Full Stack Development-Lab Assignment 4

*By- Harshil Patel, Roll No-PA26, PRN-1032211762, Batch-2*

**Aim:** Write server-side script in PHP to perform form validation and create database application using PHP and MySQL to perform insert, update, delete and search operations.

**Objectives:**

1. To understand Server-side Scripting.
2. To learn database connectivity using PHP-MySQL.
3. To perform insert, update, delete and search operations on database.

**Theory:**

1. **PHP Architecture.-**

PHP (Hypertext Preprocessor) is a widely-used open-source server-side scripting language designed for web development. PHP is particularly well-suited for creating dynamic web pages and web applications. PHP architecture can be understood at multiple levels, including the web server, PHP interpreter, and application design.

Here's a high-level overview of the PHP architecture:

1. Web Server:

- A web server (e.g., Apache, Nginx, IIS) is responsible for receiving incoming HTTP requests from clients (browsers or other clients) and routing them to the appropriate PHP scripts for processing.

- The web server typically listens on port 80 (HTTP) or 443 (HTTPS) and manages the request-response cycle.

2. PHP Interpreter:

- The PHP interpreter is responsible for executing PHP code. It takes PHP scripts, processes them, and generates HTML or other output that can be sent back to the client.

- PHP can be installed as a module (e.g., mod\_php for Apache) or as a FastCGI process (e.g., PHP-FPM for Nginx) on the web server, depending on the configuration.

3. PHP Code:

- PHP code is embedded within HTML or can exist as standalone PHP files with a ".php" extension.

- PHP code can be used to dynamically generate HTML, process form data, interact with databases, and perform a wide range of server-side tasks.

- PHP scripts can be organized into functions, classes, and libraries to improve code maintainability and reusability.

4. Database Interaction:

- PHP can connect to various databases (e.g., MySQL, PostgreSQL, SQLite, MongoDB) to retrieve or manipulate data.

- Developers often use database libraries such as PDO (PHP Data Objects) or specific extensions (e.g., mysqli) to interact with databases securely.

5. External Services and APIs:

- PHP can communicate with external services and APIs using various protocols (e.g., REST, SOAP, JSON-RPC) to fetch or send data.

6. Output:

- The PHP interpreter generates HTML, JSON, XML, or other data formats as output.

- This output is then sent back to the web server, which delivers it to the client's browser.

7. Client Browser:

- The client's browser receives the HTML and renders it to display the web page.

- PHP also provides mechanisms for setting HTTP headers, cookies, and handling sessions, which enhance the user experience.

8. Application Design:

- PHP applications can follow various architectural patterns, such as Model-View-Controller (MVC), to separate concerns and improve maintainability.

- Popular PHP frameworks like Laravel, Symfony, and CodeIgniter provide a structured way to build web applications.

It's important to note that PHP is a versatile language and can be used for a wide range of web development tasks, from simple web pages to complex web applications. The specific architecture of a PHP application can vary depending on the project's requirements and the developer's preferences.

1. **Steps for Database connectivity in PHP.**
   1. Open XAMPP and start running Apache, MySQL and FileZilla
   2. Now open your PHP file and write your PHP code to create database and a table in your database
   3. Save the file as “data.php” in htdocs folder under XAMPP folder.
   4. Then open your web browser and type localhost/data.php
   5. If you want to see your database, just type localhost/phpmyadmin in the web browser and the database can be found.

**FAQ:**

1. **What are the advantages of Server-side Scripting?**
   1. User can create one template for the entire website
   2. The site can use a content management system which makes editing simpler.
   3. Generally quicker to load than client-side scripting
   4. User is able to include external files to save coding.
   5. Scripts are hidden from view so it is more secure. Users only see the HTML output.
   6. User does not need to download plugins like Java or Flash.
2. **What is XAMPP and phpMyAdmin?**
   1. **XAMPP**-XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the Apache Friends, and its native source code can be revised or modified by the audience. It consists of Apache HTTP Server, MariaDB, and interpreter for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.
   2. **phpMyadmin**-phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.
3. **What are the two ways to connect to a database in PHP?**

There are two popular ways to connect to a MySQL database using PHP:

* With PHP’s MySQLi Extension-MySQLi is an extension that only supports MySQL databases. It allows access to new functionalities found in MySQL systems (version 4.1. and above), providing both an object-oriented and procedural interface. It supports server-side prepared statements, but not client-side prepared statements. The MySQLi extension is included PHP version 5 and newer.
* With PHP Data Objects (PDO)-PHP Data Objects (PDO) is an extension that serves as an interface for connecting to databases. Unlike MySQLi, it can perform any database functions and is not limited to MySQL. It allows flexibility among databases and is more general than MySQL. PDO supports both server and client-side prepared statements.

GITHUB link- https://github.com/stich-12/rep1