

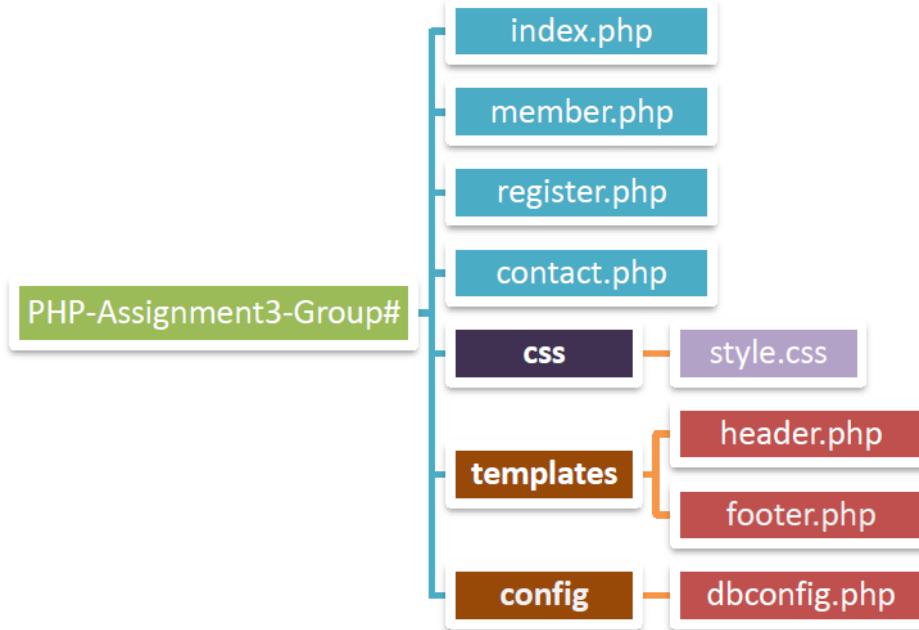
### Important Notes:

- The assignment should be **submitted through the Blackboard only (emails are not accepted) before the due date** in order not to lose marks
- Follow the **same folder and file structure or similar structure** as explained in the instructions
- **At the end of the assignment** (after answering all the questions), you should have a:
  - **folder named “PHP-Assignment3”** which contains only the following files:
    - **index.php => Public Page**
    - **member.php => For registered members only**
    - **register.php => Public Page**
    - **contact.php => Public Page**
    - **logout.php => Just contain the logout code**
    - **style.css (inside a folder named css for example)**
    - **List of images depending on your website content (inside a folder named images for example)**
    - **header.php (inside a folder named templates)**
    - **footer.php (inside the same folder templates)**
    - **dbconfig.php (inside the same folder config)**
- **Upload/Submit your files through:**
  - **The Blackboard:**
    - The PHP Files (*Not the entire folder*):
      - Use MS-Word to capture all the assignment images (screenshots) then convert it to A PDF file and upload it with your submission. Please consider the following snapshots (images):
        - The folder structure inside VS Code or any other editor you like to use
        - The Website pages before login and after
        - The table on your database on PHPMyAdmin
  - **The link to your assignment repo on your GitHub account:**
    - Upload all the assignment folders and files
    - Don't upload the assignment PDF instruction, or the screenshot PDF file to GitHub
  - **Please don't compress the files/folder ➔ Submit the PHP files and the PDF file that contains the screenshots (images) for the output (webpages)**
- **Online:** Each student must be ready with their Zoom Mic/Cam and Screen share feature to quickly show us the assignment content, Q/A, and run it to see the output (even if it's not working properly).
- **In-person:** I will make a tour to check the running application in the student's computer and asking the technical questions

**NOTE:** This presentation will be very quick without correction or feedback in order not to take the full lecture time! The assignment steps, feedback, and solutions will be completely explained to all students (after finishing all the presentations) in the next lecture

(20 Marks)

## Assignment Folder Structure:



## Assignment Contents:

This assignment will be a demonstration of your skills in using HTML5 valid form elements, PHP, and MySQL. The topics will include: OOPS, methods, form validation, conditional statements, debugging, session, and PDO SQL statements to demonstrate simple CRUD operations. You will create a simple website that contains public pages for all users (visitors) and private pages for members only (registered users).

### MySQL:

Using “**phpMyAdmin**”, consider build the following:

1. Create **Database**: “**accounts**”
2. Create **Table** (inside accounts database): “**users**” with the following fields:
  - **user\_id** → int **Primary Key (Auto Incremented)**
  - **first\_name** → maximum characters (30), NOT NULL
  - **last\_name** → maximum characters (30), NOT NULL
  - **email** → maximum characters (50), NOT NULL, and **UNIQUE**
  - **password** → maximum characters (12), NOT NULL

**NOTE:** The email field will be used as a username for logging into the member page

**HINT:** Refer to my MySQL Introduction PDF file

(10 Marks)

## Assignment Instructions:

1. You can either create a **new PHP project** (folder) using **VS Code**, NetBeans (IDE), or any IDE that you prefer with any name like “**PHP-Assignment3**” and save it inside the server folder “**htdocs**”. **(The folder of the assignment must be created inside “htdocs” which is inside “xampp” as we always do):**

2. **You have to add useful and detailed comments** to explain your code lines for these reasons:
  - ✓ to make it easy to read and understand
  - ✓ to show that you know what you are doing and how to use PHP statements
  - ✓ to be used as a proof that you're the programmers of this code
3. Create a **website of at least 5 pages**:
  - a. **Each one of these web pages has the full HTML template:**
    - **Home page (index.php):** The first default page when the user visits the website
    - **Member page (member.php):** For the logged in users (Members only)
    - **Registration page (register.php):** For registering a new user (add a new user to the database)
    - **Contact Us form page (contact.php):** For contact us form (using email)
  - b. **Except this page “logout.php” doesn’t need any HTML contents as it will be used just to logout the user and redirect them to the home page:**
    - **Logout (logout.php):** The (hidden) page that contains only PHP code for logout as explained in our lectures
4. Create a folder for the **page templates** named “**templates**” or any other name you prefer that contains only these two **.PHP** files which contains the shared code among all the browser:
  - a. **header.php:** contains the top part of the HTML Code and other PHP required code
  - b. **footer.php:** contains the bottom part of the HTML Code

5. The following code that represents the top part of the pages, must be **inside the header.php** file as shown in the example below:

```
<!DOCTYPE HTML> <!-- Using full HTML 5 -->
<html>
    <head>
        <meta charset="utf-8">
        <title>PHP Final Assignment</title>
    </head>
    <body>
        <div id="container">
            <header id="banner">
                <h1>Final Assignment</h1>
                <h3>Users' Info Using PHP with MySQL</h3>
            </header>

            <div id="nav">
                <ul>
                    <li><a href="index.php">Home</a></li>
                    <li><a href="member.php">Member</a></li>
                    <li><a href="register.php">Register</a></li> (This line is highlighted with a red border)
                    <li><a href="logout.php">Logout</a></li> (This line is highlighted with a red border)
                    <li><a href="contact.php">Contact</a></li>
                </ul>
            </div>
            <div class="main-content">
```

You will need to modify the navigation list item dynamically to display “Register” if no user has logged in yet, and display logout if a user has logged in.

**Hint:** you modify the nav item depending on the session value to check whether there is an existing user (logged in) or no.

6. The code that represents the bottom part of the page must be **inside the footer.php** file as shown in the example below:

```
</div> <!-- End div.main-content -->
</div> <!-- End div#container -->
<footer>Developed By: Group X: Alex Chow, Martin Smith, and Sam Simpson</footer>
</body>
</html>
```

**NOTE:** Please be advised that you don't have to use the same ids, classes names, or the same structure! Feel free to use any name for your CSS classes and ids if there are any. Although it's nice to style your webpages in professional way using the skills and techniques that you have learned in your course “Interface Design”.

**HINT (for questions: 3, 4, 5, and 6):**

- For creating the full website with the header and footer template files:
  - <https://github.com/anmarjarjees/php-code/tree/main/week07>

7. Create a **folder** named “**config**” or any other name you prefer that contains:
  - a. PHP file: **dbconfig.php**
  - b. This PHP file will contain the required information to connect your PHP code of your assignment to the database (the tables) using PDO. You will define the **host name**, **database name**, **user name**, and **password** plus execute and test the **PDO object**
  - c. Using the proper **PDO method** for the PDO object, set the default value for fetching your data to be in “**Associative Array**” format
  - d. Notice that this file “dbconfig.php” is required to be loaded in any page that interacts with the database (**HINT: You can include it in the header.php**)

**HINT:**

- For preparing this file:
    - <https://github.com/anmarjarjees/php-code/tree/main/week13/include>
8. In the home page (**index.php**):
    - a. Create a simple HTML content to welcome the user and simple paragraph to explain your assignment/website topic
    - b. Below the above-mentioned content (HTML elements), create a login form that is consisted of two input fields:
      - **Email ➔** of type email
      - **Password ➔** of type password
    - c. The form will be submitted to the same page, so place all your PHP code for manipulating the form submission on the top of the file (Before starting your HTML code, not in between):
      - **First:** Verify that both input boxes are **not empty**  
**(HINT: You can use the **required** attribute for HTML element)**
      - **Second:** Verify that email address matches the normal **email pattern**: **text@text.text** using the built-in PHP email check function although type=”email”  
**(HINT: You can use the **type=”email”** attribute for HTML element)**
      - **Third:** Verify that **both user’s info (email and password)** are **exists** (an existing member that has a record on the users table) using the PHP/SQL code to scan the users’ table.  
**(HINT: Use the **execute()** method with **rowCount()**)**
      - For the 3 levels of verification that mentioned above, display a meaningful message to advice the user about what he/she missed or why he/she couldn’t login
      - If the user passed the 3 verification levels (which means he/she is an existing user):
        1. Save the user’s input information (email, first name, and last name) into the session array
        2. redirect them immediately to the member page (**member.php**)
    - d. This login form should become hidden (disappear from the home page) after a user being logged successfully. So, it became visible again when the current user logged out or there is no logged in user when you visit the website

**HINTS:**

- For using session and hiding the form:
  - <https://github.com/anmarjarjees/php-code/blob/main/week11/final/index.php>
- For verifying/checking if the email and password both exist:
  - <https://github.com/anmarjarjees/php-code/blob/main/week13/04.prepare-stmt2.php>

9. In the member page (**member.php**):

- a. Welcome the logged in user by displaying:
  - User's full name
  - User's email address
- b. Notice that you need to retrieve this information from the home page "index.php" using **sessions**
- c. Add some extra contents based on your opinion

**HINT:**

- For retrieving info from the session and accessing the member page:
  - <https://github.com/anmarjarjees/php-code/blob/main/week11/final/client.php>

10. In the register page (**register.php**):

- a. Create a simple HTML content to advice the user to finish the registration from fields
- b. Below the above-mentioned content, create a registration form that is consisted of the four required input fields:
  - **First Name** → of type text
  - **Last Name** → of type text
  - **Email** → of type email
  - **Password** → of type password
- c. The form will be submitted to the same page so place all your PHP code for manipulating the form submission on the top of the file before starting your HTML code:
  - **First:** Verify that all the input boxes are **not empty**
  - **Second:** Verify that email address matches the normal **email pattern: text@text.text** using the built-in PHP email check function
  - **Third:** Verify that the **user's email is NOT existing**  
(HINT: Use the **execute()** method that returns a Boolean value if exists/or not)
  - For the 3 levels of verifications that mentioned above, display a meaningful message to advice the user about what he/she missed or why he/she couldn't login
  - If the user passed the 3 verification levels (which means he/she is an existing user):
    1. Save the user's input information (email, first name, and last name) into the session array
    2. redirect them immediately to the member page (**member.php**)

11. In the contact us page (**contact.php**):

- a. Create a contact us form that asks the user to input the following 3 required info (fields):
  - a. User's Full Name
  - b. User's Email
  - c. User's Message
- b. The form will be submitted to the same page, so place all your PHP code at the top of the page
- c. Write the appropriate code for letting the user sending/emailing their message using the example from the lecture (**Optional**)

**IMPORTANT NOTES:**

- a. Don't forget that when the user is logged in successfully:
  - i. You need to change the navigation item “**Register**” to be “**Logout**”
- b. When the user clicks the Logout link in the navigation menu, redirect them to the logout PHP file (**logout.php**). This file will clear (destroy) the SESSION and redirect the user to the home page (**index.php**)

**HINT:**

- For toggling the navigation menu items (Register ⇔ Logout):
  - <https://github.com/anmarjarjees/php-code/blob/main/week11/final/includes/header.php>

(70 Marks)

Happy Coding 😊