PHP is an acronym for "PHP: Hypertext Preprocessor"

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

A PHP script is executed on the server, and the plain HTML result is sent back to the browser.
A PHP script starts with `<?php` and ends with `?>`:
In PHP, a variable starts with the `$` sign, followed by the name of the variable

- A variable starts with the `$` sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (`$age` and `$AGE` are two different variables)

The PHP `echo` statement is often used to output data to the screen.

To get the data type of a variable, use the `var_dump()` function.

PHP has three different variable scopes:

- local
- global
- static

A variable declared outside a function has a GLOBAL SCOPE and can only be accessed outside a function:

A variable declared within a function has a LOCAL SCOPE and can only be accessed within that function:

The `global` keyword is used to access a global variable from within a function.

```
$x = 5;
$y = 10;

function myTest() {
  global $x, $y;
  $y = $x + $y;
}

myTest();

echo $y; // outputs 15
```

$_GET:
Information sent from a form with the GET method is visible to everyone (all variable names and values are displayed in the URL). GET also has limits on the amount of information to send. The limitation is about 2000 characters. However, because the variables are displayed in the URL, it is possible to bookmark the page. This can be useful in some cases.

$_POST:
Information sent from a form with the POST method is invisible to others (all names/values are embedded within the body of the HTTP request) and has no limits on the amount of information to send.However, because the variables are not displayed in the URL, it is not possible to bookmark the page.

```
<?php
// Database configuration
$servername = "localhost";
$username = "your_username";
$password = "your_password";
$dbname = "webapp";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

// Function to add a new user
function addUser($username, $password) {
    global $conn;
    $stmt = $conn->prepare("INSERT INTO users (username, password)
VALUES (?, ?)");
    $stmt->bind_param("ss", $username, $password);
    return $stmt->execute();
}

// Function to update a user
function updateUser($id, $username, $password) {
    global $conn;
    $stmt = $conn->prepare("UPDATE users SET username=?,
password=? WHERE id=?");
    $stmt->bind_param("ssi", $username, $password, $id);
    return $stmt->execute();
}

// Function to delete a user
function deleteUser($id) {
    global $conn;
    $stmt = $conn->prepare("DELETE FROM users WHERE id=?");
    $stmt->bind_param("i", $id);
    return $stmt->execute();
}

// Function to retrieve all users
function getUsers() {
    global $conn;
    $result = $conn->query("SELECT * FROM users");
    $users = array();
    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $users[] = $row;
        }
    }
    return $users;
}
```

```
$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Management</title>
</head>
<body>
    <h1>User Management</h1>

    <h2>Add User</h2>
    <form action="add_user.php" method="post">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username">
        <label for="password">Password:</label>
        <input type="password" id="password" name="password">
        <button type="submit">Add User</button>
    </form>

    <h2>Update User</h2>
    <form action="update_user.php" method="post">
        <label for="update_id">User ID to Update:</label>
        <input type="number" id="update_id" name="update_id">
        <label for="update_username">New Username:</label>
        <input type="text" id="update_username" name="update_username">
        <label for="update_password">New Password:</label>
        <input type="password" id="update_password"
name="update_password">
        <button type="submit">Update User</button>
    </form>

    <h2>Delete User</h2>
    <form action="delete_user.php" method="post">
        <label for="delete_id">User ID to Delete:</label>
        <input type="number" id="delete_id" name="delete_id">
        <button type="submit">Delete User</button>
    </form>

    <h2>Users List</h2>
    <?php
    include 'db_functions.php';

    $users = getUsers();
    if (count($users) > 0) {
        echo "<ul>";
        foreach ($users as $user) {
            echo "<li>ID: " . $user['id'] . " | Username: " . $user['username'] .
"</li>";
        }
        echo "</ul>";
    } else {
        echo "No users found.";
    }
    ?>
</body>
</html>
```

Create separate php file for each each submission handling.