

Web development Assignment 2023

Student Name: Hemant Sundarrajan

Student Number: C22440886

Requirements: To build a functional, simple library website. It should have functionality like search, view reservations, reserve, remove reservations etc.

My list of PHP pages:

- 1) config.php
- 2) databases.php
- 3) index.php
- 4) login.php
- 5) logout.php
- 6) register.php
- 7) search.php
- 8) reserve.php
- 9) view.php
- 10) remove.php

My list of Images and CSS:

- background-img.jpg
- book1.jpg
- book2.jpg
- book3.jpg
- book4.jpg
- style.css

PHP PAGES

1. config.php

```
<?php
//this is what configures us to our particular database
$serverName = "localhost";
$username = "root";
$password = "";
$dbName = "bookdb"
?>
```

The Config.PHP page is quite short. It essentially defines the configuration of a connection to a database. It has variables which set up the connection, ensure it has a username and password so it is secure, and most importantly has YOUR databases name, so the data you are receiving, or inserting into is from YOUR database. In my case, my databases name is: bookdb. I made a separate page for configuring my database to promote code reuse, you will see that in my databases.php page. This page doesn't have a corresponding HTML page on my website, as it is firstly, secure information only I should know, and there is also no need to show this information on my website.

2. databases.php

```
<?php
//we require the configure.php, to use code reuse, this way we dont need to
write it one very page
require_once "config.php";

//this is how we use our database connections via a variable
$conn = new mysqli($servername, $username, $password, $dbName);

if ($conn->connect_error)
{
    die("Connection failed" . $conn->connect_error);
}
?>
```

This PHP page is what I used to create a connection to my database. As you can see, it requires the config.php page as this is what connects me to my database, with the localhost variables etc. Once again, this page is its own PHP page, as it promotes code reuse. Just like config.php, rather than writing this out on every page, I simply need to just code: require once "databases.php". This way I reuse my code, and it makes my page more efficient. This page connects to the database using the \$conn variable, which is very important, and is present in almost all my PHP pages. This is the variable which I use, to let my code know that we are attempting to make a connection to the database when I need to use my database. This code also has a check, so that if the connection fails, we can output this message and see what went wrong.

3. index.php

```
<?php
//starts session, which tracks our users session throughout websites, if there
is or isnt one
session_start();

//this uses our databases php, to utilise code reuse, so we dont have
to manually write the connection to the database
require_once 'databases.php';

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://kit.fontawesome.com/c13e82a8d6.js"
crossorigin="anonymous"></script>
    <title>Home</title>
</head>
<body>
    <header>
        <nav>
            <input type="checkbox" id="check">
            <label for="check">
                <i class="fas fa-bars" id="btn"></i>
                <i class="fas fa-times" id="cancel"></i>
            </label>
            
            <?php
//checks if user is logged in, if not the navbar shows less
options as user isnt logged in
if (!isset($_SESSION["Username"]))
{
    echo '<ul class="bar">
        <li>
            <a href="index.php">Home</a>
        </li>
        <li>
            <a href="login.php">Login</a>
        </li>
        <li>
            <a href="register.php">Register</a>
```

```

        </li>
    </ul>';
    echo '<div class="container">
        <div class="left">
            <p>Please Log in to access all of our libraries
features! Use the Navbar to Login, or <a href="login.php">Log in here</a></p>
            <p>If you dont have an account, please register and
create one. You can <a href="register.php">Register here</a></p>
            <p>Osprey Library is your one stop shop for E learning
and E books in an online Library!</p>

            <br>
            <br>

            <h2>Please Login to Avail of these great
services!</h2>
            <p>Our library is more than just a repository of
books; its a hub for learning and collaboration. Take advantage of our library
services, including reference assistance, interlibrary loans, and research
support. Our dedicated staff is here to help you navigate the world of
information and make the most of your library experience.</p>

            <br>
            <br>

            <h2>Membership and Accessibilty</h2>
            <p>Membership and Accessibility:
            Becoming a member of our library opens the door to a
world of knowledge and resources. Discover the benefits of membership,
including borrowing privileges, exclusive events, and personalized services.
We are committed to ensuring accessibility for all, so explore our accessible
services and facilities designed to cater to diverse needs.</p>

            <br>
            <br>

            <h2>Events and Programs:</h2>
            <p>Stay tuned for our exciting events and programs
designed to engage and enrich our community. From book clubs and author talks
to workshops and lectures, theres always something happening at the library.
Here at Osprey Library, we strive to ensure that you, the user has a great
experience on this website!.</p>

        </div>
        <div class="right">
            <div class="fourbyfour">
                
                

```

```

        
        
    </div>
</div>
</div>';
}
else
{
    $loggedInUsername = $_SESSION['Username'];
    echo '<ul class="bar">
        <li>
            <a href="index.php">Home</a>
        </li>
        <li>
            <a href="search.php">Search For a Book</a>
        </li>
        <li>
            <a href="view.php">View Reserved Books</a>
        </li>
        <li>
            <a href="logout.php">Logout</a>
        </li>
    </ul>';

    echo '<div class="container">
        <div class="left">';
        echo "<h1>Welcome back to OSPREY LIBRARY
$loggedInUsername !</h1>
        <p>Now that you are logged in, you can finally avail
of our services again!</p>
        <p>We have upgraded our online services to a great
level, hopefully you can avail of them today!</p>

        <br>
        <br>";

        echo '<h2>Nice to see you again!</h2>
        <p>Our library is more than just a repository of
books; its a hub for learning and collaboration. Take advantage of our library
services, including reference assistance, interlibrary loans, and research
support. Our dedicated staff is here to help you navigate the world of
information and make the most of your library experience.</p>

        <br>
        <br>

        <h2>Membership and Accessibilty</h2>
        <p>Membership and Accessibility:

```

Becoming a member of our library opens the door to a world of knowledge and resources. Discover the benefits of membership, including borrowing privileges, exclusive events, and personalized services. We are committed to ensuring accessibility for all, so explore our accessible services and facilities designed to cater to diverse needs.</p>

<h2>Events and Programs:</h2>

<p>Stay tuned for our exciting events and programs designed to engage and enrich our community. From book clubs and author talks to workshops and lectures, theres always something happening at the library. Here at Osprey Library, we strive to ensure that you, the user has a great experience on this website!.</p>

</div>

<div class="right">

<div class="fourbyfour">

</div>

</div>

</div>';

}

\$conn->close();

?>

</nav>

</header>

<footer>

<h5>Website developed by Hemant Sundarrajan, 2023</h5>

</footer>

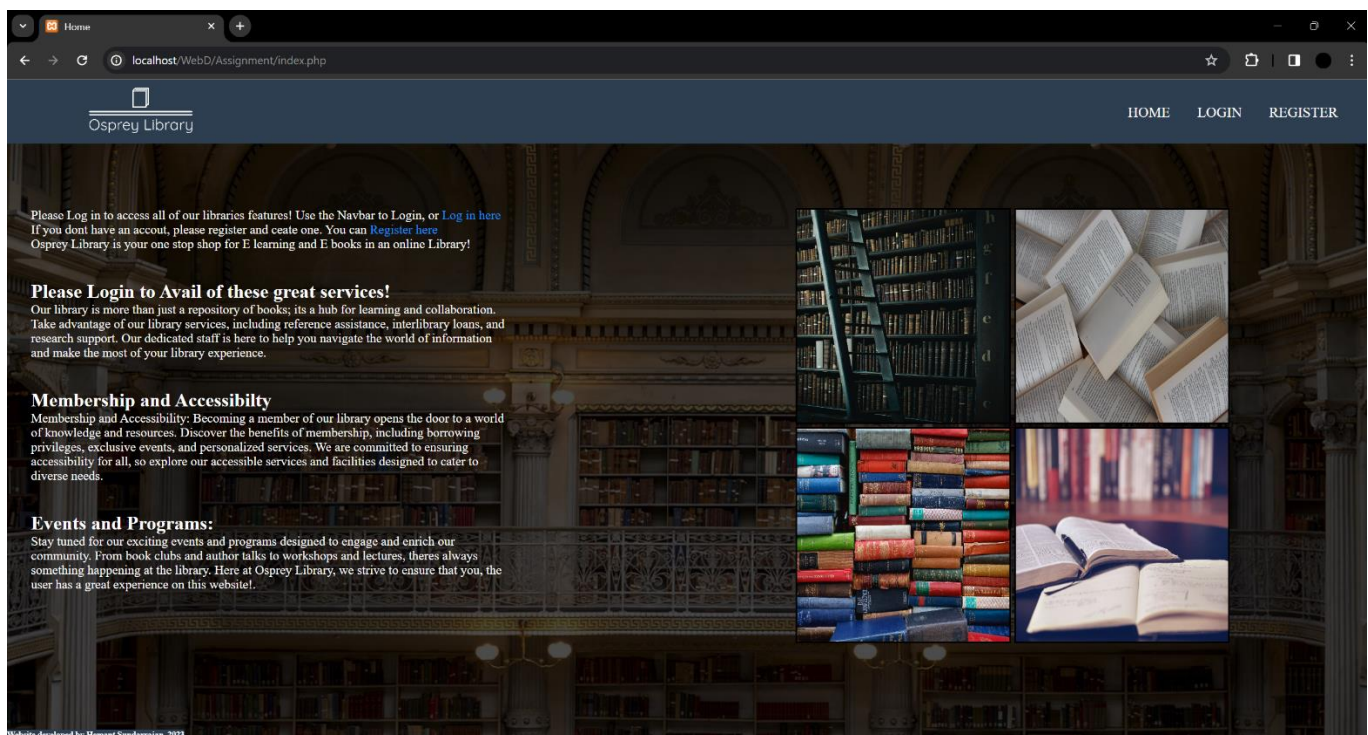
</body>

</html>

Index.php is the home page of my website. Like all my pages, the first thing you see is the session start(); and require once "databases.php". The session start is critical, as it tracks whether a user is logged in, and if a user is logged in, it ensures that consistency is kept between all the pages. This way it protects users' data integrity and allows a seamless consistency between pages on the

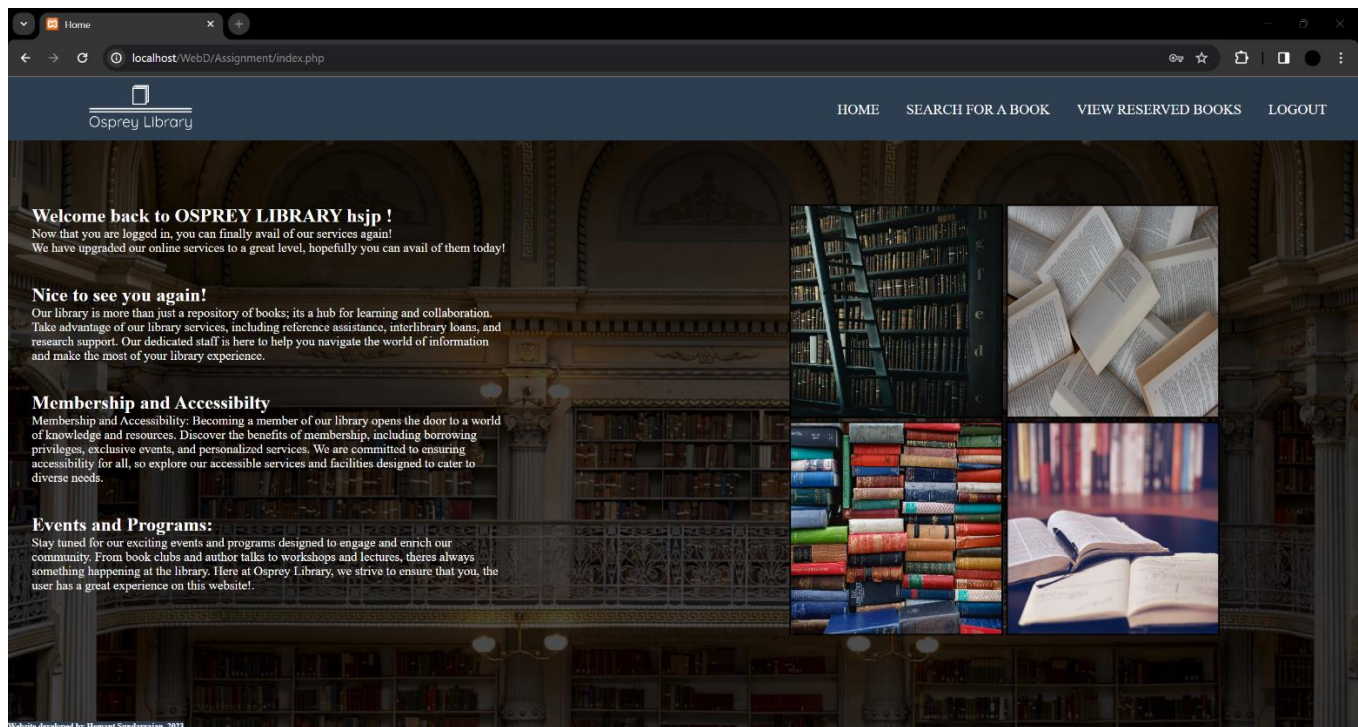
website. I then have my index page set up, so that when a user is not logged in, the information shown in the navbar, and in the content on the page is different from when the user is logged in. This is done as stated in the requirements. The user must be logged in to gain the ability to use the functionality on the website. If the user is not logged in, the website prompts the user to do so. Either by using the navbar, or by clicking on said prompts. Before the user logs in, they can only have access to the Home page (which is altered a bit compared to when they log in), the login page and the register page. The access to the login page and register page is obviously allowed, to allow the user to create, and then login to their account. I do this by using an if statement to check if a user is not logged in, and if they are, then the content shown changes. My pages also all have a consistent header/navbar and consistent footer.

How my home page (index.php) looks **BEFORE** logging in:



As you can see, before logging in, the navbar is a lot more limited, and there are prompts on the screen telling the user to login or register for an account.

How my home page (index.php) looks **AFTER** logging in



After logging in, the home page changes. It now welcomes the user with their username. There are also more options on the navbar. The user can now search for a book and reserve said book. The user can also view reserved books and remove reservations made, and the user can logout of their account. This is what the else portion of my index page does, it changes the home page to this after logging in.

4. login.php

```
<?php
//starts session, which tracks our users session throughout websites, if there
is or isnt one
session_start();

//this uses our databases php, to utilise code reuse, so we dont have
to manually write the connection to the database
require_once 'databases.php';

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://kit.fontawesome.com/c13e82a8d6"
crossorigin="anonymous"></script>
    <title>Login</title>
</head>
<body>
    <header>
        <nav>
            <input type="checkbox" id="check">
            <label for="check">
                <i class="fas fa-bars" id="btn"></i>
                <i class="fas fa-times" id="cancel"></i>
            </label>
            
            <ul class="bar">
                <li><a href="index.php">Home</a></li>
                <li><a href="login.php">Login</a></li>
                <li><a href="register.php">Register</a></li>
            </ul>
        </nav>
    </header>

    <div class="Regcontainer">
        <form method="post" novalidate>
            <div class="Regwrapper">
                <?php
                    //uses sql injection for protection, and checks with database
                    to see if username and password exists ana is correct
```

```

        if (isset($_POST["Username"]) && isset($_POST["Password"]))
        {
            $username = htmlentities($_POST["Username"]);
            $password = htmlentities($_POST["Password"]);

            //makes sure the fields arent empty
            if (!empty($username) && !empty($password))
            {
                $query = "SELECT * FROM users WHERE Username = ? AND
Password = ?";

                $stmt = $conn->prepare($query);
                $stmt->bind_param("ss", $username, $password);
                $stmt->execute();
                $result = $stmt->get_result();

                //redirects user to Home page
                if ($result->num_rows === 1)
                {
                    $_SESSION["Username"] = $username;
                    header('Location: index.php');
                    exit();
                } else {
                    $_SESSION["error"] = "Incorrect username or
password.";

                }
            }
            else
            {
                $_SESSION["error"] = "Missing Required Information";
            }
        }

        $conn->close();

    ?>
    <!-- Login form to Login to website-->
    <div class="title"><span>Login</span></div>

    <label for="Username"><b>Username</b></label>
    <input class="reg" type="text" placeholder="Enter Username"
name="Username" id="Username" required>

    <label for="Password"><b>Password</b></label>
    <input class="reg" type="password" placeholder="Enter
Password" name="Password" id="Password" required>

    <?php

```

```

        if (isset($_SESSION["error"])) {
            echo ' <p class="error-message">' . $_SESSION["error"] .
'</p>';
            unset($_SESSION["error"]);
        }
    ?>

    <hr>

    <button type="submit" class="registerbtn">Login</button>
    <p>Don't have an account? <a href="register.php">Register
here.</a></p>
    </div>
</form>
</div>
<footer>
    <h5>Website developed by Hemant Sundarrajan, 2023</h5>
</footer>
</body>
</html>

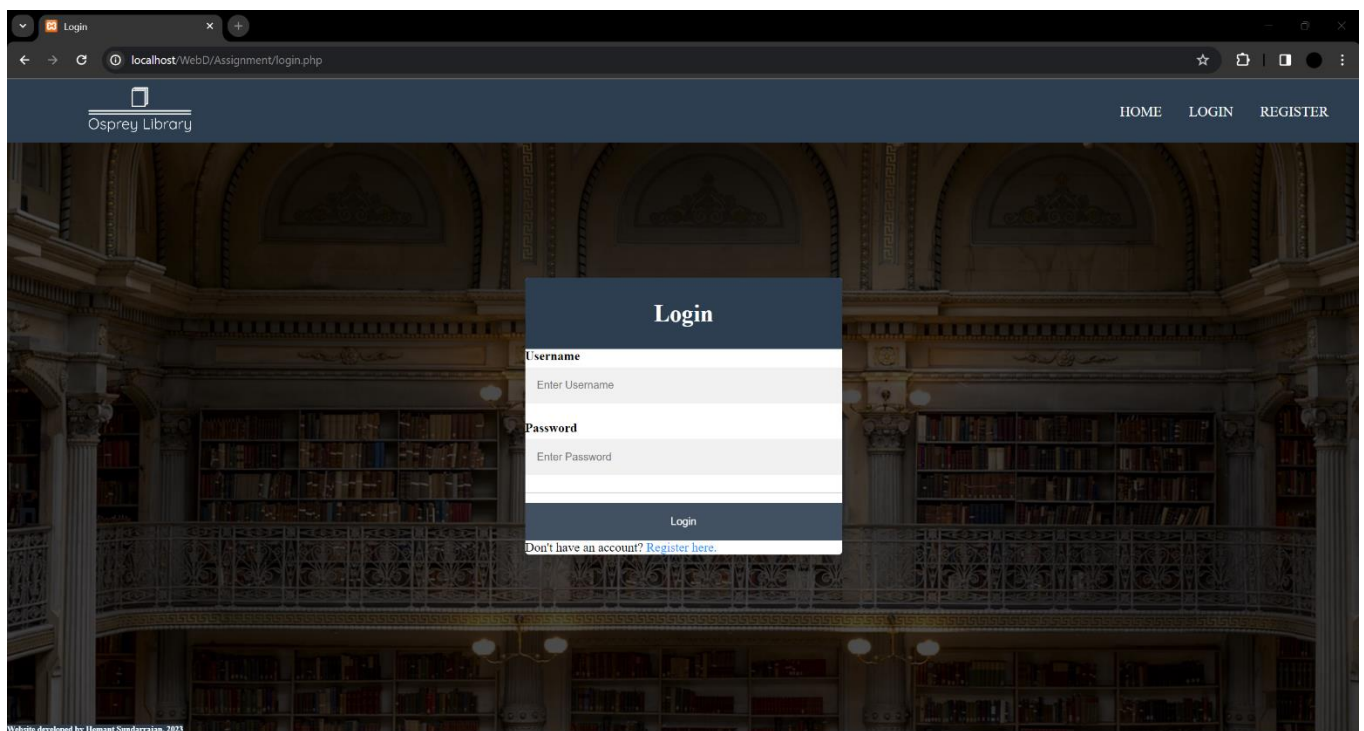
```

My login page utilises session start() and databases.php like all of my pages do. This is how I ensure that when a user is logged in, the session is ready to take that information and use it on all my pages.

The login page is quite simple yet important. It consists of a simple login form, with a username and password field, and a link to the registration page, if you do not have an account yet. The PHP works in a simple way. It connects to the database, checks if a user is in the database, and if that user's password is correct. If these conditions are not fulfilled, it will post an error. This page ensures that the user has all fields filled to proceed with the login, or once again an error is posted. To be safe, the PHP used is utilised with SQL injection. This stops all kinds of issues that may arise with someone logging in to a website. It ensures that no malicious attacks using SQL can be initiated on the database. My code then checks if the user with the certain password exists, i.e. a row for that user exists. If this is true, the user is logged in successfully. Now all the accessibility of the library site is available for the user to use. The SQL query used select

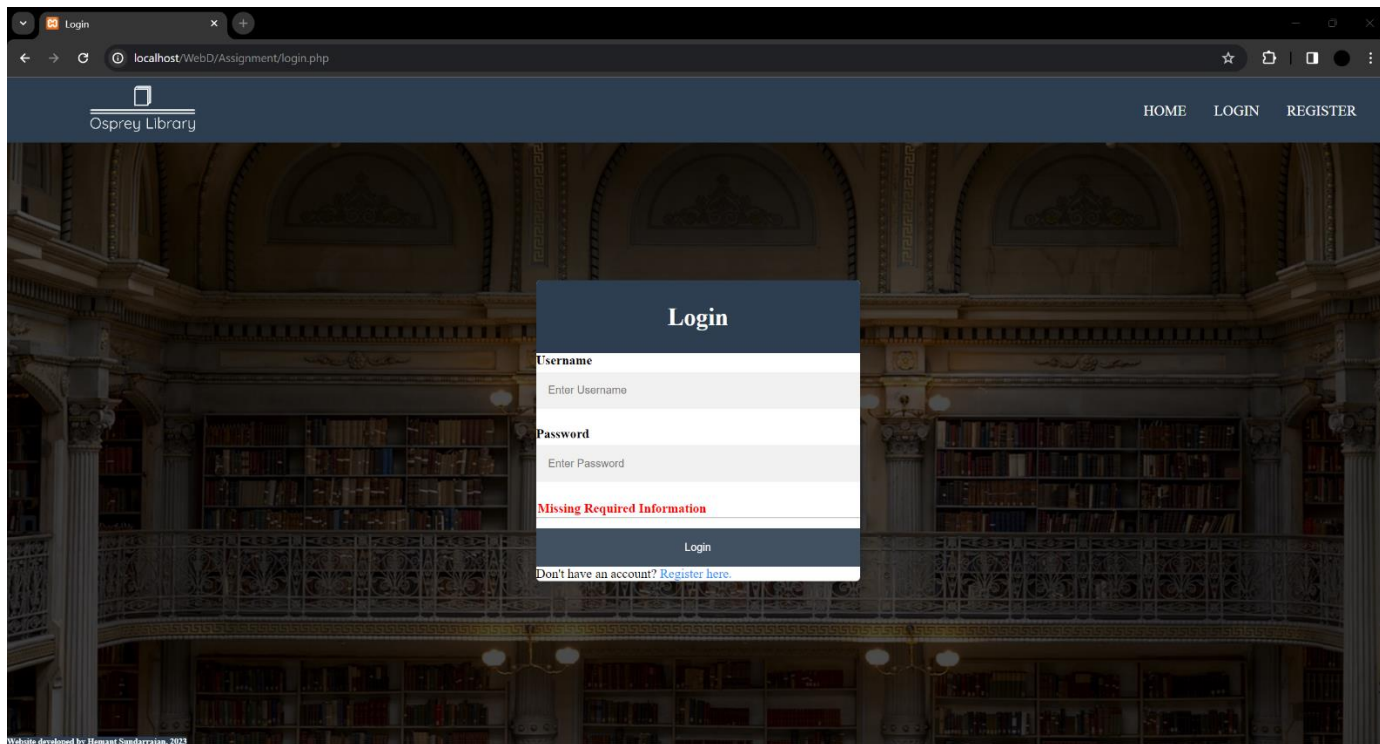
all the usernames and passwords from the database and gives it a “?” parameter, expecting a username and password. The “ss” portion of the injection essentially just means the website is expecting two strings. I also have a SESSION error message, taken from the Brightspace notes, which allows me to not hardcode errors all the time but rather just use this variable to display error messages efficiently. On successful login, the user is sent to the updated home page as seen above. The user also has access to the entire website.

How my Login page (login.php) looks in my website:



It's a simple, centred, login page. It's easy to understand and clear to the user.

How my Login Page (login.php) looks with an error message:



For example, you get this kind of error message if you do not enter anything into the login field. It is a clear and simple message.

5 logout.php

```
<?php
//Logout page which destroys session, and logs current user out, then sends
user back to Home
session_start();
session_destroy();
header("Location: index.php")
?>
```

The logout code is extremely short. It starts the session (this is the way sessions are kept consistent throughout the website) so that user who is logged in on the website is recognized when logging out. The actual logging out is done when the session is destroyed (session destroy();) This is how a session is ended. The user is then sent back to the beginning home page, where they can login or register again if they so desire. There is no actual output on this page, it is purely just a page that provides the logout functionality and nothing else.

6. register.php

```
<?php
//starts session, which tracks our users session throughout websites, if there
is or isnt one
session_start();

//this uses our databases php, to utilise code reuse, so we dont have
to manually write the connection to the database
require_once 'databases.php';

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://kit.fontawesome.com/c13e82a8d6"
crossorigin="anonymous"></script>
    <title>Register</title>
</head>
<body>
    <header>
        <nav>
            <input type="checkbox" id="check">
            <label for="check">
                <i class="fas fa-bars" id="btn"></i>
                <i class="fas fa-times" id="cancel"></i>
            </label>
            
            <ul class="bar">
                <li><a href="index.php">Home</a></li>
                <li><a href="login.php">Login</a></li>
                <li><a href="register.php">Register</a></li>
            </ul>
        </nav>
    </header>
    <?php
        if ( isset($_POST['Username']) && isset($_POST['Password']) &&
isset($_POST['PasswordConf']) && isset($_POST['FirstName'])&&
isset($_POST['Surname']) && isset($_POST['AddressLine1']) &&
isset($_POST['AddressLine2']) && isset($_POST['City']) &&
isset($_POST['Telephone']) && isset($_POST['Mobile']))
        {
```



```

//assigns the entered values to variables
$username = htmlentities($_POST['Username']);
$password = htmlentities($_POST['Password']);
$passwordc = htmlentities($_POST['PasswordConf']);
$firstname = htmlentities($_POST['FirstName']);
$surname = htmlentities($_POST['Surname']);
$addr1 = htmlentities($_POST['AddressLine1']);
$addr2 = htmlentities($_POST['AddressLine2']);
$city = htmlentities($_POST['City']);
$telephone = htmlentities($_POST['Telephone']);
$mobile = htmlentities($_POST['Mobile']);

//ensures fields arent empty
if (!empty($username) && !empty($password) && !empty($passwordc) &&
!empty($firstname) && !empty($surname) && !empty($addr1) && !empty($addr2) &&
!empty($city) && !empty($telephone) && !empty($mobile))
{
    $numberString = strval($telephone);
    $numberString2 = strval($mobile);

    //query used to check if a user already exists in the database
    $query = "SELECT * FROM users WHERE Username = '$username'";
    $result = $conn->query($query);

    //ensure the requirements are met so a user can register
    if (strlen($password) !== 6)
    {
        $_SESSION["error"] = "Password must be 6 characters long";
    }
    else if($password !== $passwordc)
    {
        $_SESSION["error"] = "The Passwords do not match!";
    }
    else if (strlen($numberString) !== 10)
    {
        $_SESSION["error"] = "Your Telephone number is not 10
digits long!";
    }
    else if (strlen($numberString2) !== 10)
    {
        $_SESSION["error"] = "Your Mobile Phone number is not 10
digits long!";
    }
    else if ($result->num_rows === 1)
    {
        $_SESSION["error"] = "This Username is Already taken!";
    }
    else

```

```

        {
            //inserts user into database
            $sql = "INSERT INTO users (Username, Password, FirstName,
Surname, AddressLine1, AddressLine2, City, Telephone, Mobile) VALUES
('$uname', '$pw', '$fname', '$lname', '$addr1', '$addr2', '$c', '$t', '$m')";

            if ($conn->query($sql) === TRUE)
            {
                header('Location: login.php');
                return;
            }
            else
            {
                echo "Error: " . $sql . "<br>" . $conn->error;
            }
        }
    }
    else
    {
        $_SESSION["error"] = "Missing Required Information";
        header('Location: register.php');
        return;
    }
}
$conn->close();
?>
<div class="Regcontainer">

    <form method="post" novalidate>
        <div class="Regwrapper">
            <div class="title"><span>Register</span></div>
            <?php

                if (isset($_SESSION["error"]))
                {
                    echo '<p class="error-messagereg">' . $_SESSION["error"] .
'</p>';

                    unset($_SESSION["error"]);
                }
            ?>

            <!-- Registration form to Register into website-->
            <label for="Username"><b>Username</b></label>
            <input class="reg" type="text" placeholder="Enter Username"
name="Username" id="Username" required>

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

```

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

        <label for="PasswordConf"><b>Password Confirmation</b></label>
        <input class="reg" type="password" placeholder="Enter Password
Confirmation" name="PasswordConf" id="PasswordConf" required>

        <label for="FirstName"><b>First Name</b></label>
        <input class="reg" type="text" placeholder="Enter First Name"
name="FirstName" id="FirstName" required>

        <label for="Surname"><b>Surname</b></label>
        <input class="reg" type="text" placeholder="Enter Surname"
name="Surname" id="Surname" required>

        <label for="AddressLine1"><b>Address Line 1</b></label>
        <input class="reg" type="text" placeholder="Enter
AddressLine1" name="AddressLine1" id="AddressLine1" required>

        <label for="AddressLine2"><b>Address Line 2</b></label>
        <input class="reg" type="text" placeholder="Enter
AddressLine2" name="AddressLine2" id="AddressLine2" required>

        <label for="City"><b>City</b></label>
        <input class="reg" type="text" placeholder="Enter City"
name="City" id="City" required>

        <label for="Telephone"><b>Telephone</b></label>
        <input class="reg" type="tel" placeholder="Enter Tel no"
name="Telephone" id="Telephone" required>

        <label for="Mobile"><b>Mobile</b></label>
        <input class="reg" type="tel" placeholder="Enter Mobile no"
name="Mobile" id="Mobile" required>

        <hr>

        <button type="submit" class="registerbtn">Register</button>
        <p>Already have an account? <a href="login.php">Log in
here</a>.</p>
    </div>
</form>
</div>
<footer>
    <h5>Website developed by Hemant Sundarrajan, 2023</h5>
</footer>
</body>
</html>
```

This page has a very simple functionality. It allows the user to register an account on the library website. It requires a certain set of information, and these sets of information have requirements. Firstly, the user must enter data into all fields. If the user doesn't enter data into all fields they cannot register (error message appears). As seen in the PHP code, the data gotten from the form is stored in variables. These variables are then checked if their requirements are met. A user must make a unique username, they cannot make a username which a previous user has already taken (already within the database). The password MUST be 6 characters long. There is a password confirmation field, this must match the first password entered. The mobile phone number, and telephone number must be 10 digits long. These requirements must be met. If even one requirement is not met, an error message will appear. For example, if a password is not met the error message "Password is not 6 characters long" will appear, and the user will not be registered. If all requirements are met, the user's information will be inserted into the database via the INSERT INTO SQL statement. The user will then be successfully registered and redirected to the login page. Once again, the error messages are handled using a SESSION variable "error" to remove redundant code.

How my Register page (register.php) looks in my website:

The screenshot displays a web browser window with the address bar showing 'localhost/WebD/Assignment/register.php'. The website's header is dark blue, featuring the 'Osprey Library' logo on the left and navigation links 'HOME', 'LOGIN', and 'REGISTER' on the right. The background of the page is a high-resolution image of a grand library interior with tall bookshelves and classical architectural details. Overlaid on this background is a white registration form titled 'Register'. The form contains the following fields:

- Username**: A text input field with the placeholder 'Enter Username'.
- Password**: A text input field with the placeholder 'Enter Password'.
- Password Confirmation**: A text input field with the placeholder 'Enter Password Confirmation'.
- First Name**: A text input field with the placeholder 'Enter First Name'.
- Surname**: A text input field with the placeholder 'Enter Surname'.
- Address Line 1**: A text input field.

At the bottom left of the page, a small footer reads 'Website developed by Hemant Sundarrajan, 2023'.

Fig 1: top half of my Register page, very similar to login page and simple in format

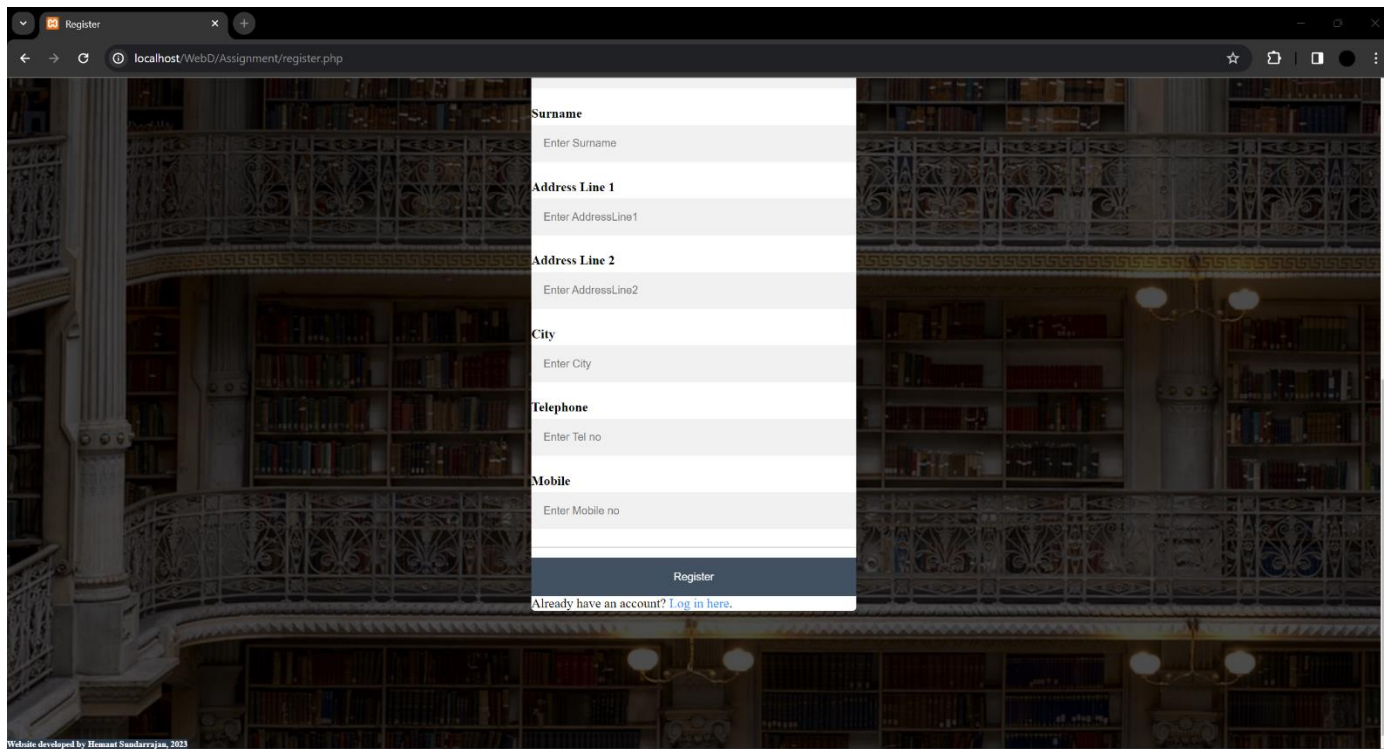


Fig 2: Bottom half of my Register page

How my Register Page (register.php) looks with an error message:

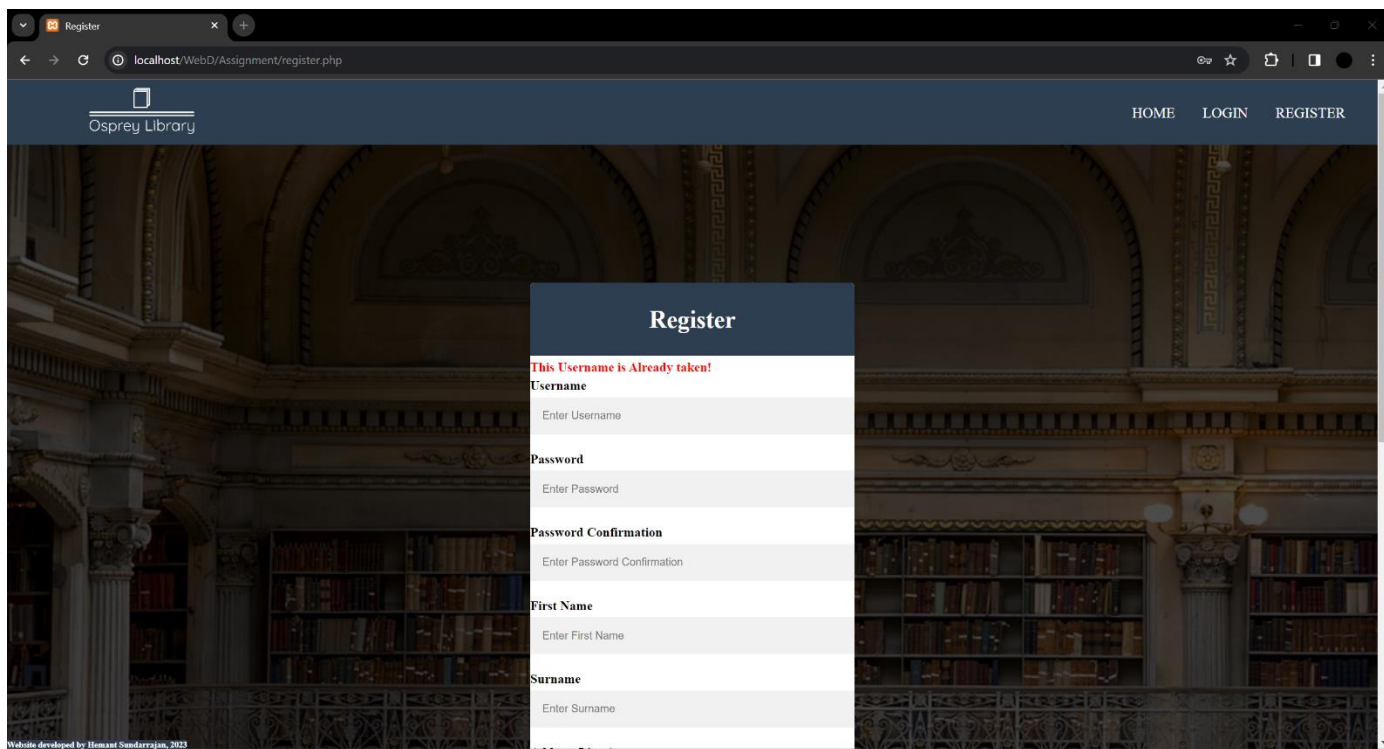


Fig 3: My error message on my register page, looks very similar to my login page, and is similar across my website to ensure consistency within my website.

7. search.php

```
<?php
//starts session, which tracks our users session throughout websites, if there
is or isnt one
session_start();

//this uses our databases php, to utilise code reuse, so we dont have
to manually write the connection to the database
require_once 'databases.php';

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://kit.fontawesome.com/c13e82a8d6"
crossorigin="anonymous"></script>
    <title>Search</title>
</head>
<body>
    <header>
        <nav>
            <input type="checkbox" id="check">
            <label for="check">
                <i class="fas fa-bars" id="btn"></i>
                <i class="fas fa-times" id="cancel"></i>
            </label>
            
            <ul class="bar">
                <li><a href="index.php">Home</a></li>
                <li><a href="search.php">Search For a Book</a></li>
                <li><a href="view.php">View Reserved Books</a></li>
                <li><a href="logout.php">Logout</a></li>
            </ul>
        </nav>
    </header>
    <?php
        //checks if user is logged in, if not forces them to, to access
        library services
        if (!isset($_SESSION["Username"]))
        {
            echo
```



```

        ' <br> <p>Please Log in to access all of our libraries features!
Use the Navbar to Login, or <a href="login.php">Log in here</a></p>
        <p>If you dont have an accout, please register and ceate one. You
can <a href="register.php">Register here</a></p>
        <p>Osprey Library is your one stop shop for E learning and E books
in an online Library!</p>';
        exit();
    }
?>
<div class="Regcontainer">

    <div class="Regwrapper">
        <form method="get" novalidate>
            <div class="title"><span>Search</span></div>

            <label for="FirstName"><b>Search By Book Title</b></label>
            <input class="reg" type="text" placeholder="Search By Book
Title" name="SearchName" id="SearchName">

            <label for="FirstName"><b>Search By Author</b></label>
            <input class="reg" type="text" placeholder="Search By Author"
name="SearchAuthor" id="SearchAuthor">

            <label for="FirstName"><b>Search By Category: </b></label>
            <select class="reg" name="SearchCategory">
                <option value="">Select Category</option>
                <?php
                $sql_category = "SELECT * FROM categories";
                $sql_category_result = $conn->query($sql_category);

                while ($categoryRow = $sql_category_result->fetch_assoc())
                {
                    echo "<option value='" .
htmlspecialchars($categoryRow["CategoryID"]) . "'>" .
htmlspecialchars($categoryRow["CategoryDescription"]) . "</option>";
                }
                ?>
            </select>
            <button type="submit" class="registerbtn">Search</button>
        </form>
    </div>
<?php

    if(isset($_GET['SearchName']) && isset($_GET['SearchAuthor']) &&
isset($_GET['SearchCategory']))
    {
        $SearchName = htmlentities($_GET["SearchName"]);
        $SearchAuthor = htmlentities($_GET["SearchAuthor"]);
    }

```

```

        $SearchCategory = htmlentities($_GET["SearchCategory"]);

        if (empty($SearchName) && empty($SearchAuthor) &&
empty($SearchCategory))
        {
            echo '<div class="Regwrapper">
            <p class="error-message">Error: please enter into at least
1 search field</p>
            </div>';
        }
        else
        {
            $sql = "SELECT books.*, categories.CategoryDescription
FROM books
LEFT JOIN reservations ON books.ISBN = reservations.ISBN
LEFT JOIN categories ON books.CategoryID =
categories.CategoryID
WHERE ";
            $sql_statements = [];

            if(!empty($SearchName))
            {
                $sql_statements[] = "books.BookTitle LIKE '%" . $conn-
>real_escape_string($SearchName) . "%'";
            }

            if(!empty($SearchAuthor))
            {
                $sql_statements[] = "books.Author LIKE '%" . $conn-
>real_escape_string($SearchAuthor) . "%'";
            }

            if (!empty($SearchCategory))
            {
                $sql_statements[] = "books.CategoryID = '" . $conn-
>real_escape_string($SearchCategory) . "'";
            }

            //implodes all sql statements in array with AND so that
they are all linked when searching
            $sql .= implode(" AND ", $sql_statements);

            //Gets page, or set default to 1
            $page = isset($_GET['page']) ? $_GET['page'] : 1;
            //max no of rows
            $PerPageResults = 5;

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

```

```

//sees how many results there is
$totalRes = $result->num_rows;
$totalPages = ceil($totalRes / $PerPageResults);

$offset = ($page - 1) * $PerPageResults;

//Limits the offset in SQL
$sql .= " LIMIT $offset, $PerPageResults";

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

if ($result->num_rows > 0)
{
    echo "<br> <br>";
    for ($i = 1; $i <= $totalPages; $i++) {
        echo "Page: ";
        echo "<a
href='search.php?page=$i&SearchName=$SearchName&SearchAuthor=$SearchAuthor&SearchCategory=$SearchCategory'>$i</a> ";
    }
    echo "<table border='1'>";
    echo "<tr>
        <th>Book Title</th>
        <th>Author</th>
        <th>Edition</th>
        <th>Year</th>
        <th>Category</th>
        <th>Reserve Status</th>
    </tr>";
    while ($row = $result->fetch_assoc())
    {
        echo "<tr>
        <td>" . htmlspecialchars($row["BookTitle"]) .
        "</td>
        <td>" . htmlspecialchars($row["Author"]) . "</td>
        <td>" . htmlspecialchars($row["Edition"]) . "</td>
        <td>" . htmlspecialchars($row["Year"]) . "</td>
        <td>" .
        htmlspecialchars($row["CategoryDescription"]) . "</td>
        <td>";
        if ($row["Reserved"] === 'N')
        {
            echo "<a href='reserve.php?id=" .
            htmlspecialchars($row["ISBN"]) . "'>Available</a>";
        }
        else if($row["Reserved"] === 'Y')
        {

```

```

        echo "Reserved";
    }
    echo "</td>
    </tr>";
}
echo "</table>";
}
else
{
    echo '<div class="Regwrapper">
    <p class="error-message">No Search Results found</p>
    </div>';
}

}
$conn->close();

}

?>
</div>
<footer>
    <h5>Website developed by Hemant Sundarrajan, 2023</h5>
</footer>

</body>
</html>

```

The search.php page is by far the most complicated page in my website. The search.php page has 3 fields that can be filled out. To search for a book its name, author, or category. The search form is extremely like the login and registration form. I wanted to keep this consistency to keep the website simple. This is a page you can only access if you have logged in, so you must be logged in to use the search bars. If you are not logged in and try to bypass it through the URL, you will just be met with a prompt to login. If you are logged in, you can now search.

The way I made the search bar work, was by utilising an array for my SQL statements. I firstly used an SQL statement which left joined my book table to my reservations, and categories tables. This way I was able to get all the data from these tables they have in common, as

well as all the data from the books table. I then used an array of SQL statements, which each had a particular query depending on which search type you used. I then utilised the IMPLode SQL command which allowed me to use AND on all the SQL statements. This is what allowed me to achieve the requirement of search by book name or author, or partial search on both. The only error messages you can get on this page is if you don't choose to use any field, i.e. you leave the search bars empty or if there are no search results entered.

After you search, a table of results shows up. This table shows the Book Title, Author, Edition, Year, Category and Reserve Status. This is done by simply checking if there is more than 1 row of data in the database (using the \$result variable) and then loops through each row with a while loop.

The next requirement was that each page of search results can only be 5 rows long. So, if there are 6 rows of books, there needs to be a second page. To do this I used Pagination. I specified how many total rows a page could have, I added to my SQL query a LIMIT \$offset statement by how many rows I wanted (the offset was calculated by taking my current page number, minus it by 1 and then multiply it by the \$PerPageResults (5). I then added this to my \$result variable through the conn->query(\$sql), and then in a for loop, I simply echo'd the number of page results as a link. The key to making this work was using GET instead of POST as this makes Pagination work a lot better. After this I was able to get search results with pages to go back and forth if there were more than 5 rows of results.

How my search page (search.php) looks in my website:

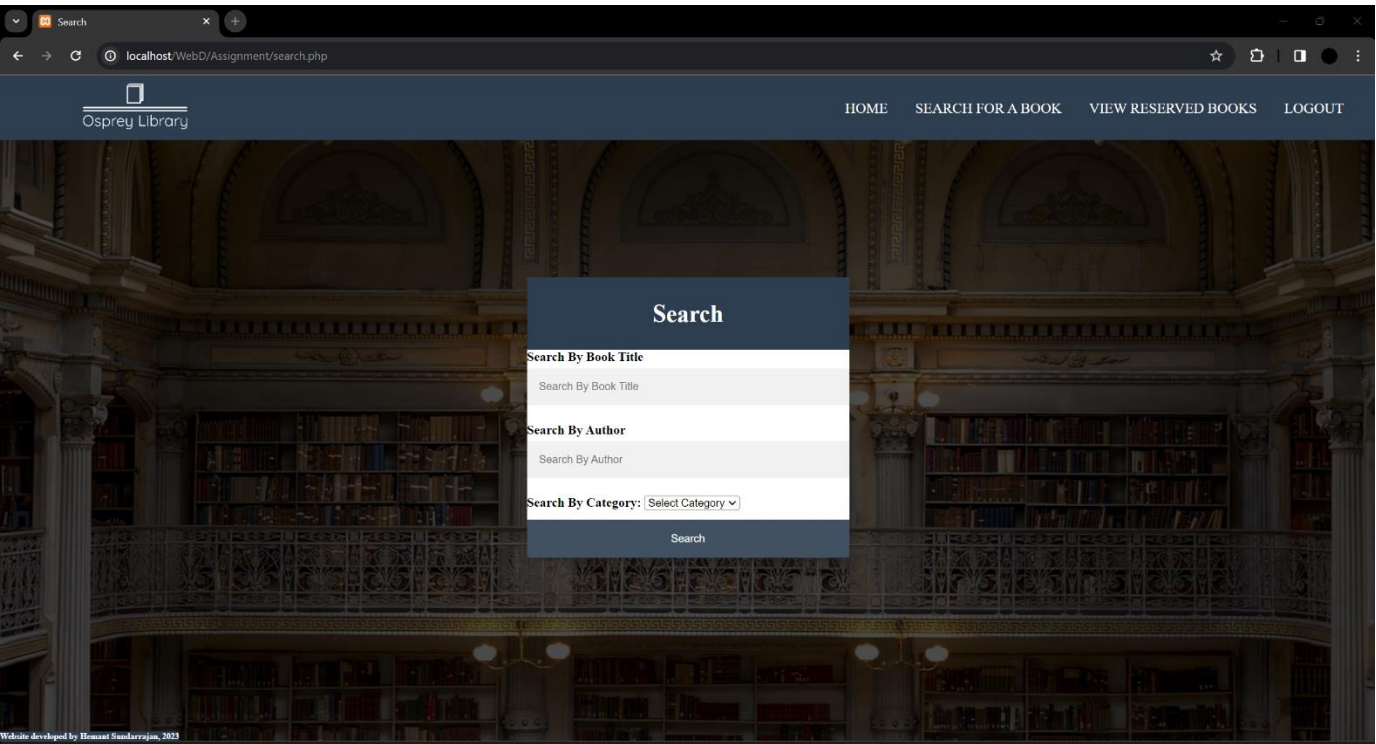


Fig 1: How my search page looks before searching.

How my search page (search.php) looks after searching for a book:

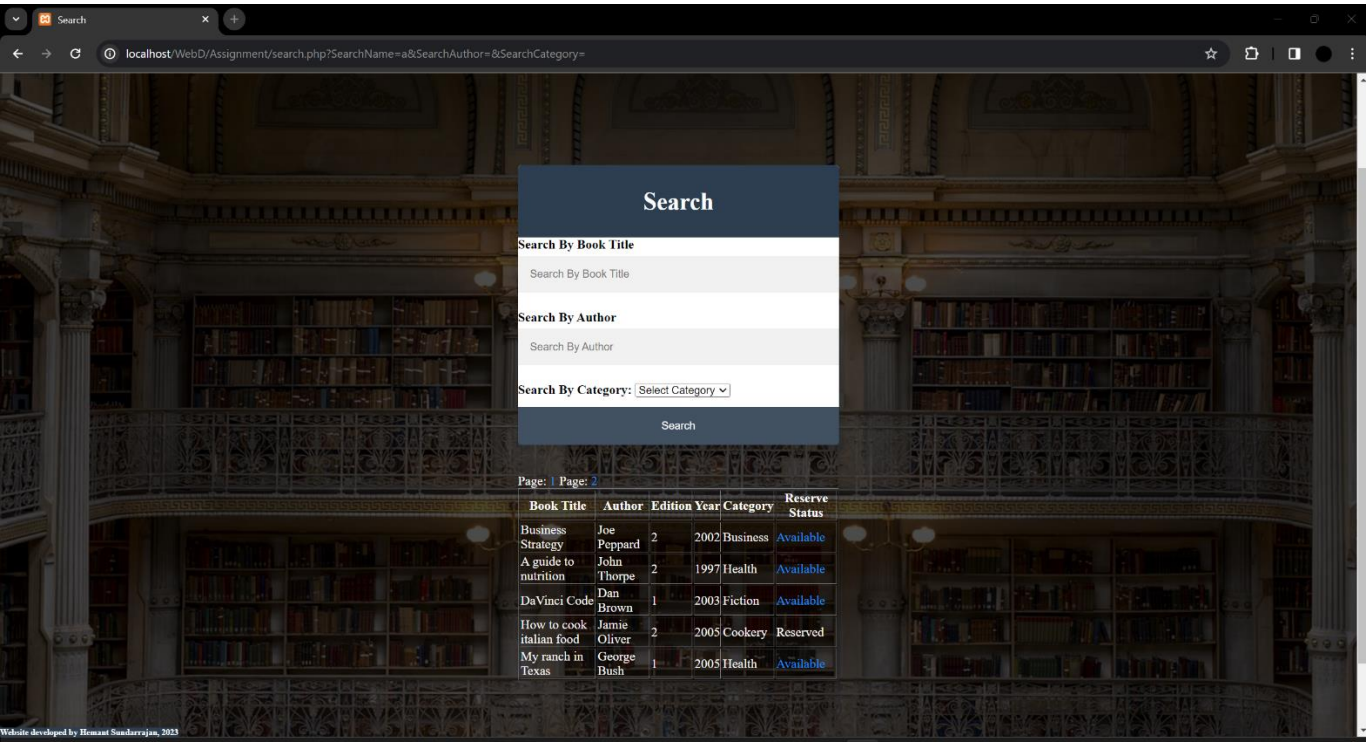


Fig 2: How my search page looks after searching for a book, it's on page 1 of the search results.

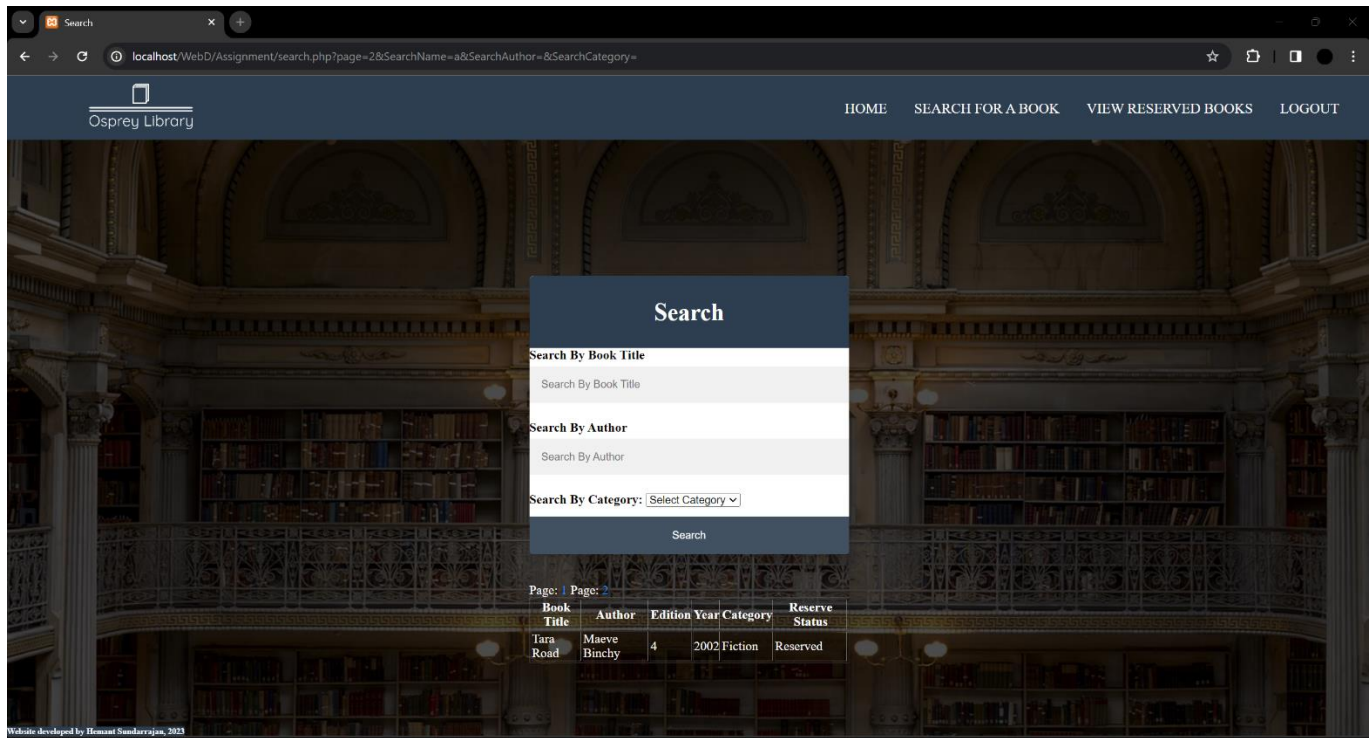


Fig 3: How my search page looks after searching for a book, it's on page 2 of the search results.

8. reserve.php

```
<?php
//starts session, which tracks our users session throughout websites, if there
is or isnt one
session_start();

//this uses our databases php, to utilise code reuse, so we dont have
to manually write the connection to the database
require_once 'databases.php';

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://kit.fontawesome.com/c13e82a8d6"
crossorigin="anonymous"></script>
    <title>Reserve Books</title>
</head>
<body>
    <header>
        <nav>
            <input type="checkbox" id="check">
            <label for="check">
                <i class="fas fa-bars" id="btn"></i>
                <i class="fas fa-times" id="cancel"></i>
            </label>
            
            <ul class="bar">
                <li><a href="index.php">Home</a></li>
                <li><a href="search.php">Search for a Book</a></li>
                <li><a href="view.php">View Reserved Books</a></li>
                <li><a href="logout.php">Logout</a></li>
            </ul>
        </nav>
    </header>
    <?php
        //checks if user is logged in, if not forces them to, to access library
        services
        if (!isset($_SESSION["Username"]))
        {
            echo
```



```

        ' <br> <p>Please Log in to access all of our libraries features! Use
the Navbar to Login, or <a href="login.php">Log in here</a></p>
        <p>If you dont have an accout, please register and ceate one. You can
<a href="register.php">Register here</a></p>
        <p>Osprey Library is your one stop shop for E learning and E books in
an online Library!</p>';
        exit();
    }
    ?>
<div class="Regcontainer2">
<?php
    $loggedInUsername = $_SESSION['Username'];

    if ( isset($_POST['reserve']) && isset($_POST['id']))
    {
        //this sql check is used to ensure a book its reserved and logged
multiple times in the database
        $id = $conn->real_escape_string($_POST['id']);
        $reservationCheck = "SELECT * FROM reservations WHERE ISBN='$id'
AND Username='$loggedInUsername'";
        $resultCheck = $conn->query($reservationCheck);

        if ($resultCheck->num_rows > 0) {
            //this makes it so that if a user tries to go back to the
previous page and reserve the book again it just sends them back to the view
page
            header('Location: view.php');
            exit();
        }
        //updates reservation status, so database knows book is reserved
and by who
        $id = $conn -> real_escape_string($_POST['id']);
        $sql = "UPDATE books SET Reserved = 'Y' WHERE ISBN = '$id'";
        $sql2 = "INSERT INTO reservations(ISBN, Username, ReservedDate)
VALUES('$id', '$loggedInUsername', curdate())";
        $conn->query($sql2);
        $conn->query($sql);
        //sends user to view.php
        header('Location: view.php');
        exit();
        echo '<footer>
            <h5>Website developed by Hemant Sundarrajan, 2023</h5>
        </footer>';
        return;
    }
    //confirms the reservation of a book, which id was gotten from search
page
    $id = $conn -> real_escape_string($_GET['id']);

```

```

        $sql = "SELECT * FROM books WHERE ISBN='$id'";
        $result = $conn->query($sql);
        $row = $result->fetch_assoc();
        echo "<h3>Confirm The Reservation of: ". $row["BookTitle"] . "</h3>\n";
        echo('<form method ="post"><input type="hidden" ');
        echo('name="id" value="'.htmlentities($row["ISBN"]).'">'. "\n");
        echo('<input type="submit" value="Reserve" name="reserve">');
        echo('<a href="search.php"> Cancel</a>');
        echo("\n</form>\n");

        $conn->close();

    ?>
</div>
<footer>
    <h5>Website developed by Hemant Sundarrajan, 2023</h5>
</footer>
</body>
</html>

```

The reserve page is a page you can only use if you are logged in. You can reserve a book by searching for one, from the search.php page.

If you choose to reserve a book it redirects you to the reserve.php page. Here you can confirm the reservation of your book or cancel the reservation. This is done to ensure the user doesn't reserve a book by accident. This is done through SQL by getting the ISBN from the search.php page, reserving it under the current users Username in the reservations table, and then setting the Reservation status to 'Y' from 'N' (to show users when they search for a book if it is already reserved). If you press cancel, the user is taken back to the search page. If you confirm the reservation the user is taken the view.php page to view their new reservation and all their reservations.

How you can reserve a book (search.php) looks like this: (from part 7. search.php)

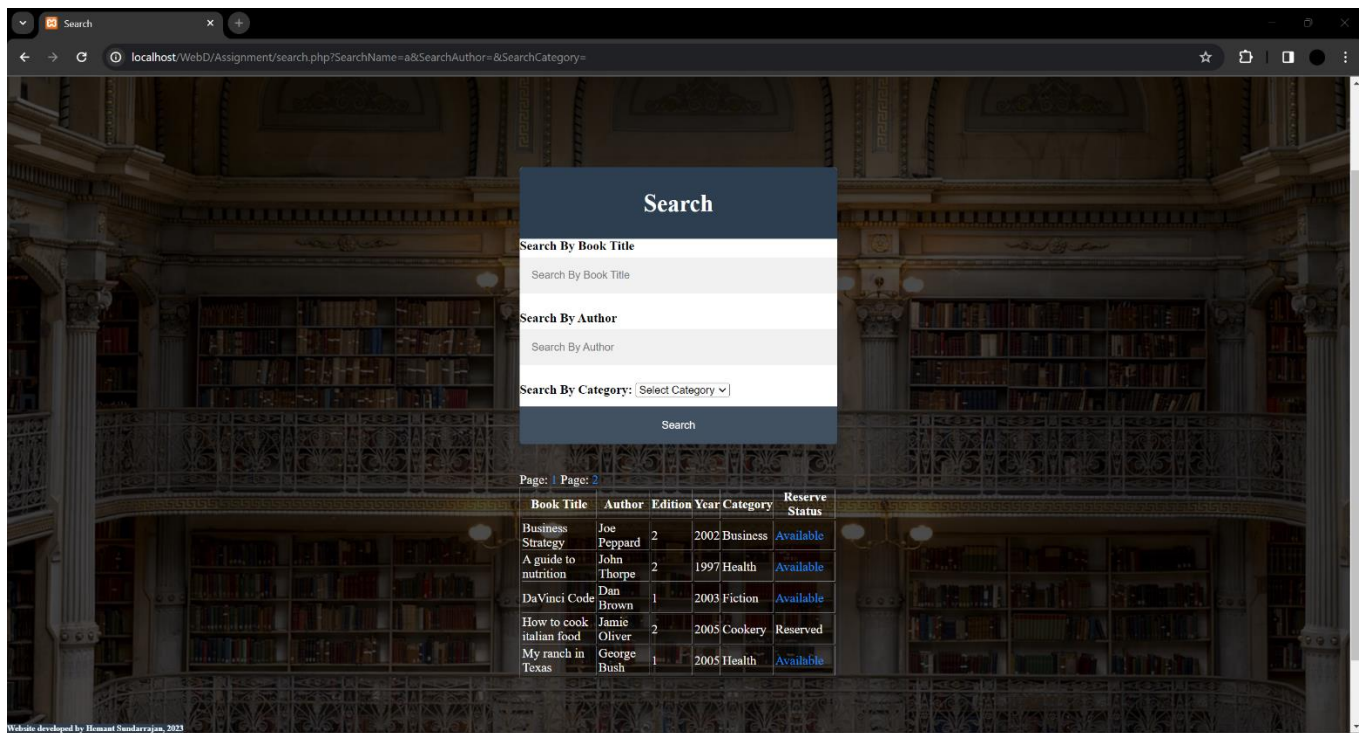


Fig 1: You can reserve a book by clicking on the “Available” links in the reserve status column. A book is already reserved if it says “Reserved”.

How you can confirm the reservation (reserve.php) of a book or cancel the reservation in my website looks like this:

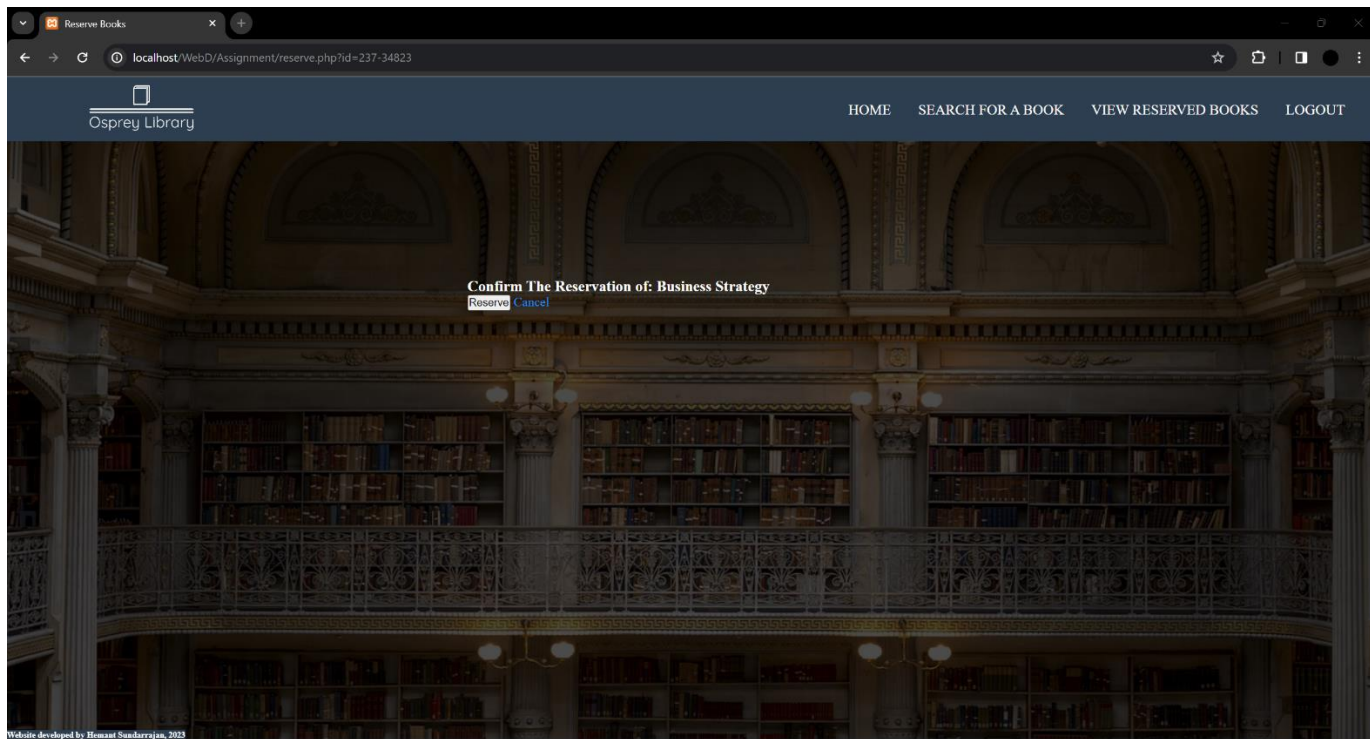


Fig 2: Here on reserve.php you can see the user has the option to reserve or cancel the book reservation. Once the user confirms their book reservation, they are taken to view.php to see their reservations.

9. view.php

```
<?php
//starts session, which tracks our users session throughout websites, if there
is or isnt one
session_start();

//this uses our databases php, to utilise code reuse, so we dont have
to manually write the connection to the database
require_once 'databases.php';

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://kit.fontawesome.com/c13e82a8d6"
crossorigin="anonymous"></script>
    <title>View Reserved Books</title>
</head>
<body>
    <header>
        <nav>
            <input type="checkbox" id="check">
            <label for="check">
                <i class="fas fa-bars" id="btn"></i>
                <i class="fas fa-times" id="cancel"></i>
            </label>
            
            <ul class="bar">
                <li><a href="index.php">Home</a></li>
                <li><a href="search.php">Search for a Book</a></li>
                <li><a href="view.php">View Reserved Books</a></li>
                <li><a href="logout.php">Logout</a></li>
            </ul>
        </nav>
    </header>
    <?php
        //checks if user is logged in, if not forces them to, to access
        library services
        if (!isset($_SESSION["Username"]))
        {
            echo
```

```

        ' <br> <p>Please Log in to access all of our libraries features!
Use the Navbar to Login, or <a href="login.php">Log in here</a></p>
        <p>If you dont have an accout, please register and ceate one. You
can <a href="register.php">Register here</a></p>
        <p>Osprey Library is your one stop shop for E learning and E books
in an online Library!</p>';
        exit();
    }
?>
<div class="Regcontainer2">
    <?php

        $loggedInUsername = $_SESSION['Username'];

        $sql = "SELECT
books.ISBN,
books.BookTitle,
books.Author,
books.Edition,
books.Year,
reservations.ReservedDate
FROM
        reservations
JOIN
        books ON reservations.ISBN = books.ISBN
WHERE
        reservations.Username = '$loggedInUsername' AND books.Reserved
= 'Y'";

        $page = isset($_GET['page']) ? $_GET['page'] : 1; // Get the
current page or default to 1
        $PerPageResults = 5; // Number of results per page

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

        $totalRes = $result->num_rows;
        $totalPages = ceil($totalRes / $PerPageResults);

        $offset = ($page - 1) * $PerPageResults;

        $sql .= " LIMIT $offset, $PerPageResults";

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

        //if books exits, in a table the books are shown
        if ($result->num_rows > 0)
        {

```

```

        echo "<h3>Hey there $loggedInUsername!, These are the books
you have reserved:</h3>";
        echo "<br> <br>";
        echo "<table border='1'>";
        echo "<tr>
            <th>Book Title</th>
            <th>Author</th>
            <th>Edition</th>
            <th>Year</th>
            <th>Date of Reservation</th>
            <th>Remove Reservation</th>
        </tr>";
        while ($row = $result->fetch_assoc())
        {
            echo "<tr>
                <td>" . htmlspecialchars($row["BookTitle"]) . "</td>
                <td>" . htmlspecialchars($row["Author"]) . "</td>
                <td>" . htmlspecialchars($row["Edition"]) . "</td>
                <td>" . htmlspecialchars($row["Year"]) . "</td>
                <td>" . htmlspecialchars($row["ReservedDate"]) . "</td>
                <td>
                    <a href='remove.php?id=" . htmlspecialchars($row["ISBN"])
. "'>Remove</a>
                </td>
            </tr>";

        }
    }
    else
    {
        echo "<h1>Hey there $loggedInUsername!, You currently have no
reserved books.";
    }
    //shows pages, which increments depending on the rows (pagination)
    for ($i = 1; $i <= $totalPages; $i++)
    {
        echo "Page: ";
        echo "<a href='view.php?page=$i'>$i</a> ";
    }

    $conn->close();
    ?>
</div>
<footer>
    <h5>Website developed by Hemant Sundarrajan, 2023</h5>
</footer>

</body>

```

```
</html>
```

The view.php page is very simple. This is where we see the books the user has reserved. It queries the database and retrieves the users' reservations in rows. It works very similar to the search.php output, however the code itself is much simpler and simply requires an SQL statement which fetches the necessary book information (Book Title, Author etc) from the database. It uses a Join statement to get data from the reservation table, and it uses the \$loggedInUsername variable in this statement to get the users' reservation data. Like the search.php page, using pagination, I was able to make it so you can only see 5 rows of reservations per page, and then needed to click to the next page to get the next rows of data. The view.php page is also used to remove a users' reservation. There is a link to the remove.php page in that book's reservation row, where they can remove that book. The date the book was reserved on is also accurately shown, if I reserve the book today, today's date will be shown in the "Date of Reservation" column.

How you can view a user's book reservations (view.php) looks like this:

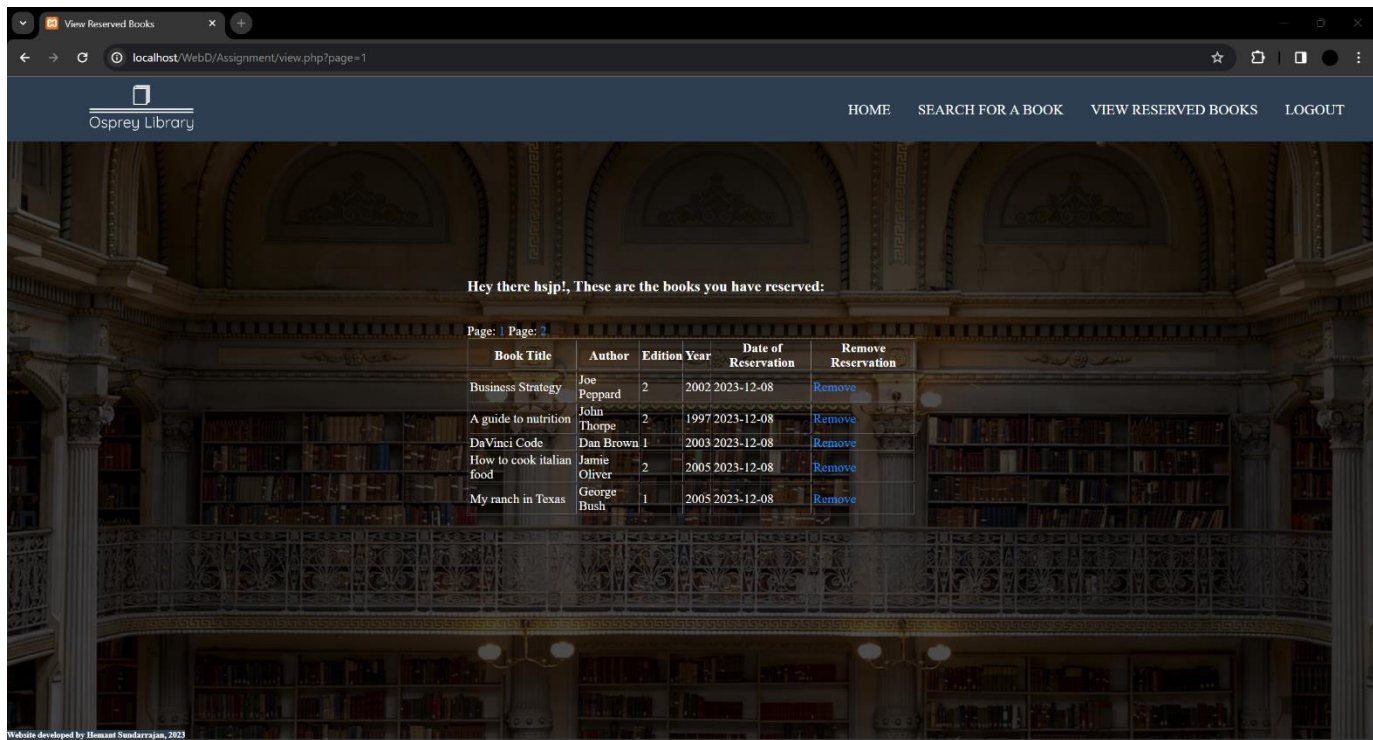


Fig 1: This is the first page of the results, just like search.php if there are more than 5 rows, there is another page pf results.

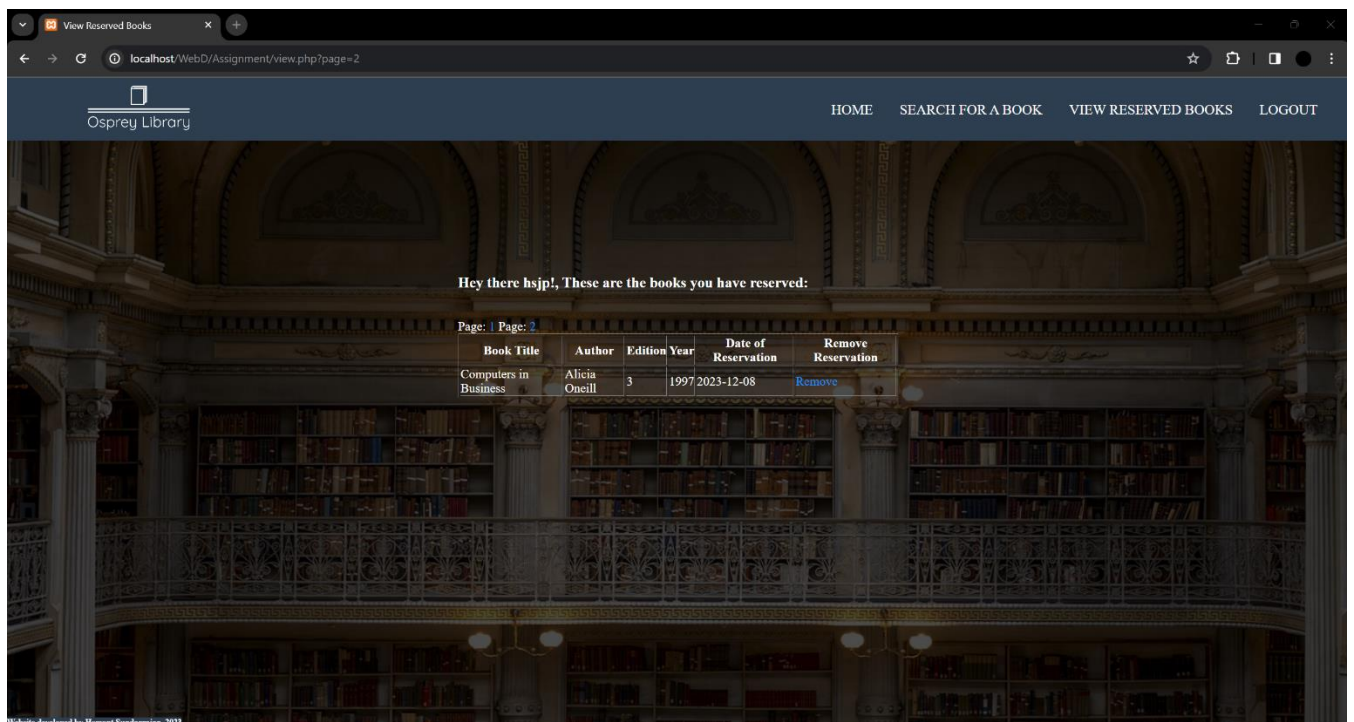


Fig 2: This is how the second page would look like if there was 6 reserved books for example.

10.remove.php

```
<?php
//starts session, which tracks our users session throughout websites, if there
is or isnt one
session_start();

//this uses our databases php, to utilise code reuse, so we dont have
to manually write the connection to the database
require_once 'databases.php';

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://kit.fontawesome.com/c13e82a8d6"
crossorigin="anonymous"></script>
    <title>Remove Reservation</title>
</head>
<body>
    <header>
        <nav>
            <input type="checkbox" id="check">
            <label for="check">
                <i class="fas fa-bars" id="btn"></i>
                <i class="fas fa-times" id="cancel"></i>
            </label>
            
            <ul class="bar">
                <li><a href="index.php">Home</a></li>
                <li><a href="search.php">Search for a Book</a></li>
                <li><a href="view.php">View Reserved Books</a></li>
                <li><a href="logout.php">Logout</a></li>
            </ul>
        </nav>
    </header>
    <?php
        //checks if user is logged in, if not forces them to, to access
        library services
        if (!isset($_SESSION["Username"])) {
            echo
```

```

        ' <br> <p>Please Log in to access all of our libraries features!
Use the Navbar to Login, or <a href="login.php">Log in here</a></p>
        <p>If you dont have an accout, please register and ceate one. You
can <a href="register.php">Register here</a></p>
        <p>Osprey Library is your one stop shop for E learning and E books
in an online Library!</p>';
        exit();
    }
?>
<div class="Regcontainer2">
    <?php
        if ( isset($_POST['remove']) && isset($_POST['id']))
        {
            //Removes reservation, so that book is now reservable again,
and no longer reserved by user
            $id = $conn -> real_escape_string($_POST['id']);
            $sql = "UPDATE books SET Reserved = 'N' WHERE ISBN = '$id'";
            $sql2 = "DELETE FROM reservations WHERE ISBN = '$id'";
            $conn->query($sql2);
            $conn->query($sql);
            echo '<h2>Reservation was Removed Successfully!</h2>';
            echo '<br> <br>';
            echo '<h4>Click here to view your Updated Reserved Books - <a
href="view.php">Reserved Books</a></h4>';
            echo '<br> <br>';
            echo '<h4>Click here to Search for New Books to reserve - <a
href="search.php">Search</a></h4>';
            echo '<footer>
                <h5>Website developed by Hemant Sundarrajan, 2023</h5>
            </footer>';
            return;
        }
        //Removal confirmation, so user doesnt accidnetally remove
reservation
        $id = $conn -> real_escape_string($_GET['id']);
        $sql = "SELECT * FROM books WHERE ISBN='$id'";
        $result = $conn->query($sql);
        $row = $result->fetch_assoc();
        echo "<h3>Confirm The Reservation Removal of: ". $row["BookTitle"]
        . "</h3>\n";
        echo('<form method ="post"><input type="hidden" ');
        echo('name="id" value="'. htmlentities($row["ISBN"]) . '>'. "\n");
        echo('<input type="submit" value="Remove" name="remove">');
        echo('<a href="view.php"> Cancel</a>');
        echo("\n</form>\n");

        $conn->close();
    ?>

```

```
</div>
<footer>
  <h5>Website developed by Hemant Sundarrajan, 2023</h5>
</footer>

</body>
</html>
```

The remove.php page is accessed through the view.php page. In the view.php page, you can click remove reservation beside a book. This will then take you to the remove.php page. On this page you can confirm the reservation removal or cancel it (in case a user accidentally pressed remove reservation). The ISBN is passed, and can be seen in the URL, this is how we delete a specific book from the users' reservations. So, if the user decided to remove the reservation, by using PHP and SQL, I took the ISBN, used my SQL query to delete it from the reservations table (so the reservation no longer exists) and then updated its reservation status from 'Y' to 'N' (so that other users could see the book is now available for reservation). Once a book is successfully removed from reservation, a confirmation page loads, where the user can now choose to view their reservations (on the view.php page) or search for new books to reserve (on the search.php page).

How you can view a user's book reservations so you can remove the book reservation (from 9. view.php) looks like this:

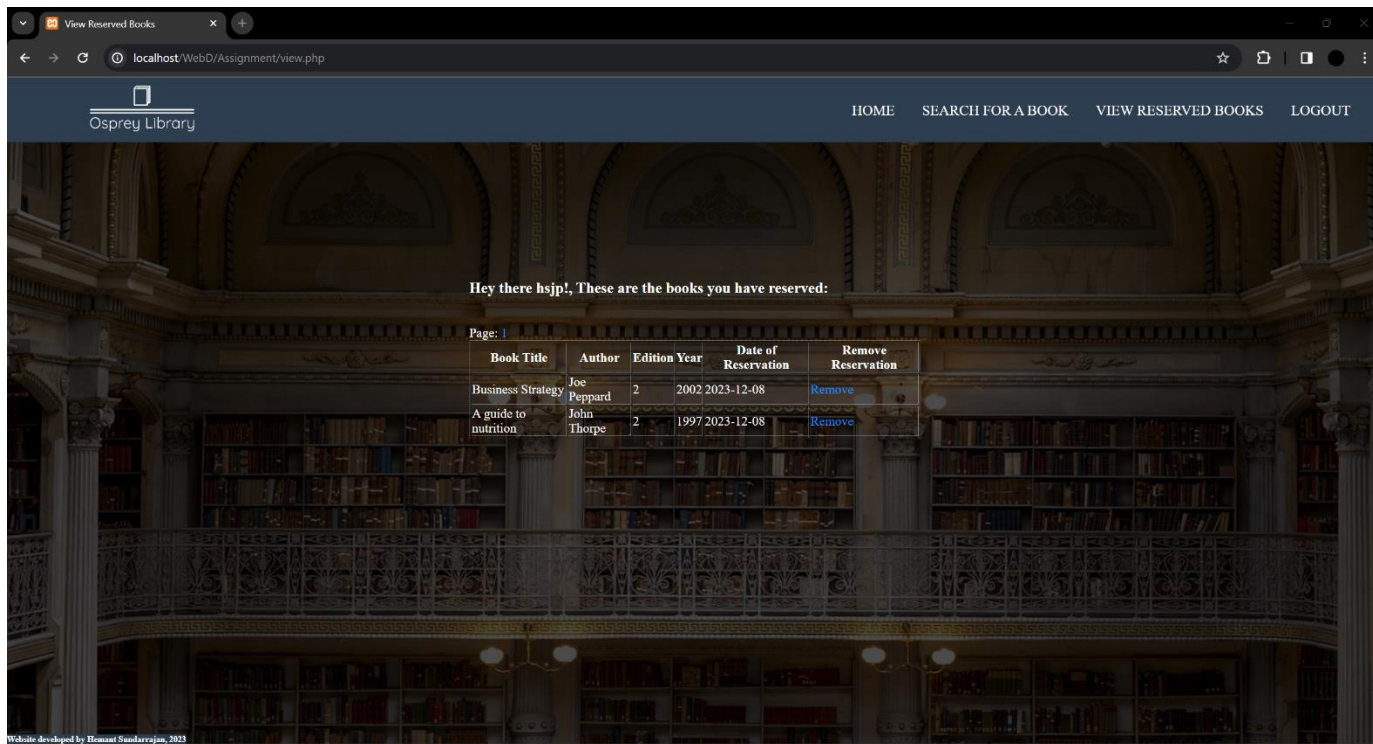


Fig 1: As you can see, there is a remove reservation column on view.php, where you can choose to remove a book reservation.

How you can confirm a reservation removal or cancel it (remove.php):

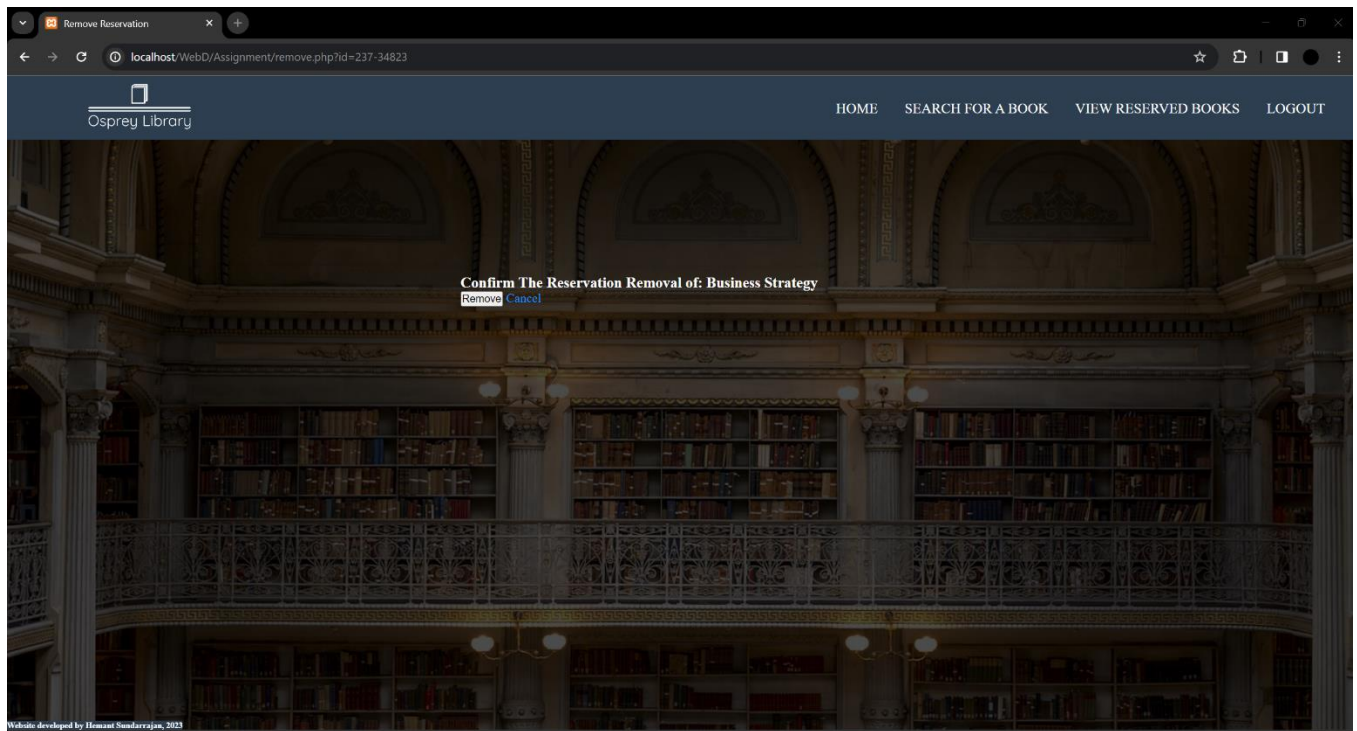


Fig 2: This is what it looks like when you press “Remove” beside a book in the view.php page.

How it looks on my website when a reservation is removed successfully (reserve.php):

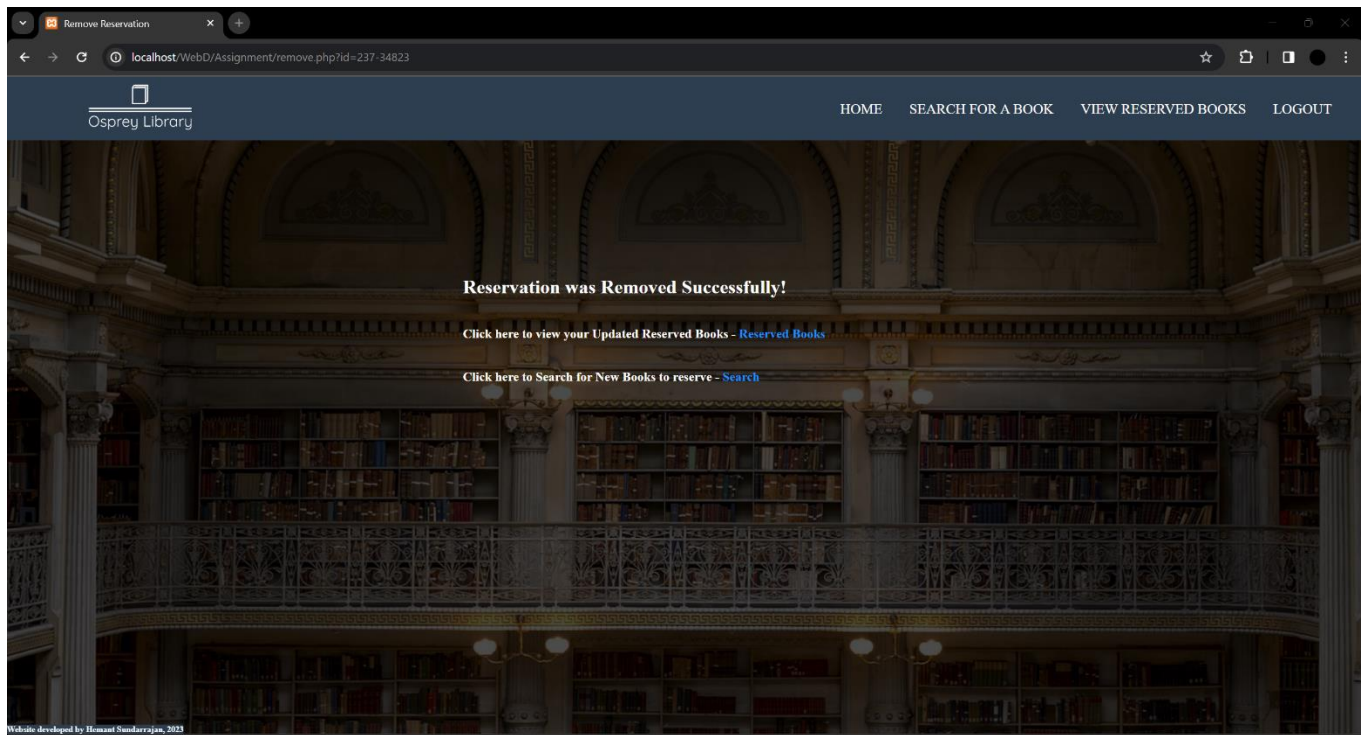
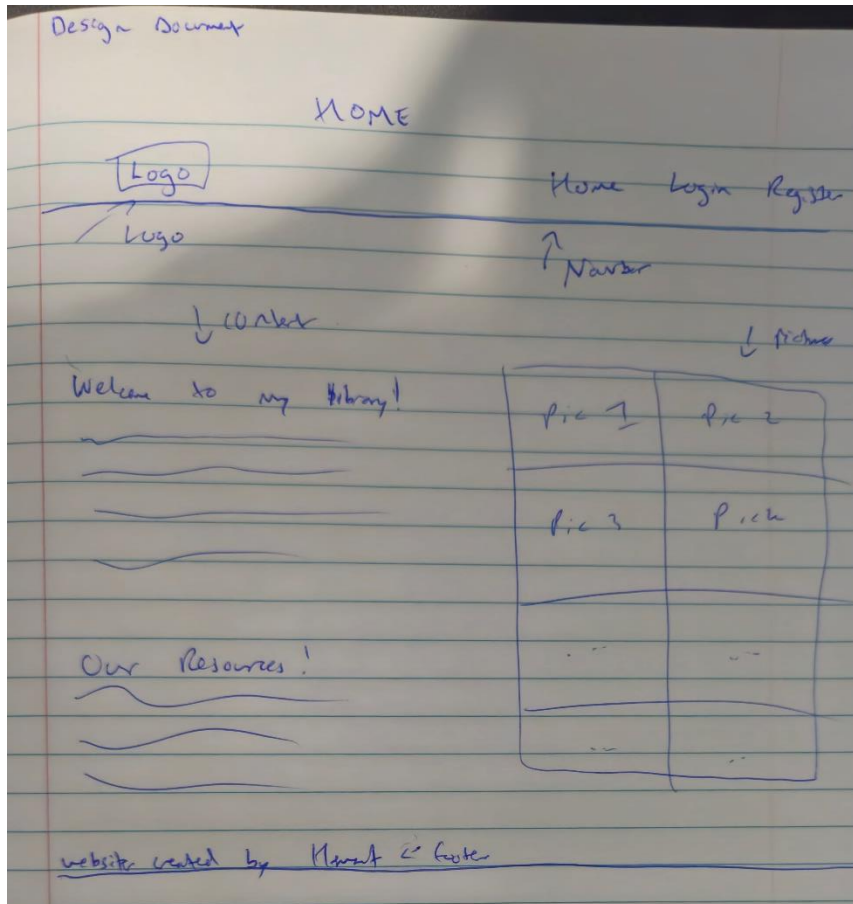


Fig 3: As you can see, you get a success message, and are prompted to search for new books to reserve or go view your reserved books after you removed a reservation.

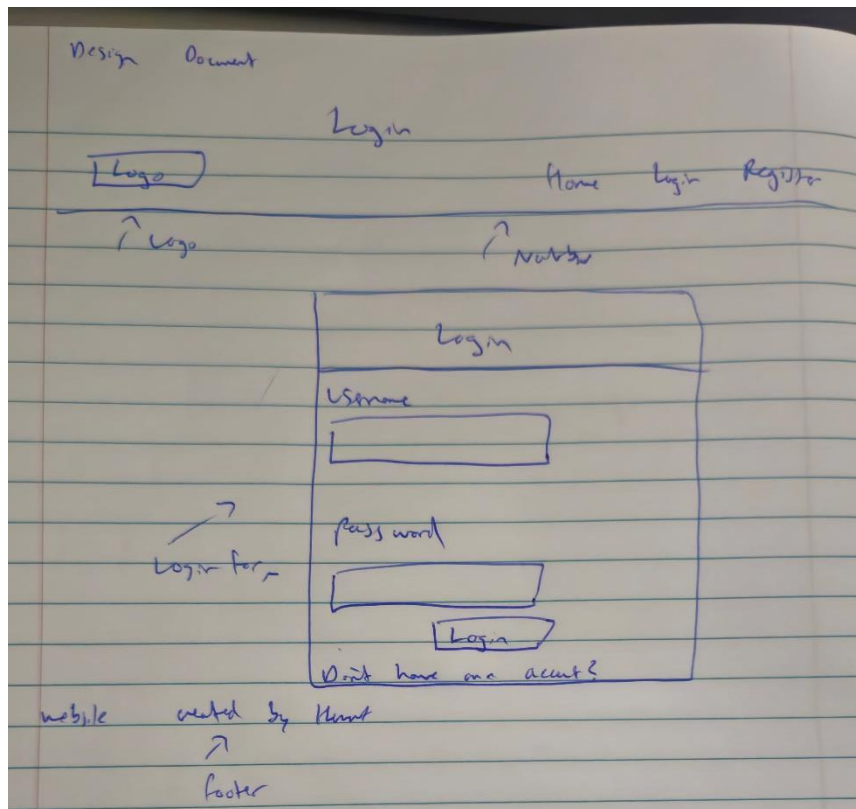
Design Document

Just a simple Wire Frame design of what I thought my website might look like:

Example of my home page:



Example of my Login Page:



My CSS code: (style.css)

```
*{
  margin: 0;
  padding: 0;
  list-style: none;
  text-decoration: none;
  box-sizing: border-box;
}
body{
  background-image: linear-gradient(rgba(0,0,0,0.7),rgba(0,0,0,0.7)),
url(background-img.jpg);
  background-size: cover;
  image-rendering: -webkit-optimize-contrast;
  color: white;
  background-size: cover;
  width: 100%;
  min-height: 100vh;
  image-rendering: -webkit-optimize-contrast;
  position: relative;
  overflow-y: auto;
}
nav{
  height: 80px;
  background: #2c3e50;
}
nav .imgcl1{
  width: 130px;
  position: absolute;
  top: 10px;
  left: 6%;
}
nav .bar{
  float: right;
  margin-right: 25px;
}
nav .bar li{
  display: inline-block;
  line-height: 80px;
  margin: 0 15px;
}
nav .bar li a{
  position: relative;
  color: white;
  font-size: 18px;
  padding: 5px 0;
  text-transform: uppercase;
}
```

```
nav .bar li a:before{
  position: absolute;
  content: '';
  left: 0;
  bottom: 0;
  height: 3px;
  width: 100%;
  background: white;
  transform: scaleX(0);
  transform-origin: right;
  transition: transform .2s linear;
}
nav .bar li a:hover:before{
  transform: scaleX(1);
  transform-origin: left;
}

label #btn,
label #cancel{
  color:white;
  font-size: 30px;
  float: right;
  line-height: 80px;
  margin-right: 40px;
  cursor: pointer;
  display: none;
}
#check{
  display: none;
}
@media (max-width: 1118px) {
  nav img{
    left: 8%;
  }
}

@media (max-width: 944px) {
  nav img{
    left: 4%;
    top: 5px;
    width: 110px;
  }
  nav .bar li a{
    font-size: 17px;
  }
}

@media (max-width: 860px) {
  label #btn{
```

```

        display: block;
    }
    .bar{
        position: fixed;
        width: 100%;
        height: 100vh;
        background: #34495e;
        top: 80px;
        left: -100%;
        text-align: center;
        transition: all .5s;
    }
    nav .bar li{
        display: block;
        margin: 50px 0;
        line-height: 30px;
    }
    nav .bar li a{
        font-size: 20px;
    }
    #check:checked ~ ul{
        left: 0;
    }
    #check:checked ~ label #btn{
        display: none;
    }
    #check:checked ~ label #cancel{
        display: block;
    }
}
::selection{
    background: rgba(26,188,156,0.3);
}

.Regcontainer{
    max-width: 440px;
    padding: 0 20px;
    margin: 170px auto;
}
.Regcontainer2{
    max-width: 600px;
    padding: 0 20px;
    margin: 170px auto;
}
.Regwrapper{
    width: 100%;
    background: #fff;
    border-radius: 5px;

```

```
    box-shadow: 0px 4px 10px 1px rgba(0,0,0,0.1);
}
.Regwrapper .title{
    height: 90px;
    background: #2c3e50;
    border-radius: 5px 5px 0 0;
    color: #fff;
    font-size: 30px;
    font-weight: 600;
    display: flex;
    align-items: center;
    justify-content: center;
}

.Regwrapper label b{
    color: black;
}

.Regwrapper p{
    color: black;
}

input.reg, input.reg {
    width: 100%;
    padding: 15px;
    margin: 5px 0 22px 0;
    display: inline-block;
    border: none;
    background: #f1f1f1;
}

input.reg:focus, input.reg:focus {
    background-color: #ddd;
    outline: none;
}

.registerbtn {
    background-color: #2c3e50;
    color: white;
    padding: 16px 20px;
    margin-top: 12px;
    border: none;
    cursor: pointer;
    width: 100%;
    opacity: 0.9;
}
```

```
}

.registerbtn:hover {
opacity: 1;
}

a {
color: dodgerblue;
}

footer h5{
    text-align: center;
    background-color: #2c3e50;
    padding: 0px;
    color: white;
    bottom: 0;
    left: 0;
    position: fixed;
    font-size: 10.5px;
}

.container {
    padding-top: 80px;
    width: 100%;
    display: flex;
}

.left {
flex: 50%;
margin-left: 30px;
}

.left p{
    max-width: 600px;
}

.right {
flex: 50%;
}

.container .right .fourbyfour{
    width: 600px;
    height: 600px;
    margin: 0 auto;
}

.fourbyfour img {
```

```
border: 2px solid black;
width: 45%;
height: 45%;
}

.Regwrapper .error-message {
  color: red;
  font-weight: bold;
  padding-top: 2px;
  padding-bottom: 2px;
  padding-left: 2px;
}

.Regwrapper .error-messagereg {
  color: red;
  font-weight: bold;
  padding-top: 5px;
  padding-bottom: 5px;
}
```