SEMESTER-V

COURSE 13 A: WEB APPLICATION DEVELOPMENT USING PHP & MySQL

Theory **Credits: 3** **3 hrs/week**

Course Objectives:

1. Understand the foundational elements of PHP, including variables, data types, operators, and flow control functions.
2. Develop proficiency in managing arrays, objects, strings, dates, and time functionalities using PHP.
3. Design and process HTML forms integrated with PHP, including advanced operations like file uploads, redirection, and exception handling.
4. Implement session management and cookie handling to preserve user state and provide secure, personalized experiences.
5. Connect PHP with MySQL databases, enabling learners to perform CRUD operations and build dynamic web applications.

Course Outcomes

At the end of the course, students will be able to:

1. Demonstrate effective use of PHP building blocks such as variables, expressions, constants, control structures, and functions with appropriate scope and argument handling.
2. Manipulate complex data structures including arrays and objects, and utilize PHP string and date/time functions for dynamic content generation.
3. Create functional web forms using PHP, retrieve form inputs, manage file uploads, and execute form-based operations like redirection and email dispatch.
4. Apply secure techniques for maintaining user sessions and cookies, including session lifecycle control, session variable manipulation, and user authentication workflows.
5. Integrate PHP with MySQL to build database-driven web components, perform record management, and architect structured menu-based data operations.

Unit 1. The building blocks of PHP:
Variables, Data Types, Operators and Expressions, Constants.

Flow Control Functions in PHP: Switching Flow, Loops, Code Blocks and Browser Output.

Working with Functions: Creating functions, Calling functions, Returning the values from User-Defined Functions, Variable Scope, Saving state between Function calls with the static statement, arguments of functions

Unit 2.Working with Arrays:
Creating Arrays, Some Array-Related Functions.

Working with Objects: Creating Objects, Accessing Object Instances,

Working with Strings, **Dates and Time:** Formatting strings with PHP, Manipulating Strings with PHP, Using Date and Time Functions in PHP.

Unit 3.Working with Forms:
Creating Forms, Accessing Form Input with User defined Arrays, Combining HTML and PHP code on a single Page, Using Hidden Fields to save state, Page redirection, Sending Mail on Form Submission, **Working with File Uploads**, Managing files on server, **Exception handling.**

Unit 4. Working with Cookies and User Sessions:
Introducing Cookies, setting a Cookie with PHP, Session Function Overview, starting a Session, working with session variables, passing session IDs in the Query String, Destroying Sessions and Unsetting Variables, Using Sessions in an Environment with Registered Users.

Unit 5.Interacting with MySQL using PHP:
MySQL Versus MySQLi Functions, connecting to MySQL with PHP, Working with MySQL Data. Planning and Creating Database Tables, Creating Menu, Creating Record Addition Mechanism, Viewing Records, Creating the Record Deletion Mechanism.

Text Book(s)

1. SAMS Teach yourself PHP MySQL and Apache,, Julie C. Meloni, Pearson Education
2. PHP: The Complete Reference, Steven Holzner , McGraw-Hill

Reference Books

1. Learning PHP, MySQL, JavaScript, CSS & HTML5, Robin Nixon, Third Edition, O'reilly, 2014
2. The web warrior guide to Web Programming, Xue Bai Michael Ekedahl,, Thomson, 2006.

Activities:

Outcome: Demonstrate effective use of PHP building blocks such as variables, expressions, constants, control structures, and functions with appropriate scope and argument handling.

Activity: Write a PHP script that:

- Declares variables and constants
- Uses expressions and control structures (if, switch, loops)
- Defines and calls functions with arguments and return values
- Demonstrates variable scope (global, local)

Evaluation Method: Evaluate on a 10-point scale based on code review checklist to verify the

- Correct syntax and use of each building block
- Logical flow using control structures
- Proper function definition and scope handling
- Output accuracy and readability

Outcome: Manipulate complex data structures including arrays and objects, and utilize PHP string and date/time functions for dynamic content generation.

Activity: Create a PHP script that:

- Stores student data in arrays and objects
- Formats and manipulates strings (e.g., name formatting)
- Displays current date/time and calculates age from DOB

Evaluation Method: Rubric-based assessment on a 10-point scale to check the:

- Correct use of arrays and objects
- Effective string and date/time functions
- Dynamic output generation
- Code clarity and structure

Outcome: Create functional web forms using PHP, retrieve form inputs, manage file uploads, and execute form-based operations like redirection and email dispatch.

Activity: Build a contact form that:

- Accepts name, email, message
- Uploads a file (e.g., resume)
- Sends an email confirmation
- Redirects to a thank-you page

Evaluation Method: Evaluate on a 10-point scale based on functional testing to perform:

- Form input retrieval and validation
- File upload success
- Email dispatch and redirection
- Error handling and user feedback

Outcome: Apply secure techniques for maintaining user sessions and cookies, including session lifecycle control, session variable manipulation, and user authentication workflows.

Activity: Develop a login system that:

- Starts a session on login
- Stores user data in session variables
- Sets a cookie for Remember Me
- Logs out and destroys session securely

Evaluation Method: Security checklist (10-point scale):

- Session lifecycle control
- Cookie setup with secure flags
- Authentication logic
- Session/cookie cleanup on logout

Outcome: Integrate PHP with MySQL to build database-driven web components, perform record management, and architect structured menu-based data operations.

Activity: Create a student management system:

- Connect to MySQL database
- Add, view, update, delete student records
- Display menu-based navigation (e.g., by class or grade)

Evaluation Method: Database interaction test on a 10-point scale:

- Successful CRUD operations
- Structured menu navigation
- SQL query correctness

SEMESTER-V

COURSE 13 A: WEB APPLICATION DEVELOPMENT USING PHP & MySQL

Practical **Credits: 1** **2 hrs/week**

List of Experiments:

1. Write a PHP program to Display Hello
2. Write a PHP Program to display today's date.
3. Write a PHP program to display Fibonacci series.
4. Write a PHP Program to read the employee details.
5. Write a PHP program to prepare the student marks list.
6. Create student registration form using text box, check box, radio button, select, submit button. And display user inserted values in the new PHP page.
7. Create Website Registration Form using text box, check box, radio button, select, submit button. And display user inserted values in the new PHP page.
8. Write a PHP script to demonstrate passing variables with cookies.
9. Write a PHP script to connect to the MySQL server from your website.
10. Write a program to keep track of how many times a visitor has loaded the page.
11. Write a PHP application to perform CRUD (Create, Read, Update and Delete) operations on a database table.
12. Create a web site using any open-source framework built on PHP and MySQL – It is a team activity wherein students are divided into multiple groups and each group comes up with their own website with basic features.