**Development document**

**Signup form using PHP and MySQL with validation**

**By Thanuj Kumar S**

For any kind of web application login, signup is the most important thing for security reason. If we not have a login features in our web application then any one can access our data and services.

In this project, I have created user registration in PHP with the login script.

On the landing page, it shows a login form with a signup link. The registered user can enter their login details with the login form. Once done, he can get into the dashboard after successful authentication.

If the user does not have an account, then he can click the signup option to create a new account.

The user registration form requests username, email, password from the user. On submission, PHP code allows registration if the email does not already exist.

The code we used has client-side validation for validating the entered user details. And also, it includes contains the server-side uniqueness test. The user email is the base to check uniqueness before adding the users to the MySQL database.

Now let’s see the work flow and following files we used and descriptions

**1.login.php**

**Create user registration and login form**

I have created three HTML view login form, registration form and the dashboard for this code.

Below HMTL code is for displaying the login form to the user. In this form, it has two inputs to allow the user to enter their username and password.

Without these details, a validation code will not allow the login to proceed. The login form tag’s on-click attribute is with loginValidation(). This function contains the login form validation script.

On submitting this login form, the PHP code will validate the user. If the users clear the authentication, then it will redirect him to the dashboard.

If the user attempts to log in with the wrong data, then the code will display a login error message in the login form.

**2. user-registration.php**

Following is a user registration form to get minimal user data from the user. All form fields are mandatory.

It will pass-through a JavaScript validation before processing the user registration in PHP.

On submitting the registration form fields, it will invoke the signupValidation() JavaScript method. In this method, it validates with the non-empty check, email format, and the password match.

After validation, the PHP registration will take place with the posted form data.

**3. index.php**

The index. php file is the skeleton of the website. ... php" fleshed out with a selection of content inserted from the database. The index. php file for a template contains a mixture of code that will be delivered as it is, and php code, which will be modified before it is delivered

<?php

require\_once \_\_DIR\_\_ . "/login.php";

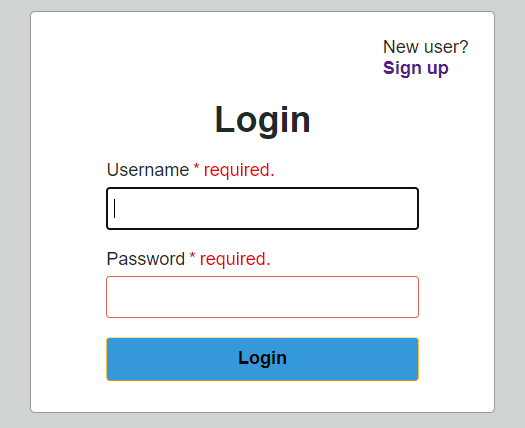
**4. css files**

For give good look and feel

**Registration and login form validation**

JavaScript is used to validate the form before submitting the request to PHP backend.

On invalid data submission, the code will return a boolean false. It forces the user to enter the required fields by highlighting them.



**5. Model/Member.php**

**Process user registration in PHP**

After submitting the form details, it processes user registration in the PHP code.

This code uses default form submit to post data to the PHP. If you want the user registration code with AJAX, then we have to prevent the default submit with a script.

We have added code at the beginning of the user-registration-form.php. It checks if the user submitted the form. Then, it invokes the registerMember() method defined in the Member model.

In the registerMember() function, it checks if the posted email already exists. If so, it truncates the registration flow and returns the error. Otherwise, it creates the Insert query to add the member record into the MySQL database.

The loginMember() function checks if there is any match for the entered login details. If the match found, it clears the authentication and allows the user to access the dashboard.

**PHP login authentication code**

The code in the login-form.php file above invokes the authentication function after login

**6. logout.php**

Logout

Following logout script is common code. You can use it any of your PHP projects. It clears the complete user session and redirects the user to the home / login page.

<?php

// clear all the session variables and redirect to index

session\_start();

session\_unset();

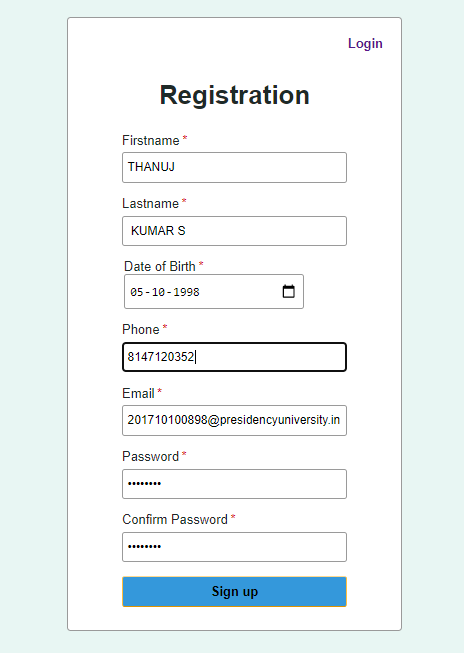
session\_write\_close();

$url = "./index.php";

header("Location: $url");

**MySQL database script**

We create a database in MySQL (script: user-registeration.sql)



Application link : [This contains all the files to create signup dashboard](https://github.com/MrThanuj/HaviAssesment/blob/main/Application.zip)

Thank You.