**Registration and Login form with Email Validation and Password Reset**

**Prerequisites:** PHP version 8.1.6 and phpMyAdmin SQL server both operated by XAMPP server. HTML and CSS.

**Installation of XAMPP server:** <https://www.apachefriends.org/>

**Open-Source Software for sending emails (swiftmailer):** <https://getcomposer.org/download/>

**Instructions to use XAMPP server:** After installing the XAMPP server, go to C:\xampp\xampp-control.exe and start it. Start the Apache and MySQL services. The .php files must be stored in any folder in C:\xampp\htdocs in order to run. After typing localhost in chrome, the htdocs folder will open. For opening the SQL server, admin action must be started, it redirects to <http://localhost/phpmyadmin/>. The databases and tables can be created here.

**Contents of the Project:**

1. A **Registration Form,** where the users can be able to register themselves by providing their username, year of passing out, email address, phone number along with their password. A registered user can verify himself by clicking on the verify yourself link which will be received on their email address. Thus, a registered user can become a verified user.
2. A **Login Form,** where a verified user can log into his account. A non-verified user can also login but he needs to verify his email after logging in.
3. An **Admin Panel**, where the data of a registered user / a verified user can be found. The data can also be viewed in the phpMyAdmin portal but this Admin Panel is password protected and only authorized users can see the data.
4. A **Forgot Password** option is available in the login form, which means a user can reset his password without logging in. When he enters his email address, a mail will be received, with which he can reset his password and then login.
5. A **Change Password** and Change Username option is available after the user logs into his account.
6. **Email Validation** is also available for security reasons. The registered user cannot access all the options in his homepage before verifying his Email. Also, a user cannot use a random email address.

**Setting your Email Address for Sending Emails:** Log into your Gmail Account, click on the Manage your Google Account below your profile on the right top. Go to Security > Signing in into Google > App Passwords. Make sure that the two-step verification for your account is turned on. Then create an app password and note it down. In the sendEmails.php file in lines 6 and 48, change your email address and in line 7, change your password (both must be in double quotes).

**How is the data going into the Registration Form showing up in the Admin Panel?:** The data entered in the Registration Form is sent into a Table named **users** in the **verify\_user** Database. The data is reflected into the Admin Panel page. The data can also be viewed in the phpMyAdmin service but in order to be password protected it is operated by an authorized person like a Backend Engineer.

**Username:** The username field is set to be unique along with the ID (which is a primary key), Email ID and Phone Number. A user cannot use the username which is already used by another used. The same goes with Email ID and Phone Number.

**Token:** The token field is one of the many fields present in the **users** table. The token is a string which contains of 50 characters. This is used to verify a user through his email address. While the user registers himself, a unique token is automatically generated and that token is stored in the token column in the table. The unique token is attached to the link in which the user verifies himself. When this token matches with that token in the database, the validation is done. Thus, the user is verified.

**Password:** We all know the function of the password field. But while observing the password field in the database, it is very complex. In order to secure the private details of the user, the passwords have been encoded using a hash code. This hash code only has the ability of encrypting, it is a one-way hashing algorithm. Decrypting is not possible. Only the user can be able to reset and change his password. Even the backend engineer doesn’t have access to the user’s password.

**Verified:** The 0 under the verified column conveys that the user has not verified the link through his email whereas the 1 says he/she has verified.

**Databases and Tables:** For the Admin Panel, we need to create a database and a table inside it where we can insert the login credentials for the admin who can access the data of the users. I named them as **admin** and **adminlogin** respectively. For the Users, the database was named as **verify\_user** and the tables were named as **users** and **password\_resets**. The **users** table takes data from the registration form and the **password\_resets** table notes down the history of when the user changes his/her password.

**Changing your Password:** While you click on the Forgot your Password button, you’ll be redirected to a page where you need to verify your email address in order to get a mail where you can reset your password. After verifying your email, an email will be sent to your mail. You can reset your password and then login with the new credentials. The changed data will be simultaneously changed in the database too.

**Files included for the User Panel:**

1. login.php – Login Page
2. register.php – Registration Page
3. logged\_in.php – Page which we get into while we log in
4. logout.php – Page which logs out the user, redirects into login.php
5. authControllers.php – Page where the functionalities are present, tell what to do when a particular button is clicked, sends the data to the database
6. sendEmails.php – Sends emails, the pattern of the email is also set here, using CSS(must change lines 6,7 and 48 to receive emails from the account you wish to)
7. verify\_email.php – Generates a message in the logged\_in.php page when a particular account is verified
8. bootstrap.min.css and main.css – Helps in styling the forms
9. styles.css – Helps in styling the logged\_in.php page
10. forgot\_password.php – Your email address is checked whether if it’s in the database or not
11. password\_message.php – A message is displayed saying that a link has been sent to your email address
12. reset\_password.php – This page open when you click on the link sent to your email address. You can reset your password here
13. change\_username.php – You can change your username in this page after logging in
14. change\_password.php – You can change your password in this page after logging in

**Files included for the Admin Panel:**

1. adminlogin.php – Login Page for the Admin
2. adminpage.php – Page which opens after the Admin logs in, the data entered by users can be accesses from here
3. connection.php – Page which connects the adminlogin.php page and the database where the admin credentials are present
4. validate.php – This page is used to verify if the admin has logged in with the correct details
5. database.php – This page is used to connect the **verify\_user** database with the adminpage.php
6. developers.php – This page allocates the data from the **verify\_user** database into the adminpage.php
7. logout.php – Page which logs out the Admin, redirects to adminlogin.php
8. login.css – Used for styling the adminlogin.php page
9. Also, various .css files were used for styling the adminpage.php page