

Building a complete login, logout, and sign-up registration system involves multiple steps and components. Below is a detailed guide for creating such a system using HTML, CSS, JavaScript, PHP, and MySQL. This guide covers the following:

1. Setting up the MySQL database
2. Creating the HTML forms
3. Adding CSS for styling
4. Writing JavaScript for client-side validation
5. Writing PHP scripts for server-side processing

Step 1: Setting Up the MySQL Database

First, create a MySQL database and a table to store user information.

```
```sql
```

```
CREATE DATABASE user_management;
```

```
USE user_management;
```

```
CREATE TABLE users (
```

```
 id INT AUTO_INCREMENT PRIMARY KEY,
```

```
 username VARCHAR(50) NOT NULL UNIQUE,
```

```
 password VARCHAR(255) NOT NULL,
```

```
 email VARCHAR(100) NOT NULL UNIQUE,
```

```
 created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
```

```
);
```

```
```
```

Step 2: Creating the HTML Forms

Create separate HTML files for the login, sign-up, and logout forms.

****sign_up.html****

```html

<!DOCTYPE html>

<html>

<head>

    <title>Sign Up</title>

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

</head>

<body>

    <form action="sign\_up.php" method="POST" id="signUpForm">

        <h2>Sign Up</h2>

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

        <input type="text" id="username" name="username" required>

        <label for="email">Email:</label>

        <input type="email" id="email" name="email" required>

        <label for="password">Password:</label>

        <input type="password" id="password" name="password" required>

        <button type="submit">Sign Up</button>

    </form>

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

</body>

</html>

```

****login.html****

```html

<!DOCTYPE html>

<html>

```
<head>

 <title>Login</title>

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

</head>

<body>

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

 <h2>Login</h2>

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

 <input type="text" id="username" name="username" required>

 <label for="password">Password:</label>

 <input type="password" id="password" name="password" required>

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

 </form>

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

</body>

</html>

...


```

**\*\*logout.php\*\***

```
` `` `php

<?php

session_start();

session_destroy();

header("Location: login.html");

exit();

?>

...


```

### ### Step 3: Adding CSS for Styling

Create a CSS file to style the forms.

**\*\*style.css\*\***

``` `css`

`body{`

`font-family: Arial, sans-serif;`

`background-color: #f2f2f2;`

`display: flex;`

`justify-content: center;`

`align-items: center;`

`height: 100vh;`

`}`

`form {`

`background: #fff;`

`padding: 20px;`

`border-radius: 5px;`

`box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);`

`width: 300px;`

`}`

`h2 {`

`margin-bottom: 20px;`

`color: #333;`

`}`

`label {`

```
display: block;
margin-bottom: 8px;
color: #333;
}
```

```
input {
  width: 100%;
  padding: 10px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 3px;
}
```

```
button {
  width: 100%;
  padding: 10px;
  background-color: #28a745;
  border: none;
  border-radius: 3px;
  color: #fff;
  font-size: 16px;
  cursor: pointer;
}
```

```
button:hover {
  background-color: #218838;
}
...
```

Step 4: Writing JavaScript for Client-Side Validation

Create a JavaScript file for client-side validation.

****validation.js****

````\njs`

```
document.getElementById('signUpForm').addEventListener('submit', function(event) {\n const password = document.getElementById('password').value;\n if (password.length < 6) {\n alert('Password must be at least 6 characters long.);\n event.preventDefault();\n }\n});\n```\n
```

### ### Step 5: Writing PHP Scripts for Server-Side Processing

Create PHP scripts for handling user registration, login, and session management.

**\*\*sign\_up.php\*\***

````\nphp`

`<?php`

`session_start();`

`include 'db.php';`

`if ($_SERVER["REQUEST_METHOD"] == "POST") {`

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

`$email = $_POST['email'];`

`$password = password_hash($_POST['password'], PASSWORD_BCRYPT);`

```
$sql = "INSERT INTO users (username, email, password) VALUES ('$username',  
'$email', '$password')";
```

```
if (mysqli_query($conn, $sql)) {  
    $_SESSION['username'] = $username;  
    header("Location: welcome.php");  
} else {  
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);  
}
```

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

****login.php****

`` `php

<?php

session_start();

include 'db.php';

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

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

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

```
$sql = "SELECT * FROM users WHERE username = '$username'";
```

```
$result = mysqli_query($conn, $sql);
```

```

$user = mysqli_fetch_assoc($result);

if ($user && password_verify($password, $user['password'])) {
    $_SESSION['username'] = $username;
    header("Location: welcome.php");
} else {
    echo "Invalid username or password.";
}

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

**db.php**
` `` `php
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "user_management";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
?>

```



```

**\*\*welcome.php\*\***

```php

<?php

session_start();

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

header("Location: login.html");

exit();

}

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

echo '
Logout';

?>

```

### ### Conclusion

This setup provides a complete login, logout, and sign-up registration system using HTML, CSS, JavaScript, PHP, and MySQL. Customize and expand upon it based on your requirements, such as adding additional security measures, form validations, and user experience enhancements.

If you need further assistance or have specific questions about any part of the setup, feel free to ask!