**Step 1: Create the Login Form (HTML)**

**Step 2: Handle Login (PHP)**

We need to handle form submission and authenticate the user against the database. Here’s how we’ll do it in PHP.

**Step 3: Create the Welcome Page (PHP)**

Once the user is successfully logged in, redirect them to a welcome page. This page will greet them with their name.

**Step 4: Logout Feature**

You should provide a way for the user to log out, which will destroy the session and redirect them to the login page.

**Step 5: Testing the Application**

1. **Place the files:**
   * Place login.html, login.php, welcome.php, and logout.php in the same directory in your local server.
2. **Start Apache and MySQL:**
   * Make sure that **Apache** and **MySQL** are running in your local XAMPP or WAMP environment.
3. **Open the login form:**
   * Go to http://localhost/login.html in your browser.
4. **Test the login functionality:**
   * Use the same credentials that were registered in the **registration system** (from the previous step).
   * If the login is successful, you will be redirected to welcome.php and greeted with a message like "Welcome, manali!".
5. **Logout:**
   * You can test the logout functionality by clicking the **Logout** link, which will destroy the session and take you back to the login page.

**Step 6: Security Enhancements**

1. **Prepared Statements:**
   * We already used **prepared statements** to prevent SQL injection in the login process.
2. **Session Handling:**
   * Always check if a user is logged in before allowing access to protected pages (like welcome.php).
3. **HTTPS:**
   * For production, use HTTPS to encrypt user data during form submission.

Output:

