**Web Development 2 Library Project**

List of PHP pages in my project and their functionaliy:

**Register.php**: This is the first page you interact with, the user is given the option to register for the website, or it they already have an account, they can clikc log in, whihc will redirect them to the login page, once either of these actions is complete they are brought to the search page.

**Login.php:**This file allows the user to log in ny checking if the users inputted username exists in the tabla and if the passwrd matches.

**Logout1.php**: This page is used to log the user out of the website, clearing the currently user session and allowing for a different user to log in. It also helps prevent non logged in users accessing elements of the website they should not.

**Index:php:** Home page to navigate through the different functions.

**Search.php**: This is the page where the user can search through the libraries database to find books, they can search by Book title, Author and by Category, they also have the option to reserve a book if it is not already reserved, during searching.

**Reserve.php**: On the reserve.php page, the currently logged in user can see what books they have reserved, and can un-reserve them if they wish. This page is accessed through the nav-bar on each of the pages

**Header.php:** This is used to create the header/nav-bar on each page and is called at the top of each of the pages  
  
**Footer.php**: Similarly to the header, it is called on each of the pages and sets what the footer of each page looks like.

**Database.php:** This connects the page to the databse which is used throughout the library, this is called as require\_once ‘database.php’ on each of the .php files throughout the website.

**Code**

**Register.php**

<!-- register.php (PHP Registration Script) -->

<?php

// Connects database page

require "database.php";

// Check connection

// Check the database connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get the input values from the form

$username = $\_POST['username'];

$password = $\_POST['password'];

$confirm\_password = $\_POST['confirm\_password'];

$firstname = $\_POST['firstname'];

$surname = $\_POST['surname'];

$addressline1 = $\_POST['addressline1'];

$addressline2 = $\_POST['addressline2'];

$city = $\_POST['city'];

$telephone = $\_POST['telephone'];

$mobilenumber = $\_POST['mobilenumber'];

// Used to prevent SQL injection

$username = mysqli\_real\_escape\_string($conn, $username);

$password = mysqli\_real\_escape\_string($conn, $password);

$firstname = mysqli\_real\_escape\_string($conn, $firstname);

$surname = mysqli\_real\_escape\_string($conn, $surname);

$addressline1 = mysqli\_real\_escape\_string($conn, $addressline1);

$addressline2 = mysqli\_real\_escape\_string($conn, $addressline2);

$city = mysqli\_real\_escape\_string($conn, $city);

$telephone = mysqli\_real\_escape\_string($conn, $telephone);

$mobilenumber = mysqli\_real\_escape\_string($conn, $mobilenumber);

// Check if password length is exactly 6 characters

if (strlen($password) != 6) {

echo "Invalid Input, Password must be six characters";

} else if ($password !== $confirm\_password) {

echo "Password and Confirm Password do not match";

} else {

// Prepare the SQL statement using a prepared statement to prevent SQL injection

$stmt = $conn->prepare("INSERT INTO users (Username, Password, FirstName, Surname, AddressLine1, AddressLine2, City, Telephone, Mobile)

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)");

// Bind the parameters to the prepared statement

$stmt->bind\_param("sssssssss", $username, $password, $firstname, $surname, $addressline1, $addressline2, $city, $telephone, $mobilenumber);

// Execute the query and check if it was successful

if ($stmt->execute()) {

echo "Registered Successfully!";

header("location: login.php");

} else {

echo "Error: " . $stmt->error;

}

// Close the statement and connection

$stmt->close();

$conn->close();

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://www.phptutorial.net/app/css/style.css">

<link rel="stylesheet" href="register.css">

<title>Register</title>

</head>

<body>

<main>

<form action="register.php" method="POST">

<div class="container">

<h1>Register</h1>

<p>Please fill in this form to create an account.</p>

<hr>

<label for="username"><b>Username</b></label>

<input type="text" placeholder="Enter Username" name="username" id="username" required>

<label for="psw"><b>Password (6 Characters)</b></label>

<input type="password" placeholder="Enter Password" name="password" id="password" required>

<label for="psw-confirm"><b>Confirm Password</b></label>

<input type="password" placeholder="Confirm Password" name="confirm\_password" id="confirm\_password" required>

<label for="fName"><b>First Name</b></label>

<input type="text" placeholder="Enter First Name" name="firstname" id="firstname" required>

<label for="Surname"><b>Surname</b></label>

<input type="text" placeholder="Enter Surname" name="surname" id="surname" required>

<label for="line1"><b>Address Line 1</b></label>

<input type="text" placeholder="Enter Address Line 1" name="addressline1" id="addressline1" required>

<label for="line2"><b>Address Line 2</b></label>

<input type="text" placeholder="Enter Address Line 2" name="addressline2" id="addressline2" required>

<label for="city"><b>City</b></label>

<input type="text" placeholder="Enter City" name="city" id="city" required>

<label for="telephone"><b>Telephone</b></label>

<input type="text" placeholder="Enter Telephone" name="telephone" id="telephone" required pattern="^[0-9]+$" title="Please enter a valid telephone number">

<label for="mobile"><b>Mobile Number</b></label>

<input type="text" placeholder="Enter Mobile Number" name="mobilenumber" id="mobilenumber" required pattern="^[0-9]+$" title="Please enter a valid mobile number">

<hr>

<p>By creating an account you agree to our <a href="#">Terms & Privacy</a>.</p>

<button type="submit" class="registerbtn" name="register">Register</button>

</div>

<div class="container signin">

<p>Already have an account? <a href="login.php">Login</a>.</p>

</div>

</form>

</main>

</body>

</html>

**Login.php**

<?php

// Start the session to track the logged-in user

session\_start();

// Database connection

require "database.php";

if (isset($\_POST['username']) && isset($\_POST['password'])) {

$username = $conn->real\_escape\_string($\_POST['username']);

$password = $conn->real\_escape\_string($\_POST['password']);

// Use a basic query with escaped inputs

$sql = "SELECT password FROM users WHERE username = '$username'";

$result = $conn->query($sql);

// Check if username exists and password matches

if ($result->num\_rows > 0) {

$row = $result->fetch\_assoc();

$dbPassword = $row['password'];

if ($password == $dbPassword) {

// Set session variables

$\_SESSION['username'] = $username; // Store the username in the session

echo "Login successful!";

// Redirect to search page after successful login

header("Location: search.php");

exit();

} else {

echo "Invalid password.";

}

} else {

echo "The username $username does not exist in the database.";

}

$result->free();

}

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://www.phptutorial.net/app/css/style.css">

<link rel="stylesheet" href="register.css">

<title>Login</title>

</head>

<body>

<main>

<form action="login.php" method="POST">

<div class="container">

<h1>Login</h1>

<p>Please enter your details.</p>

<hr>

<label for="username"><b>Username</b></label>

<input type="text" placeholder="Enter Username" name="username" id="username" required>

<label for="psw"><b>Password</b></label>

<input type="password" placeholder="Enter Password" name="password" id="password" required>

<button type="submit" class="loginbtn" >Login</button>

</div>

</form>

</main>

</body>

</html>

**Logout1.php**

<?php

// Start the session

session\_start();

// Destroy the session and clear all session data

session\_unset();

session\_destroy();

// Clear the session cookie by setting the expiration date to the past

setcookie(session\_name(), '', time() - 3600, '/');

// Redirect the user to the login page

header("Location: login.php");

exit();

?>

**Header.php**

<!-- header.php -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="register.css">

<title>Book Search Website</title>

</head>

<body>

<header>

<h1></h1>

<nav>

<ul>

<li><a href="index.php">Home</a></li>

<li><a href="register.php">Register</a><li>

<li><a href="search.php">Search Books</a></li>

<li><a href="reserve.php">Reservations</a></li>

<li><a href="logout1.php">Logout</a></li>

</ul>

</nav>

</header>

**Footer.php**

<!-- footer.php -->

<footer>

<p>&copy; 2024 Book Search Website. All rights reserved.</p>

</footer>

</body>

</html>

**Search.php**

<?php

include('header.php');

// Enable error reporting to help with debugging

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

// Start the session to maintain login status

session\_start();

// database connection

require\_once "database.php";

// Check if the user is logged in, if not, show an error or redirect to login page

if (!isset($\_SESSION['username'])) {

// Redirect to login page if the user is not logged in

header("Location: login.php");

exit(); }

// Check the connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

// Pagination settings

$booksPerPage = 5; // Number of books to display per page

// Get the current page number, default is 1

$page = isset($\_GET['page']) ? (int)$\_GET['page'] : 1;

$offset = ($page - 1) \* $booksPerPage; // Calculate the offset for the SQL query

// Query to fetch categories for the dropdown

$sqlCategories = "SELECT CategoryID, CategoryDescription FROM categories"; // Ensure both CategoryID and CategoryDescription are selected

$categoriesResult = $conn->query($sqlCategories);

// Check if the query was successful

if ($categoriesResult === false) {

die("Error fetching categories: " . $conn->error . "<br>SQL Query: $sqlCategories");

}

// Handle search query if submitted

$BookTitle = isset($\_POST['BookTitle']) ? $conn->real\_escape\_string($\_POST['BookTitle']) : (isset($\_GET['BookTitle']) ? $conn->real\_escape\_string($\_GET['BookTitle']) : '');

$Author = isset($\_POST['Author']) ? $conn->real\_escape\_string($\_POST['Author']) : (isset($\_GET['Author']) ? $conn->real\_escape\_string($\_GET['Author']) : '');

$CategoryID = isset($\_POST['CategoryID']) ? $conn->real\_escape\_string($\_POST['CategoryID']) : (isset($\_GET['CategoryID']) ? $conn->real\_escape\_string($\_GET['CategoryID']) : '');

// Perform the search query with LIKE for partial matches (only at the beginning)

$sql = "SELECT ISBN, BookTitle, Author, CategoryID, Reserved FROM books WHERE 1=1";

// Match BookTitle starting with the given value

if (!empty($BookTitle)) {

$sql .= " AND BookTitle LIKE '$BookTitle%'"; // Match if BookTitle starts with the search term

}

// Match Author starting with the given value

if (!empty($Author)) {

$sql .= " AND Author LIKE '$Author%'"; // Match if Author starts with the search term

}

// Exact match for CategoryID

if (!empty($CategoryID)) {

$sql .= " AND CategoryID = '$CategoryID'"; // Exact match for CategoryID

}

// Get total number of books for pagination

$totalResult = $conn->query($sql);

if ($totalResult === false) {

die("Error fetching total books: " . $conn->error);

}

$totalBooks = $totalResult->num\_rows;

$totalPages = ceil($totalBooks / $booksPerPage); // Calculate total pages

// Add LIMIT and OFFSET to the query for pagination

$sql .= " LIMIT $booksPerPage OFFSET $offset";

// Fetch the search results

$result = $conn->query($sql);

// Handle book reservation

if (isset($\_GET['reserve'])) {

$isbn = $conn->real\_escape\_string($\_GET['reserve']); // Get the ISBN of the book to reserve

$username = $\_SESSION['username']; // Get the logged-in username

$reservedDate = date('Y-m-d H:i:s'); // Current date and time as the reservation date

// Check if the book is already reserved

$checkReservedSql = "SELECT Reserved FROM books WHERE ISBN = '$isbn'";

$checkReservedResult = $conn->query($checkReservedSql);

$book = $checkReservedResult->fetch\_assoc();

if ($book && $book['Reserved'] === 'N') {

// Update the books table to mark the book as reserved

$updateReservedSql = "UPDATE books SET Reserved = 'Y' WHERE ISBN = '$isbn'";

// Insert a new record into the reservations table

$reserveSql = "INSERT INTO reservations (ISBN, Username, ReservedDate) VALUES ('$isbn', '$username', '$reservedDate')";

if ($conn->query($updateReservedSql) === TRUE && $conn->query($reserveSql) === TRUE) {

echo "Book reserved successfully.";

} else {

echo "Error reserving the book: " . $conn->error;

}

} else {

echo "This book is already reserved.";

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://www.phptutorial.net/app/css/style.css">

<link rel="stylesheet" href="register.css">

<title>Book Search</title>

</head>

<body>

<main>

<h1>Search for Books</h1>

<div class="container">

<!-- Search form -->

<form action="search.php" method="POST">

<label for="BookTitle"><b>Book Title</b></label>

<input type="text" placeholder="Enter Book Title" name="BookTitle" id="BookTitle" value="<?php echo htmlspecialchars($BookTitle); ?>">

<label for="Author"><b>Author</b></label>

<input type="text" placeholder="Enter Author" name="Author" id="Author" value="<?php echo htmlspecialchars($Author); ?>">

<label for="CategoryID"><b>Category</b></label>

<select name="CategoryID" id="CategoryID">

<option value="">Select Category</option>

<?php

// Check if categories are available

if ($categoriesResult->num\_rows > 0) {

// Loop through the result and display both CategoryID and CategoryDescription

while ($category = $categoriesResult->fetch\_assoc()) {

$selected = ($CategoryID == $category['CategoryID']) ? 'selected' : '';

// Display CategoryID and CategoryDescription in the dropdown

echo "<option value='" . $category['CategoryID'] . "' $selected>" . htmlspecialchars($category['CategoryID']) . " - " . htmlspecialchars($category['CategoryDescription']) . "</option>";

}

} else {

echo "<option value=''>No categories available</option>";

}

?>

</select>

<button type="submit" class="registerbtn">Search</button>

</form>

</div>

<div class="search-results">

<?php if (!empty($BookTitle) || !empty($Author) || !empty($CategoryID)): ?>

<form action="search.php" method="GET">

<?php

// Check if there are search results

if ($result->num\_rows > 0) {

echo "<ul>"; // Start unordered list

while ($row = $result->fetch\_assoc()) {

echo "<li><strong>Book Title:</strong> " . htmlspecialchars($row['BookTitle']) . "<br><strong>Author:</strong> " . htmlspecialchars($row['Author']) . "<br>";

// Check if the book is already reserved

if ($row['Reserved'] === 'N') {

echo "<a href='search.php?page=$page&BookTitle=" . urlencode($BookTitle) . "&Author=" . urlencode($Author) . "&CategoryID=" . urlencode($CategoryID) . "&reserve=" . urlencode($row['ISBN']) . "'>Reserve this book</a>"; // Using ISBN

} else {

echo "<span>Reserved</span>";

}

echo "</li>";

}

echo "</ul>"; // End unordered list

} else {

echo "No books found matching the search criteria.";

}

// Pagination controls

if ($totalPages > 1) {

echo "<div class='pagination'>";

// Previous page link

if ($page > 1) {

echo "<a href='search.php?page=" . ($page - 1) . "&BookTitle=" . urlencode($BookTitle) . "&Author=" . urlencode($Author) . "&CategoryID=" . urlencode($CategoryID) . "'>Previous</a>";

}

// Next page link

if ($page < $totalPages) {

echo "<a href='search.php?page=" . ($page + 1) . "&BookTitle=" . urlencode($BookTitle) . "&Author=" . urlencode($Author) . "&CategoryID=" . urlencode($CategoryID) . "'>Next</a>";

}

echo "</div>";

}

?>

</form>

<?php endif; ?>

</div>

<!-- Close the database connection -->

<?php $conn->close(); ?>

<?php include('footer.php'); ?>

</body>

</main>

</html>

**Reserve.php**

<?php

include('header.php');

// Enable error reporting to help with debugging

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

// Start the session to maintain login status

session\_start();

// Include database connection

require\_once "database.php";

// Check if the user is logged in, if not, show an error or redirect to login page

if (!isset($\_SESSION['username'])) {

// Redirect to login page if the user is not logged in

header("Location: login.php");

exit();

}

// Check the connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

// Fetch reserved books for the logged-in user

$username = $\_SESSION['username'];

$reservedBooksSql = "SELECT r.ISBN, b.BookTitle, b.Author, r.ReservedDate

FROM reservations r

JOIN books b ON r.ISBN = b.ISBN

WHERE r.Username = '$username' AND b.Reserved = 'Y'";

$reservedBooksResult = $conn->query($reservedBooksSql);

// Handle unreserve action

if (isset($\_GET['unreserve'])) {

$isbnToUnreserve = $conn->real\_escape\_string($\_GET['unreserve']); // Get the ISBN of the book to unreserve

// Update the books table to mark the book as not reserved

$updateReservedSql = "UPDATE books SET Reserved = 'N' WHERE ISBN = '$isbnToUnreserve'";

// Delete the reservation record from the reservations table

$deleteReservationSql = "DELETE FROM reservations WHERE ISBN = '$isbnToUnreserve' AND Username = '$username'";

if ($conn->query($updateReservedSql) === TRUE && $conn->query($deleteReservationSql) === TRUE) {

echo "Book unreserved successfully.";

} else {

echo "Error unreserving the book: " . $conn->error;

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://www.phptutorial.net/app/css/style.css">

<link rel="stylesheet" href="register.css">

<title>Reserved Books</title>

</head>

<body>

<main>

<h1>Your Reserved Books</h1>

<div class="container">

<?php if ($reservedBooksResult->num\_rows > 0): ?>

<ul>

<?php while ($row = $reservedBooksResult->fetch\_assoc()): ?>

<li>

<strong>Book Title:</strong> <?php echo htmlspecialchars($row['BookTitle']); ?><br>

<strong>Author:</strong> <?php echo htmlspecialchars($row['Author']); ?><br>

<strong>Reserved Date:</strong> <?php echo htmlspecialchars($row['ReservedDate']); ?><br>

<a href="reserve.php?unreserve=<?php echo urlencode($row['ISBN']); ?>">Unreserve</a>

</li>

<?php endwhile; ?>

</ul>

<?php else: ?>

<p>You have no reserved books at the moment.</p>

<?php endif; ?>

</div>

<p>If you're not logged in, you'll be redirected to the login page.</p>

</main>

</body>

</html>

<?php

// Close the database connection

$conn->close();

include('footer.php');

?>

**Database.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "library";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

?>

**Index.php**

<?php

// Start the session and ensure the session cookie expires when the browser is closed

ini\_set('session.cookie\_lifetime', 0); // Set the session cookie to expire when the browser closes

session\_start();

// Check if the user is logged in

$isLoggedIn = isset($\_SESSION['username']);

?>

<?php include('header.php'); ?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://www.phptutorial.net/app/css/style.css">

<link rel="stylesheet" href="test.css">

<title>Home</title>

<style>

.center-text {

text-align: center;

margin: 10px 0;

}

</style>

</head>

<body>

<main>

<h1>Welcome to the Book Search Website</h1>

<?php if ($isLoggedIn): ?>

<p>Hello, <?php echo htmlspecialchars($\_SESSION['username']); ?>! You are logged in.</p>

<p>On this website you can search for and resver books, please follow the buttons below!</p>

<!-- Button to navigate to the search page -->

<form action="search.php">

<button type="submit" class="gotosearchbtn">Go to Book Search</button>

</form>

<!-- Button to navigate to reservations page -->

<form action="reserve.php">

<button type="submit" class="gotoreservebtn">Go to Reservations</button>

</form>

<!-- Logout button -->

<form action="logout1.php" method="POST">

<button type="submit" class="logoutbtn">Logout</button>

</form>\*

<?php else: ?>

<p>You are not logged in. Please log in to access the book search and reservations.</p>

<!-- Button to navigate to the login page -->

<form action="login.php">

<button type="submit" class="loginbtn">Go to Login</button>

</form>

<!-- Centered text -->

<p class="center-text">Don't have an account, register here!</p>

<!-- Button to navigate to the registration page -->

<form action="register.php">

<button type="submit" class="registerbtn">Register an Account</button>

</form>

<?php endif; ?>

</main>

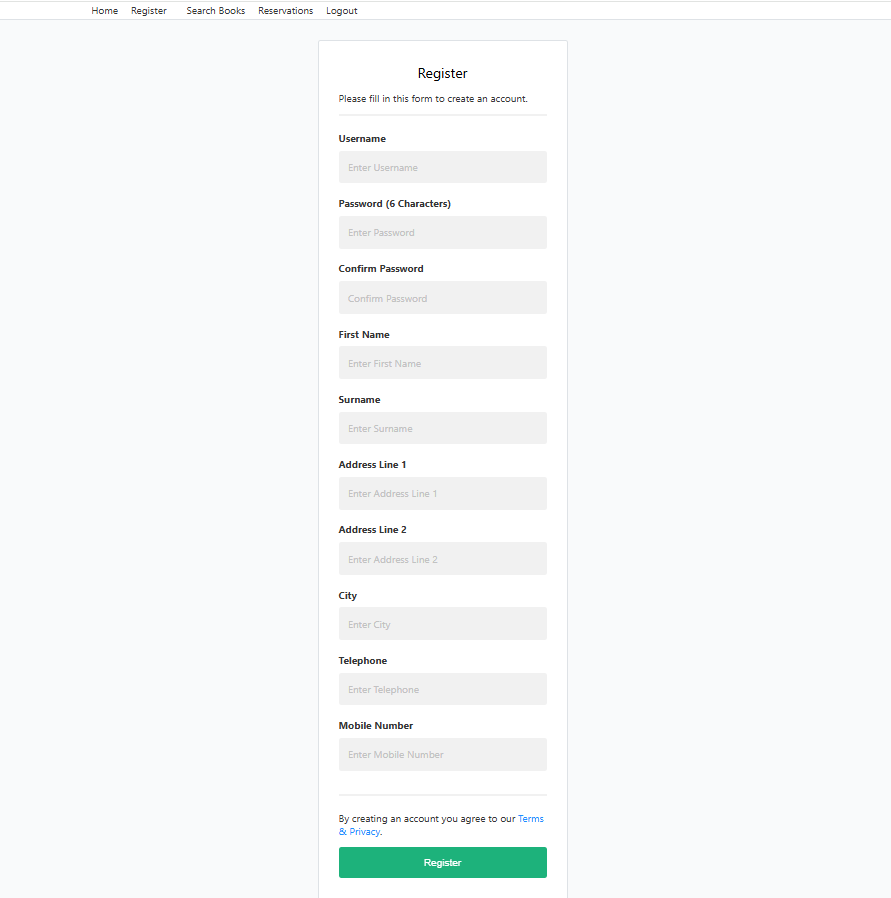
</body>

</html>

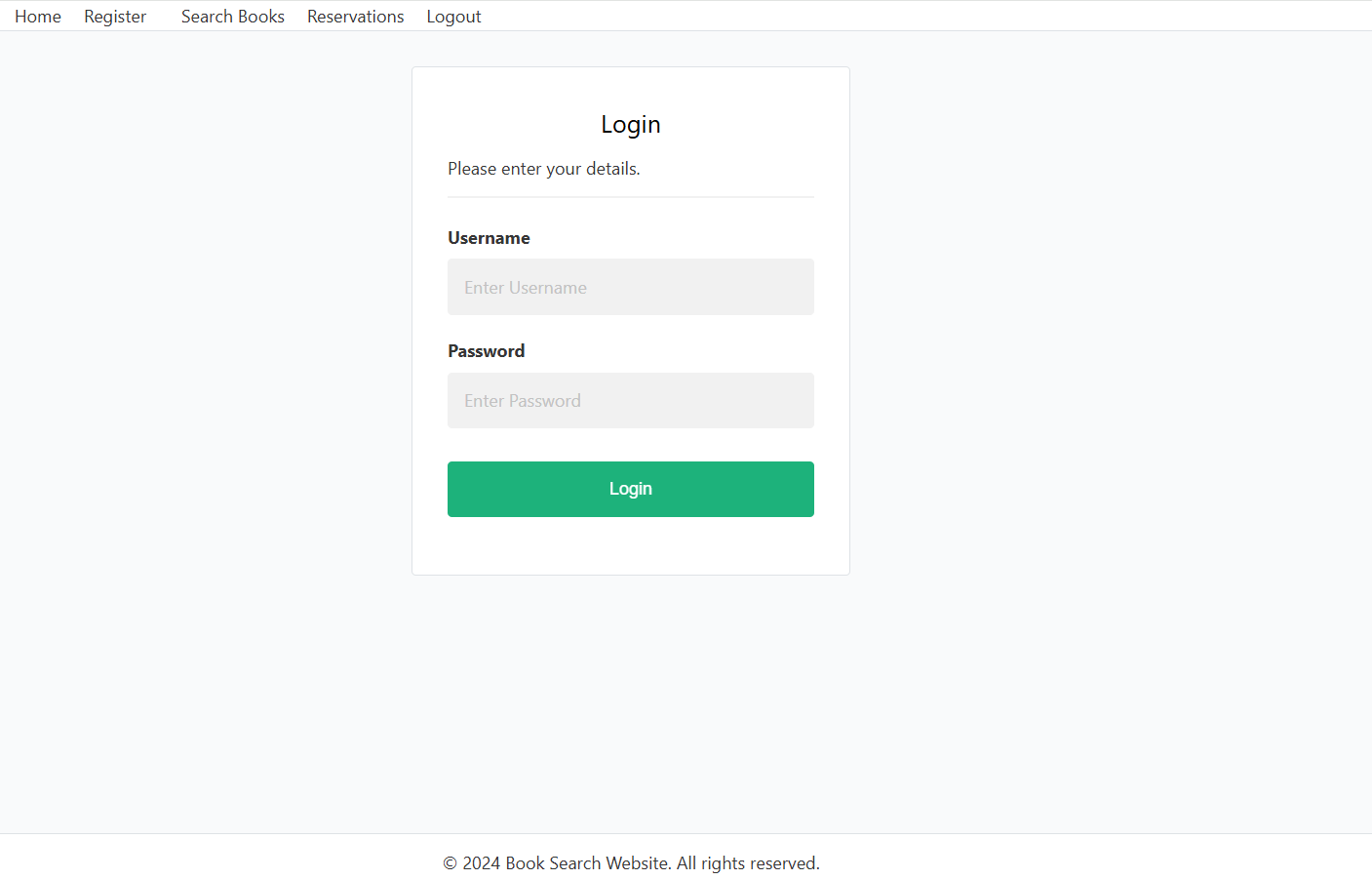
<?php include('footer.php'); ?>

**Screen Samples**

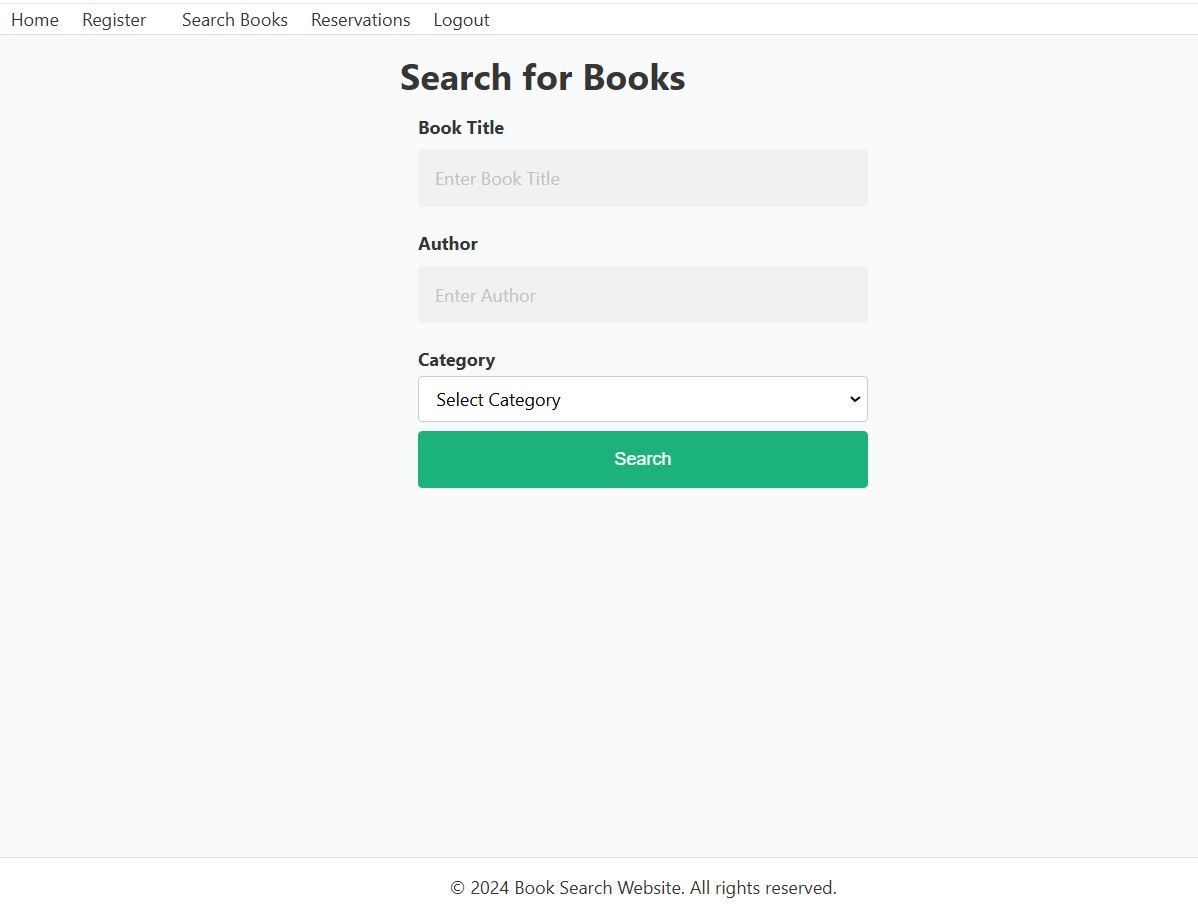
**Register page**

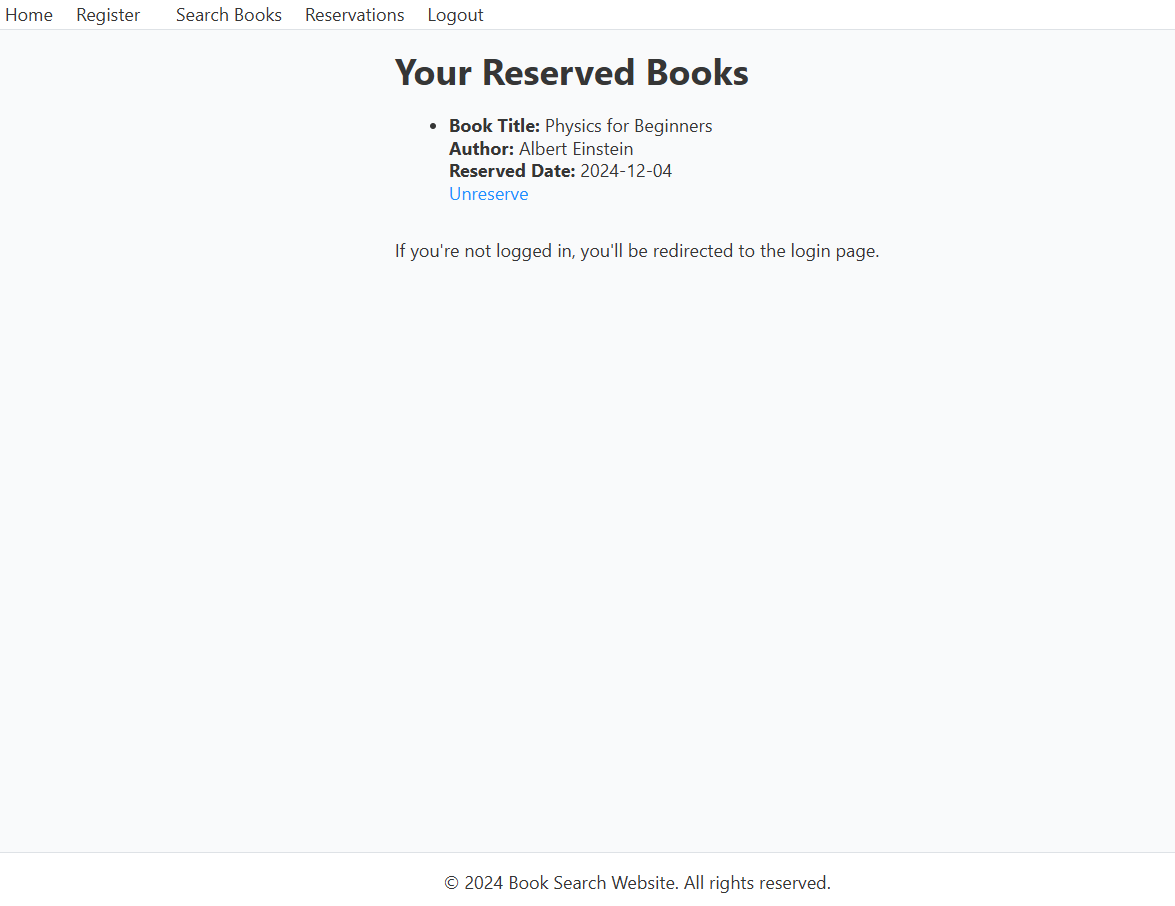


**Login page**

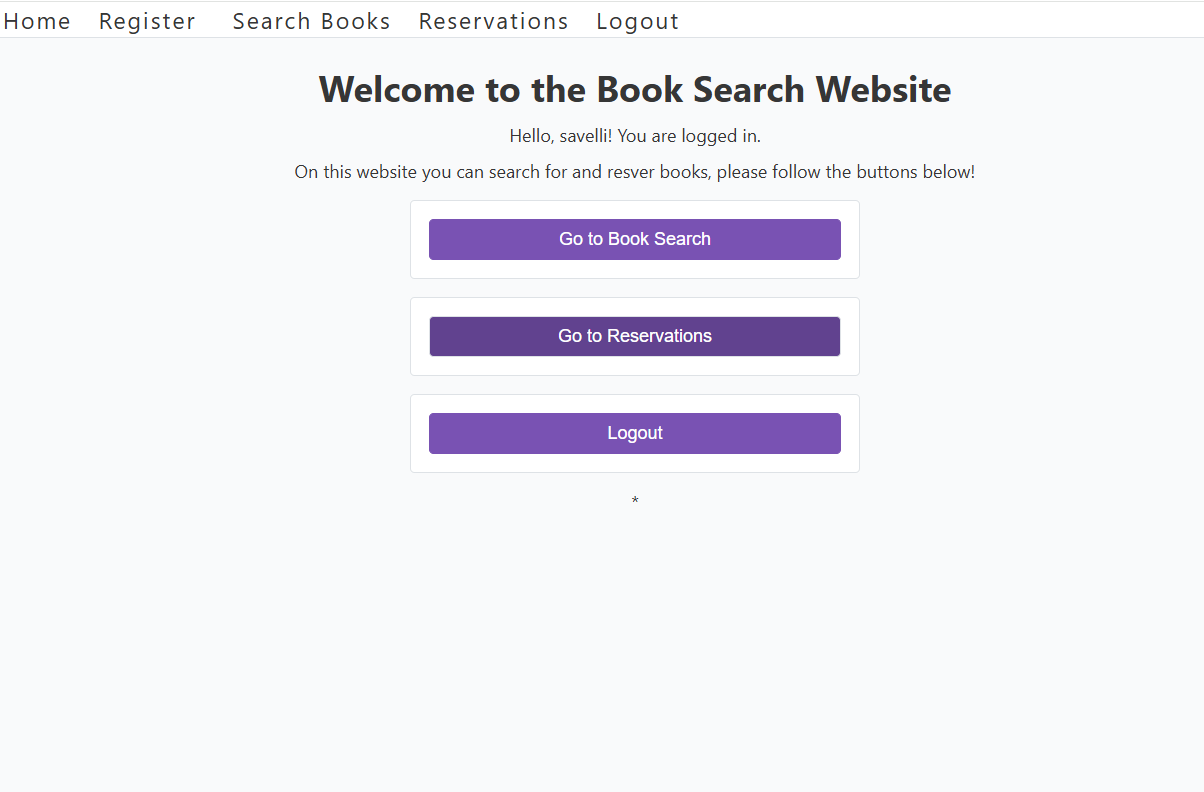


**Search Page**



**Reservations**  


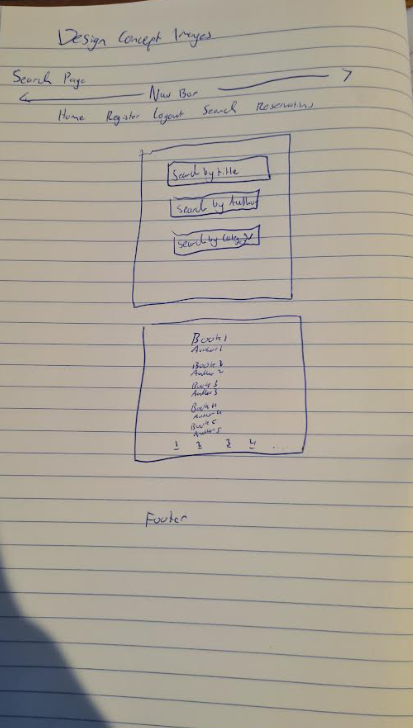
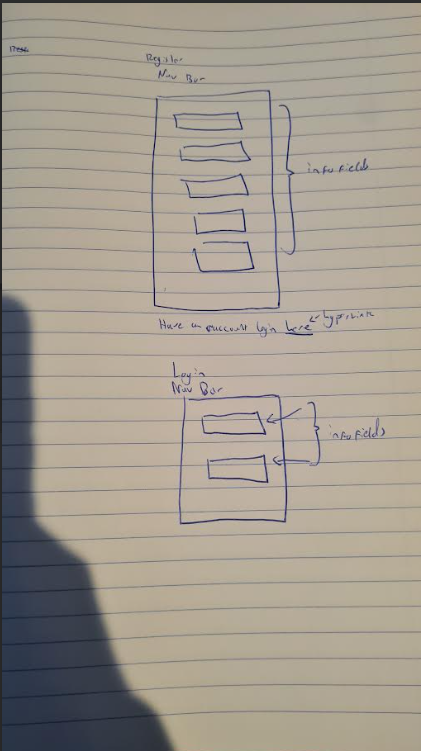
**Home Page**



**Design Document**

When designing the website, how the pages would interact with each other was an important consideration, I decided that beginning on the register page, which would have easy access to the login page would be best. From there the search page would be the main ‘hub’ form where everything could be accessed. A sleek design was also selected so that the user can carry out their desired task with ease.

Early Design concept images are below



These were used as a rough outline for the structure of the page and how it would look.