|  |
| --- |
| DT228/2 |
| Web Development II |
| Assignment 2015 - 2016 |

|  |
| --- |
| Vlad Zat |

Contents

[PHP Pages 3](#_Toc436938901)

[Source Code 4](#_Toc436938902)

[Login.php 4](#_Toc436938903)

[Register.php 6](#_Toc436938904)

[Books.php 9](#_Toc436938905)

[Search.php 10](#_Toc436938906)

[Reservations.php 13](#_Toc436938907)

[Functions.php 15](#_Toc436938908)

[db.php 17](#_Toc436938909)

[Logout.php 17](#_Toc436938910)

[Username.php 17](#_Toc436938911)

[ReserveBook.php 17](#_Toc436938912)

[Return.php 18](#_Toc436938913)

[Header.html 19](#_Toc436938914)

[Footer.html 19](#_Toc436938915)

[AccessDenied.html 19](#_Toc436938916)

[mystyle.css 19](#_Toc436938917)

[script.js 24](#_Toc436938918)

[Screen Samples 25](#_Toc436938919)

[Login Page 25](#_Toc436938920)

[Register 25](#_Toc436938921)

[Login and Register 26](#_Toc436938922)

[Books 26](#_Toc436938923)

[Search 27](#_Toc436938924)

[Reservations 27](#_Toc436938925)

[Books, Search and Reservations 28](#_Toc436938926)

[Design 28](#_Toc436938927)

# PHP Pages

* **Login.php** – This is the first page of the application. This is where the user can login. This is an independent page but it assigns the username that is used throughout the application.
* **Register.php** – This is the page where a guest can create an account. It also verifies the information entered in the form.
* **Books.php** – In this page all the books can be viewed. There are 5 books displayed per page. It shows the book title, author, category, publication date and edition. In addition it also contains a button on the right where the user can reserve the book.
* **Search.php** – This page works similarly to the Books page with the addition that the user can choose which books to display. It can search by book title and/or author and it can also search by book category.
* **Reservations.php** – In this page the reserved books of the currently logged in user are displayed in a similar fashion as in the Books page.
* **Functions.php** – This file is used to store functions that are used throughout the pages. It contains a function to read the books that adhere to some condition, a function to display the books read and a function to return the page number.
* **db.php** – This file is used to open the database. It is used in all the pages of the application.
* **Logout.php** – This php page destroys the session and redirects the user to the Login page. It is used on every page as a Logout button when a user is Logged in.
* **Username.php** – This page is included in the 3 pages that display data to show the username and a logout button just under the header. It’s used in the Books, Search and Reservations pages to show the current user.
* **ReserveBook.php** – This page is accessed when a book is reserved in either the Books page or in the Search page. It adds the book that the user wants to reserve to the Reservation table and it also sets the Reserved column of the book in the Books table to Y.
* **Return.php** – This page is accessed when a book is returned from the Reservations Page. It removes the book from the Reservations table and it also sets the Reserved column of the book in the Books table to N.

# Source Code

## Login.php

<!DOCTYPE html>

<html>

<head>

<title>Library - Login</title>

<link type="text/css" href="http://fonts.googleapis.com/css?family=Open+Sans">

<link rel="stylesheet" type="text/css" href="../CSS/mystyle.css">

</head>

<body>

<div id="container">

<?php include "../HTML/Header.html"; ?>

<div id="content">

<?php

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

{

// Start the session

session\_start();

// Get the inputed username and password

$username = mysql\_real\_escape\_string($\_POST["username"]);

$password = mysql\_real\_escape\_string($\_POST["password"]);

// Verify if the fields are empty

if (empty($username) || empty($password))

{

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

echo "Please enter a Username and a Password.";

echo "</div>";

}

else

{

// Connect to the database

require\_once("db.php");

// SQL to get all the informations on the users

$result = mysql\_query("SELECT Username, Password

FROM Users

WHERE Username = '$username' AND Password = '$password'");

// Check if the row exists

$row = mysql\_fetch\_row($result);

if (!empty($row))

{

// Set the username session

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

// Close the database

mysql\_close($db);

// Refresh the page

header("Location: Login.php");

}

else

{

$success = false;

}

// Print an error message if the username and password aren't in the database

if (!$success)

{

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

echo "Invalid Username or Password";

echo "</div>";

}

// Close the database

mysql\_close($db);

}

}

else

{

// Display the login page

session\_start();

if (empty($\_SESSION['username'])) { ?>

<div id="login">

<h2>Login</h2>

<form method = "POST">

<p>

<input type="text" name="username" placeholder="Username" autofocus>

</p>

<p>

<input type="password" name="password" placeholder="Password">

</p>

<p> <input type="Submit"> </p>

</form>

</div>

<?php } else { ?>

<div class = "success">

Hello <?php echo $\_SESSION['username'] ?>!

<form action = "Logout.php">

<button> Logout </button>

</form>

</div>

<?php }

} ?>

</div>

<?php include "../HTML/Footer.html"; ?>

</div>

</body>

</html>

## Register.php

<!DOCTYPE html>

<html>

<head>

<title>Library - Register</title>

<link type="text/css" href="http://fonts.googleapis.com/css?family=Open+Sans">

<link rel="stylesheet" type="text/css" href="../CSS/mystyle.css">

</head>

<body>

<div id="container">

<?php include "../HTML/Header.html"; ?>

<div id="content">

<?php

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

isset($\_POST['passwordConfirm']) && isset($\_POST['address']) && isset($\_POST['city']) && isset($\_POST['telephone']) &&

isset($\_POST['mobile']))

{

// Get the register information

$first = mysql\_real\_escape\_string($\_POST['firstName']);

$last = mysql\_real\_escape\_string($\_POST['lastName']);

$username = mysql\_real\_escape\_string($\_POST['username']);

$password = mysql\_real\_escape\_string($\_POST['password']);

$passwordConfirm = mysql\_real\_escape\_string($\_POST['passwordConfirm']);

$address = mysql\_real\_escape\_string($\_POST['address']);

$address2 = mysql\_real\_escape\_string($\_POST['address2']);

$city = mysql\_real\_escape\_string($\_POST['city']);

$telephone = mysql\_real\_escape\_string($\_POST['telephone']);

$mobile = mysql\_real\_escape\_string($\_POST['mobile']);

// Verify if the fields are empty

if (empty($first) || empty($last) || empty($username) || empty($password) || empty($passwordConfirm) || empty($address) || empty($city) || empty($telephone) || empty($mobile))

{

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

echo 'Please fill in all the fields.';

echo "</div>";

}

else

{

// Verify if the password and mobile phone number are the correct length and if the mobile is numeric

if ( (strlen($password) != 6) || (strlen($mobile) != 10) || (!is\_numeric($mobile)) )

{

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

echo 'The password and/or mobile number are of incorrect length.<br>';

echo 'The password must have 6 characters and the mobile number must have 10 numbers.';

echo "</div>";

}

else

{

// Verify if both password fields are identical

if ($password != $passwordConfirm)

{

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

echo 'The password confirmation field is incorrect.';

echo "</div>";

}

else

{

// Connect to the database

require\_once("db.php");

// SQL to get all the information about the users from the database

$result = mysql\_query("SELECT Username

FROM Users

WHERE Username = '$username'");

// Check if the row exists

$success = false;

$row = mysql\_fetch\_row($result);

if (empty($row))

{

$success = true;

}

// Verify if the username is unique

if (!$success)

{

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

echo "The username is already taken.";

echo "</div>";

}

else

{

// Insert the user's information in the database

$sql = "INSERT INTO Users (Username, Password, FirstName, Surname, AddressLine1, AddressLine2, City, Telephone, Mobile)

VALUES ('$username', '$password', '$first', '$last', '$address', '$address2', '$city', '$telephone', '$mobile')";

// Insert the data and check if it works

if (mysql\_query($sql))

{

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

echo "You have registered successfully!";

echo "</div>";

}

else

{

// Display error message if the insert didn't work

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

echo "Cannot insert the data into the database";

echo "</div>";

}

// Close the database

mysql\_close($db);

}

}

}

}

}

else

{

// Start the session

session\_start();

if (empty($\_SESSION['username'])) { ?>

<div id="registerForm">

<h2>Register</h2>

<form method="POST">

<p>

<input type="text" name="firstName" placeholder="First name" autofocus>

<input type="text" name="lastName" placeholder="Last name">

</p>

<p>

<input type="text" name="username" placeholder="Username" class="longField">

</p>

<p>

<input type="password" name="password" placeholder="Password" class="longField">

</p>

<p>

<input type="password" name="passwordConfirm" placeholder="Confirm Password" class="longField">

</p>

<p>

<input type="text" name="address" placeholder="Address Line 1" class="longField">

</p>

<p>

<input type="text" name="address2" placeholder="Address Line 2" class="longField">

</p>

<p>

<input type="text" name="city" placeholder="City" class="longField">

</p>

<p>

<input type="text" name="telephone" placeholder="Telephone number" class="longField">

</p>

<p>

<input type="text" name="mobile" placeholder="Mobile number" class="longField">

</p>

<p>

<input type="submit" class="longField">

</p>

</form>

</div>

<?php } else { ?>

<div class = "success">

Hello <?php echo $\_SESSION['username'] ?>!

<form action = "Logout.php">

<button> Logout </button>

</form>

</div>

<?php }

} ?>

</div>

<?php include "../HTML/Footer.html"; ?>

</div>

</body>

</html>

## Books.php

<!DOCTYPE html>

<html>

<head>

<title>Library - Books</title>

<link type="text/css" href="http://fonts.googleapis.com/css?family=Open+Sans">

<link rel="stylesheet" type="text/css" href="../CSS/mystyle.css">

<script src="../JS/script.js"></script>

</head>

<body>

<div id="container">

<?php include "../HTML/Header.html"; ?>

<?php include "../PHP/Username.php"; ?>

<div id="content">

<?php

// External PHP

require\_once("Functions.php");

// Get the current page number

$pageNo = getPageNumber();

// Check if the user is logged in

if ( empty($\_SESSION['username']) )

{

include "../HTML/AccessDenied.html";

}

else

{

// Open the database

require\_once("db.php");

// SQL to get all the books

$sql = "SELECT b.ISBN, b.BookTitle, b.Author, b.Edition, b.Year, c.CategoryDescription, b.Reserved

FROM Books b JOIN Categories c ON(b.Category = c.CategoryID)";

// Read the data into the variable books

readData($sql, $books);

// Print the books array

printBooks($pageNo, $books, "Reserve");

// Create a form with two buttons that has the value of the next or previous page

echo '<form type = "GET">';

if ( $pageNo != 0 )

{

echo '<button name = "page" value = '.($pageNo - 1).' class = "leftArrow"> < </button>';

}

if ( (($pageNo+1) \* 5) < count($books) )

{

echo '<button name = "page" value = '.($pageNo + 1).' class = "rightArrow"> > </button>';

}

echo '</form>';

// Close the database

mysql\_close($db);

}

?>

</div>

<?php include "../HTML/Footer.html"; ?>

</div>

</body>

</html>

## Search.php

<!DOCTYPE html>

<html>

<head>

<title>Library - Search</title>

<link type="text/css" href="http://fonts.googleapis.com/css?family=Open+Sans">

<link rel="stylesheet" type="text/css" href="../CSS/mystyle.css">

<script src="../JS/script.js"></script>

</head>

<body>

<div id="container">

<?php include "../HTML/Header.html"; ?>

<?php include "../PHP/Username.php"; ?>

<div id="content">

<?php // Check if the user is logged in

if ( empty($\_SESSION['username']) )

{

include "../HTML/AccessDenied.html";

}

else

{ ?>

<form method = "GET">

<input type = "hidden" name = "page" value = "0">

<input type = "text" id = "bookTitle" name = "bookTitle" placeholder = "Title" autofocus>

<input type = "text" id = "bookAuthor" name = "bookAuthor" placeholder = "Author">

<select name = "category">

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

<?php

// Open the database

require\_once("db.php");

$result = mysql\_query("SELECT \* FROM Categories");

// Create the drop down category list

while($row = mysql\_fetch\_row($result))

{

echo "<option value=$row[0]";

// Select the category

if ((isset($\_GET['category']) && ($row[0] == $\_GET['category'])))

{

echo " selected ";

}

echo ">$row[1]</option>";

}

?>

</select>

<input type = "submit" id = "searchSubmit">

</form>

<br>

<?php } ?>

<?php

if ( isset($\_GET['bookTitle']) || isset($\_GET['bookAuthor']) || isset($\_GET['category']) )

{

// External PHP

require\_once("Functions.php");

// Get the current page number

$pageNo = getPageNumber();

// Set the title variable

if (!empty($\_GET['bookTitle']))

{

$bookTitle = strtoupper(mysql\_real\_escape\_string($\_GET['bookTitle']));

echo "Title: ".$\_GET['bookTitle'];

}

else

{

$bookTitle = "";

}

// Set the author variable

if (!empty($\_GET['bookAuthor']))

{

$bookAuthor = strtoupper(mysql\_real\_escape\_string($\_GET['bookAuthor']));

if (!empty($\_GET['bookTitle']))

echo " | ";

echo "Author: ".$\_GET['bookAuthor'];

}

else

{

$bookAuthor = "";

}

if (isset($\_GET['category']))

{

$category = $\_GET['category'];

}

else

{

$category = "";

}

// SQL to get all the books that contain the searched title and/or author

$sql = "SELECT b.ISBN, b.BookTitle, b.Author, b.Edition, b.Year, c.CategoryDescription, b.Reserved

FROM Books b JOIN Categories c ON(b.Category = c.CategoryID)

WHERE b.bookTitle LIKE '%$bookTitle%' AND b.Author LIKE '%$bookAuthor%' AND c.categoryID LIKE '%$category%'";

// Read the data into the variable books

readData($sql, $books);

// Print the books array

printBooks($pageNo, $books, "Reserve");

// Create a form with two buttons that has the value of the next or previous page

echo '<form type = "GET">';

echo '<input type = "hidden" name = "bookTitle" value = "'.$bookTitle.'">';

echo '<input type = "hidden" name = "bookAuthor" value = "'.$bookAuthor.'">';

echo '<input type = "hidden" name = "category" value = "'.$category.'">';

if ( $pageNo != 0 )

{

echo '<button name = "page" value = '.($pageNo - 1).' class = "leftArrow"> < </button>';

}

if ( (($pageNo+1) \* 5) < count($books) )

{

echo '<button name = "page" value = '.($pageNo + 1).' class = "rightArrow"> > </button>';

}

echo '</form>';

// Close the database

mysql\_close($db);

}

?>

</div>

<?php include "../HTML/Footer.html"; ?>

</div>

</body>

</html>

## Reservations.php

<!DOCTYPE html>

<html>

<head>

<title>Library - Reservations</title>

<link type="text/css" href="http://fonts.googleapis.com/css?family=Open+Sans">

<link rel="stylesheet" type="text/css" href="../CSS/mystyle.css">

<script src="../JS/script.js"></script>

</head>

<body>

<div id="container">

<?php include "../HTML/Header.html"; ?>

<?php include "../PHP/Username.php"; ?>

<div id="content">

<?php // Check if the user is logged in

if ( empty($\_SESSION['username']) ) {

include "../HTML/AccessDenied.html";

}

else

{

// External PHP

require\_once("Functions.php");

// Get the current page number

$pageNo = getPageNumber();

// Get the username

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

// Open the database

require\_once("db.php");

// SQL to get all the reservations made by the current user

$sql = "SELECT b.ISBN, b.BookTitle, b.Author, b.Edition, b.Year, c.CategoryDescription, b.Reserved

FROM Books b JOIN Categories c ON(b.Category = c.CategoryID)

JOIN Reservations r ON(b.ISBN = r.ISBN)

WHERE r.Username = '$username'";

// Read the data into the variable books

readData($sql, $books);

// Display message if there are no reservations made by the current user

if (!isset($books))

{

echo '<h3 style = "text-align: center">You don\'t have any reservations.</h3>';

}

else

{

// Check if there are any books to display on the page

if (count($books) < $pageNo \* 5 + 1)

{

$pageNo--;

if ($pageNo == -1)

$pageNo = 0;

header("Location: Reservations.php?page=".$pageNo);

}

// Print the books array

printBooks($pageNo, $books, "Return");

// Create a form with two buttons that has the value of the next or previous page

echo '<form type = "GET">';

if ( $pageNo != 0 )

{

echo '<button name = "page" value = '.($pageNo - 1).' class = "leftArrow"> < </button>';

}

if ( (($pageNo+1) \* 5) < count($books) )

{

echo '<button name = "page" value = '.($pageNo + 1).' class = "rightArrow"> > </button>';

}

echo '</form>';

}

// Close the database

mysql\_close($db);

}

?>

</div>

<?php include "../HTML/Footer.html"; ?>

</div>

</body>

</html>

## Functions.php

<?php

function printBooks($pageNo, $books, $state)

{

// Print the array (5 per page)

for ($i = $pageNo \* 5 ; ($i < $pageNo \* 5 + 5) && !empty($books[$i]) ; $i++)

{

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

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

echo '<br>';

echo '<span class = "title">';

echo $books[$i]['Title'];

echo '</span>';

echo '<span class = "author">';

echo ' by '.$books[$i]['Author'];

echo '</span>';

echo '<br>';

echo $books[$i]['Category']." | ".$books[$i]['Year']." | ";

switch($books[$i]['Edition'])

{

case 1:

{

echo '1st';

break;

}

case 2:

{

echo '2nd';

break;

}

case 3:

{

echo '3rd';

break;

}

default:

{

echo $books[$i]['Edition'].'th';

}

}

echo ' Edition';

echo '</div>';

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

if ( ($books[$i]['Reserved'] == 'N') && ($state == "Reserve") )

{

echo '<button name = "ISBN" value="'.$books[$i]['ISBN'].'" class = "reserve" onclick = "confirmReservation(this.value, '.$pageNo.')"> Reserve </button>';

}

else

if ($state == "Reserve")

echo '<button class = "notAvailable" disabled> Not Available </button>';

else

echo '<button name = "ISBN" value="'.$books[$i]['ISBN'].'" class = "reserve" onclick="confirmReturn(this.value, '.$pageNo.')"> Return </button>';

echo '</div>';

echo '</div>';

}

}

function readData($sql, &$books)

{

// SQL to return all the book records

$result = mysql\_query($sql);

// Put the books into a 2D array

while ($row = mysql\_fetch\_row($result))

{

$temp['ISBN'] = htmlentities($row[0]);

$temp['Title'] = htmlentities($row[1]);

$temp['Author'] = htmlentities($row[2]);

$temp['Edition'] = htmlentities($row[3]);

$temp['Year'] = htmlentities($row[4]);

$temp['Category'] = htmlentities($row[5]);

$temp['Reserved'] = htmlentities($row[6]);

$books[] = $temp;

}

}

function getPageNumber()

{

// Get the page number if it exists

if (isset($\_GET['page']))

{

return $\_GET['page'];

}

else

{

return 0;

}

}

?>

## db.php

<?php

$db = mysql\_connect('localhost', 'root', '');

if ( $db === FALSE ) die('Failed to connect to the database');

if ( mysql\_select\_db("assignment") === FALSE ) die ('Failed to connect to the database');

?>

## Logout.php

<?php

session\_start();

session\_destroy();

header("Location: ../PHP/Login.php");

die();

?>

## Username.php

<div class = "user">

<?php

session\_start();

if (!empty($\_SESSION['username']))

{

echo "Hello " . $\_SESSION['username'] . "! ";

echo '<form action = "Logout.php" style = "display: inline; float: right; margin-right: 30px;">';

echo '<button> Logout </button>';

echo '</form>';

}

?>

</div>

## ReserveBook.php

<?php

// Start the session

session\_start();

// Get the variables needed

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

$ISBN = $\_GET['ISBN'];

$page = $\_GET['page'];

$title = $\_GET['bookTitle'];

$author = $\_GET['bookAuthor'];

$category = $\_GET['category'];

// Assign the category variable a blank space if the category is not set

if ((isset($\_GET['category'])) && (empty($\_GET['category'])))

{

$category = " ";

}

// Add a new reservation to the Reservations Table and set the reserved fields to Y

require\_once("db.php");

$sql = "INSERT INTO Reservations (ISBN, Username, ReservedDate)

VALUES ('$ISBN', '$username', '2015-08-15')";

if (mysql\_query($sql))

{

$sql = "UPDATE Books

SET Reserved = 'Y'

WHERE ISBN = '$ISBN'";

mysql\_query($sql);

}

mysql\_close($db);

// Return to the previous page

if (!empty($title) || !empty($author) || !empty($category))

header("Location: ../PHP/Search.php?bookTitle=".$title."&bookAuthor=".$author."&page=".$page."&category=".$category);

else

header("Location: ../PHP/Books.php?page=".$page);

?>

## Return.php

<?php

// Get the ISBN and the page number

session\_start();

$ISBN = $\_GET['ISBN'];

$page = $\_GET['page'];

// Connect to the database

require\_once("db.php");

// Delete the entry from the Reservations Table

$sql = "DELETE FROM Reservations

WHERE ISBN = '$ISBN'";

if (mysql\_query($sql))

{

$sql = "UPDATE Books

SET Reserved = 'N'

WHERE ISBN = '$ISBN'";

mysql\_query($sql);

}

mysql\_close($db);

header("Location: ../PHP/Reservations.php?page=".$page);

?>

## Header.html

<header>

<h1>Library</h1>

<div id="menu">

<ul >

<li class="select"> <a href="../PHP/Login.php">Login</a> </li>

<li class="separator">|</li>

<li class="select"> <a href="../PHP/Register.php">Register</a> </li>

<li class="separator">|</li>

<li class="select"> <a href="../PHP/Books.php?page=0">Books</a> </li>

<li class="separator">|</li>

<li class="select"> <a href="../PHP/Search.php?page=0">Search</a> </li>

<li class="separator">|</li>

<li class="select"> <a href="../PHP/Reservations.php?page=0">Reservations</a></li>

</ul>

</div>

</header>

## Footer.html

<footer>

&copy; 2015 Vlad Zat

<br>

<!--<button> <a href = "Logout.php"> Logout </a> </button>-->

</footer>

## AccessDenied.html

<div class = "error" style = "margin-top: -120px">

Access Denied!

<br>

<form action = "Login.php">

<button> Login </button>

</form>

</div>

## mystyle.css

body {

background-color: #d6d6d6;

position: relative;

font-family: 'Open Sans';

}

#container {

width: 960px;

//height: 700px;

background-color: #FFFFFF; //#7E8AA2;

margin: auto;

top: 0; bottom: 1; left: 0; right: 0;

position: absolute;

}

header {

width: 100%;

height: 120px;

background-color: #000000; //#263248;

color: #FFFFFF;

text-align: center;

position: absolute;

}

footer {

width: 100%;

//height: 50px;

background-color: #000000; //#263248;

color: #FFFFFF;

bottom: 0;

text-align: center;

//position: absolute;

}

#content {

width: 900px;

//height: 100px;

margin-left: 30px;

//margin-top: 150px;

//position: absolute;

min-height: 650px;

height: auto !important;

margin-bottom: -50px;

}

#content:after {

content: "";

display: block;

}

footer, #content:after {

height: 50px;

}

#menu {

width: 100%;

margin: auto;

text-align: center;

position: absolute;

}

#menu ul {

margin: auto;

margin-left: -40px;

list-style-type: none;

display: inline-block;

}

#menu li {

padding: 5px;

float: left;

display: block;

border-radius: 5px;

margin-left: 50px;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

}

.separator {

cursor: default;

}

.select {

cursor: pointer;

}

.select:hover {

color: #E74C3C;

}

h2 {

color: #2980B9;

}

#login {

width: 500px;

height: 300px;

margin: auto;

color: #FF9800;

margin-left: 200px;

margin-top: 350px;

padding-bottom: 230px;

text-align: center;

}

#login form {

width: 100%;

height: 100%;

margin: auto;

padding-top: 5px;

font-size: 120%;

}

#login input[type=submit], #registerForm input[type=submit] {

background-color: #2C3E50;

border: 0;

cursor: pointer;

//border-radius: 10px;

color: #FFFFFF;

padding: 5px;

font-weight: bold;

}

header a {

text-decoration: none;

color: #FFFFFF;

}

header a:hover, a:active {

color: #FF9800;

}

input{

height: 30px;

width: 200px;

}

#registerForm {

width: 500px;

margin: auto;

margin-top: 280px;

margin-bottom: 160px;

margin-left: 200px;

text-align: center;

}

.longField {

height: 30px;

width: 408px;

}

.error, .success {

width: 100%;

text-align: center;

padding-top: 300px;

}

.error {

color: red;

}

.success {

color: green;

}

.tableStyle {

width: 900px;

height: 500px;

border: thin solid black;

}

.tableStyle td {

height: 100px;

}

.record {

width: 900px;

height: 90px;

border: 1px solid black;

}

.record:hover {

background-color: #FF9800;

}

.left {

width: 765px;

height: 100%;

margin-left: 20px;

float: left;

cursor: default;

}

.title, .author {

display: inline;

}

.title {

font-weight: bold;

}

.author {

color: grey;

}

.right {

width: 95px;

height: 100%;

margin-right: 20px;

float: right;

text-align: center;

}

.right button {

margin-top: 30px;

}

.reserve {

cursor: pointer;

}

.notAvailable {

cursor: default;

}

.reserve, .notAvailable {

border-radius: 5px;

}

.leftArrow {

float: left;

}

.rightArrow {

float: right;

}

.leftArrow, .rightArrow {

cursor: pointer;

margin-top: 10px;

background-color: #2C3E50;

border: 0;

padding-left: 10px;

padding-right: 10px;

padding-top: 5px;

padding-bottom: 5px;

color: #FFFFFF;

}

.user {

margin-top: 150px;

margin-bottom: 20px;

margin-left: 30px;

}

#bookTitle {

width: 350px;

}

#bookAuthor {

width: 350px;

}

#searchSubmit {

width: 83px;

}

select {

width: 95px;

height: 30px;

}

## script.js

// Create a confirmation popup window

function confirmReservation(value, page)

{

var answer = confirm("Are you sure you want to reserve it?");

if (answer)

window.location.href = "ReserveBook.php" + window.location.search + "&ISBN=" + value;

}

// Create a confirmation popup window

function confirmReturn(value, page)

{

var answer = confirm("Are you sure you want to return it?");

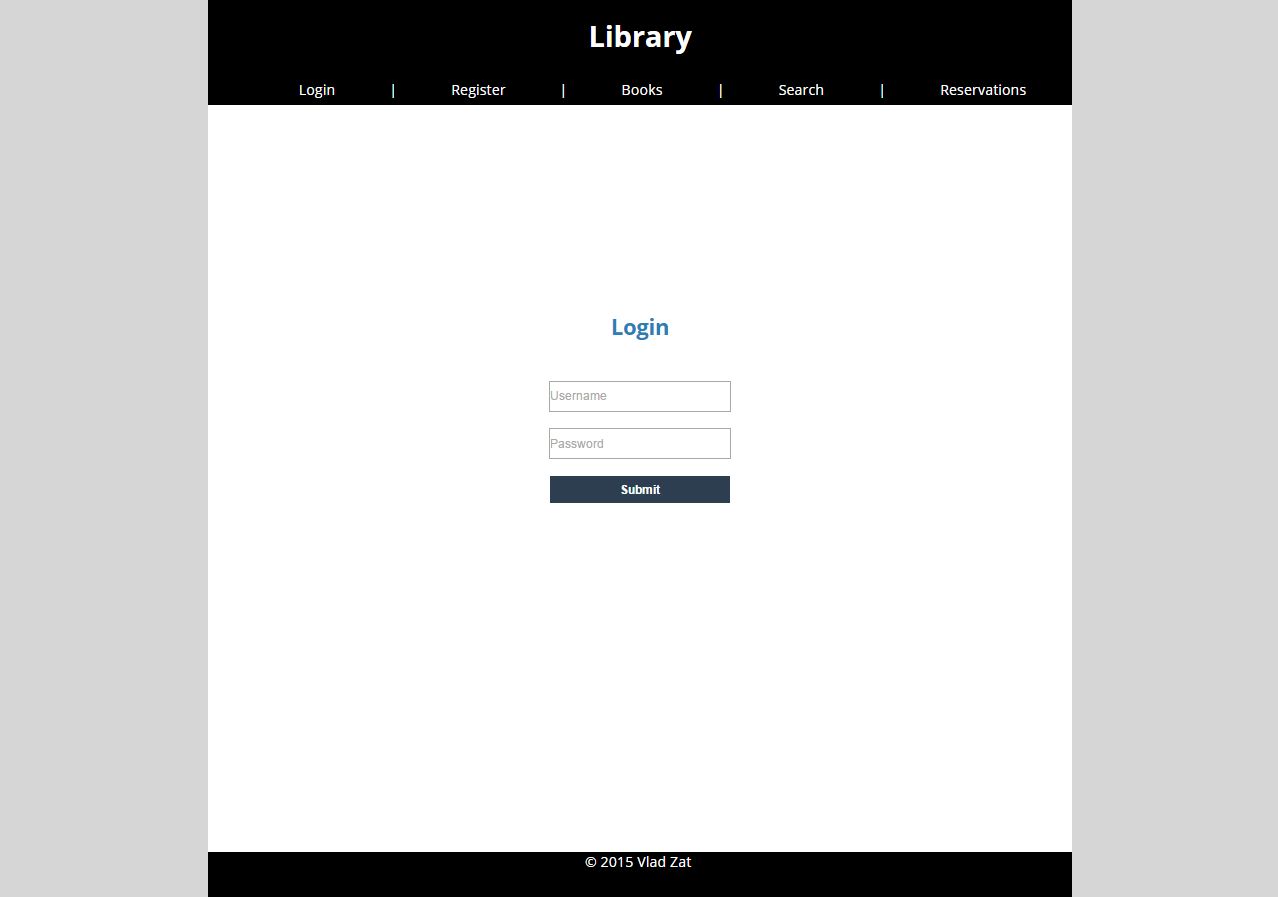
if (answer)

window.location.href = "Return.php?ISBN=" + value + "&page=" + page;

}

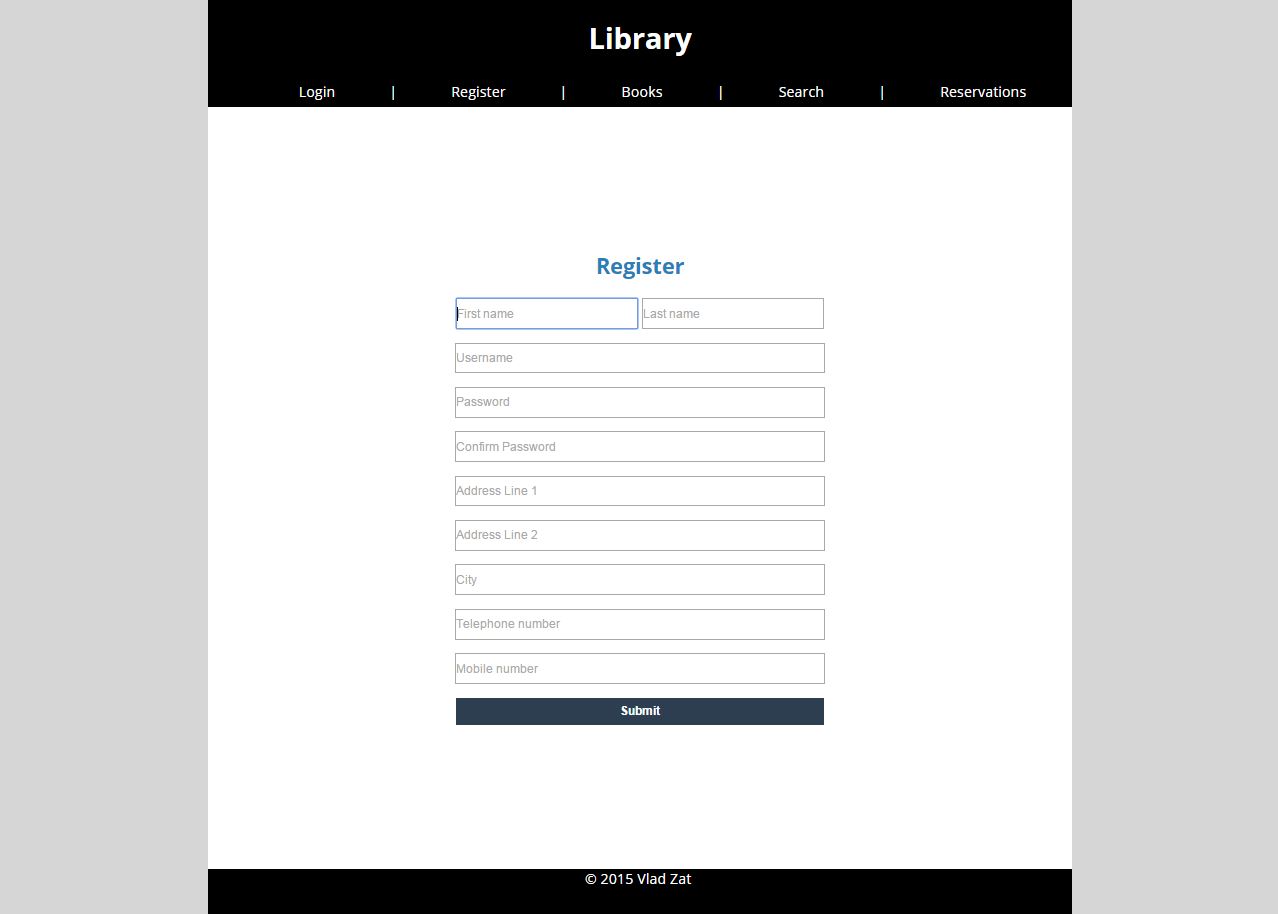
# Screen Samples

## Login Page



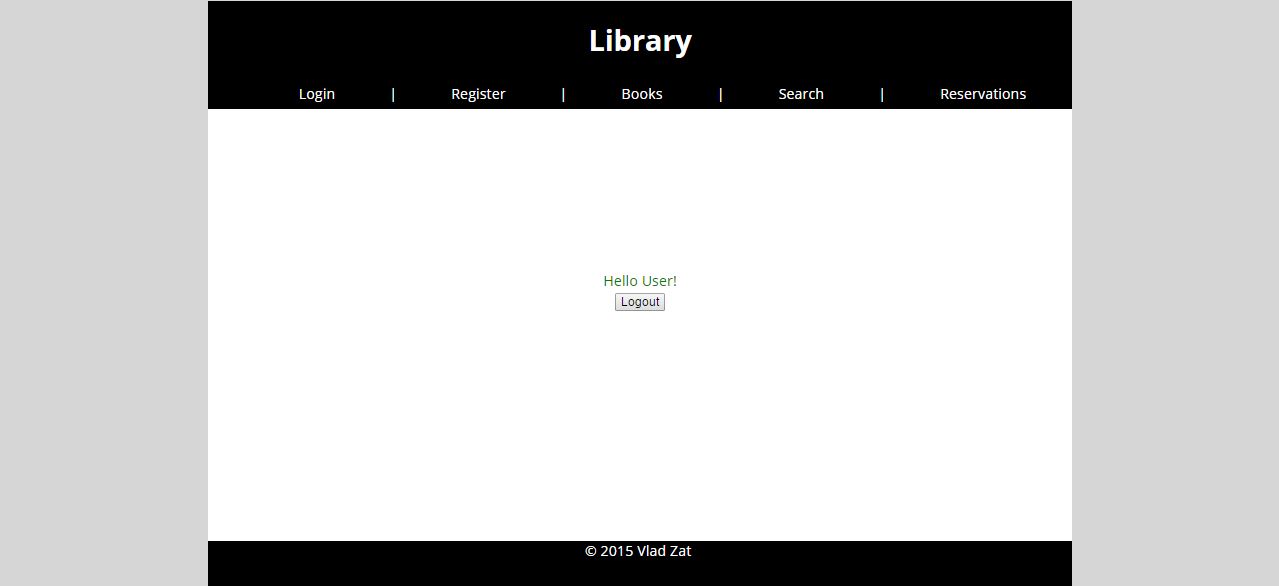
Login page when no user is logged in

## Register



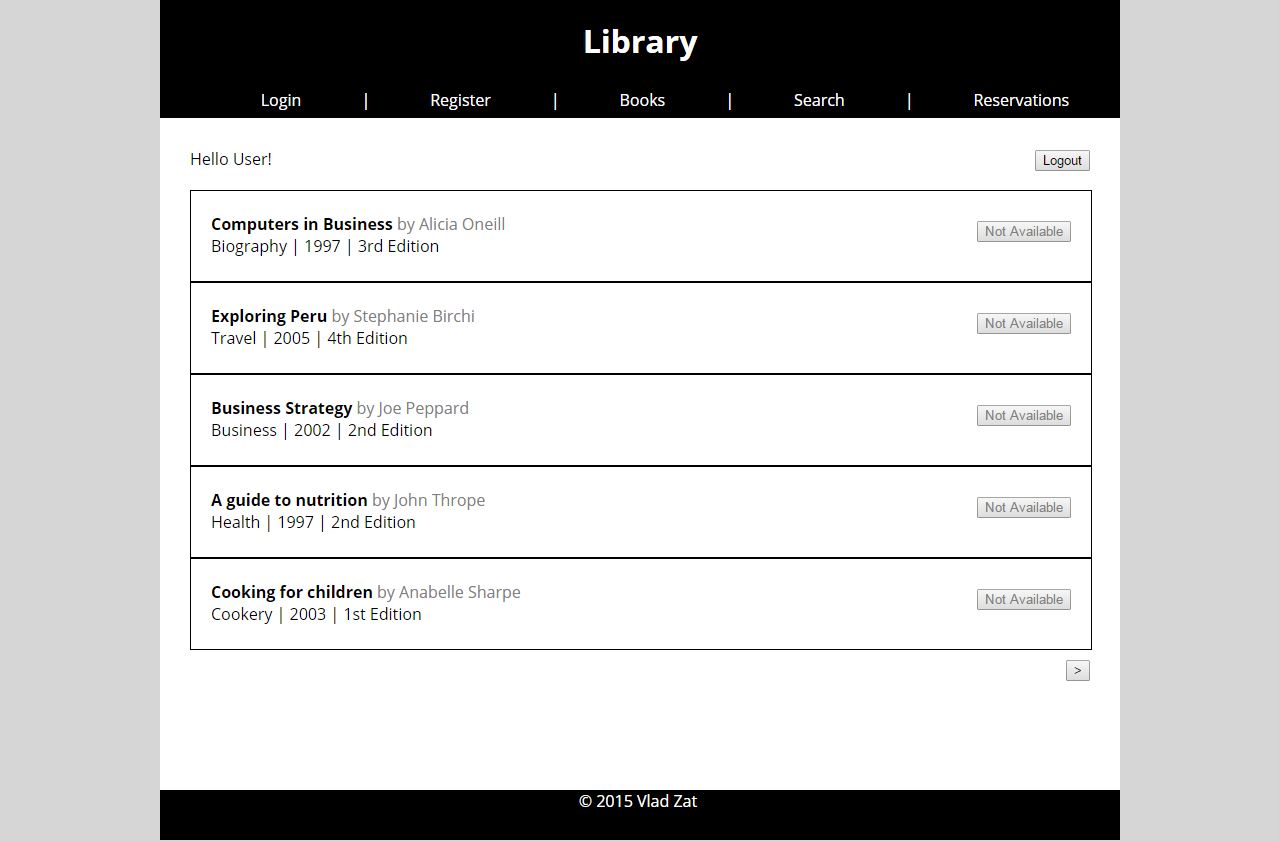
Register page when no user is logged in

## Login and Register



Login and Register page when a user is logged in

## Books

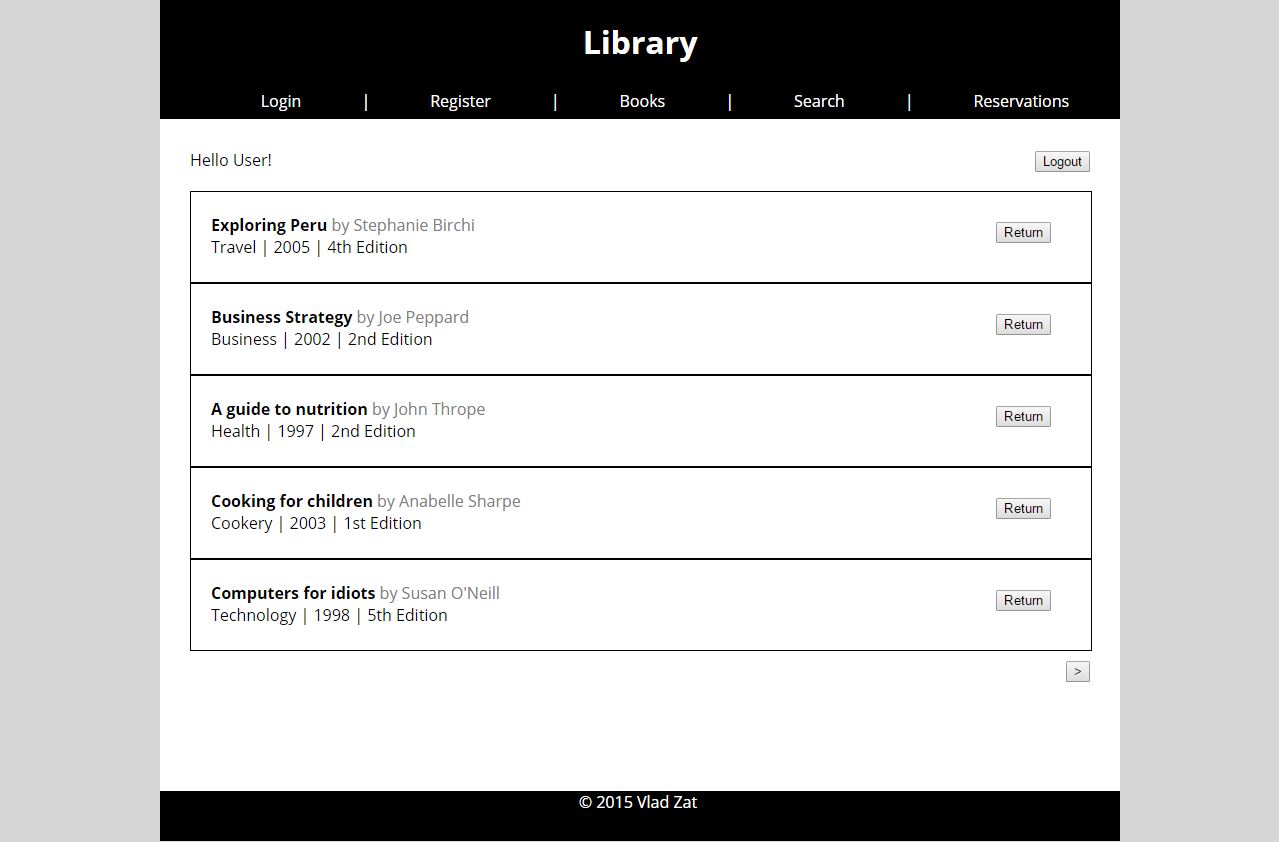


Books page when a user is logged in

## SearchC:\Users\Jean-Luc Picard\Desktop\Search.JPG

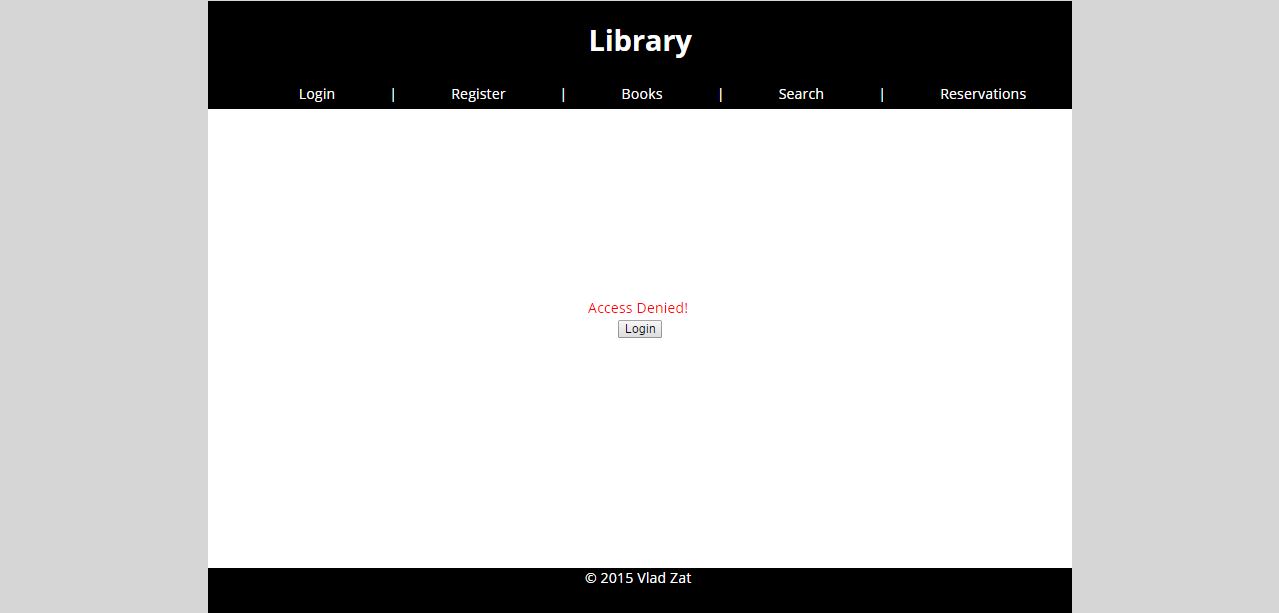
Search page when a user is logged in

## Reservations



Reservations page when a user is logged in

## Books, Search and Reservations



Books, Search and Reservations pages when no user is logged in

## Design

