Simple User Authentication System

Chohra Chemseddine

Objective

Your task is to build a simple PHP-based user authentication system that allows users to register, log in, and log out using sessions. The system should ensure that only authenticated users can access a protected dashboard page.

Requirements

1. **User Registration:**
    - Create a registration form where users can sign up by providing their full name, username, and password.
    - Store the user's credentials securely in a MySQL database.
    - The passwords should be hashed before storing them in the database.
2. **User Login:**
    - Create a login form where users can enter their username and password.
    - Validate the credentials against the database.
    - If the credentials are correct, start a session and redirect the user to their dashboard page.
3. **Protected Dashboard:**
    - Create a dashboard page that is only accessible to logged-in users.
    - Display a welcome message with the user's full name.
    - Provide a logout option.
4. **User Logout:**
    - Implement a logout feature that destroys the session and redirects the user to the login page.
    - If the user tries to access the dashboard directly via URL without being logged in, redirect them to the login page.
5. **Default Pages:**
    - The default page (index.php) should display the login form with an option to register.
    - If a user session is active, redirect them to the dashboard page.

Deliverables

1. A MySQL database with a `users` table to store user credentials.

2. PHP scripts for:

   - Registration (`register.php`)
   - Login (`login.php`)
   - Dashboard (`dashboard.php`)
   - Logout (`logout.php`)

3. A `config.php` file to handle database connections.

Bonus (Optional)

1. Add input validation to ensure usernames and passwords meet specific criteria (e.g., minimum length).

2. Display error messages for invalid login attempts or registration errors.

3. Style the forms and pages using CSS to make the application visually appealing.

Submission Guidelines

- Export the database structure and data as an SQL file.

- Zip all PHP files and the exported SQL file into a single archive.

- Upload the zip file to the university's moodle platform in the appropriate assignment submission area.

- You can share the project files via a GitHub repository and provide the link in the submission if you prefer.

- For late submissions, a penalty of 5% per day will be applied.