

HTML / PHP Pages

Login Page (loginpage.php):



The screenshot shows a login page with a light gray background. A white rectangular box with a black border contains the login form. On the left side of the box, the text 'Please login to reserve a book!' is displayed in bold, followed by 'Please login or register to start taking out a book!' in a smaller font. On the right side, the title 'Log in' is centered at the top. Below it are two input fields: 'Username:' and 'Password:'. A 'Log In' button is positioned below the password field. At the bottom of the right section, there is a link that says 'Not a member of our site yet? [Register now](#)'.

Description: Upon opening the Library Website, the user is greeted with a simple login box where they are asked to enter their Username and Password. If the user does not have an account, they are prompted to click on the “Register Now” hyperlink which is connected to the makeaccountpage.php file where the user can now make an account and register. When the user logs in, they are brought to the homepageindex.php file which is the main home page of the library website being used. This login page was styled by implementing a CSS file into the loginpage.php file. The CSS file (pages.css in folder CSS) was then coded with the use of <div class=“wrapper2”> to give the basic box with border and text.

loginpage.php code:

```
<?php include 'login.php'; ?>

<!doctype html>
<html>
  <head>
    <title> Welcome!</title>
    <link rel="stylesheet" href="CSS/pages.css">
  </head>

  <body>
    <div class="wrapper2">
      <div class="column">
        <h1>Please login to reserve a book!</h1>
        <p>Please login or register to start taking out a
book!</p>
      </div>
```

```

        <?php include 'formforlogin.php'; ?>
    </div>

</body>
</html>

```

login.php is connected to loginpage.php and this allows for the background running of the form in order to input the Username and Password credentials. Within this php file, you can see that it connects to the myphpadmin database needed to store and pull the credentials entered when registering. The credentials are pulled by connecting to the database within the connectdb.php file which is then implemented in the login.php file as seen below in line 12.

login.php code:

```

<!DOCTYPE html>

<html>
    <head>
        <title>Welcome!</title>
    </head>

    <body>

        <?php
        include_once 'sessionstart.php';
        require_once('connectdb.php');

        if (isset($_POST['submitLogin'])){
            $username=$_POST["username"];
            $password=$_POST["password"];

            if ($stmt = $con -> prepare("SELECT ID_user, username, password FROM users
WHERE username = ?")) {
                $stmt -> bind_param('s', $username);
                $stmt -> execute();
                $stmt -> bind_result($userid, $user, $pass);
                $stmt -> fetch();}

            if ($user == $username and password_verify($password, $pass)){
                $_SESSION['user'] = $user;
                $_SESSION['userid'] = $userid;
                header('Location: homepageindex.php');
            }
            else {
                global $error;
                $error = "Wrong username or password";
            }
            $stmt->free_result();

```

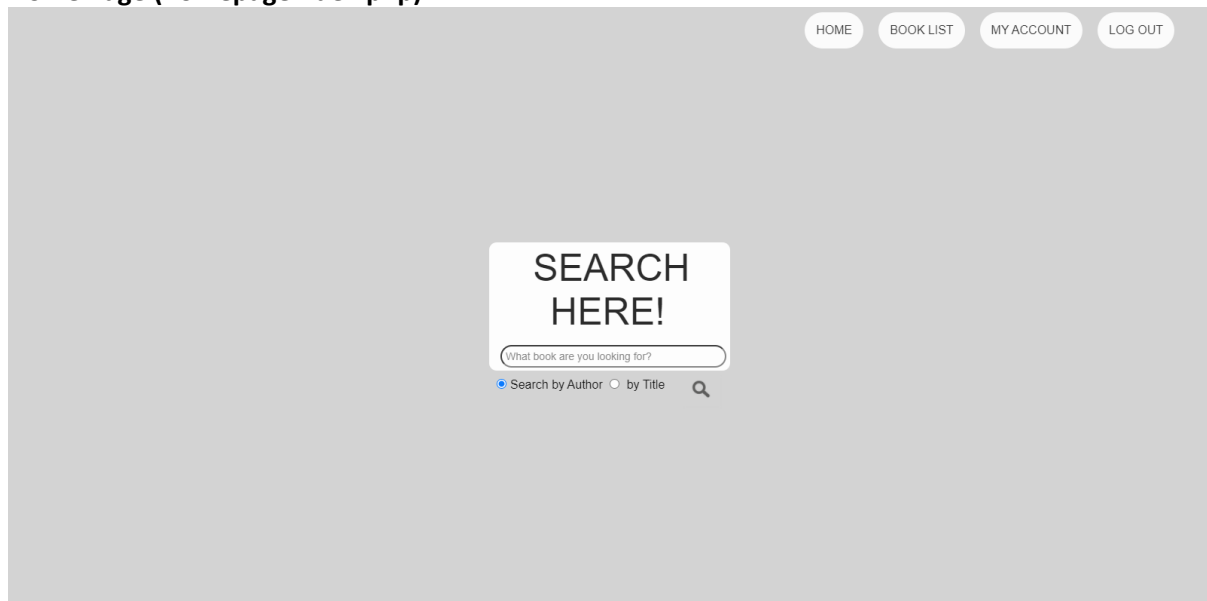
```

$stmt->close();
}
$con->close();
?>

</body>
</html>

```

Home Page (homepageindex.php):



Description: Once the user finally inputs their credentials correctly and logs in, they see this screen as shown above. This is the index/home page of the library website. On this screen, the user has a few options to choose from at the top in the navigation bar. This nav bar is made with the header.php file which is then implemented in every other php file within the website therefore making the nav bar always accessible to the user for quick access to everything on the website. The nav bar consists of a “Home”, “Book list”, “My account” and a “logout” button. Each button takes you to the new respective PHP page. On the home page is a search option (searchfunction.php) as well which allows the user to search for any book by either Author or title of the book. A search.png is used for the little magnifying glass image implemented with CSS.

homepageindex.php code:

```

<?php
include_once 'sessionstart.php';
if ($_SESSION['user'] == NULL) {
    header('Location: loginpage.php');
}
?>

<!doctype html>
<html>
<head>
<title>Homepage</title>
<link rel="stylesheet" href="CSS/pages.css">
<script>
window.onload = function() {
document.getElementById("searchInput").focus();

```

```

};
</script>
</head>
<body>
<?php include 'header.php'; ?>
<div class="wrapper">
    <div class="logo">
        <a class="navlink" href="homepageindex.php" title="Your Web
Library">SEARCH HERE!</a>
    </div>
    <form class="search" action="searchfunction.php" method="post">
        <div>
            <input id="searchInput" type="search" name="q" value=""
placeholder="What book are you looking for?">
            <button id="searchButton" type="submit" style="border: 0;
background: transparent">
                
            </button>
        </div>
        <input type="radio" name="search_type" value="author_last_name"
checked>Search by Author
        <input type="radio" name="search_type" value="title"> by Title
    </form>
</div>

</body>
</html>

```

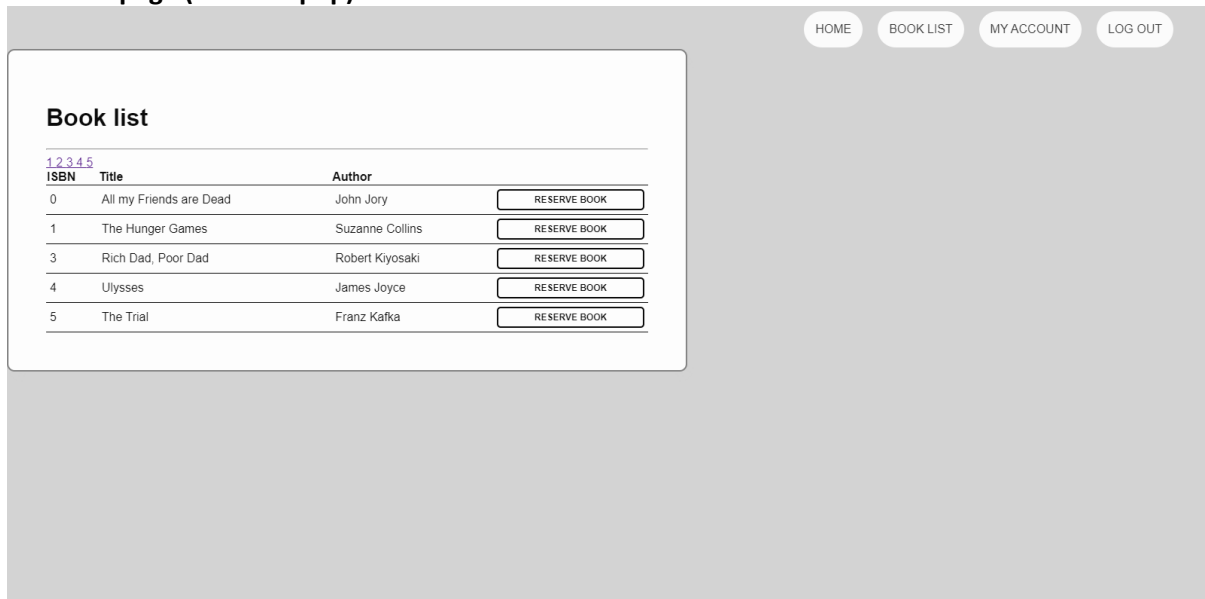
Header.php code:

```

<?php
echo '<header>
    <nav class="top-navbar">
        <ul>
            <li><a class = "navlink" href="logout.php">LOG OUT</a></li>
            <li><a class = "navlink" href="myaccountpage.php">MY ACCOUNT</a></li>
            <li><a class = "navlink" href="booklist.php">BOOK LIST</a></li>
            <li><a class = "navlink" href="homepageindex.php">HOME</a></li>
        </ul>
    </nav>
</header>';
?>

```

Book list page (booklist.php):



Description: When the user clicks on the “booklist” button in the navigation bar, they are taken to booklist.php which presents them with this page shown above. From here, the user can see all the books available for reservation on the library website. Each book has an ISBN code along with the Title of the book and the Author of each book. At the end is a big reserve book button that allows the user to reserve the book. Clicking the button takes the user to reservebooksuccessfully.php where they are given a “Book was reserved successfully” message. On the booklist page, only 5 books are presented to the user per page but due to pagination, the user can cycle through different pages via the hyperlinked numbers at the top. These numbers present different pages with different books on each page. The booklist page is also linked to the database in order for the user to see if the book is reserved by other users.

booklist.php code:

```
<?php
include_once 'sessionstart.php'; //Calling the file once to be used throughout
the running of this booklist PHP file.

if ($_SESSION['user'] == NULL) {
    header('Location: loginpage.php');
}

if (isset($_POST["button"]))
{
    $_SESSION['bookid'] = $_POST["button"];
    header('Location: reservebook.php'); //Directed to this PHP file/page once
user wants to reserve a book.
}
?>

<!DOCTYPE html>
<html>
    <head>
        <title>List of Books </title>
```

```

        <link rel="stylesheet" href="CSS/pages.css">
    </head>

<body>
    <?php include 'header.php'; ?>
    <div class="wrapper3">
        <h1>Book list</h1>
        <hr>
        <table>

            <tr>
                <th>ISBN</th>
                <th>Title</th>
                <th>Author</th>
                <th></th>
            </tr>
            <?php
                require("connectdb.php"); //Connecting to the database needed
to call the books.

                $sql = "SELECT ID_book, title, CONCAT(author_first_name,'
',author_last_name) AS author from books";

                $results_per_page = 5;
                $query = "select *from books";
                $result = mysqli_query($con, $query);
                $number_of_result = mysqli_num_rows($result);

                $number_of_page = ceil ($number_of_result /
$results_per_page);

                if (!isset ($_GET['page'])) {
                    $page = 1;
                } else {
                    $page = $_GET['page'];
                }

                $page_first_result = ($page-1) * $results_per_page;

                $query = "SELECT ID_book, title, CONCAT(author_first_name,'
',author_last_name) AS author from books LIMIT " . $page_first_result . ',' .
$results_per_page;
                $result = mysqli_query($con, $query);

                while($row = $result->fetch_assoc()){
                    echo "<tr>";

```

```

        echo "<td>" . $row['ID_book'] . "</td>";
        echo "<td>" . $row['title'] . "</td>";
        echo "<td>" . $row['author'] . "</td>";
        // echo "<td>" . $row['category'] . "</td>";
        echo "<td>" . '<form action=' . htmlspecialchars($_SERVER["PHP_SELF"])
. ' method="post"> <button type="submit" name="button" value=' .
$row["ID_book"] . ' class="rentbutton">RESERVE BOOK</button> </form>' .
"</td>";
        echo "</tr>";
    }

    for($page = 1; $page<= $number_of_page; $page++) {
        echo '<a href = "booklist.php?page=' . $page . '">' . $page . '
</a>';
    }

    ?>

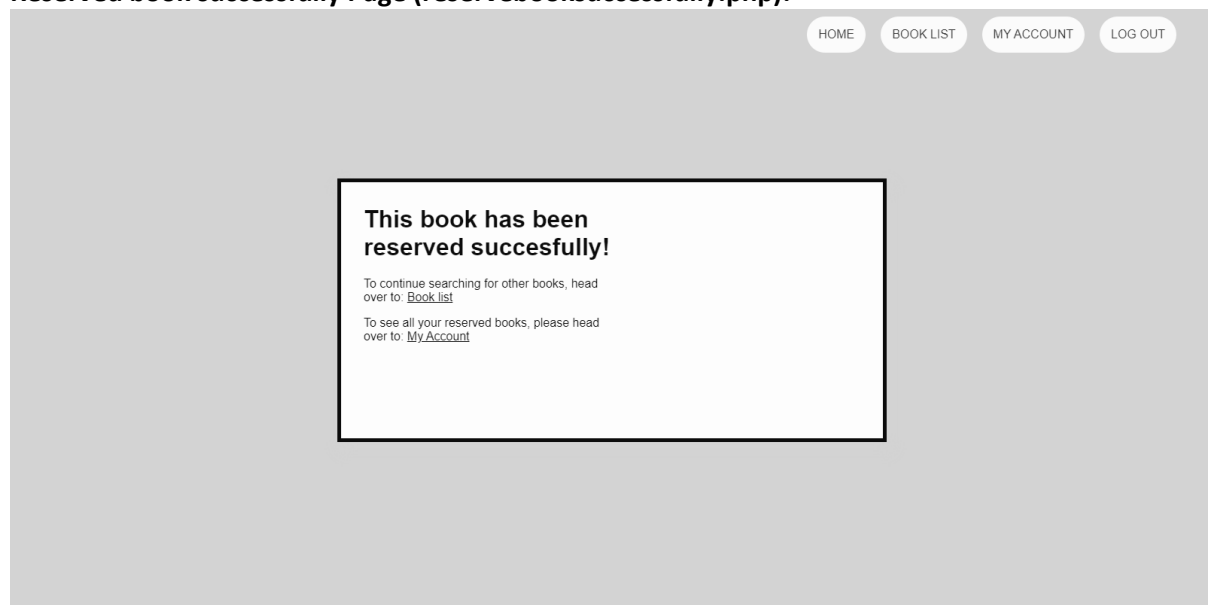
</table>
</div>

</body>

</html>

```

Reserved book successfully Page (reservebooksuccessfully.php):



Description: The page shown above is the page presented to the user when they reserve a book successfully. They are taken to this reservebooksuccessfully.php page where they are given a nice message along with clickable hyperlinks that can take the user back to the booklist to reserve more books, or to the My Account page (myaccountpage.php) where every user can see their book reservations.

reservebooksuccessfully.php code:

```

<?php
include_once 'sessionstart.php';
if ($_SESSION['user'] == NULL){
    header('Location: loginpage.php');
}
?>
<!DOCTYPE html>
<html >
    <head>
        <title>Book List</title>
        <link rel="stylesheet" href="CSS/pages.css">
    </head>
    <body>
        <?php include 'header.php'; ?>
        <div class="wrapper2">
            <div class="column">
                <h1>This book has been reserved succesfully!</h1>
                <p>To continue searching for other books, head over to: <a
class="link" href="booklist.php">Book list</a>
                <p>To see all your reserved books, please head over to: <a
class="link" href="myaccountpage.php">My Account</a>
            </div>
            <div class="column">
            </div>
        </div>

    </body>
</html>

```

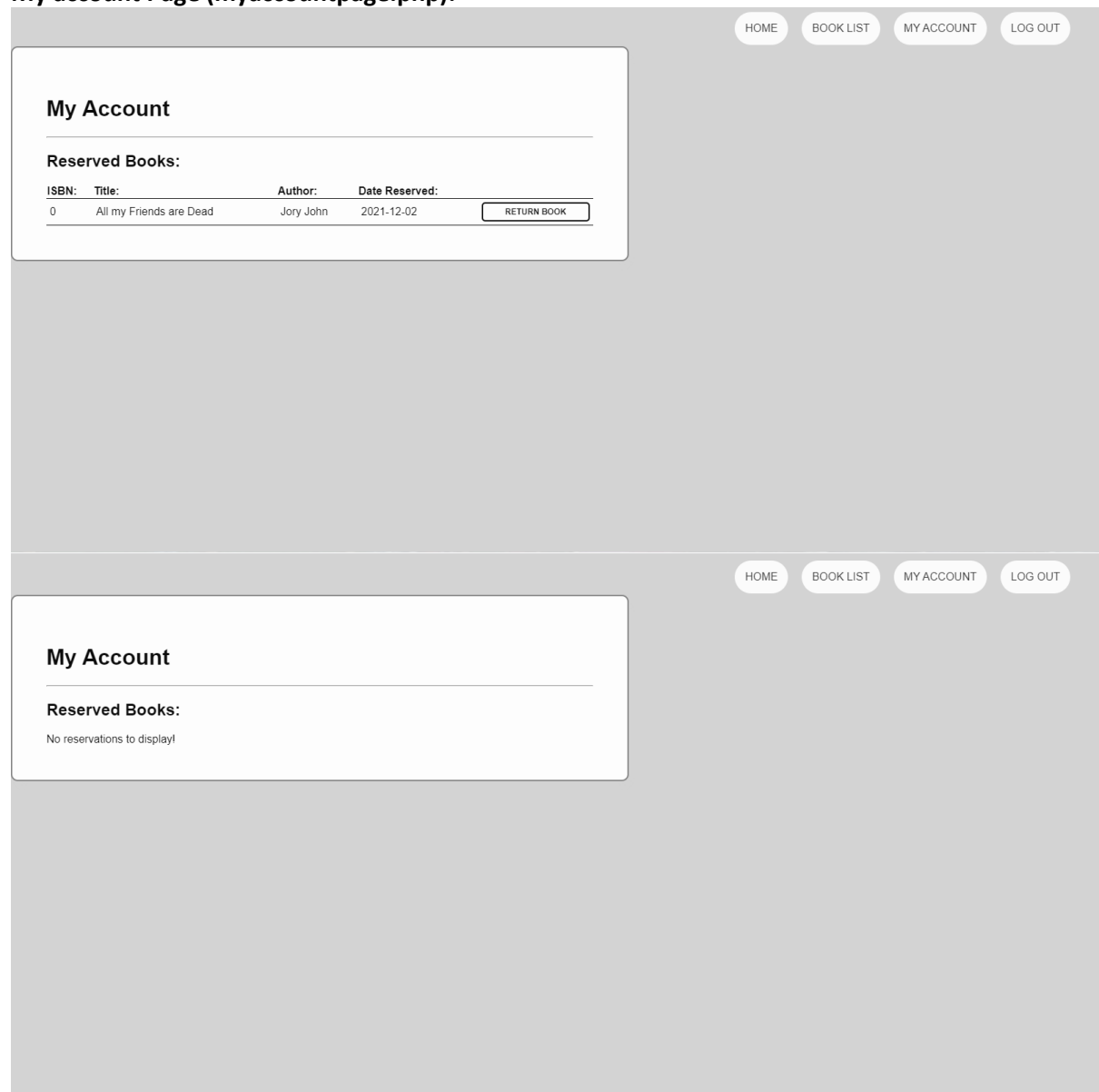
reservebook.php code:

```

<?php
include_once 'sessionstart.php';
if ($_SESSION['user'] == NULL) {
    header('Location: loginpage.php');
}
require_once('connectdb.php');
$userid = $_SESSION['userid'];
$sql = "INSERT INTO transactions (Date_start, ID_book, ID_user) values
(CURDATE(), '$_SESSION[bookid]', '$userid')";
if ($con->query($sql) === TRUE) {
    header('Location: reservebooksuccessfully.php');
} else {
    echo "Error: " . $sql . "<br>" . $con->error;
}
$_SESSION['bookid'] = NULL;
$con->close();
?>

```


My account Page (myaccountpage.php):



Description: This is the My Account page where users are able to see all the books they have reserved. If they do not have any books reserved, the user is just presented with a “No reservations to display!” message. If the user has reserved books, once again it displays the ISBN number along with title and author. The user also has the option to return the book to the library once they are doing using it. When the return book button is clicked, the user is taken to returnbooksuccess.php which just present the user with a confirmation that their book was returned successfully. The returnbooksuccess.php file is connected to the returnbook.php file which connects the page to the database allowing for the program to see what books are reserved by what users and to put the book back in the library once returned.

myaccountpage.php code:

```
<?php
include_once 'sessionstart.php';
require_once('connectdb.php');
if ($_SESSION['user'] == NULL) {
```

```

    header('Location: loginpage.php');
}
if (isset($_POST["button"]))
{
    $_SESSION['transactionid'] = $_POST["button"];
    header('Location: returnbook.php');
}
?>
<!DOCTYPE html>
<html>
    <head>
        <title>My Account</title>
        <link rel="stylesheet" href="CSS/pages.css">
    </head>
    <body>
        <?php include 'header.php'; ?>
        <div class="wrapper3">

            <h1>My Account</h1>
            <hr>
            <h2>Reserved Books:</h2>
            <?php
                $userid = $_SESSION['userid'];

                $sql = "
                SELECT transactions.ID_transaction, books.ID_book, books.title,
                CONCAT(books.author_last_name, ' ', books.author_first_name) AS author,
                transactions.Date_start FROM transactions NATURAL JOIN books
                WHERE transactions.Date_end IS NULL AND transactions.ID_User = '$userid'
                ";

                if(!$result = $con->query($sql)){
                    die('There was an error running the query [' . $con->error . ']');
                }
                if($result->num_rows > 0) {
                    echo "<table>
                        <th>ISBN:</th>
                        <th>Title:</th>
                        <th>Author:</th>
                        <th>Date Reserved:</th>
                    </tr>";
                    while($row = $result->fetch_assoc()){
                        echo "<tr>";
                        echo "<td>" . $row['ID_book'] . "</td>";
                        echo "<td>" . $row['title'] . "</td>";
                        echo "<td>" . $row['author'] . "</td>";
                        echo "<td>" . $row['Date_start'] . "</td>";

```

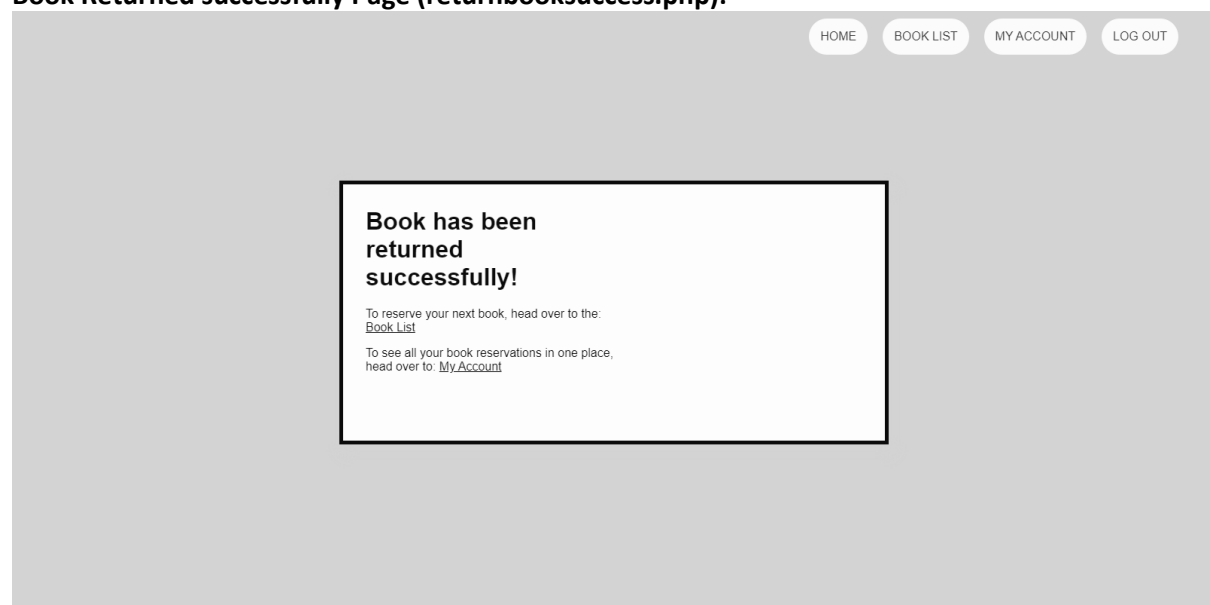
```

        echo "<td>" . '<form action=' . htmlspecialchars($_SERVER["PHP_SELF"]) . '
method="post"> <button type="submit" name="button" value=' .
$row["ID_transaction"] . ' class="rentbutton">RETURN BOOK</button> </form>' .
"</td>";
        echo "</tr>";
    }
    echo "</table>";
}
else {
    echo "No reservations to display!";
}
$result->free();
$con->close();
?>
</div>

</body>
</html>

```

Book Returned successfully Page (returnbooksuccess.php):



returnbooksuccess.php code:

```

<?php
include_once 'sessionstart.php';
if ($_SESSION['user'] == NULL){
    header('Location: loginpage.php');
}
?>
<!DOCTYPE html>
<html >
    <head>
        <title>Book List</title>
        <link rel="stylesheet" href="CSS/pages.css">
    
```

```

</head>
<body>
  <?php include 'header.php'; ?>
  <div class="wrapper2">
    <div class="column">
      <h1>Book has been returned successfully!</h1>
      <p>To reserve your next book, head over to the: <a class="link"
href="booklist.php">Book List</a>
      <p>To see all your book reservations in one place, head over to: <a
class="link" href="myaccountpage.php">My Account</a>
    </div>
    <div class="column">
    </div>
  </div>

</body>
</html>

```

returnbook.php code:

```

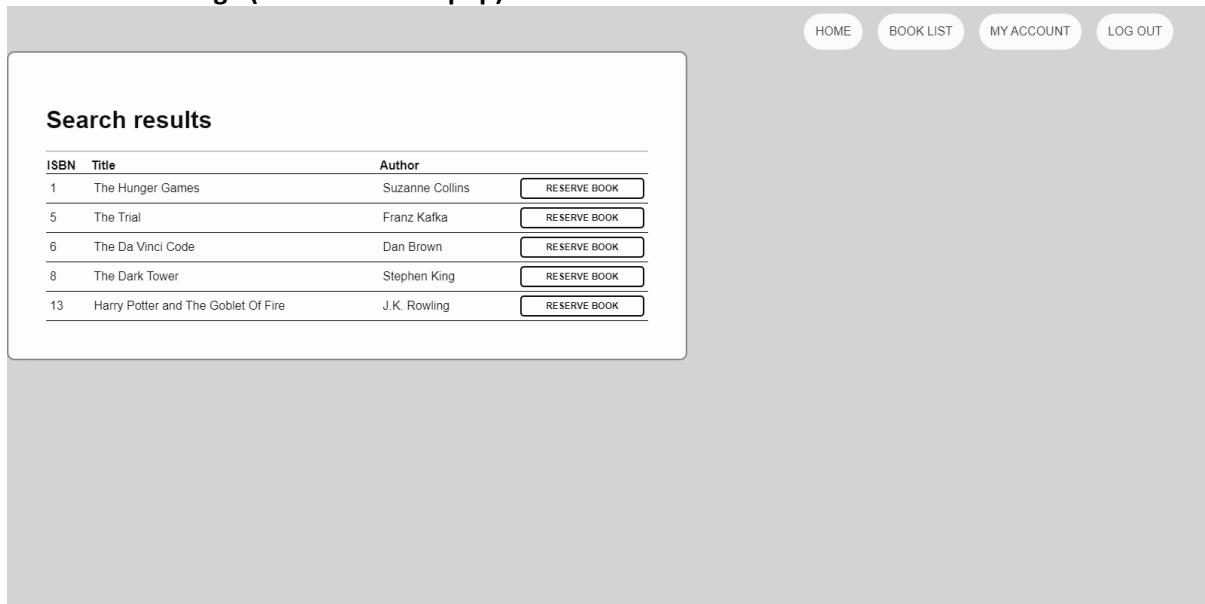
<?php
include_once 'sessionstart.php';
if ($_SESSION['user'] == NULL) {
  header('Location: loginpage.php');
}

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "library";

$con = new mysqli($servername,$username, $password, $dbname);
$sql = "UPDATE transactions SET Date_end=CURDATE() WHERE
ID_transaction='$_SESSION[transactionid]'";
if ($con->query($sql) === TRUE) {
  header('Location: returnbooksuccess.php');
} else {
  echo "Error: " . $sql . "<br>" . $con->error;
}
$con->close();
?>

```

Search results Page (searchfunction.php):



Description: From the main page, if the user searches for a book by author or title, they are presented with this page as shown above which is similar to the booklist. Here you can see the ISBN number along with the title and author as well as a button to reserve which follows the exact same steps as explained before.

searchfunction.php code:

```
<?php
include_once 'sessionstart.php';
if ($_SESSION['user'] == NULL) {
    header('Location: loginpage.php');
}
if (isset($_POST["button"]))
{
    $_SESSION['bookid'] = $_POST["button"];
    header('Location: reservebook.php');
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Book List</title>
    <link rel="stylesheet" href="CSS/pages.css">
</head>
<body>
    <?php include 'header.php'; ?>
    <div class="wrapper3">
        <h1>Search results</h1>
        <hr>
        <?php
            require_once('connectdb.php');
            $query = $_POST["q"];
```

```

$search_type = $_POST["search_type"];
if ($search_type == 'author_last_name'){
$search_type = "author_last_name LIKE '%{$query}%' OR author_first_name";
}

$sql = "
SELECT ID_book, title, CONCAT(author_first_name, ' ',author_last_name) AS
author from books
WHERE $search_type LIKE '%{$query}%' " ;
if(!$result = $con->query($sql)){
die('There was an error running the query [' . $con->error .']');
}
if($result->num_rows > 0) {
    echo "<table>
        <th>ISBN</th>
        <th>Title</th>
        <th>Author</th>
        <th></th>
    </tr>";
    while($row = $result->fetch_assoc()){
        echo "<tr>";
        echo "<td>" . $row['ID_book'] . "</td>";
        echo "<td>" . $row['title'] . "</td>";
        echo "<td>" . $row['author'] . "</td>";
        echo "<td>" . '<form action="searchfunction.php" method="post"> <button
type="submit" name="button" value=' . $row["ID_book"] . '
class="rentbutton">RESERVE BOOK</button> </form>' . "</td>";
        echo "</tr>";
    }
    echo "</table>";
}
else {
    echo "No records to display!";
}
$result->free();
$con->close();
?>
</div>

</body>
</html>

```

Other files used to make the library system:

CSS file to customise the website:

```

/* GLOBAL */
footer {
    bottom: 0;

```

```
position: absolute;
background-color: grey;
opacity: 0.85;
width: 100%;
height: 20px;
margin: 0 auto;
text-align: center;
}

.error {
  color: red;
}

.correct {
  border-color: green;
}

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

li a {
  display: block;
}

.navlink {
  text-decoration: none;
}

body {
  font-family: sans-serif;
  background-color: lightgrey;
  background-repeat: no-repeat;
  background-size: cover;
  width: 100%;
  height: 100%;
  margin: 0;
}

.navlink:link {
  color: #222222;
  background-color: transparent;
  text-decoration: none;
}

.navlink:visited {
```

```
    color: #222222;
    background-color: transparent;
    text-decoration: none;
}

.navlink:hover {
    color: black;
    background-color: transparent;
    text-decoration: underline;
}

.navlink:active {
    color: black;
    background-color: transparent;
    text-decoration: none;
}

.button{
    background-color: transparent;
    font-weight: bold;
    color: black;
    border: 1px solid black;
    border-radius: 5px;
    padding: 2px;
    cursor: pointer;
}

.button2 {
    background-color: transparent;
    font-weight: bold;
    color: black;
    border: 1px solid black;
    width: 100%;
    border-radius: 5px;
    height: 100%;
    padding: 2px;
    cursor: pointer;
}

.button:hover {
    background-color: midnightblue;
    color: white;
    transition-duration: 0.3s;
    cursor: pointer;
}

.button2:hover {
    background-color: midnightblue;
```



```
    color: white;
    transition-duration: 0.3s;
    cursor: pointer;
}

/* HOME PAGE */

.top-navbar {
    margin-top: 10px;
    width: 100%;
    height: 20%;
}

.top-navbar .navlink {
    padding: 15px;
    float: right;
    background-color: white
    ;
    border-radius: 100px;
    margin: 0 10px;
    opacity: .9;
}

.top-navbar li::after {
    padding-left: 10px;
}

.top-navbar ul:first-child {
    padding-right: 40px;
}

#searchButton{
    float: right;
    box-shadow: 2px 2px 2px 1px lightgray;
    cursor: pointer;
}

#searchButton:hover {
    cursor: pointer;
    box-shadow: 1.5px 1.5px 2px 2px lightgray;
    cursor: pointer;
}

.wrapper {
    width: 320px;
    height: 170px;
    position: absolute;
```

```
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
    margin: auto;
    background-color: white;
    opacity: 0.95;
    border-radius: 10px;
}

.wrapper *{
    margin: 5px;
}

.logo {
    font-size: 50px;
    font-weight: 100;
    font-family: sans-serif;
    text-align: center;
    padding-bottom: 5px;
}

input[type=search] {
    width: 100%;
    font-size: 14px;
    border-radius: 15px;
    border-width: 2px;
    padding: 5px;
    opacity: 0.9;
}

.searchEngine {
display: inline;
}

input:focus {
    outline: none;
}

/* LOGIN PAGE */

.wrapper2 {
    display: flex;
    width: 700px;
    height: 320px;
    position: absolute;
    top: 0;
    bottom: 0;
```

```
    left: 0;
    right: 0;
    margin: auto;
    padding: 10px 10px;
    border: 5px solid black;
    border-radius: 0px;
    box-shadow: 5px 10px 15px lightgray;
    opacity: 0.95;
    background-color: white;
}

.column {
    flex: 50%;
    margin-left: 20px;
}

.loginForm * {
    border-radius: 10px;
    width: 75%;
    padding: 10px;
    margin: 5px 0;
    font-size: 16px;
}

.column hr {
    width: 80%;
    float: left;
}

.column h2 {
    padding-left: 85px;
}

.register {
    color: navy;
    font-weight: bold;
}

/*BOOK LIST PAGE*/
.wrapper3 {
    width: 800px;
    height: auto;
    display: block;
    position: absolute;
    margin: 50px auto auto auto;
    padding: 50px 50px;
    border: 2px solid gray;
    border-radius: 10px;
```

```
    box-shadow: 5px 10px 15px lightgray;
    opacity: 0.95;
    background-color: white;
}

td {
    padding: 5px;
}

tr {
    text-align: left;
    border-bottom: 1px solid black;
}

table {
    width: 100%;
    border-collapse: collapse;
}

tr:hover {
    background-color: lightgray;
    transition-duration: 0.2s;
}

.rentbutton {
    width: 100%;
    border-radius: 5px;
    background-color: transparent;
    color: black;
    font-weight: bold;
    border: 2px solid black;
    padding: 5px;
    font-size: 12px;
    cursor: pointer;
}

.rentbutton:hover{
    background-color: navy;
    color: white;
    transition-duration: 0.2s;
    cursor: pointer;
}

/* -- RESERVE/RETURN BOOK SUCCESS --*/

.link {
    color: black;
}
```

```

.link:visited {
    color: black;
}

.link:active {
    color: black;
}

.link:hover {
    color:black;
}

```

programend.php code:

```

<?php
include_once 'login.php';
if ($_SESSION['user'] == NULL) {
    header('Location: loginpage.php');
}
?>

<!doctype html>
<html>

    <head>
        <title> Goodbye!</title>
        <link rel="stylesheet" href="CSS/pages.css">
    </head>

<body>
    <div class="wrapper2">
        <div class="intro column">
            <h1>P L A C E H O L D E R</h1>
        </div>
        <?php include 'formforlogin.php'; ?>
    </div>

</body>
</html>

```

Description: This piece of code functions to allow the user to log out once they are done using the library website. Programend.php takes the user to logout.php which eventually brings the user back to the login screen seen at the start of the program waiting for another user to log in. The logout.php file destroys the session previously used by another user.

logout.php code:

```
<?php
include_once 'sessionstart.php';
if ($_SESSION['user'] == NULL) {
    header('Location: loginpage.php');
}
$_SESSION = array();
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}
session_destroy();
header('Location: programend.php');
?>
```

DESIGN DOCUMENT

Below are wireframe illustrations of each screen that can be seen throughout the use of the library website:

A very simplistic design was used throughout the library website that allowed the user to very clearly read everything without being bombarded with different books and information. For the booklist and search function, only a max of 5 books were displayed per page. Pages did not have much text as to not overwhelm the user reading and hyperlinks as well as a navigation bar was present on every page allowing for quick manoeuvrability around the website.

Log in Page:



Home page:

A diagram of a computer monitor and keyboard. At the top, there is a horizontal menu bar with four rectangular buttons, each containing a single horizontal line. Below this, the monitor is represented by a large rectangle with a thick border. Inside the monitor's frame, there are three horizontal lines near the top and a single horizontal line near the bottom. Below the monitor, there is a horizontal bar representing the keyboard, which is divided into two sections: a larger section on the left and a smaller section on the right.

Booklist/Search Pages:

<div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> </div>			
<div> <div>12345</div> <div> <div></div> <div></div> <div></div> </div> </div>			
-			
-			
-			
-			
-			

My Account page:

Book reserved/returned successfully page:

The diagram illustrates a simple neural network architecture. At the top, there are four input nodes, each represented by a rectangle with a horizontal line inside. These nodes are connected to a hidden layer consisting of two nodes, also represented by rectangles with horizontal lines. The hidden layer nodes are connected to a single output node at the bottom, which is a rectangle with a horizontal line. The entire network is set against a light gray background.