

# Documentation of Forms for Online Bookstore Project

The provided forms are part of the authentication and user management system for an online bookstore. The forms include sign-in, sign-up, and password retrieval functionalities. Below is the detailed documentation for each form:

## Sign-In Form (`sign-in.php`)

PHP Script for Authentication

```
<?php
session_start();
$server_name = "localhost";
$username = "root";
$password = "";
$database_name = "online_bookstore";
$conn = mysqli_connect($server_name, $username, $password,
$database_name);
if (!$conn) {
    die("Connection Failed: " . mysqli_connect_error());
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $Email = mysqli_real_escape_string($conn, $_POST['Email']);
    $Password = mysqli_real_escape_string($conn, $_POST['Password']);

    if (!empty($Email) && !empty($Password) && !is_numeric($Email)) {
        $query = "SELECT * FROM signup_data WHERE Email = '$Email' LIMIT
1";
        $result = mysqli_query($conn, $query);

        if ($result) {
            if (mysqli_num_rows($result) > 0) {
                $user_data = mysqli_fetch_assoc($result);

                if ($Password == $user_data['Password']) {
                    $_SESSION['user_name'] = $user_data['Username'];
                    $_SESSION['Email'] = $user_data['Email'];
                    header("Location: index.php");
                    die();
                } else {
                    echo "<script type='text/javascript'>alert('Wrong
username or password')</script>";
                }
            } else {

```

```

        echo "<script type='text/javascript'>alert('User not
found')</script>";
    }
    } else {
        echo 'Query error: ' . mysqli_error($conn);
    }
    } else {
        echo "<script type='text/javascript'>alert('Invalid email or
password')</script>";
    }
}
?>

```

- Purpose: Authenticates user credentials.
- Details:
  - Database Connection: Connects to the `online\_bookstore` database.
  - Input Sanitization: Uses `mysqli\_real\_escape\_string` to prevent SQL injection.
  - Query Execution: Checks if the user exists and validates the password.
  - Session Management: Stores user information in session variables upon successful login.
  - Error Handling: Displays appropriate alerts for different error scenarios.

### HTML Form for Sign-In

```

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>signin-form</title>
    <link rel="shortcut icon" href="../images/open-book.png"
type="image/png">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css">
</head>
<body>
    <header class="header">
        <div class="header-1">
            <a href=" " class="logo"><i class="fas fa-book"></i>
BookHeaven </a>
        </div>
    </header>
    <div class="login-form-container">

```

```

        <form action="sign-in.php" method="post">
            <h2>Sign In</h2>
            <span>Username</span>
            <input type="email" name="Email" class="box"
placeholder="enter your email" id="" required>
            <span>Password</span>
            <input type="password" name="Password" class="box"
placeholder="enter your password" id="" required>
            <div class="checkbox">
                <input type="checkbox" name="" id="remember-me">
                <label for="remember-me"> remember me</label>
            </div>
            <input type="submit" value="sign in" class="btn">
            <p><i>forget password ? <a href="forget.php">click
here</a></i></p>
            <p><i>dont't have an account ? <a href="sign-up.php">create
one</a></i></p>
        </form>
    </div>
</body>
</html>

```

- Purpose: Collects user credentials for login.
- Details:
  - Input Fields: Email and password fields.
  - Additional Options: Checkbox for "remember me," links for password recovery and account creation.

### Sign-Up Form (`sign-up.php`)

#### HTML Form for Sign-Up

```

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>signup-form</title>
    <link rel="shortcut icon" href="../images/open-book.png"
type="image/png">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css">
</head>

```

```

<body>
  <header class="header">
    <div class="header-1">
      <a href=" " class="logo"><i class="fas fa-book"></i>
BookHeaven </a>
    </div>
  </header>
  <div class="signup-form-container">
    <form action="sign-up.php" method="post">
      <h2>Sign Up</h2>
      <span>User Name</span>
      <input type="text" name="Username" class="box"
placeholder="Enter your User-Name" id="full_name" maxlength="20"
pattern="^\S+$" title="No spaces allowed" required>
      <span>Full Name</span>
      <input type="text" name="Fullname" class="box"
placeholder="Enter your Full Name" id="full_name" maxlength="25" required>
      <span>Email</span>
      <input type="email" name="Email" class="box"
placeholder="Enter your email" id="email" maxlength="35" pattern="^\S+$"
title="No spaces allowed" required>
      <span>Phone</span>
      <input type="tel" name="Phone" class="box" placeholder="Enter
your phone number" id="phone" maxlength="10" pattern="[0-9]{10}"
title="Please enter a 10-digit mobile number" required>
      <span>Password</span>
      <input type="password" name="Password" class="box"
placeholder="Enter your password" id="password" maxlength="20"
pattern=".{8,}" title="Password must be at least 8 characters long"
required>
      <input type="submit" name="sign_up" value="Sign Up"
class="btn">
      <p class="message"><i>Already have an account? <a href="sign-
in.php">Sign in</a></i></p>
    </form>
    <p class="message"><i>People who use our service may have uploaded
your contact information to BookHeaven. <a href=" ">Learn
more</a></i></p><br>
    <p class="message"><i>By signing up, you agree to our <a href="
">Terms</a>,<a href=" ../html/help.html">Privacy policy</a> and <a href="
">Cookies Policy</a></i></p>
  </div>
</body>
</html>

```

- Purpose: Collects user information for account creation.

- Details:

- Input Fields: Username, full name, email, phone, and password fields with validation rules.
- Links: Prompts existing users to sign in.

### PHP Script for Registration

```
<?php
$server_name="localhost";
$username="root";
$password="";
$database_name="online_bookstore";
$conn = mysqli_connect($server_name, $username, $password,
$database_name);
if (!$conn) {
    die("Connection Failed: " . mysqli_connect_error());
}

if (isset($_POST['sign_up'])) {
    $Username = mysqli_real_escape_string($conn, $_POST['Username']);
    $Fullname = mysqli_real_escape_string($conn, $_POST['Fullname']);
    $Email = mysqli_real_escape_string($conn, $_POST['Email']);
    $Phone = mysqli_real_escape_string($conn, $_POST['Phone']);
    $Password = mysqli_real_escape_string($conn, $_POST['Password']);

    $sql_query = "INSERT INTO signup_data (Username, Fullname, Email,
Phone, Password)
VALUES ('$Username', '$Fullname', '$Email', '$Phone',
'$Password')";

    if (mysqli_query($conn, $sql_query)) {
        echo "<script type='text/javascript'>alert('Successfully
registered')</script>";
    } else {
        echo "<script type='text/javascript'>alert('Enter valid
information')</script>";
    }

    mysqli_close($conn);
}
?>
```

- Purpose: Handles the registration process and stores user data in the database.
- Details:

- Database Connection: Connects to the `online\_bookstore` database.
- Input Sanitization: Uses `mysqli\_real\_escape\_string` to prevent SQL injection.
- Data Insertion: Inserts user data into the `signup\_data` table.
- Feedback: Displays alerts based on the success or failure of the registration process.

## Password Retrieval Form (`forget.php`)

HTML Form for Password Retrieval

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Forget Password Form</title>
  <link rel="shortcut icon" href="../images
/open-book.png" type="image/png">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css">
</head>
<body>
  <header class="header">
    <div class="header-1">
      <a href=" " class="logo"><i class="fas fa-book"></i>
BookHeaven </a>
    </div>
  </header>
  <div class="signup-form-container">
    <form action="" method="post">
      <h2>Forget Password</h2>
      <span>Email</span>
      <input type="email" name="Email" class="box"
placeholder="Enter your email" id="email" required>
      <input type="submit" name="submit" value="Submit" class="btn">
      <p class="message"><i>Remember your password? <a href="sign-
in.php">Sign in</a></i></p>
    </form>
    <p class="message"><i>By submitting, you agree to our <a href="
">Terms</a>, <a href=" " ">Privacy policy</a>, and <a href=" " ">Cookies
Policy</a></i></p>
```

```
</div>
</body>
</html>
```

- Purpose: Collects user email for password retrieval.
- Details:
  - Input Field: Email field to receive the user's registered email address.
  - Links: Prompts users to sign in if they remember their password.

## PHP Script for Password Retrieval

```
<?php
$server_name = "localhost";
$username = "root";
$password = "";
$database_name = "online_bookstore";
$conn = mysqli_connect($server_name, $username, $password,
$database_name);
if (!$conn) {
    die("Connection Failed: " . mysqli_connect_error());
}

if (isset($_POST['submit'])) {
    $Email = $_POST['Email'];
    $sql_query = "SELECT PASSWORD FROM `signup_data` WHERE Email =
'$Email'";
    $result = mysqli_query($conn, $sql_query);

    if ($result && mysqli_num_rows($result) > 0) {
        $row = mysqli_fetch_assoc($result);
        $password = $row['PASSWORD'];
        echo "<p class='password-display'>Your password is:
<strong>$password</strong></p>";
    } else {
        echo "<p class='password-display'>Invalid email address. Please
try again.</p>";
    }

    mysqli_close($conn);
}
?>
```

- Purpose: Retrieves and displays the user's password.

- Details:
  - Database Connection: Connects to the ``online bookstore`` database.
  - Email Validation: Checks if the provided email exists in the database.
  - Password Retrieval: Fetches the user's password from the database and displays it.
  - Feedback: Provides appropriate messages based on the success or failure of the email lookup.

## Documentation of Edit Profile Form for Online Bookstore Project

The edit profile form allows users to update their personal information, including their username, full name, phone number, and password. Below is the detailed documentation for the PHP script and HTML form.

### PHP Script for Profile Update

#### Initialization and Database Connection

```
<?php
session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Edit Profile</title>
</head>
<body>
```

- Purpose: Starts the session and sets up the initial HTML document structure.
- Details:
  - ``session_start();`` initializes the session to access session variables.
  - Basic HTML head setup with meta tags and title.



## Header and Form Structure

```
<header class="header">
  <div class="header-1">
    <a href=" " class="logo"><i class="fas fa-book"></i> BookHeaven </a>
  </div>
</header>
<div class="signup-form-container">
  <h2>Edit Profile</h2>
  <form action="profile.php" method="post">
    <p class="message"><i>Enter your email that you want to change
details</i></p>
    <span>Email</span>
    <input type="email" name="Email" class="box" placeholder="Enter your
email" id="email" value="<?php echo $_SESSION['Email']; ?>" disabled>
    <hr style="color: black;">
    <span>User Name</span>
    <input type="text" name="Username" class="box" placeholder="Enter
your userName" id="full_name" value="<?php echo $_SESSION['user_name'];
?>">
    <span>Full Name</span>
    <input type="text" name="Fullname" class="box" placeholder="Enter
your fullName" id="full_name" required>
    <span>Phone</span>
    <input type="tel" name="Phone" class="box" placeholder="Enter your
phone number" id="phone" required>
    <span>Password</span>
    <input type="password" name="Current_Password" class="box"
placeholder="Current Password" required>
    <span>New Password</span>
    <input type="password" name="New_Password" class="box"
placeholder="Enter new password">
    <input type="submit" name="update_profile" value="Update Profile"
class="btn">
  </form>
  <p class="message"><i>Return to <a
href="http://localhost/website/html/help.html">Help</a></i></p>
</div>
```

- Purpose: Creates a form to update user profile information.
- Details:
  - Header: Contains the website logo.

- Form: Includes input fields for username, full name, phone, current password, and new password.
- Email Field: Displayed as disabled to show the user's current email, which cannot be changed.
- Submit Button: Triggers the form submission to update the profile.
- Help Link: Provides a link to the help page.

### PHP Script for Processing Form Submission

```
<?php
$server_name = "localhost";
$username = "root";
$password = "";
$database_name = "online_bookstore";
$conn = mysqli_connect($server_name, $username, $password,
$database_name);

if (!$conn) {
    die("Connection Failed: " . mysqli_connect_error());
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form data
    $Username = $_POST['Username'];
    $Fullname = $_POST['Fullname'];
    $Phone = $_POST['Phone'];
    $Current_Password = $_POST['Current_Password'];
    $New_Password = $_POST['New_Password']; // Optional
    $Email = $_POST['Email'];

    // Validate current password before update
    $query = "SELECT * FROM signup_data WHERE Email = '$Email' LIMIT 1";
    $result = mysqli_query($conn, $query);

    if ($result) {
        if (mysqli_num_rows($result) > 0) {
            $user_data = mysqli_fetch_assoc($result);
            if ($user_data['Password'] == $Current_Password) {
                // Update user details
                $update_query = "UPDATE signup_data SET Username =
'$Username', Fullname = '$Fullname', Phone = '$Phone'";
                if (!empty($New_Password)) {
                    $update_query .= ", Password = '$New_Password'";
                }
                $update_query .= " WHERE Email = '$Email'";
            }
        }
    }
}
```

```

        if (mysqli_query($conn, $update_query)) {
            echo "<script type='text/javascript'>alert('Profile
updated successfully!')</script>";
        } else {
            echo "Error updating profile: " . mysqli_error($conn);
        }
    } else {
        echo "<script type='text/javascript'>alert('Wrong current
password')</script>";
    }
    } else {
        echo "<script type='text/javascript'>alert('User not
found')</script>";
    }
}
mysqli_close($conn);
}
?>
</body>
</html>

```

- Purpose: Handles the form submission and updates user profile information in the database.

- Details:

- Database Connection: Connects to the `online\_bookstore` database.
- Form Data Retrieval: Retrieves data from the form using `\$\_POST`.
- Password Validation: Checks if the current password matches the stored password for the provided email.
- Profile Update: If the current password is correct, updates the user's profile details in the `signup\_data` table. If a new password is provided, it updates that as well.
- Feedback: Provides alerts to inform the user about the success or failure of the update process.
- Connection Closure: Closes the database connection after the operations are completed.

#### Documentation for Logout Script (`logout.php`)

The `logout.php` script is designed to handle user logout functionality for the online bookstore project. The script ensures that the user's session is properly terminated, and then redirects the user to the sign-in page. Below is the detailed documentation of the script.

Script: ``logout.php``

```
<?php
session_start();
session_unset();
session_destroy();
header("Location: sign-in.php");
exit();
?>
```

## Breakdown and Explanation

### 1. Starting the Session

```
session_start();
```

- Purpose: Initializes the session or resumes the current session based on a session identifier passed via a GET or POST request, or passed via a cookie.
- Details: This function must be called before any output is sent to the browser.

### 2. Unsettling All Session Variables

```
session_unset();
```

- Purpose: Frees all session variables.
- Details: This function does not destroy the session itself, but it clears all variables in the session.

### 3. Destroying the Session

```
session_destroy();
```

- Purpose: Destroys all data registered to a session.
- Details: This function does not unset any of the global variables associated with the session, nor does it unset the session cookie. To fully destroy a session, you also need to unset the session cookie.

#### 4. Redirecting to the Sign-In Page

```
header("Location: sign-in.php");  
exit();
```

- Purpose: Redirects the user to the `sign-in.php` page.
- Details:
  - The `header` function sends a raw HTTP header to the browser, causing the browser to redirect to the specified location.
  - `exit();` ensures that no further code is executed after the redirection.

#### Usage

- When to Use: This script should be triggered when a user chooses to log out of the application, for example, by clicking a "Logout" button or link.
- How to Use: Typically, you would have a link or button in your application like this:

```
<?php  
<a href="logout.php">Logout</a>  
?>
```

- Clicking this link will take the user to `logout.php`, where their session will be terminated, and they will be redirected to the sign-in page.

#### Security Considerations

- Session Handling: Properly handling sessions is crucial for security. By calling `session_unset()` and `session_destroy()`, the script ensures that all session data is completely removed.

- Redirection: Redirecting users to the sign-in page ensures that they cannot access protected areas of the application after logging out.

## Conclusion

These forms collectively manage the user authentication and account management aspects of the online bookstore. The sign-in form validates user credentials and initiates a session, the sign-up form collects and stores new user data, and the forget password form retrieves and displays the user's password upon email verification. The implementation includes security measures like input sanitization to prevent SQL injection.

The edit profile form is a crucial part of the user account management system in the online bookstore project. It allows users to update their personal information securely, ensuring that the current password is validated before making any changes. The form and PHP script together provide a seamless user experience while maintaining the integrity of the user's data.

The **`logout.php`** script is a simple yet crucial part of user session management. It ensures that user sessions are properly terminated, preventing unauthorized access to restricted areas of the application after a user logs out. The script provides a secure way to manage user sessions and improve the overall security of the application.