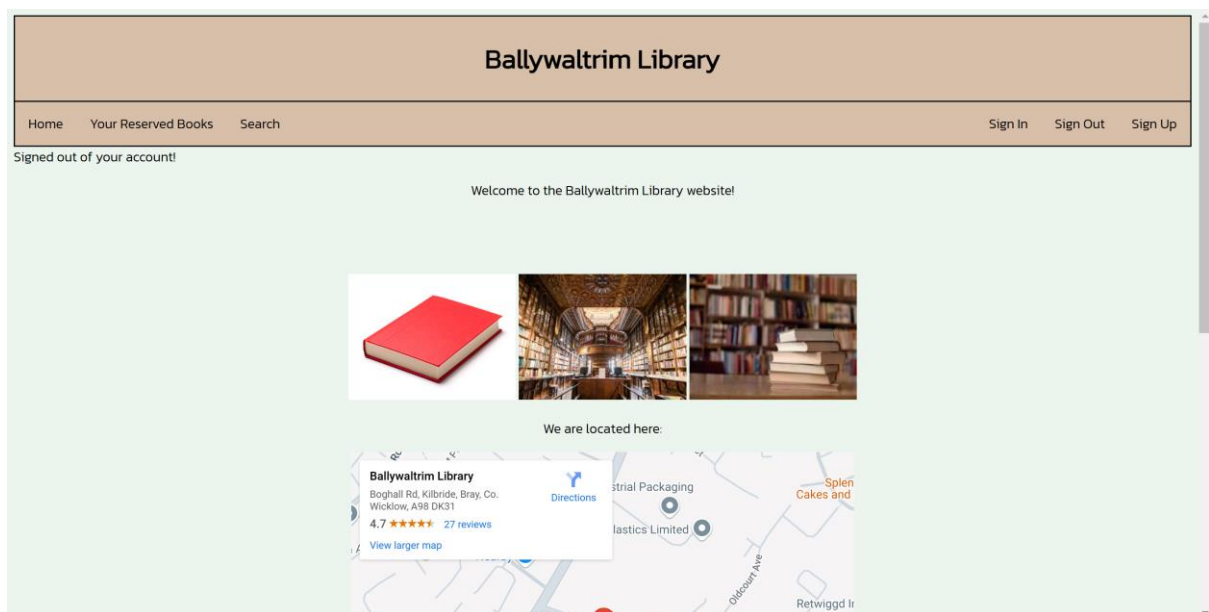
1 2

-This page should display all the reservations that the user has. The user should be able to cancel their reservations here.

# My Pages

## Index.php

-This is the home page for the website. It contains links to all the other pages. The user is returned to the index if they try to go to Your Books or Search without signing in. This page displays messages to the user when they sign in, sign out and sign up.



```
1. <?php
2. session_start();
3. ?>
4.
5. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
6. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
7. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
8. <meta http-equiv="content-type" content="text/html; charset=iso-8859" />
9. <head>
10. <title>Library Index</title>
11. </head>
12. <header>
13. <div>
```



```

14.         <h1>Ballywaltrim Library</h1>
15.     </div>
16.     <link rel="stylesheet" href="css/style.css">
17. </header>
18. <!--Links to other pages. yourBooks and search will redirect to signIn -->
19. <nav class="nav">
20.     <a href = "index.php">Home</a>
21.     <a href = "php/yourBooks.php">Your Reserved Books</a>
22.     <a href = "php/search.php">Search</a>
23.     <div>
24.         <a href = "php/signIn.php">Sign In</a>
25.         <a href = "php/signOut.php">Sign Out</a>
26.         <a href = "php/signUp.php">Sign Up</a>
27.     </div>
28. </nav>
29. <body>
30. <?php
31. //shows message when user logs in
32. if ( isset($_SESSION["loggedIn"]) )
33. {
34.     echo($_SESSION["loggedIn"]."</p>\n");
35.     unset($_SESSION["loggedIn"]);
36. }
37. //shows message when user logs out
38. if ( isset($_SESSION["signedOut"]) )
39. {
40.     echo($_SESSION["signedOut"]."</p>\n");
41.     unset($_SESSION["signedOut"]);
42. }
43. //shows message when user creates a new account
44. if (isset($_SESSION["accountCreated"]))
45. {
46.     echo($_SESSION["accountCreated"]."</p>\n");
47.     unset($_SESSION["accountCreated"]);
48. }
49. ?>
50. <div class = "main">
51.     <p>Welcome to the Ballywaltrim Library website!</p>
52.     <br><br><br>
53.     <tr>
54.         <td>
55.             
56.         </td>
57.         <td>
58.             
59.         </td>
60.         <td>
61.             
62.         </td>
63.     </tr>
64.     <br>
65.     <p>We are located here:</p>
66.     <section>
67.         <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2390.603758464319!2d-
6.126641923611656!3d53.18908748647892!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x4867a8
80d56e87af%3A0xa3ce7bcd4aa2b857!2sBallywaltrim%20Library!5e0!3m2!1sen!2sie!4v1733432102482!5m2!1
sen!2sie" width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy"
referrerpolicy="no-referrer-when-downgrade"></iframe>
68.     </section>
69.     <br><br>
70.     <section>
71.         <iframe width="560" height="315"
src="https://www.youtube.com/embed/4vIQON2fDWM?si=cKPLU5inD85lt3tP" title="YouTube video player"
frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope;
picture-in-picture; web-share" referrerpolicy="strict-origin-when-cross-origin"
allowfullscreen></iframe>
72.     </section>
73. </div>
74.

```



```
75. </body>
76. <footer>
77.     @2024 Created by Fionn Brien
78. </footer>
79. </html>
80.
```

## signIn.php

-This page allows the user to sign in to the website. It posts the user input from a form and compares the username and password entered to those stored in the database. If the username exists and the password is correct, the username is assigned to a Session variable and the user can access the pages they couldn't before. A message is also sent to display when the user is returned to the index. If the user gets the username or password wrong the appropriate error message is displayed.

| Ballywaltrim Library                  |                                                      |
|---------------------------------------|------------------------------------------------------|
| Home                                  |                                                      |
| Sign in to your Library Account       |                                                      |
| Username:                             | <input type="text" value="Enter your username"/>     |
| Password:                             | <input type="password" value="Enter your password"/> |
| <input type="submit" value="Submit"/> |                                                      |
| @2024 Created by Fionn Brien          |                                                      |

```
1. <?php
2.
3. // Start a session
4. session_start();
5.
6. // Include the database connection
7. require_once 'database.php';
8.
9. // Check if the form was submitted
10. if ( isset($_POST["username"]) && isset($_POST["password"]) )
11. {
12.     // Retrieve posted form data
13.     $uname = $conn->real_escape_string($_POST['username']);
14.     $pass = $conn->real_escape_string($_POST['password']);
```

```

15.
16.     $sql = "SELECT UserName, Password FROM users"; //gets username and password from the
table
17.     echo "<pre>\n$sql</pre>\n";
18.     $result = $conn->query($sql);
19.     //while loop will loop through the database and will compare the posted username and
password to all in the database
20.     while(True)
21.     {
22.         $row = $result->fetch_assoc();
23.         if ($row['UserName'] == $uname)
24.         {
25.             break; //loop will break when it gets to the matching data
26.         }
27.         elseif ($row['UserName'] == null)
28.         {
29.             break; // else if the loop breaks when it reaches null data at the end of the
database
30.         }
31.     }
32.     if ( $_POST['username'] == $row['UserName'] ) //if username matches
33.     {
34.         if ( $_POST['password'] == $row['Password'] ) //if password matches
35.         {
36.             //user is logged in with SESSION and a message is added to $_SESSION[loggedIn]
37.             $_SESSION["Username"] = $_POST["username"];
38.             $_SESSION["loggedIn"] = "Welcome, ". $_POST["username"] . "!";
39.             header( 'Location: ../index.php' ); //user is returned to index page
40.             return;
41.         }
42.         else //password does not match
43.         {
44.             $_SESSION["error"] = "Incorrect password.";
45.             header( 'Location: signIn.php' );
46.             return;
47.         }
48.     }
49.     else //username doesn't match
50.     {
51.         $_SESSION["error"] = "User doesn't exist.";
52.         header( 'Location: signIn.php' );
53.         return;
54.     }
55. }
56.
57. else if ( count($_POST) > 0 ) //user enters no data
58. {
59.     $_SESSION["error"] = "Missing Required Information";
60.     header( 'Location: signIn.php' );
61.     return;
62. }
63.

64. $conn->close();
65. ?>
66.
67. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
68. "http://www.w3.org/TR/xhtml1-transitional.dtd">
69. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
70. <meta http-equiv="content-type" content="text/html; charset=iso-8859" />
71. <head>
72.     <title>Library Sign In</title>
73. </head>
74. <header>
75.     <div>
76.         <h1>Ballywaltrim Library</h1>
77.     </div>
78.     <link rel="stylesheet" href="../css/style.css">
79. </header>

```

```

80. <nav class="nav">
81.     <a href = "../index.php">Home</a>
82. </nav>
83. <body>
84.
85. <p>Sign in to your Library Account</p>
86.
87. <?php
88. //displays error message for user
89. if ( isset($_SESSION["error"]) )
90. {
91.     echo('<p style="color:red">Error: ' .$_SESSION["error"]."</p>\n");
92.     unset($_SESSION["error"]);
93. }
94. ?>
95.
96. <!--This form posts username and password-->
97. <div class = "signIn">
98.     <form action = "" method = "post">
99.         <p>Username: <input type = "text" name = "username" value = "" placeholder="Enter
your username" required></p>
100.         <p>Password: <input type = "text" name = "password" value = "" placeholder="Enter
your password" required></p>
101.         <p><input type = "submit" value = "Submit"></p>
102.     </form>
103. </div>
104.
105. </body>
106. <footer>
107.     @2024 Created by Fionn Brien
108. </footer>
109. </html>
110.

```

## signOut.php

-This page ends the session and restarts it. This deletes the Session variables, so the user is signed out. When the session is restarted, a message is assigned to a Session variable and is displayed when the user is returned to the index page, so they know they signed out successfully.

-No display for this page. It runs and returns to the index page.

```

1. <?php
2. session_start();
3. session_destroy();
4. session_start();
5. $_SESSION["signedOut"] = "Signed out of your account!";
6. header("Location: ../index.php");
7. ?>
8.

```

## signUp.php

-This page allows the user to put in their information in a form and then it will add this as a row in the users table. However, the data will only be added if all the fields are filled, to prevent null values, and if error checking is passed e.g. username is checked to if it is already in the database, password must be entered correctly twice and must be 6 characters long. Appropriate error messages are displayed to help the user. If these tests are passed, the users table gains the new row, and the user is returned to the index page. A message is displayed on the index page to tell the user their account was created successfully.

Ballywaltrim Library

Home

Create your Library Account

Username:

Password:

Confirm Password:

First Name:

Surname:

Address Line 1:

Address Line 2:

City:

Telephone:

Mobile:

Submit

@2024 Created by Fionn Brien

```
1. <?php
2. session_start();
3. require_once "database.php";
4.
5. //only runs if all fields are submitted
6. if (isset($_POST['username']) && isset($_POST['password1']) && isset($_POST['password2'])
&& isset($_POST['firstName']) && isset($_POST['surname']) && isset($_POST['address1']) &&
isset($_POST['address2']) && isset($_POST['city']) && isset($_POST['telephone']) &&
isset($_POST['mobile']))
7. {
8.     //all the posted data is saved to variables
9.     $user = $conn -> real_escape_string(string: $_POST['username']);
10.    $pas = $conn -> real_escape_string(string: $_POST['password1']);
11.    $pas2 = $conn -> real_escape_string(string: $_POST['password2']);
12.    $fname = $conn -> real_escape_string(string: $_POST['firstName']);
13.    $sname = $conn -> real_escape_string(string: $_POST['surname']);
14.    $add1 = $conn -> real_escape_string(string: $_POST['address1']);
15.    $add2 = $conn -> real_escape_string(string: $_POST['address2']);
16.    $city = $conn -> real_escape_string(string: $_POST['city']);
17.    $tel = $conn -> real_escape_string(string: $_POST['telephone']);
18.    $mob = $conn -> real_escape_string(string: $_POST['mobile']);
19.
20.    //error checking
21.    $cor = 0;
22.    $_SESSION["err3"] = "";
23.
24.    //username check
25.    $sql="SELECT * FROM users WHERE UserName='$user'";
26.
27.    //checks if the username appears already and wont allow a new row to be added if the
name is taken
```

```

28.     if($result=mysqli_query($conn,$sql))
29.     {
30.         if(mysqli_num_rows($result)>0)
31.         {
32.             $_SESSION["err3"] .= "This username is taken.\n";
33.         }
34.
35.         else
36.         {
37.             $cor += 1;
38.         }
39.     }
40.
41.     //password check
42.     if ($pas == $pas2 && strlen($pas) == 6)
43.     {
44.         $cor += 1;
45.     }
46.
47.     else
48.     {
49.         $_SESSION["err3"] .= "The passwords much match and be 6 characters long.\n";
50.     }
51.
52.     //telephone check
53.     if (strlen($tel) == 7 && is_numeric($tel))
54.     {
55.         $cor += 1;
56.     }
57.
58.     else
59.     {
60.         $_SESSION["err3"] .= "Telephone must be 7 numbers.\n";
61.     }
62.
63.
64.     //phone check
65.     if (strlen($mob) == 10 && is_numeric($mob))
66.     {
67.         $cor += 1;
68.     }
69.     else
70.     {
71.         $_SESSION["err3"] .= "Mobile must be 10 numbers.\n";
72.     }
73.
74.
75.     //if all the tests are passed, new row is added to the database
76.     if ($cor == 4)
77.     {
78.         $sql = "INSERT INTO users (Username, Password, FirstName, Surname, AddressLine1,
AddressLine2, City, Telephone, Mobile) VALUES('$user', '$pas', '$fname', '$sname', '$add1',
'$add2', '$city', '$tel', '$mob')";
79.         echo "<pre>\n$sql\n</pre>\n";
80.         $conn->query(query: $sql);
81.         $_SESSION["accountCreated"] = "Account creation successful! Please sign in!";
82.         header( 'Location: ../index.php' );
83.         return;
84.     }
85.
86.
87. }
88.
89. ?>
90.
91. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
92. "http://www.w3.org/TR/xhtml1-transitional.dtd">
93. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
94. <meta http-equiv="content-type" content="text/html; charset=iso-8859" />
95. <head>

```

```

96.     <title>Library Sign Up</title>
97. </head>
98. <header>
99.     <div>
100.         <h1>Ballywaltrim Library</h1>
101.     </div>
102.     <link rel="stylesheet" href="../css/style.css">
103. </header>
104. <nav class="nav">
105.     <a href = "../index.php">Home</a>
106. </nav>
107. <body>
108.
109. <?php
110. //displays errors for user
111. if ( isset($_SESSION["err3"]) )
112. {
113.     echo('<p style="color:red">'.$_SESSION["err3"]."</p>\n");
114.     unset($_SESSION["err3"]);
115. }
116. ?>
117.
118. <!--form posts the user data-->
119. <div class = "signUp">
120.     <p>Create your Library Account</p>
121.     <form method = "post">
122.         <p>Username:
123.         <input type = "text" name = "username"></p>
124.         <p>Password:
125.         <input type = "text" name = "password1"></p>
126.         <p>Confirm Password:
127.         <input type = "text" name = "password2"></p>
128.         <p>First Name:
129.         <input type = "text" name = "firstName"></p>
130.         <p>Surname:
131.         <input type = "text" name = "surname"></p>
132.         <p>Address Line 1:
133.         <input type = "text" name = "address1"></p>
134.         <p>Address Line 2:
135.         <input type = "text" name = "address2"></p>
136.         <p>City:
137.         <input type = "text" name = "city"></p>
138.         <p>Telephone:
139.         <input type = "number" name = "telephone"></p>
140.         <p>Mobile:
141.         <input type = "number" name = "mobile"></p>
142.         <br><br>
143.         <p><input type = "submit" value = "Submit"></p>
144.     </form>
145. </div>
146. </body>
147. <footer>
148.     @2024 Created by Fionn Brien
149. </footer>
150. </html>
151.

```

## yourBooks.php

-This page shows all the books that the user has reserved and allows them to cancel the reservation. This page can only be accessed once the user is signed in. It will redirect to the

index page if the Session['Username'] isn't set. Once the user is signed in, it queries the database to get the rows that are reserved by the user. It does this with a subquery that gets the ISBNs from reserved table where the username is the same as the users. Then it gets the rows from the books table with same ISBNs. It displays the rows in a table. While displaying the rows, the category table is queried so that the names of the genres are displayed instead of just the ID numbers. I didn't finish the pagination, so it shows all possible rows, not just 5 at a time. At the end of each row, there is a button to cancel the reservation. If pressed, the user is sent to resCancel.php and GET is used to send the ISBN to resCancel.php. Once the user returns, a message is displayed to show the cancellation was successful.

Ballywaltrim Library

Your Reserved Books

[Home](#) [Your Reserved Books](#) [Search](#)

@2024 Created by Fionn Brien

| ISBN        | Book Title               | Author       | Edition | Year | Category | Reserved                           |
|-------------|--------------------------|--------------|---------|------|----------|------------------------------------|
| 9823-98345  | How to cook Italian food | Jamie Oliver | 2       | 2005 | Cookery  | <a href="#">Cancel Reservation</a> |
| 23u8-923849 | A guide to nutrition     | John Thorpe  | 2       | 1997 | Health   | <a href="#">Cancel Reservation</a> |

```
1. <?php
2. session_start();
3.
4. if ( isset( $_SESSION['Username'] ) ) {
5.     // Grab user data from the database using the user_id
6.     // Let them access the "logged in only" pages
7. } else {
8.     // Redirect them to the login page
9.     header("Location: signIn.php");
10. }
11. require_once "database.php";
12.
13. $username = $conn->real_escape_string($_SESSION['Username']);
14.
15. $sql = "SELECT * FROM books WHERE ISBN IN (SELECT ISBN FROM reserved WHERE Username =
'$username')"; //gets all the rows in books that have been reserved by the current user
16. $result = $conn->query($sql);
17. if ($result->num_rows > 0)
18. {
19.
20.     echo "<table border='2'>";
21.     echo "<tr><td>";
22.     echo "ISBN";
23.     echo "</td><td>";
24.     echo "Book Title";
25.     echo "</td><td>";
26.     echo "Author";
27.     echo "</td><td>";
28.     echo "Edition";
29.     echo "</td><td>";
30.     echo "Year";
```



```

31.     echo "</td><td>";
32.     echo "Category";
33.     echo "</td><td>";
34.     echo "Reserved";
35.     echo "</td></tr>\n";
36.
37.     //prints all the rows in reserved associated with the user
38.     while($row = $result->fetch_assoc())
39.     {
40.         echo "<tr><td>";
41.         echo(htmlentities($row["ISBN"]));
42.         echo "</td><td>";
43.         echo(htmlentities($row["BookTitle"]));
44.         echo "</td><td>";
45.         echo(htmlentities($row["Author"]));
46.         echo "</td><td>";
47.         echo(htmlentities($row["Edition"]));
48.         echo "</td><td>";
49.         echo(htmlentities($row["Year"]));
50.         echo "</td><td>";
51.         //code below selects the Name of the category from category table using Category
numberfrom the books table
52.         $sql2 = "SELECT CategoryDesc FROM category WHERE CategoryID =" . $row["Category"].
"";
53.         $result2 = $conn->query($sql2);
54.         $row2 = $result2->fetch_assoc();
55.         echo(htmlentities($row2["CategoryDesc"]));
56.         echo "</td>";
57.         //puts a link in the table that allows the user to cancel a resevation
58.         //sends the ISBN to resCancel.php with GET
59.         echo "<td>
60.             <form method = 'GET' action = 'resCancel.php'>
61.                 <input type = 'hidden' name = 'isbn' value = '" .
htmlspecialchars($row["ISBN"])."'">
62.                 <input type = 'submit' name = 'reserve' value = 'Cancel Reservation'>
63.             </form>
64.         </td>";
65.         echo "</td></tr>\n";
66.     }
67. }
68.
69. else
70. {
71. echo "0 results";
72. }
73.
74. $conn->close();
75. ?>
76. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
77. "http://www.w3.org/TR/xhtml1-transitional.dtd">
78. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang = "en" lang = "en">
79. <meta http-equiv="content-type" content = "text/html; charset = iso-8859" />
80. <head>
81.     <title>Your Reserved Boooks</title>
82. </head>
83. <header>
84.     <div>
85.         <h1>Ballywaltrim Library<h1>
86.         <h2>Your Reserved Books</h2>
87.     </div>
88.     <link rel="stylesheet" href = "../css/style.css">
89. </header>
90. <nav class="nav">
91.     <a href = "../index.php">Home</a>
92.     <a href = "yourBooks.php">Your Reserved Books</a>
93.     <a href = "search.php">Search</a>
94. </nav>
95.
96. <?php
97. //displays message when user cancels a reservation

```

```

98. if ( isset($_SESSION["resCancelled"]) )
99. {
100.     echo($_SESSION["resCancelled"]."</p>\n");
101.     unset($_SESSION["resCancelled"]);
102. }
103. ?>
104. </body>
105. <footer>
106.     @2024 Created by Fionn Brien
107. </footer>
108. </html>
109.

```

## resCancel.php

-The user is sent to this page once they click on the cancel reservation button on a row in yourBooks.php. This page removes the associated row from reserved table using the ISBN of the row that the user clicked on that was sent using GET. After removing the row from the reserved table, it also updates the Reserved column in the books table from N to Y so it can be reserved now in Search.php. The user is returned to yourBooks.php with a message that displays so they know the cancellation was a success.

-Nothing displays as the user is sent back to yourBooks instantly with header.

```

1. <?php
2. //removes reservations on books
3. session_start();
4.
5. //if someone entered the link without being signed in they would be redirected to sign in
page
6. if ( isset( $_SESSION['Username'] ) ) {
7.
8. } else {
9.     header("Location: signIn.php");
10. }
11.
12. require_once "database.php";
13. $username = $conn->real_escape_string($_SESSION['Username']); //gets the username from the
session
14. $date = date("Y-m-d"); //uses date() to get current date
15.
16. if (isset($_GET['isbn']))
17. {
18.     $cancel = $conn-> real_escape_string($_GET['isbn']); //uses GET to bring the ISBN from
the row clicked
19.     $sql = $sql = "DELETE FROM reserved WHERE ISBN = '$cancel'"; //deletes the row that the
user chose from reserved
20.     echo "<pre>\n$sql</pre>\n";
21.     $conn->query($sql);
22.     $sql = "UPDATE books SET Reserved = 'N' WHERE ISBN = '$cancel'"; //changes status of the
book in books table
23.     echo "<pre>\n$sql</pre>\n";
24.     $conn->query($sql);
25.     $_SESSION["resCancelled"] = "You cancelled your reservation!"; //message for when user
is brought back to yourBooks to show the success
26.     header( 'Location: yourBooks.php' ) ; //brings user back to yourBooks
27.     return;
28. }
29.
30. $conn->close();
31. ?>
32.

```

## search.php

-This page allows the user to search the books table of the database by book title or author and by category with a dropdown menu. It can only be accessed when the user is logged in to stop unauthorised people from editing the database. It allows for partial search using WHERE e.g. author LIKE “\_\_\_\_%”. A radial option must be picked for the text search, or the page won’t return a table and will give an error message. If a radial is chosen but no text is entered, the entire database will display. A partial search will work because of the Like operator. The rows don’t display 5 at a time because I didn’t get to finish pagination. If the user uses the dropdown menu, all books of the chosen category will be displayed. A subquery gets the ID from category table where the Description is equal to the dropdown option picked. The main query then gets all the rows from books with that categoryID. These rows are then displayed. If there aren’t any matching results, the page will display “0 Results” to the user. If Reserved in each row is ‘Y’, then “Reserved” is printed. If Reserved is “Y”, a button that links to reserve.php is printed. This conditional statement prevents the user from attempting to reserve a book that has already been reserved. Once the reserve button is pressed the user will be sent to reserve.php. When they return, a message will be displayed to say that the reservation was a success.

Ballywaltrim Library

Search

[Home](#) [Your Reserved Books](#) [Search](#)

Filter by:

☐ Title ☐ Author

Search for:

Submit

Genre: Health

Submit

@2024 Created by Fionn Brien

| ISBN       | Book Title               | Author          | Edition | Year | Category | Reserve? |
|------------|--------------------------|-----------------|---------|------|----------|----------|
| 2983-3494  | Cooking for children     | Anabelle Sharpe | 1       | 2003 | Cookery  | Reserve  |
| 9823-98345 | How to cook Italian food | Jamie Oliver    | 2       | 2005 | Cookery  | Reserved |

1. `<?php`
- 2.
3. `session_start();`

```

4.
5. if ( isset( $_SESSION['Username'] ) ) {
6.     // Grab user data from the database using the user_id
7.     // Let them access the "logged in only" pages
8. } else {
9.     // Redirect them to the login page
10.    header("Location: signIn.php");
11. }
12.
13. require_once "database.php";
14.
15. ?>
16. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
17. "http://www.w3.org/TR/xhtml1-transitional.dtd">
18. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
19. <meta http-equiv="content-type" content="text/html; charset=iso-8859" />
20. <head>
21.     <title>Search</title>
22. </head>
23. <header>
24.     <div>
25.         <h1>Ballywaltrim Library</h1>
26.         <h2>Search</h2>
27.     </div>
28.     <link rel="stylesheet" href="../css/style.css">
29. </header>
30. <nav class="nav">
31.     <a href = "../index.php">Home</a>
32.     <a href = "yourBooks.php">Your Reserved Books</a>
33.     <a href = "search.php">Search</a>
34. </nav>
35. <body>
36.
37.     <?php
38.     //displays the message that reserve.php sends when successful
39.     if ( isset($_SESSION["reserved"]) )
40.     {
41.         echo($_SESSION["reserved"]. "</p>\n");
42.         unset($_SESSION["reserved"]);
43.     }
44.     //displays the errors
45.     if ( isset($_SESSION["err2"]) )
46.     {
47.         echo('<p style="color:red">Error: ' . $_SESSION["err2"] . "</p>\n");
48.         unset($_SESSION["err2"]);
49.     }
50.     ?>
51.
52.     <!--Forms for search that POST the user input -->
53.     <div class = "search">
54.         <form action = "" method = "post">
55.             <p>Filter by:</p>
56.             <input type="radio" id="title" name="filter" value="title">
57.             <label for="title">Title</label>
58.             <input type="radio" id="author" name="filter" value="author">
59.             <label for="author">Author</label>
60.             <p>Search for: <input type = "text" name = "search" value = ""></p>
61.             <p><input type = "submit" value = "Submit"></p>
62.         </form>
63.         <form action = "" method = "post">
64.             <label for="category">Genre: </label>
65.             <select name="category" id="category">
66.                 <option value="Health">Health</option>
67.                 <option value="Business">Business</option>
68.                 <option value="Biography">Biography</option>
69.                 <option value="Technology">Technology</option>
70.                 <option value="Travel">Travel</option>
71.                 <option value="Self-Help">Self-Help</option>
72.                 <option value="Cookery">Cookery</option>
73.                 <option value="Fiction">Fiction</option>

```



```

139.                 echo "</td><td>";
140.                 $sql2 = "SELECT CategoryDesc FROM category WHERE CategoryID ='" .
$row["Category"]. "'";
141.                 $result2 = $conn->query($sql2);
142.                 $row2 = $result2->fetch_assoc();
143.                 echo(htmlentities($row2["CategoryDesc"]));
144.                 echo "</td>";
145.                 if ($row["Reserved"] == 'N')
146.                 {
147.                     echo "<td>
148.                         <form method = 'GET' action = 'reserve.php'>
149.                             <input type = 'hidden' name = 'isbn' value = '" .
htmlspecialchars($row['ISBN']). "'>
150.                                 <input type = 'submit' name = 'reserve' value =
'Reserve'>
151.                                     </form>
152.                                 </td>";
153.                 }
154.                 else
155.                 {
156.                     echo"<td>Reserved</td>";
157.                 }
158.                 echo"</tr>\n";
159.             }
160.         }
161.
162.         else
163.         {
164.             //if there are no results this is displayed
165.             $_SESSION["err2"] = "0 Results";
166.             header( 'Location: search.php' );
167.             return;
168.         }
169.
170.     }
171.
172.     else
173.     {
174.         //user can't search unless they pick a filter
175.         $_SESSION["err2"] = "You must select a filter!";
176.         header( 'Location: search.php' );
177.         return;
178.     }
179.
180. }
181.
182.
183. //displays the books by category
184. else if ( isset($_POST["category"]))
185. {
186.     $cat = $conn->real_escape_string($_POST['category']);
187.     $sql = "SELECT * FROM books WHERE Category IN (SELECT CategoryID FROM category
WHERE CategoryDesc = '$cat')"; //uses a subquery to get the rows in books where the number is
equal to the categoryid in category
188.     $result = $conn->query($sql);
189.     //displays the rows in a table
190.     if ($result->num_rows > 0)
191.     {
192.
193.         echo "<table border='2'>";
194.         echo "<tr><td>";
195.         echo "ISBN";
196.         echo "</td><td>";
197.         echo "Book Title";
198.         echo "</td><td>";
199.         echo "Author";
200.         echo "</td><td>";
201.         echo "Edition";
202.         echo "</td><td>";
203.         echo "Year";

```

```

204.         echo "</td><td>";
205.         echo "Category";
206.         echo "</td><td>";
207.         echo "Reserve?";
208.         echo "</td></tr>\n";
209.
210.         //prints all the rows in reserved associated with the user
211.         while($row = $result->fetch_assoc())
212.         {
213.             echo "<tr><td>";
214.             echo(htmlentities($row["ISBN"]));
215.             echo "</td><td>";
216.             echo(htmlentities($row["BookTitle"]));
217.             echo "</td><td>";
218.             echo(htmlentities($row["Author"]));
219.             echo "</td><td>";
220.             echo(htmlentities($row["Edition"]));
221.             echo "</td><td>";
222.             echo(htmlentities($row["Year"]));
223.             echo "</td><td>";
224.             $sql2 = "SELECT CategoryDesc FROM category WHERE CategoryID =' " .
$row["Category"]. "'";
225.             $result2 = $conn->query($sql2);
226.             $row2 = $result2->fetch_assoc();
227.             echo(htmlentities($row2["CategoryDesc"]));
228.             if ($row["Reserved"] == 'N')
229.             {
230.                 //puts a link in the table that allows the user to reserve
books that are available
231.                 //sends the ISBN to reserve.php with GET
232.                 echo "<td>
233.                     <form method = 'GET' action = 'reserve.php'>
234.                         <input type = 'hidden' name = 'isbn' value = '" .
htmlspecialchars($row["ISBN"])." '>
235.                         <input type = 'submit' name = 'reserve' value =
'Reserve'>
236.                     </form>
237.                     </td>";
238.             }
239.             else
240.                 //books that are reserved don't have a link
241.                 echo "<td>Reserved</td>";
242.             echo "</tr>\n";
243.         }
244.     }
245.
246.     else
247.     {
248.         $_SESSION["err2"] = "0 Results";
249.         header( 'Location: search.php' );
250.         return;
251.     }
252. }
253. $conn->close();
254. ?>
255.
256. </body>
257. <footer>
258.     @2024 Created by Fionn Brien
259. </footer>
260. </html>
261.

```



## reserve.php

-This page reserves a book for a user. GET is used in search.php to send the ISBN to this page. The user's username is taken from the SESSION variable used for being signed in. The current date is retrieved from the function date("Y-m-d"). These three variables are then added as a new row in the reserved table and the Reserved in the books table on the ISBN's row is updated to 'Y'. This book will now appear in yourBooks.php for the user and nobody will be able to reserve it anymore in search.php. The user is sent back to search.php and a message is sent to display there.

```
1. <?php
2. session_start();
3.
4. //if someone entered the link without being signed in they would be redirected to sign in
page
5. if ( isset( $_SESSION['Username'] ) ) {
6.
7. } else {
8.     header("Location: signIn.php");
9. }
10.
11. require_once "database.php";
12.
13. $username = $conn->real_escape_string($_SESSION['Username']); //gets the username from the
session
14. $date = date("Y-m-d"); //uses date() to get current date
15.
16. if (isset($_GET['isbn']))
17. {
18.
19.     $res = $conn->real_escape_string($_GET['isbn']); //uses GET to bring the ISBN from the
row clicked
20.     $sql = "INSERT INTO reserved (ISBN, UserName, ReservedDate)
VALUES('$res','$username','$date')"; //adds a new row to reserved
21.     echo "<pre>\n$sql</pre>\n";
22.     $conn->query($sql);
23.     $sql = "UPDATE books SET Reserved = 'Y' WHERE ISBN = '$res'"; //updates status of book
in books
24.     echo "<pre>\n$sql</pre>\n";
25.     $conn->query($sql);
26.     $_SESSION["reserved"] = "This book is reserved for you now!"; //message for when user is
brought back to search
27.     header( 'Location: search.php' );
28.     return;
29. }
30.
31. $conn->close();
32. ?>
33.
```

## database.php

-This page creates a connection to the project database. It checks that the database is connected and will print an error message if it fails. All of the other pages include\_once this page so they can connect to the database and perform CRUD on it.

```

1. <?php
2. //all the php files use this file to connect to the database
3.
4. $servername = "localhost";
5. $username = "root";
6. $password = "";
7. $dbname = "project";
8.
9. // Create connection
10. $conn = new mysqli($servername, $username, $password, $dbname);
11. // Check connection
12. if ($conn->connect_error) {
13.     die("Connection failed: " . $conn->connect_error);
14. }
15. ?>
16.

```

## style.css

-All of my pages connect to this css file to give them a cleaner appearance and to move certain elements.

```

1. @import url('https://fonts.googleapis.com/css2?family=Kanit:wght@300&display=swap');
2.
3. header {
4.     background-position: center;
5.     background-color: #D8BFAA;
6.     background-size: cover;
7.     color: black;
8.     text-align: center;
9.     padding: 5px 0;
10.    border-style: solid solid hidden solid;
11.    font-family: 'Kanit', sans-serif;
12. }
13.
14. body {
15.     background-color: #EBF5EE;
16.     font-family: 'Kanit', sans-serif;
17. }
18.
19. .nav a {
20.     float: left;
21.     display: block;
22.     color: black;
23.     text-align: center;
24.     padding: 14px 16px;
25.     text-decoration: none;
26.     font-family: 'Kanit', sans-serif
27. }
28.
29. .nav div {
30.     float: right;
31. }
32.
33. .main {
34.     text-align: center;
35. }
36. }
37.
38. .nav {
39.     background-color: #D8BFAA;
40.     border-style: solid;
41.     overflow: hidden;
42. }
43.

```

```
44. .signIn {
45.     text-align: center;
46. }
47.
48. .signUp {
49.     text-align: center;
50. }
51.
52. .search {
53.     text-align: center;
54. }
55. footer {
56.     left: 0;
57.     bottom: 0;
58.     height: auto;
59.     width: auto;
60.     text-align: center;
61. }
62.
```