

Module 5: PHP & MySQL (Classes 17–21)

Module Duration: 10 Hours (5 Classes × 2 Hours)

Software: Visual Studio Code, XAMPP/WAMP, PHP 8.x, MySQL, phpMyAdmin

Module Learning Objectives

By the end of this module, students will be able to:

- Understand server-side web development using PHP.
 - Write PHP programs using variables, operators, conditions, loops, arrays, and functions.
 - Connect PHP applications to a MySQL database.
 - Perform CRUD (Create, Read, Update, Delete) operations.
 - Build dynamic web applications with forms and database integration.
 - Develop a basic user authentication (login) system.
-

Class 17 (2 Hours)

Introduction to PHP

Learning Outcomes

After completing this class, students will be able to:

- Understand PHP and server-side scripting.
 - Install and configure XAMPP.
 - Write basic PHP programs.
 - Process HTML form data using PHP.
-

17.1 What is PHP?

PHP (**H**ypertext **P**reprocessor) is a server-side scripting language used to create dynamic and interactive websites.

Advantages of PHP

- Open-source and free

- Easy to learn
 - Fast and lightweight
 - Cross-platform compatibility
 - Excellent MySQL support
 - Large developer community
 - Widely supported by hosting providers
-

17.2 Client-Side vs Server-Side

Client-Side	Server-Side
Runs in the browser	Runs on the web server
HTML	PHP
CSS	PHP
JavaScript	MySQL
User Interface	Business Logic

17.3 Installing XAMPP

Components

- Apache Web Server
- PHP
- MySQL (MariaDB)
- phpMyAdmin

Steps

1. Install XAMPP.
 2. Start Apache and MySQL.
 3. Place project files inside the **htdocs** folder.
 4. Open `http://localhost/project-name`.
-

17.4 PHP Syntax

Example:

```
<?php
echo "Welcome to PHP!";
?>
```

17.5 Variables

```
<?php
$name = "John";
$age = 22;

echo $name;
?>
```

17.6 Data Types

- String
 - Integer
 - Float
 - Boolean
 - Array
 - Object
 - NULL
-

17.7 Operators

Arithmetic

- -
 -
 -
 -
 -
- /
- %

Comparison

- ==
- ===
- !=
-
- <
- - =
- <=

Logical

- &&
 - ||
 - !
-

17.8 Processing HTML Forms

Example:

```
<?php
$name = $_POST['name'];
echo "Welcome, " . $name;
?>
```

Practical Exercise

Create a contact form that accepts:

- Name
- Email
- Mobile Number
- Message

Display the submitted information on another page.

Assignment

Develop a student information form using PHP.

Class 18 (2 Hours)

PHP Control Statements & Functions

Learning Outcomes

Students will be able to:

- Make decisions using conditional statements.
 - Repeat tasks with loops.
 - Create reusable functions.
 - Organize projects with include files.
-

18.1 Conditional Statements

- if
 - if...else
 - else if
 - switch
-

18.2 Loops

- for
- while
- do...while
- foreach

Example:

```
<?php
for($i=1;$i<=10;$i++){
    echo $i."<br>";
}
?>
```

18.3 Arrays

Types:

- Indexed Array
 - Associative Array
 - Multidimensional Array
-

18.4 Functions

Example:

```
<?php
function greet($name){
    return "Hello ".$name;
}

echo greet("John");
?>
```

18.5 Include & Require

- include
- require
- include_once
- require_once

Purpose:

- Header
 - Footer
 - Navigation
 - Database Connection
-

Practical Exercise

Create a multi-page website using:

- header.php
 - footer.php
 - menu.php
 - index.php
-

Assignment

Create a simple calculator using PHP functions.

Class 19 (2 Hours)

Introduction to MySQL

Learning Outcomes

Students will be able to:

- Understand relational databases.
 - Create databases and tables.
 - Perform basic SQL operations.
 - Use phpMyAdmin.
-

19.1 What is MySQL?

MySQL is a Relational Database Management System (RDBMS) used to store and manage application data.

19.2 Database Concepts

- Database
 - Table
 - Record (Row)
 - Field (Column)
 - Primary Key
 - Foreign Key
-

19.3 phpMyAdmin

Features:

- Create databases
 - Create tables
 - Insert records
 - Execute SQL queries
 - Export and import databases
-

19.4 SQL Basics

Create Database

```
CREATE DATABASE student_db;
```

Create Table

```
CREATE TABLE students(  
id INT AUTO_INCREMENT PRIMARY KEY,  
name VARCHAR(100),  
email VARCHAR(100),  
department VARCHAR(50)  
);
```

Insert Data

```
INSERT INTO students(name,email,department)  
VALUES('John','john@email.com','CSE');
```

Select Data

```
SELECT * FROM students;
```

Practical Exercise

Create a **Student Management** database and add sample records.

Assignment

Create two tables:

- students
- courses

Insert at least five records into each table.

Class 20 (2 Hours)

PHP & MySQL Database Connection

Learning Outcomes

Students will be able to:

- Connect PHP to MySQL.
 - Insert records from HTML forms.
 - Display database records on a webpage.
-

20.1 Database Connection

Using MySQLi:

```
<?php
$conn = new mysqli("localhost","root","","student_db");

if($conn->connect_error){
    die("Connection Failed");
}
?>
```

20.2 Insert Data

Receive data from an HTML form and save it to the database.

20.3 Display Data

Retrieve records using:

```
SELECT * FROM students;
```

Display the results in an HTML table.

20.4 Error Handling

Check for:

- Connection errors
 - SQL errors
 - Empty form fields
-

Practical Exercise

Develop a **Student Registration System** that:

- Accepts student details
 - Saves them to MySQL
 - Displays all registered students
-

Assignment

Add a search feature to the student registration system.

Class 21 (2 Hours)

CRUD Operations & Basic Login System

Learning Outcomes

Students will be able to:

- Update and delete records.
 - Search database records.
 - Create a simple login system.
 - Build a complete CRUD application.
-

21.1 CRUD Operations

Create

Insert new records.

Read

Display records.

Update

Modify existing records.

Delete

Remove records.

21.2 Search Functionality

Search records by:

- Name
 - Email
 - Department
-

21.3 User Authentication

Basic Login Process:

1. User enters username and password.
 2. PHP checks the credentials.
 3. Session starts if valid.
 4. Redirect to dashboard.
 5. Logout destroys the session.
-

21.4 PHP Sessions

Example:

```
<?php
session_start();

$_SESSION['username']="Admin";
?>
```

21.5 Logout

```
<?php
session_start();
session_destroy();
header("Location:index.php");
?>
```

Practical Exercise

Build a **Student Management System** with:

- Add Student
 - View Student List
 - Edit Student
 - Delete Student
 - Search Student
 - Login & Logout
-

Module Project

Students will develop a **Dynamic Student Management System** with the following features:

Frontend

- Responsive Login Page
- Dashboard
- Student Registration Form
- Student List
- Search Bar

Backend

- PHP Form Processing
- MySQL Database
- CRUD Operations

- Session-Based Authentication

Database

- users
- students

Features

- Add Records
- Edit Records
- Delete Records
- Search Records
- Secure Login
- Logout

Module Assessment

Assessment	Marks
PHP & MySQL Quiz	10
Practical Exercises	20
Assignments	20
Student Management System Project	40
Presentation & Viva	10
Total	100

Learning Outcomes

After completing **Module 5: PHP & MySQL**, students will be able to:

- Develop dynamic websites using PHP.
- Process and validate HTML forms on the server.
- Connect PHP applications to MySQL databases using MySQLi.
- Design and manage relational databases with phpMyAdmin.
- Perform complete CRUD operations (Create, Read, Update, Delete).
- Implement basic user authentication using PHP sessions.
- Build a complete database-driven web application following industry-standard development practices.

- Prepare for advanced topics such as MVC frameworks, REST APIs, and full-stack web development.