

WEB DEVELOPMENT 2

PROJECT REPORT

*TU 856/2
CMPU 2022*

D22127506 Heeyeon Yoon

1. PHP Pages

a) db.php:

This file handles the database connection. It defines the connection parameters such as the host, username, password, and database name and establishes a connection to the LibraryDataBase hosted on phpMyAdmin. Most pages (e.g., index.php, login.php) include db.php, ensuring that all database related operations are performed without reconnecting each time.

- **Key Functionality:** mysqli_connect(\$host, \$username, \$password, \$dbname)

: Ensures a persistent connection to the database and checks for errors

```
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

b) header.php:

This file creates the page header, which includes the navigation bar and a banner image. It displays different menus for logged-in users and guests by checking the session variable \$_SESSION['Username']

- **Key Functionality**

: Determines the user's status and dynamically adjust the navigation menu accordingly.

```
<?php if ($isLoggedIn): ?>
    <!-- Navigation bar for logged in users -->
    <li><a href="index.php">Home</a></li>
    <li><a href="view_reservations.php">My Reservations</a></li>
    <li><a href="logout.php">Logout</a></li>
<?php else: ?>
    <!-- Navigation bar for users who are not logged in -->
    <li><a href="index.php">Home</a></li>
    <li><a href="login.php">Login</a></li>
    <li><a href="register.php">Register</a></li>
<?php endif; ?>
```

c) footer.php:

This file displays the footer section, which includes copyright information and a contact link at the bottom of most pages.

d) index.php:

This file is the homepage of the application. For logged-in users, it displays a welcome message, search functionality and a link to their reservation list. For guests, it lets users to log in or register. This file includes header/footer.php, db.php, and search.php.

- **Key Functionality**

: Greeting logged-in users

```
<?php if ($isLoggedIn): ?>
    <div class="welcome-message">
        <h2>Hello, <?php echo htmlspecialchars($_SESSION['Username']); ?>!</h2>
        <h2>Welcome to our Library System</h2>
    </div>
```

e) login.php:

This file handles user login. It verifies that the user exists in the Users table in the database. If the login is successful, it stores the username in the session and redirects the user to the homepage. This file includes db.php, header/footer.php, and index.php.

- Key Functionality

: Validating user credentials

```
if ($result->num_rows > 0) {
    $user = $result->fetch_assoc();
    if ($password === $user['Password']) {
        // if password is correct, store user in the session
        $_SESSION['Username'] = $username;
        header("Location: index.php"); // redirection after log in
        exit();
    } else {
        $error = "Invalid username or password.";
    }
} else {
    $error = "User not found.";
}
```

f) logout.php:

This file logs the user out by ending the session using session_destroy() and then redirects them to the login page.

- Key Functionality : session_destroy()

g) register.php:

This file allows new users to register by providing required details. It validates the inputs (e.g., password length, unique username) and stores the information in the Users table. If successful, logs the user in and redirects to the homepage. If the username is already taken, it displays an error.

- Key Functionality

: Ensuring unique usernames

```
$stmt = $conn->prepare("SELECT * FROM Users WHERE Username = ?");
$stmt->bind_param("s", $username);
$stmt->execute();
$result = $stmt->get_result();
if ($result->num_rows > 0) {
    $error = "Username is already taken.";
} else if
```

h) search.php:

This file processes book search requests, allowing users to search by title, author, or category. It implements pagination for large(> 5 rows) search results. It includes db.php to query the Books and Category tables.

- Key Functionality

: Constructing dynamic search queries

```
if ($search_term) {
    $query .= " AND (Books.BookTitle LIKE ? OR Books.Author LIKE ?)";
    $params[] = $search_term;
    $params[] = $search_term;
    $types .= "ss";
}
```

i) reservation.php:

This file handles book reservations, if the selected book is available, it updates the Books table to mark it as reserved and records the reservation in the ReservedBook table. It uses db.php to access the Books and ReservedBook tables and redirects to view_reservations.php after a successful reservation.

- Key Functionality

: Reserving a book

```
// Update the reservation status to reserved(1)
$stmt = $conn->prepare("UPDATE Books SET Reserved = 1 WHERE ISBN = ?");
$stmt->bind_param("s", $isbn);
$stmt->execute();
```

j) view_reservations.php:

This file displays a logged-in user's current reservations with details like book title, reservation date and cancellation button. It uses db.php to query the Reservedbook and Books tables and includes header.php and footer.php for UI consistency.

- Key Functionality

: Fetching user reservations

```
$username = $_SESSION['Username']; // Get the username
$stmt = $conn->prepare("SELECT ReservedBook.*, Books.BookTitle FROM ReservedBook JOIN Books ON
ReservedBook.ISBN = Books.ISBN WHERE ReservedBook.Username = ?");
$stmt->bind_param("s", $username); // Bind the useername to the query
$stmt->execute();
$result = $stmt->get_result(); // Execute the query and get the result
```

k) cancel_reservation.php:

This file cancels a user's reservation by removing the record from the ReservedBook table and updating the reserved status of the book in the Books table. It uses db.php to update the ReservedBook and Book tables and redirects to view_reservations.php after cancellation.

- Key Functionality

: Canceling a reservation

```
// Update the reservation status to not reserved(0)
$stmt = $conn->prepare("UPDATE Books SET Reserved = 0 WHERE ISBN = ?");
$stmt->bind_param("s", $isbn);
$stmt->execute();
```

2. Source Code

a) db.php:

```
<?php
// Database connection configuration
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "LibraryDataBase";

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

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
} else {
    // echo "Connected successfully!";
}
?>
```

b) header.php:

```
<?php
$isLoggedIn = isset($_SESSION['Username']); // Check if the user is logged or not
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Library</title>
    <!-- link to css file -->
    <link rel="stylesheet" href="../css/style.css">
</head>
<body>

<div class="banner">
    
</div>

<header>
    <nav>
        <ul>
            <?php if ($isLoggedIn): ?>
                <!-- Navigation bar for logged in users -->
                <li><a href="index.php">Home</a></li>
                <li><a href="view_reservations.php">My Reservations</a></li>
                <li><a href="logout.php">Logout</a></li>
            <?php else: ?>
                <!-- Navigation bar for users who are not logged in -->
                <li><a href="index.php">Home</a></li>
                <li><a href="login.php">Login</a></li>
                <li><a href="register.php">Register</a></li>
            <?php endif; ?>
        </ul>
    </nav>
</header>

</body>
</html>
```

c) footer.php:

```
<footer>
    <p>&copy; <?php echo date("Y"); ?> Library System. All rights reserved.</p>
    <p>Contact: <a href="mailto:d22127506@mytudublin.ie">d22127506@mytudublin.ie</a></p>
</footer>
```

d) index.php:

```
<?php
session_start(); // Session start

// Connect to DB
require_once "db.php";

// Check if the user is logged in
$isLoggedIn = isset($_SESSION['Username']);

// bring the book categories from databases
$categories = [];
$stmt = $conn->prepare("SELECT CategoryID, CategoryDetails FROM Category ORDER BY CategoryDetails ASC");
$stmt->execute();
$result = $stmt->get_result();

while ($row = $result->fetch_assoc()) {
    $categories[] = $row;
}
$stmt->close();

?>

<?php require_once "header.php"; ?>

<main>
    <!-- For logged in users -->
    <?php if ($isLoggedIn): ?>
        <div class="welcome-message">
            <h2>Hello, <?php echo htmlspecialchars($_SESSION['Username']); ?>!</h2>
            <h2>Welcome to our Library System</h2>
        </div>
        <!-- Search form -->
        <form action="search.php" method="GET">
            <p>Search the book by book title or author</p>
            <input type="text" name="search" placeholder="Enter book title or author...">
            <select name="category" class="category-dropdown">
                <option value="">Select Category (Optional)</option>
                <?php foreach ($categories as $category): ?>
```

```

        <option value=<?php echo
htmlspecialchars($category['CategoryID']); ?>>
        <?php echo
htmlspecialchars($category['CategoryDetails']); ?>
        </option>
    <?php endforeach; ?>
</select>
<button type="submit">Search</button>
</form>
<div class="reservation_container">
    <a href="view_reservations.php" id="viewReserve">View My
Reservations</a>
</div>
<?php else: ?>
    <!-- Login and Register buttons for non logged in users -->
    <div class="library-banner">
        
        <div class="btn_container">
            <p>Please login or register to use our services</p>
            <div class="index_container">
                <a href="login.php"><button
class="index_button">Login</button></a>
                <a href="register.php"><button
class="index_button">Register</button></a>
            </div>
        </div>
    </div>
<?php endif; ?>
</main>

<?php require_once "footer.php"; ?>

```

e) login.php:

```

<?php
session_start();
require_once "db.php"; // connect databases

/*
// check if user logged in
if (isset($_SESSION['Username'])) {
    header("Location: index.php"); // redirection if user already logged in
    exit();
}
*/

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['Username'];
    $password = $_POST['Password'];

    $stmt = $conn->prepare("SELECT * FROM Users WHERE Username = ?");
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $result = $stmt->get_result();

```

```

if ($result->num_rows > 0) {
    $user = $result->fetch_assoc();
    if ($password === $user['Password']) {
        // if password is correct, store user in the session
        $_SESSION['Username'] = $username;
        header("Location: index.php"); // redirection after log in
        exit();
    } else {
        $error = "Invalid username or password.";
    }
} else {
    $error = "User not found.";
}
}

?>

<?php require_once "header.php"; ?>

<main>
    <h2>Login</h2>
    <?php if (isset($error)) { echo "<p style='color:red;'>$error</p>"; } ?>
    <form method="POST">
        <label for="username">Username</label>
        <input type="text" name="Username" id="username" required>
        <label for="password">Password</label>
        <input type="password" name="Password" id="password" required>
        <button type="submit">Login</button>
    </form>
</main>

<?php require_once "footer.php"; ?>

```

f) logout.php:

```

<?php
session_start();
session_destroy();
header('Location: login.php');
exit;
?>

```

g) register.php:

```
<?php
session_start();
require_once "db.php"; // connect to DB

// checking if a session has already started
if (isset($_SESSION['Username'])) {
    header("Location: index.php");
    exit();
}

// Handle the registration process
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['Username'];
    $password = $_POST['Password'];
    $confirm_password = $_POST['confirm_password'];
    $firstname = $_POST['FirstName'];
    $surname = $_POST['Surname'];
    $address1 = $_POST['AddressLine1'];
    $address2 = $_POST['AddressLine2'];
    $city = $_POST['City'];
    $telephone = $_POST['Telephone'];
    $mobile = $_POST['Mobile'];

    // Check if required fields are empty
    if (empty($username) || empty($password) || empty($firstname) ||
empty($surname) || empty($address1) || empty($city) || empty($telephone) ||
empty($mobile)) {
        $error = "All fields are required.";
    }

    // Validate password length
    elseif ($password !== $confirm_password) {
        $error = "Passwords do not match.";
    }
    // Validate if passwords match
    elseif (strlen($password) < 6) {
        $error = "Password must be at least 6 characters.";
    }
    // Validate mobile number format
    elseif (strlen($mobile) != 10 || !ctype_digit($mobile)) {
        $error = "Mobile number must contain 10 numeric characters.";
    }

    else {
        // Check if the username already exists in the database
        $stmt = $conn->prepare("SELECT * FROM Users WHERE Username = ?");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $result = $stmt->get_result();
        if ($result->num_rows > 0) {
```

```

{
    $error = "Username is already taken.";
} else {
    // Insert the new user into the database
    $stmt = $conn->prepare("INSERT INTO Users (Username, Password,
FirstName, Surname, AddressLine1, AddressLine2, City, Telephone, Mobile) VALUES
(?, ?, ?, ?, ?, ?, ?, ?, ?)");
    $stmt->bind_param("ssssssss", $username, $password, $firstname,
$surname, $address1, $address2, $city, $telephone, $mobile);
    if ($stmt->execute()) {
        $_SESSION['Username'] = $username; // Auto login after registration
        header("Location: index.php");
        exit();
    } else {
        $error = "Error: " . $stmt->error;
    }
}
}

?>

<?php require_once "header.php"; ?>

<main>
    <h2>Register</h2>
    <h4>• Choose a unique username.</h4>
    <h4>• Password must contain six characters.</h4>
    <h4>• Mobile phone numbers must contain 10 numeric characters.</h4>
    <?php if (isset($error)) { echo "<p style='color:red;'>$error</p>"; } ?>
    <form method="POST">
        <label for="username">Username</label>
        <input type="text" name="Username" id="username" required>

        <label for="password">Password</label>
        <input type="password" name="Password" id="password" required>

        <label for="confirm_password">Confirm Password</label>
        <input type="password" name="confirm_password" id="confirm_password"
required>

        <label for="firstname">First Name</label>
        <input type="text" name="FirstName" id="firstname" required>

        <label for="surname">Surname</label>
        <input type="text" name="Surname" id="surname" required>

        <label for="address1">Address Line 1</label>
        <input type="text" name="AddressLine1" id="address1" required>

        <label for="address2">Address Line 2</label>
        <input type="text" name="AddressLine2" id="address2">

        <label for="city">City</label>
        <input type="text" name="City" id="city" required>

        <label for="telephone">Telephone</label>
        <input type="text" name="Telephone" id="telephone" required>

        <label for="mobile">Mobile</label>
        <input type="text" name="Mobile" id="mobile" required>

        <button type="submit">Register</button>
    </form>
</main>

<?php require_once "footer.php"; ?>

```

h) search.php:

```
<?php
// checking if a session has already started
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

require_once "db.php"; // Connect to DB

// Number of rows to show per page
$rows_per_page = 5;

$search_term = isset($_GET['search']) && $_GET['search'] !== "" ? "%" .
$_GET['search'] . "%" : null; // Prepare the search term
$category = isset($_GET['category']) && $_GET['category'] !== "" ? 
$_GET['category'] : null;

// If both search term and category are empty, no results (set to 0)
if ($search_term === null && $category === null) {
    $total_rows = 0;
} else {
    $query = "SELECT Books.*, Category.CategoryDetails
              FROM Books
              JOIN Category ON Books.CategoryID = Category.CategoryID
              WHERE 1=1";

    // Add search term condition
    $params = [];
    $types = ""; // Initialize as an empty string

    if ($search_term) {
        $query .= " AND (Books.BookTitle LIKE ? OR Books.Author LIKE ?)";
        $params[] = $search_term;
        $params[] = $search_term;
        $types .= "ss";
    }

    // Add category condition
    if ($category) {
        $query .= " AND Books.CategoryID = ?";
        $params[] = $category;
        $types .= "s";
    }

    // Get total number of results
    $count_query = "SELECT COUNT(*) as total FROM Books
                   JOIN Category ON Books.CategoryID = Category.CategoryID
                   WHERE 1=1";

    if ($search_term) {
        $count_query .= " AND (Books.BookTitle LIKE ? OR Books.Author LIKE ?)";
    }
    if ($category) {
        $count_query .= " AND Books.CategoryID = ?";
    }
}
```

```

// Execute the query
$count_stmt = $conn->prepare($count_query);
if ($count_stmt === false) {
    die('Error in prepare: ' . $conn->error);
}

// Ensure $types is not empty before binding parameters
if (!empty($types)) {
    $count_stmt->bind_param($types, ...$params);
}
$count_stmt->execute();
$count_result = $count_stmt->get_result();
$total_rows = $count_result->fetch_assoc()['total']; // Get the total number of
results
$count_stmt->close();
}

// If no results, set $result to null
if ($total_rows == 0) {
    $result = null;
}

// Pagination variables
$total_pages = $total_rows > 0 ? ceil($total_rows / $rows_per_page) : 0; //
Calculate total pages only if results exist
$current_page = isset($_GET['page']) ? (int)$_GET['page'] : 1; // Get the current
page
$offset = ($current_page - 1) * $rows_per_page; // Calculate the OFFSET value

// Execute the query for results
if ($total_rows > 0) {
    $query .= " LIMIT ?, ?";
    $params[] = $offset;
    $params[] = $rows_per_page;
    $types .= "ii";

    // Prepare and execute the query
    $stmt = $conn->prepare($query);
    if ($stmt === false) {
        die('Error in prepare: ' . $conn->error);
    }
    if (!empty($types)) {
        $stmt->bind_param($types, ...$params);
    }
    $stmt->execute();
    $result = $stmt->get_result();
}
?>

<?php require_once "header.php"; ?>

```

```

<main>
    <h2>Search Results</h2>
    <?php if ($result && $result->num_rows > 0): ?>
        <table>
            <tr>
                <th>Title</th>
                <th>Author</th>
                <th>Edition</th>
                <th>Year</th>
                <th>Category</th>
                <th>Availability Status</th>
            </tr>
            <?php while ($row = $result->fetch_assoc()): ?>
                <tr>
                    <td><?php echo htmlspecialchars($row['BookTitle']); ?></td>
                    <td><?php echo htmlspecialchars($row['Author']); ?></td>
                    <td><?php echo htmlspecialchars($row['Edition']); ?></td>
                    <td><?php echo htmlspecialchars($row['Year']); ?></td>
                    <td><?php echo
                        htmlspecialchars($row['CategoryDetails']);
                    ?></td>
                    <td>
                        <?php if ($row['Reserved'] > 0): ?>
                            <span>Not Available: Already Reserved</span>
                        <?php else: ?>
                            <span><a href="reservation.php?ISBN=<?php echo
                                $row['ISBN'];
                            ?>">Available: Reserve</a></span>
                        <?php endif; ?>
                    </td>
                </tr>
            <?php endwhile; ?>
        </table>

        <!-- Pagination -->
        <div class="pagination">
            <?php if ($current_page > 1): ?>
                <a href="search.php?search=<?php echo
                    urlencode($search_term);
                ?>&category=<?php echo
                    urlencode($category);
                ?>&page=1">&laquo; First</a>
                <a href="search.php?search=<?php echo
                    urlencode($search_term);
                ?>&category=<?php echo urlencode($category);
                ?>&page=<?php
                    echo $current_page - 1;
                ?>">Previous</a>
            <?php endif; ?>

            <span>Page <?php echo $current_page; ?> of <?php echo
                $total_pages;
            ?></span>

            <?php if ($current_page < $total_pages): ?>
                <a href="search.php?search=<?php echo
                    urlencode($search_term);
                ?>&category=<?php echo urlencode($category);
                ?>&page=<?php
                    echo $current_page + 1;
                ?>">Next</a>
                <a href="search.php?search=<?php echo
                    urlencode($search_term);
                ?>&category=<?php echo urlencode($category);
                ?>&page=<?php
                    echo $total_pages;
                ?>">Last &raquo;</a>
            <?php endif; ?>
        </div>
        <?php else: ?>
            <p>No books found.</p>
        <?php endif; ?>
    </main>

    <?php require_once "footer.php"; ?>

```

i) reservation.php:

```
<?php
session_start();
require_once "db.php"; // Connect to DB

// Only allow access to logged-in users
if (!isset($_SESSION['Username'])) {
    header("Location: login.php"); // Redirect to login page
    exit();
}

// Handle book reservation
if (isset($_GET['ISBN'])) {
    $isbn = $_GET['ISBN'];
    $username = $_SESSION['Username'];

    // Check if the book exists and is available for reservation
    $stmt = $conn->prepare("SELECT * FROM Books WHERE ISBN = ?");
    $stmt->bind_param("s", $isbn);
    $stmt->execute();
    $result = $stmt->get_result();
    $book = $result->fetch_assoc();

    if ($book && $book['Reserved'] == 0) {
        // If the book is available, insert reservation into the ReservedBook table
        $stmt = $conn->prepare("INSERT INTO ReservedBook (ISBN, Username,
ReservedDate) VALUES (?, ?, NOW())");
        $stmt->bind_param("ss", $isbn, $username);
        $stmt->execute();

        // Update the reservation status to reserved(1)
        $stmt = $conn->prepare("UPDATE Books SET Reserved = 1 WHERE ISBN = ?");
        $stmt->bind_param("s", $isbn);
        $stmt->execute();

        // Redirect to the view_reservation page after successful reservation
        header("Location: view_reservations.php");
        exit();
    }
}
?>

<?php require_once "header.php"; ?>

<main>
    <h2>Reservation Success</h2>
    <p>Your reservation has been successfully made!</p>
    <a href="view_reservations.php">View My Reservations</a>
</main>

<?php require_once "footer.php"; ?>
```

j) view_reservatons.php:

```
<?php
session_start();
require_once "db.php"; // Connect to DB

// Check if the user is logged in
if (!isset($_SESSION['Username'])) {
    header("Location: login.php"); // Redirect to login page if not logged in
    exit();
}

$username = $_SESSION['Username']; // Get the username
$stmt = $conn->prepare("SELECT ReservedBook.*, Books.BookTitle FROM ReservedBook
JOIN Books ON ReservedBook.ISBN = Books.ISBN WHERE ReservedBook.Username = ?");
$stmt->bind_param("s", $username); // Bind the usuername to the query
$stmt->execute();
$result = $stmt->get_result(); // Execute the query and get the result
?>

<?php require_once "header.php"; ?>

<main>
    <h2>My Reservations</h2>
    <?php if ($result->num_rows > 0): ?>
        <table>
            <tr>
                <th>Book Title</th>
                <th>Reserved Date</th>
                <th>Action</th>
            </tr>
            <?php while ($row = $result->fetch_assoc()): ?>
                <tr>
                    <td><?php echo htmlspecialchars($row['BookTitle']); ?></td>
                    <td><?php echo htmlspecialchars($row['ReservedDate']); ?></td>
                    <td>
                        <a href="cancel_reservation.php?ISBN=<?php echo
$row['ISBN']; ?>">Cancel</a>
                    </td>
                </tr>
            <?php endwhile; ?>
        </table>
    <?php else: ?>
        <p>No reservations yet.</p>
    <?php endif; ?>
</main>
```

k) cancel_reservation.php:

```
<?php
session_start();
require_once "db.php"; // Connect to DB

// Only allow access to logged-in users
if (!isset($_SESSION['Username'])) {
    header("Location: login.php"); // Redirect to the login page if the user is not
logged in
    exit();
}

if (isset($_GET['ISBN'])) {
    $isbn = $_GET['ISBN']; // Get the ISBN of the book to cancel the reservation
    $username = $_SESSION['Username'];

    // Cancel the reservation by deleting the record from the ReservedBook table
    $stmt = $conn->prepare("DELETE FROM ReservedBook WHERE ISBN = ? AND Username
= ?");
    $stmt->bind_param("ss", $isbn, $username);
    $stmt->execute();

    // Update the reservation status to not reserved(0)
    $stmt = $conn->prepare("UPDATE Books SET Reserved = 0 WHERE ISBN = ?");
    $stmt->bind_param("s", $isbn);
    $stmt->execute();

    header("Location: view_reservations.php"); // Redirect to view_reservation page
after canceling the reservation
    exit();
}
?>
```

I) style.css:

```
/* Basic style for the entire page */
body {
    display: flex;
    flex-direction: column;
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
    min-height: 100vh;
}

main {
    flex: 1;
    text-align: center;
}

/* Header style */
header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    background-color: #503154;
    padding: 10px 20px;
    border-bottom: 1px solid #efefef;
}

.banner img{
    width: 100%;
    height: 100%;
    display: block;
    object-fit: cover;
}

/* Navigation style */
nav {
    margin-top: 10px;
}

nav ul {
    list-style: none;
    display: flex;
    justify-content: flex-end;
    padding: 0;
}

nav ul li {
    margin-left: 20px;
    display: inline;
}
```

```
nav ul li a {
    text-decoration: none;
    color: white; /* 기본 메뉴 색상 */
    font-size: 1.1em;
    font-weight: bold;
}

nav ul li a:hover {
    color: #a877bf;
    text-decoration: underline;
}

/* Reservation list table style */
table {
    width: 80%;
    margin: 20px auto;
    border-collapse: collapse;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

table th, table td {
    padding: 15px;
    text-align: left;
    border: 1px solid #ddd;
}

table th {
    background-color: #f2f2f2;
}

table tr:nth-child(even) {
    background-color: #f9f9f9;
}

table tr:hover {
    background-color: #e9e9e9;
}

/* Common form style for login, signup, and search forms */
form {
    display: flex;
    flex-direction: column;
    align-items: center;
    max-width: 400px;
    margin: 30px auto;
    padding: 50px;
    background-color: white;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    border-radius: 8px;
}
```

```
form input[type="text"], form input[type="password"], form input[type="email"],
form input[type="number"] {
    width: 100%;
    padding: 12px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 4px;
    font-size: 1em;
}

form input[type="text"]:focus, form input[type="password"]:focus, form
input[type="email"]:focus, form input[type="number"]:focus {
    border-color: #007bff;
    outline: none;
}

form button {
    margin-top: 30px;
    padding: 12px 20px;
    background-color: #503154;
    color: white;
    border: none;
    border-radius: 8px;
    font-size: 1.1em;
    cursor: pointer;
    transition: background-color 0.3s, transform 0.2s;
}

form button:hover {
    background-color: #7b5079;
    transform: scale(1.05);
}

/* Library banner image style */
.library-banner {
    position: relative;
    width: 100%;
    height: auto;
    max-height: 800px;
    overflow: hidden;
}

.library-banner img {
    width: 100%;
    height: 100%;
    object-fit: cover;
    object-position: bottom;
    display: block;
}
```

```
/* Login and signup button container style */
.library-banner .btn_container {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    display: flex;
    flex-direction: column;
    align-items: center;
    width: 100%;
    max-width: 400px;
    padding: 40px;
    background-color: rgba(255, 255, 255, 0.8);
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    border-radius: 20px;
}

/* Login and signup message style */
.library-banner .btn_container p {
    font-size: 1.3em;
    font-weight: bold;
    color: #503154;
    margin-bottom: 20px;
    text-align: center;
}

/* Button style */
.library-banner .index_button {
    width: 125px;
    padding: 12px;
    margin: 10px 25px;
    background-color: #503154;
    color: white;
    border: none;
    border-radius: 8px;
    font-size: 1.1em;
    cursor: pointer;
    transition: background-color 0.3s, transform 0.2s;
    text-align: center;
}

.library-banner .index_button:hover {
    background-color: #7b5079;
    transform: scale(1.05);
}
```

```
/* Error message style for input fields */
form p {
    color: red;
    font-size: 1em;
    margin: 0;
}

/* Mobile number and password requirements */
h4, p {
    font-size: 1.1em;
    color: #333;
    text-align: center;
    margin-top: 20px;
}

/* Message shown after login */
main p {
    font-size: 1.2em;
    text-align: center;
    color: #333;
}

/* User welcome message */
.welcome-message {
    text-align: center;
    margin-bottom: 20px;
}

.welcome-message h2 {
    font-size: 2em;
    color: #503154;
    margin: 5px 0;
}

/* reservation button container */
.reservation_container {
    display: flex;
    justify-content: center;
    margin-top: 30px;
}

#viewReserve {
    padding: 15px 25px;
    background-color: #503154;
    color: white;
    font-size: 1.2em;
    font-weight: bold;
    text-decoration: none;
    border-radius: 10px;
    transition: background-color 0.3s, transform 0.2s;
}
```

```
#viewReserve:hover {
    background-color: #7b5079;
    transform: scale(1.05);
}

/* Footer style */
footer {
    text-align: center;
    padding: 10px 20px;
    margin-top: auto;
    background-color: #503154;
    color: white;
    font-size: 0.9em;
    bottom: 0;
    width: 100%;
    box-shadow: 0 -2px 5px rgba(0, 0, 0, 0.1);
}

footer a {
    color: #ffffff;
    text-decoration: underline;
}

footer a:hover {
    color: #00bcd4;
    text-decoration: none;
}

footer p {
    color: white;
}

/* Label style for input fields */
label {
    font-size: 1em;
    font-weight: bold;
    margin-bottom: 5px;
    color: #333;
    text-align: left;
    width: 100%;
}

main {
    padding: 30px;
}

h2 {
    text-align: center;
    color: #333;
    font-size: 2em;
}
```

```
/* Dropdown style */
select {
    width: 107%;
    padding: 12px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 4px;
    font-size: 1em;
}

select:focus {
    border-color: #007bff;
    outline: none;
}

/* Pagination style */
.pagination {
    display: flex;
    justify-content: center;
    align-items: center;
    margin-top: 20px;
    font-size: 18px;
    position: relative;
}

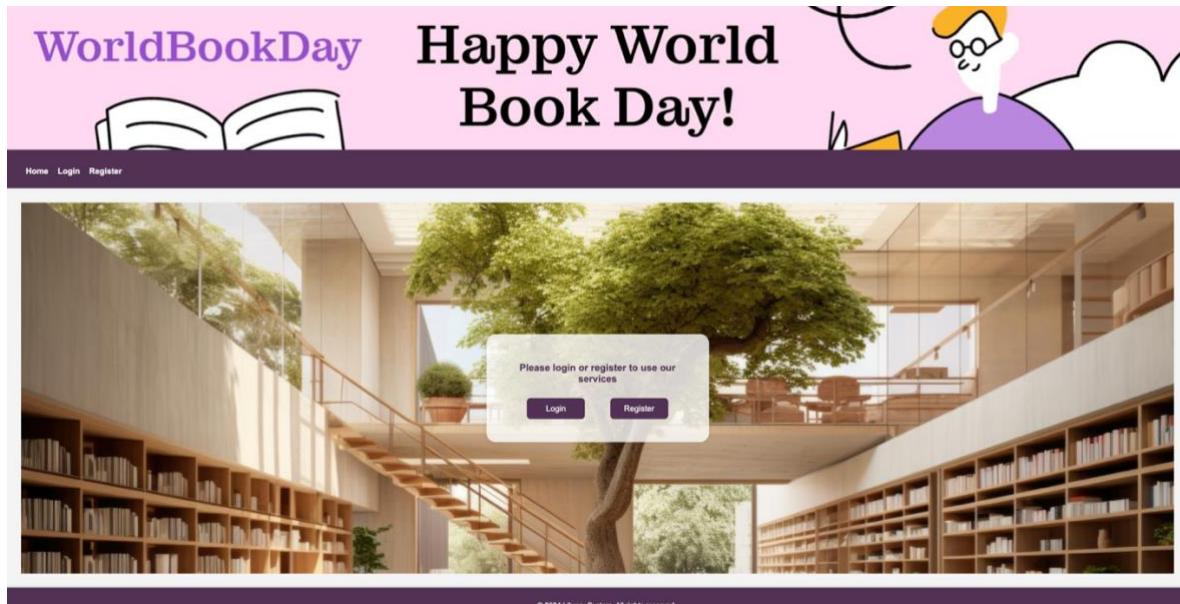
.pagination a {
    margin: 0 10px;
    padding: 8px 16px;
    background-color: #f4f4f4;
    border-radius: 5px;
    color: #333;
    text-decoration: none;
}

.pagination a:hover {
    background-color: #7b5079;
    color: white;
}

.pagination span {
    margin: 0 10px;
    font-size: 16px;
}
```

3. Snaps Of The Application

1-a) index page (home page without login or register):



The user can log in or register using the login and register buttons. Other services, such as book search or book registration, are not available until the user logs in.

1-b) index page (home page when the user login or register):



Once the user logs in or registers, the homepage directly displays a greeting message and the book search form. The user can also view their reservations by clicking the reservation button.

2-a) login page:

The image shows a login page for a library system. The header features the text "WorldBookDay" and "Happy World Book Day!" with a cartoon illustration of a person reading a book. The main content area is titled "Login" and contains fields for "Username" and "Password" with a "Login" button. The footer includes copyright information: "© 2024 Library System. All rights reserved." and "Contact: d2127506@myyuhu.ie".

WorldBookDay

Happy World Book Day!

Home Login Register

Login

Username

Password

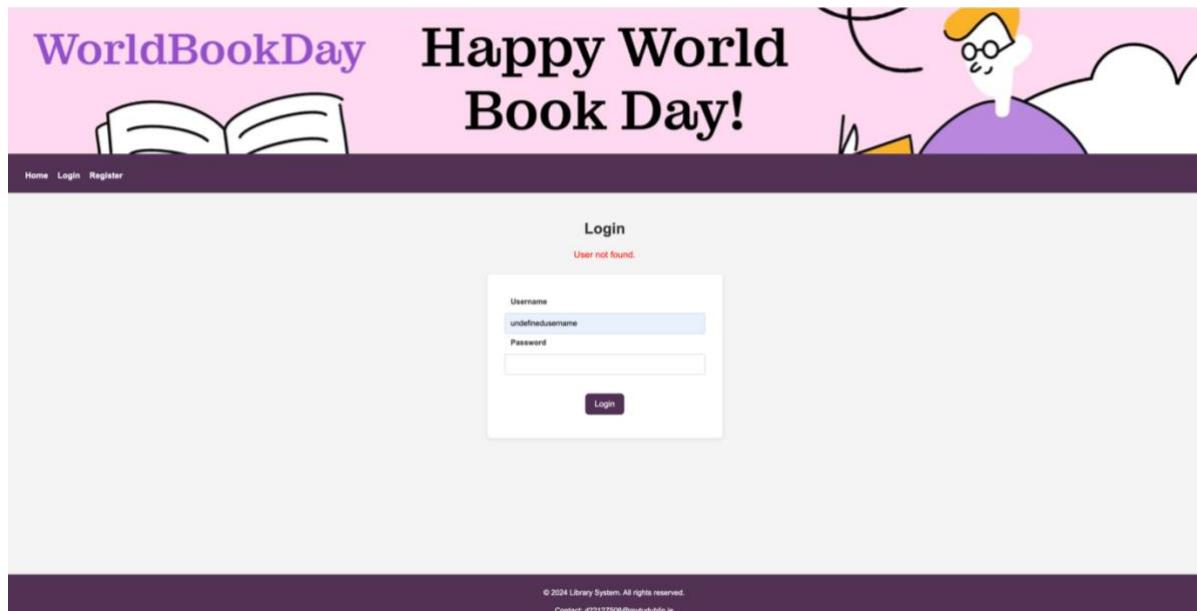
Login

© 2024 Library System. All rights reserved.
Contact: d2127506@myyuhu.ie

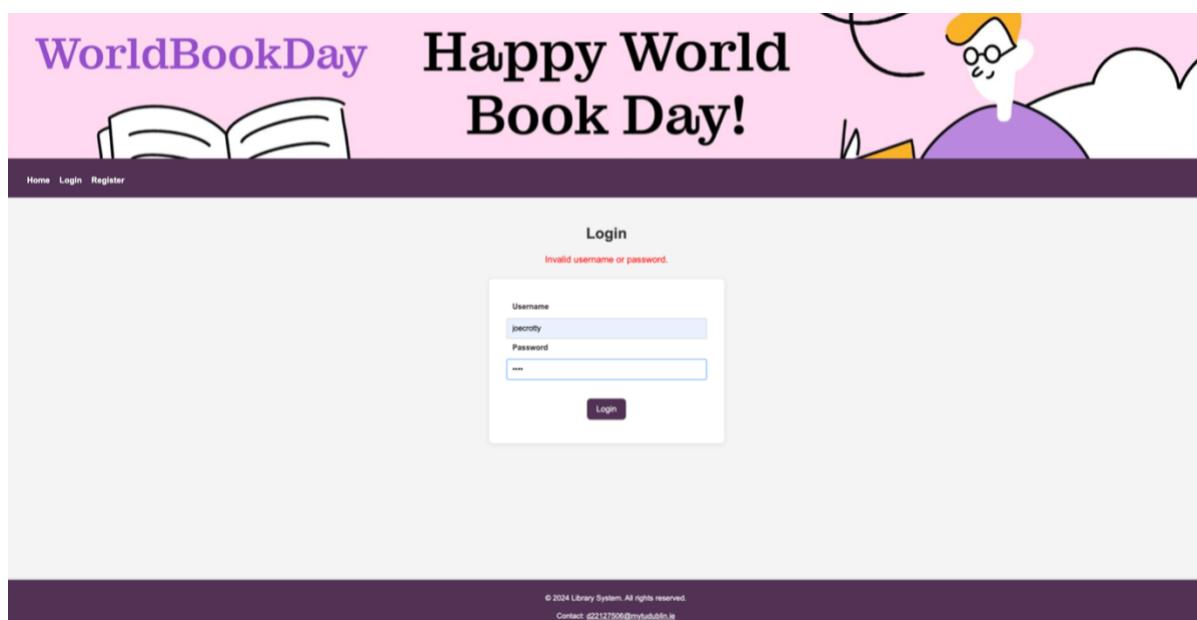
2-b) login page – validation check:

Check if the username exists in the database and verify if the user has entered the correct password.

When the username doesn't exist



When the user enters wrong password



3-a) Register page:

The screenshot shows the homepage of a website for "Happy World Book Day". The header features the text "WorldBookDay" and "Happy World Book Day!" with a cartoon illustration of a person reading a book. Below the header is a dark purple navigation bar with links for "Home", "Login", and "Register". The main content area is titled "Register" and contains the following validation rules:

- * Choose a unique username.
- * Password must contain six characters.
- * Mobile phone numbers must contain 10 numeric characters.

A registration form is displayed with fields for Username, Password, Confirm Password, First Name, Surname, Address Line 1, Address Line 2, City, Telephone, and Mobile. A "Register" button is at the bottom of the form.

3-b) Register page (Validation check):

When the user submits the form without filling in the required fields.

This screenshot shows the same registration form as above, but with validation errors. The "Password" field is highlighted in blue, indicating it is required. The validation rules remain the same as in the previous screenshot:

- * Choose a unique username.
- * Password must contain six characters.
- * Mobile phone numbers must contain 10 numeric characters.

The other fields (Username, First Name, Surname, Address Lines, City, Telephone, and Mobile) are empty and have no validation errors.

When the password is less than 6 characters.

The screenshot shows a registration form titled "Register". Below the title are three validation errors in red text:

- Choose a unique username.
- Password must contain six characters.
- Mobile phone numbers must contain 10 numeric characters.

A red message at the bottom right states "Password must be at least 6 characters." The "Username" field is empty, and the "Password" field is also empty.

When the username already exists in the database

The screenshot shows a registration form titled "Register". Below the title are three validation errors in red text:

- Choose a unique username.
- Password must contain six characters.
- Mobile phone numbers must contain 10 numeric characters.

A red message at the bottom right states "Username is already taken." The "Username" field contains the value "joecrotty".

When the password and confirm password do not match

The screenshot shows a registration form titled "Register". Below the title are three validation errors in red text:

- Choose a unique username.
- Password must contain six characters.
- Mobile phone numbers must contain 10 numeric characters.

A red message at the bottom right states "Passwords do not match." The "Username" field contains "joecrotty". The "Password" field contains "*****" and the "Confirm Password" field also contains "*****".

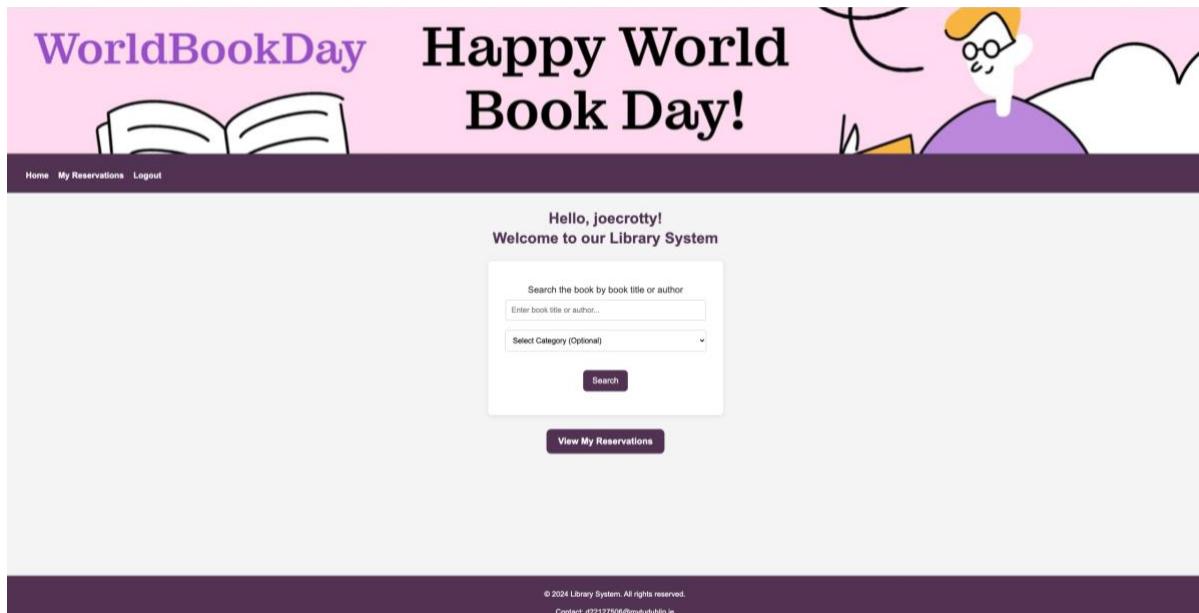
When the mobile number contains alphabetical characters or has fewer than 10 numeric characters

The screenshot shows a registration form titled "Register". Below the title are three validation errors in red text:

- Choose a unique username.
- Password must contain six characters.
- Mobile phone numbers must contain 10 numeric characters.

A red message at the bottom right states "Mobile number must contain 10 numeric characters." The "Username" field contains "heeyeon".

4-a) Search page:



4-b) Search page:

When the user searches for the name of a book.

The screenshot shows the search results page. At the top, it displays the same "Hello, joecrotty!" and "Welcome to our Library System" message as the previous page. Below this is the search form, which now has "Computers in Business" entered into the title field. The search button is visible below the form. A "View My Reservations" link is also present. The main content area is titled "Search Results" and contains a table with one row of data. The table columns are "Title", "Author", "Edition", "Year", "Category", and "Availability Status". The data in the table is as follows:

Title	Author	Edition	Year	Category	Availability Status
Computers in Business	Alicia O'Neill	3	1997	Biography	Available: Reserve

Below the table, a small note says "Page 1 of 1".

The user can reserve the book if the availability status is “Available: Reserve” by clicking on it.

When there are more than 5 rows of results, pagination is displayed (e.g., "Page 1 of 3"). Users can navigate to the next page by clicking the buttons.

Hello, joecrotty!

Welcome to our Library System

Search the book by book title or author

Select Category (Optional)

[View My Reservations](#)

Search Results

Title	Author	Edition	Year	Category	Availability Status
A guide to nutrition	John Thorpe	2	1997	Health	Available: Reserve
My life in picture	Kevin Graham	8	2004	Health	Available: Reserve
My ranch in Texas	George Bush	1	2005	Health	Not Available: Already Reserved
Exploring Peru	Stephanie Birchie	4	2005	Business	Available: Reserve
Business Strategy	Joe Peppard	2	2002	Business	Available: Reserve

Page 1 of 3 [Next](#) [Last »](#)

When there are no matching results in the dataset.

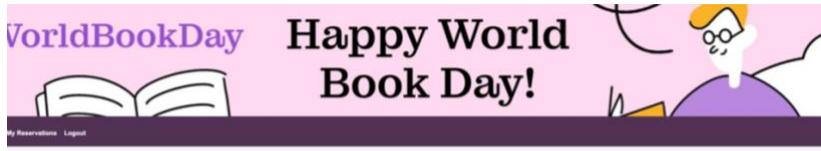
Hello, joecrotty!

Welcome to our Library System

Search the book by book title or author

Business

[View My Reservations](#)



WorldBookDay

Happy World Book Day!

Search Results

No books found.

5-a) My reservation page:

When the user clicks “Available: Reserve,” they are directly taken to the "My Reservations" page. Alternatively this page can be accessed through the menu bar or the “View My Reservations” button below the search form.

Book Title	Reserved Date	Action
Computers in Business	2024-12-05	Cancel
My ranch in Texas	2008-10-11	Cancel

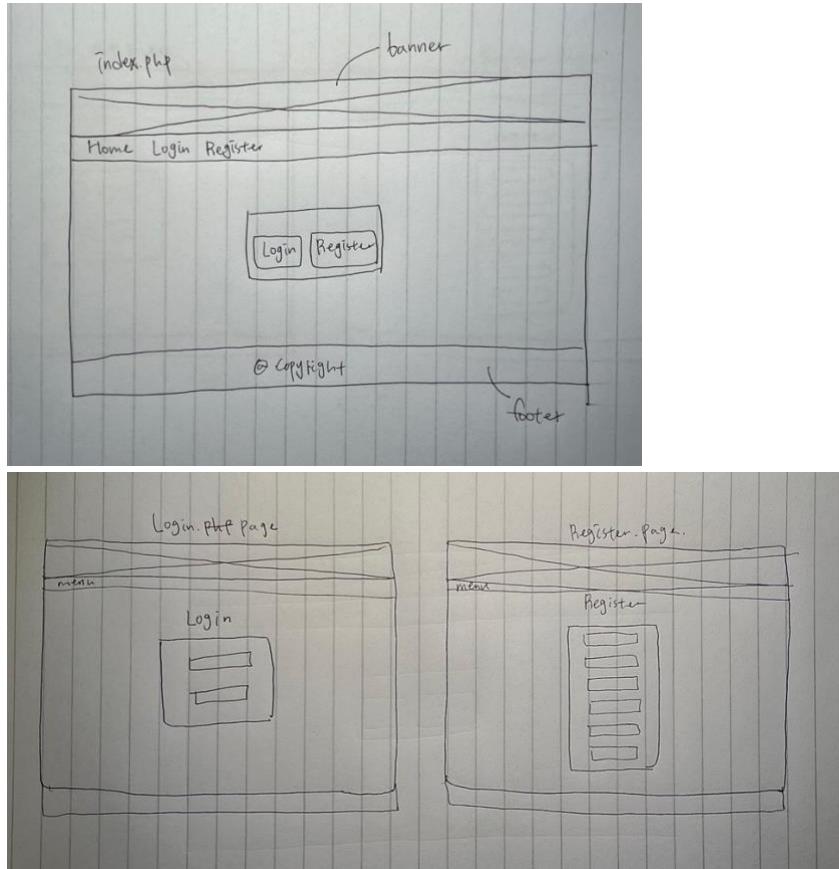
You can cancel a reservation by clicking the “Cancel” button. When the button is clicked, the reservation is released in the database, and the cancellation is completed.
(Reservation for *Computers in Business* after cancellation.)

Book Title	Reserved Date	Action
My ranch in Texas	2008-10-11	Cancel

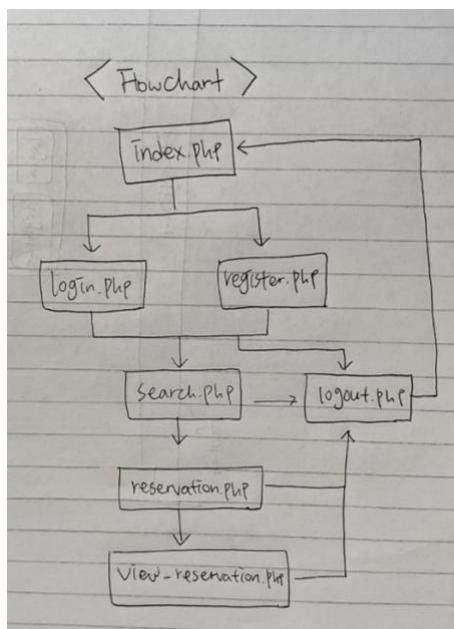
6) When the user clicks the logout button, they are redirected to the index page

4. Design Document

1) Low Fidelity Prototype



2) Flowchart



3-a) Database – Books Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ISBN	varchar(255)	utf8mb4_general_ci		No	None			More
2	BookTitle	varchar(255)	utf8mb4_general_ci		Yes	NULL			More
3	Author	varchar(255)	utf8mb4_general_ci		Yes	NULL			More
4	Edition	int(11)			Yes	NULL			More
5	Year	int(11)			Yes	NULL			More
6	CategoryID	int(11)			Yes	NULL			More
7	Reserved	int(1)			No	None			More

	ISBN	BookTitle	Author	Edition	Year	CategoryID	Reserved
	093-403992	Computers in Business	Alicia O'Neill	3	1997	3	0
	23472-8729	Exploring Peru	Stephanie Birchie	4	2005	2	0
	237-34823	Business Strategy	Joe Peppard	2	2002	2	0
	238-923849	A guide to nutrition	John Thorpe	2	1997	1	0
	2983-3494	Cooking for children	Anabelle Sharpe	1	2003	7	0
	82n8-308	computers for idiots	Susan O'Neill	5	1998	4	0
	9823-23984	My life in picture	Kevin Graham	8	2004	1	0
	9823-2403-0	DaVinci Code	Dan Brown	1	2003	8	0
	9823-98345	How to cook Italian food	Jamie Oliver	2	2005	7	1
	9823-98487	Optimising your business	Cleo Blair	1	2001	2	0
	98234-029384	My ranch in Texas	George Bush	1	2005	1	1
	988745-234	Tara Road	Maeve Binchy	4	2002	8	0
	993-004-00	My life in bits	John Smith	1	2001	1	0
	9987-0039882	Shooting History	Jon Snow	1	2003	1	0

3-b) Database – Users Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Username	varchar(30)	utf8mb4_general_ci		No	None			More
2	Password	text	utf8mb4_general_ci		No	None			More
3	FirstName	text	utf8mb4_general_ci		No	None			More
4	Surname	text	utf8mb4_general_ci		No	None			More
5	AddressLine1	text	utf8mb4_general_ci		No	None			More
6	AddressLine2	text	utf8mb4_general_ci		No	None			More
7	City	text	utf8mb4_general_ci		No	None			More
8	Telephone	int(11)			No	None			More
9	Mobile	int(11)			No	None			More

	Username	Password	FirstName	Surname	AddressLine1	AddressLine2	City	Telephone	Mobile
	alanjmckenna	t1234s	Alan	McKenna	38 Cranley Road	Fairview	Dublin	9998377	8887889
	joecrotty	kj7889	Joseph	Crotty	Apt 5 Clyde Road	Donnybrook	Dublin	8887889	876654456
	tommy100	123456	Tom	Bahan	14 Hyde Road	Dalkey	Dublin	9983747	876738782

3-c) Database – ReservedBook Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□ 1	ISBN	varchar(255)	utf8mb4_general_ci		No	None		Change Drop More	
□ 2	Username	varchar(50)	utf8mb4_general_ci		No	None		Change Drop More	
□ 3	ReservedDate	date			No	None		Change Drop More	

	ISBN	Username	ReservedDate
□	Edit Copy Delete 9823-98345	tommy100	2008-10-11
□	Edit Copy Delete 98234-029384	joecrotty	2008-10-11

3-d) Database – Category Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□ 1	CategoryID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
□ 2	CategoryDetails	varchar(222)	utf8mb4_general_ci		No	None			Change Drop More

	CategoryID	CategoryDetails
□	Edit Copy Delete 1	Health
□	Edit Copy Delete 2	Business
□	Edit Copy Delete 3	Biography
□	Edit Copy Delete 4	Technology
□	Edit Copy Delete 5	Travel
□	Edit Copy Delete 6	Self-Help
□	Edit Copy Delete 7	Cookery
□	Edit Copy Delete 8	Fiction