

Assignment: User Registration & Login System

Objective: Create a simple user registration and login system using PHP, JavaScript, and MySQL. This assignment will help you understand form validation, data processing, and database interaction.

Submission link: <https://forms.gle/kXVESmGq2CLNf63o6>

Deadline: 27/03/2025

Requirements:

- **Forms**

- Create a **registration form** (index.php)
- Create a **login form** (login.php)
- Use **Tailwind CSS** for styling

You can reuse the previous assignment's design

- **Validation**

- Ensure fields are not empty
- Password must be at least **6 characters** long
- Confirm password must match the entered password.

- **Backend (server.php)**

- Use **MySQLi** or **PDO** to connect to the database
- Process all **backend logic** in server.php
- **Do not use sessions**

- **JavaScript (Frontend Requests)**
 - Use the **Fetch API** to send and receive data in **JSON format**
 - **jQuery is optional**, but **Fetch API is required for sending requests**
 - **Database**
 - Table name: users
 - Fields: **id, name, email, password**
-

Process Flow

- **User Registration**
 - Users register via index.php
 - On successful registration, redirect to login.php
 - **User Login**
 - Users log in using their email and password
 - If login is successful, redirect to dashboard.php
 - **dashboard.php** *(The page should exist but does not need any design or content.)*
-

Submission Instructions:

- After completing the assignment, **compress all project files** (including the database .sql file) **into a .zip file.**
 - **Upload the .zip file** to the [provided link](#) before the deadline.
-

Evaluation Criteria:

Criteria	Points	Description
<i>Form Design & Styling</i>	10	Uses Tailwind CSS properly. The forms are structured and visually appealing.
<i>Frontend Validation</i>	15	Validates empty fields, password length (min. 6 chars), and matching passwords before sending data to the server.
<i>JavaScript Fetch Usage</i>	15	Correctly sends requests via Fetch API and processes JSON responses.
<i>PHP Backend Processing</i>	20	Properly handles form submissions and database interactions in server.php.
<i>Database Integration</i>	20	Successfully connects to MySQL, inserts user data, and verifies login credentials.
<i>Redirects & Process Flow</i>	10	Registration redirects to login.php, and successful login redirects to dashboard.php.
<i>Code Quality & Organization</i>	10	Code is clean, structured, and well-commented.