



Syllabus
Gyanmanjari Institute of Technology
Semester-4 (B. Tech.)

Subject: Fundamentals of Web Development - BETICE14313

Type of course: Multidisciplinary Open Professional Elective Courses

Prerequisite: Basic Knowledge of HTML, CSS and JavaScript

Rationale:

This course introduces the fundamentals of web development using PHP, MySQL, and Apache. Students will learn about webpage and website concepts, PHP features, advantages, limitations, and best coding practices. The syllabus covers PHP syntax, HTML integration, variables, constants, loops, conditional statements, functions, arrays, and file handling, along with forms, GET/POST methods, validations, sessions, cookies, file uploads, error and exception handling, and PHP-MySQL interaction. PHP, a powerful server-side scripting language, enables dynamic web content and supports frameworks that enhance development speed, security, and performance, equipping students with practical skills to build interactive web applications and manage databases efficiently.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks		Total Marks
CI	T	P		SEE	CCE	
4	0	2	5	100	50	150

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; SEE - Semester End Evaluation; CCE-Continuous and Comprehensive Evaluation.



Course Content:

Sr. No	Course Content	Hrs.	% Weightage
1	<p>Introduction to PHP</p> <p><u>Theory Topics:</u></p> <p>Fundamental of webpage, website and apache server, introduction to Apache, MySQL, PHP (AMP Module), PHP Features, advantages and limitations, Brief History of Open Source Initiatives. Installing Apache, MySQL and PHP, Overview of PHP structure and Syntax, PHP and HTML, PHP Structure and Syntax, Importance of Coding Practices, Configuration of PHP, PHP Data Types, String, integer, float, object, array, NULL value, Integrating HTML with PHP, Using Constants and Variables, Passing Variables between pages, Passing Variables through a URL, Using Includes for Efficient Code, Alternate Syntax for PHP: Alternates to the echo Command, Alternates to Logical Operator, Alternates to Double Quotes, Alternates to Increment/Decrement Values.</p> <p><u>Practical:</u></p> <ol style="list-style-type: none"> 1. Develop a simple webpage that outputs the text Hello World using a PHP script. 2. Write a PHP script to display "Welcome to PHP Programming" using proper PHP structure and syntax. 3. Design a webpage that illustrates how to use single-line and multi-line comments in PHP code. 4. Write a PHP script to demonstrate all PHP data types (string, integer, float, array, object, and NULL) and display their values. 5. Create a PHP program that integrates HTML with PHP, uses variables and constants, and performs basic arithmetic operations. 6. Create a webpage with two variables — one holding a numeric value and another a string — and print both values on the screen. 7. Develop a webpage that uses the rand() function to generate and display a random number between 1 and 100. 8. Create a PHP page that generates a random number and calculates its square root using the sqrt() function. 9. Build your personal resume webpage using HTML, including headings, lists, and different text formatting styles. 10. Develop a PHP-based text story game by providing an input form for user choices and displaying customized story outcomes. 11. Design a form that accepts user inputs for page title, background color, font color, and content text, and then use PHP to generate a styled HTML page. 12. Write a PHP script demonstrating alternate PHP syntax including alternatives to echo, logical operators, double quotes, and increment/decrement operators. 	18	20%



Evaluation Method:			
Sr. No.	Evaluation Methods	SEE	CCE
1	Build Your Own Dynamic Webpage Students have to create an interactive PHP webpage on give topic that displays a personalized welcome message using form input and basic styling.	20	
2	Active Learning Assignment The Web Demystified: Static vs Dynamic Website Demo Students will practically demonstrate the difference between a static website and dynamic website through hands-on implementation. Save your work as a PDF and upload it to the GMIU portal.		10
	Total	20	10

Examination Style:

Build Your Own Dynamic Webpage (20 Marks)
Design an interactive webpage using PHP, HTML, and CSS according to the topic or screenshot shared by the faculty. The page should include a form where users enter their name, department, and interests. Upon submission, PHP will display a personalized welcome message (e.g., "Welcome, Ram from CSE!"). This activity helps students build a foundation for creang personal portfolio or professional profile webpages. Use of AI in any form is strictly prohibited in the creation of the web pages.

The Web Demystified: Static vs Dynamic Website Demo (10 Marks)
Students will design and deploy two mini projects—one static website using HTML and CSS and one dynamic website using PHP on a local server (XAMPP)—to perform, observe, and clearly differentiate between static and dynamic websites, understand the roles of webpages, websites, and server-side scripting, analyze client-side versus server-side processing, and demonstrate how PHP enables dynamic, user-driven content. highlighting how modern websites integrate static design with dynamic functionality for interactive user experiences. Save your work as a PDF and upload it to the GMIU portal.



<p>2</p>	<p>Conditional Statements, Looping, PHP Functions and Arrays</p> <p><u>Theory Topics:</u></p> <p>Conditional Statements: If-else statement, If-else if statement, Switch statement, Loops: While loop, Do..while loop, For loop, For-each loop, Functions: Create a function, Function with one argument, Function with two argument, Function with default argument value, Functions returns a value, Arrays: Introduction to Array, Creating Arrays, Storing values in an array, Initializing Arrays, Creating Numeric Arrays, Associative Arrays, Multidimensional Arrays, Array related Function, PHP Server variables.</p> <p><u>Practical:</u></p> <ol style="list-style-type: none"> 13. Write a PHP script to enter a number and check whether it is prime. 14. Write a PHP script to input two numbers and print all Armstrong numbers within the given range. 15. Write a PHP function that checks if a number is greater than 30, 20, or 10 using a ternary operator and displays the result. 16. Write a PHP function that checks whether a given number is prime or not and displays the result. 17. Write a PHP script that uses a for loop to calculate the sum of all integers from 0 to 30 and display the final total. 18. Create a PHP program that uses a for loop to compute and display the factorial of a given number. 19. Write a PHP script to generate and display the Fibonacci series. 20. Create a PHP script that iterates through numbers 1 to 100 and: <ul style="list-style-type: none"> • Prints "Fizz" for multiples of 3, • Prints "Buzz" for multiples of 5, • Prints "FizzBuzz" for numbers divisible by both 3 and 5. 21. Write a PHP script to calculate 5! using a function. 22. Create a PHP function that accepts a non-negative integer as an argument and returns its factorial. 23. Create a PHP web application with fruits array and display each fruit name using a foreach loop. 24. Develop a PHP web application that stores an array of animals along with their heights. Use comparison operators to determine and display the tallest animal. 25. Write a PHP script to extract and display the first element from the array: <code>\$color = array(4 => 'white', 6 => 'blue', 11 => 'red');</code> Output: white 	<p>18</p>	<p>20%</p>
----------	---	-----------	------------

26. Write a PHP script that sorts an array of colors in ascending (alphabetical) order and displays them as:
 blue
 red
 white
27. Develop a PHP program that uses an associative array of countries and their capitals (e.g., "France" => "Paris"). Sort the array alphabetically by country name and display each country-capital pair.

Evaluation Method:

Sr. No.	Evaluation Methods	SEE	CCE
1	Dynamic E-Commerce Style Product Display in PHP: Students will display a list of products with details from an associative array. User can apply filter on it.	20	
2	Active Learning Assignment PHP Household Budget Calculator Students will create a small calculator for monthly expense tracking: input rent, groceries, transport, etc. Save your work as a PDF and upload it to the GMIU portal.		10
Total		20	10

Examination Style:

Dynamic E-Commerce Style Product Display in PHP (20 Marks)
 Students are required to build a product (like laptop, phone, accessories) listing page in PHP that uses associative arrays and loops to show product information, including name, price, and description. A category filter may be added to display selected groups of products, reflecting common e-commerce functionality.

PHP Household Budget Calculator (10 Marks)
 Students will build a PHP monthly expense tracker that calculates total expenses from user input and uses conditional statements to display remarks like "Under budget" or "Over budget," applying arithmetic and control structures in a real-life scenario. Save your work as a PDF and upload it to the GMIU portal.



3	<p>Built-in Functions and Working with Forms</p> <p><u>Theory Topics:</u></p> <p>Built-in Functions :String Functions and Formatting Strings with PHP, Mathematical Functions Dates and Time Function, Array Function, File Function. Forms: Form elements-TextBox, TextArea, Password, RadioButton, CheckBox, Image, Buttons-Submit and Reset, Multiple Submit Buttons, Creating Simple Input Form, Accessing Form Input with Use-Defined Arrays, Uploading file to web server, GET & POST method, PHP include and require statement. Submitting form values, using \$_Get and \$_Post Method, \$_REQUEST, Accessing form inputs with Get/Post functions, PHP Form Validations: Preg_Match and Preg_Match_all, PHP Required Field Validation, PHP Email Validation, Using CSS in Forms, Using hidden fields to Save State, Combining HTML and PHP codes together on single page, Redirecting to the user.</p> <p><u>Practical:</u></p> <p>28. Write a PHP script demonstrating all string and math functions. 29. Write a PHP script to remove the first part of a string up to a specific character. 30. Example: For the string 'gmiu@domain.com', the output should be 'domain.com'. 31. Write a PHP script that displays a string and its corresponding values in a table format:</p> <table border="1" style="margin-left: 40px;"> <tr> <td>Name</td> <td>Salary</td> </tr> <tr> <td>Ram</td> <td>10000 Rs.</td> </tr> <tr> <td>Sita</td> <td>25000 Rs.</td> </tr> <tr> <td>Laxman</td> <td>15000 Rs.</td> </tr> </table> <p>32. Write a PHP script demonstrating all date, time, array, and file handling functions. 33. Create a PHP script for a birthday countdown, calculating the number of days between the current date and the user's birthday. 34. Create a PHP script to display the current date in multiple formats using the date() function. Example formats (assuming the date is March 15, 2025): 2025/03/15 25.03.15 15-03-25</p> <p>35. Write a PHP script to calculate the difference between two dates. Example: If date1 = 2025-03-01 and date2 = 2025-03-15, difference = 14 days. 36. Create a PHP script to convert a date from yyyy-mm-dd to dd-mm-yyyy.</p>	Name	Salary	Ram	10000 Rs.	Sita	25000 Rs.	Laxman	15000 Rs.	18	20%
Name	Salary										
Ram	10000 Rs.										
Sita	25000 Rs.										
Laxman	15000 Rs.										



<p>37. Create a PHP script to convert a given number of seconds into days, hours, minutes, and seconds.</p> <p>38. Develop a web page that dynamically generates a year selection list using PHP, allowing the user to choose a year from the dropdown.</p> <p>39. Develop a PHP script that counts and displays the number of lines in a file.</p> <p>40. Design a simple HTML form that accepts a username and displays the entered name using a PHP echo statement.</p> <p>41. Create a student registration form using text boxes, checkboxes, radio buttons, drop-down select, and a submit button, then display the submitted values on a new PHP page.</p> <p>42. Create a form with radio buttons for gender selection, a textbox for the user's name, and a textarea for feedback, then display the submitted data using the \$_POST method.</p> <p>43. Create a form to collect user information (name, age, email) using both GET and POST methods, then display the submitted data.</p> <p>44. Develop a web application form containing fields for nickname, firstname, and memo. Prompt the user to enter details and display the submitted information on the webpage.</p> <p>45. Create a form with a submit button to save user data (name, email, hobbies, gender, city, known languages) and display it on the same page.</p> <p>46. Create a PHP script that will have firstname and lastname fields and use JavaScript validation to ensure only letters are accepted.</p> <p>47. Create a PHP script that prompts the user to enter a username and password and validates the password to ensure it is 7-16 characters, contains at least one numeric digit, one uppercase, and one lowercase letter.</p> <p>48. Design a Travel Booking Form with the following fields and apply appropriate validations:</p> <ul style="list-style-type: none">• Full Name*• Email Address*• Select Destination Package*• Departure Date*• Number of Travelers*• Additional Services to Include*• Agree to Terms and Conditions* <p><i>(Fields marked with * are mandatory)</i></p>	
---	--

Evaluation Method:			
Sr. No.	Evaluation Methods	SEE	CCE
1	PHP Input Validation and Statistical Processing Application: Develop a PHP system to collect, validate, store, and analyze user data, highlighting records that fall below a defined limit.	20	
2	Active Learning Assignment AI-Powered Feedback Analysis Web App Build a PHP-MySQL web application that collects user feedback and uses an AI tool to analyze sentiment or summarize comments. Save your work as a PDF and upload it to the GMIU portal.		10
	Total	20	10

Examination Style:

PHP Input Validation and Statistical Processing Application (20 Marks)
Build a PHP application to record and validate user inputs, store the data in a file, compute relevant statistics, and highlight entries that fall below a specified threshold. This task practices working with forms, arrays, file handling, string formatting, and date/time functions.

AI-Powered Feedback Analysis Web App (10 Marks)
students will develop a PHP-MySQL web application that collects user feedback through a form capturing Name, Email, Rating, and Comments. The feedback will be stored in a MySQL database, and an AI API (such as OpenAI or Hugging Face) will be used to analyze the comments, detecting sentiment or generating summaries. The application will display feedback in a table along with AI-generated insights, average ratings, and sentiment statistics, with optional features like highlighting negative feedback or generating weekly reports. This project helps students practice PHP forms, CRUD operations, database handling, API integration, and data analysis. Save your work as a PDF and upload it to the GMIU portal.



<p>4</p>	<p>Sessions, Cookies, Images, Files and Error Handling</p> <p><u>Theory Topics:</u></p> <p>Sessions and Cookies: Introduction to Session, Working with Session Variables, creating and deleting session variable, Starting and Destroying session, Passing session IDs, Introduction to Cookies, Creating Cookies with PHP, Deleting Cookies, Creating session cookie working with the query string, Allowing User to Upload Images, Working with file and directories, Understanding files and directory, Opening and Closing file, Coping, renaming and deleting file, Working with directories, File uploading and downloading. Error types in PHP, Exception handling in PHP, Error Handling – Try, Catch and Throw block, die() function.</p> <p><u>Practical:</u></p> <p>49. Create a PHP script that sets two session variables, color and animal, assigns values to them, and accesses the session information on another page to display the stored values.</p> <p>50. Develop a PHP script that creates a session variable called counter and increments its value each time the user visits the page, showing the updated count.</p> <p>51. Write a PHP script to start a session, store the user's name in a session variable, display the stored value, and then destroy the session to remove all stored information.</p> <p>52. Create a PHP script that explicitly passes the session ID in the URL to maintain session state across multiple pages.</p> <p>53. Create a simple web page with Login and Logout functionality using PHP sessions to manage user authentication.</p> <p>54. Create a PHP script to define a cookie named "user" with the value "BTech". The cookie should expire after 30 days.</p> <p>55. Write a PHP script to set a cookie with the user's name and an expiration time of 1 hour, then display the cookie value on the page.</p> <p>56. Write a PHP script to determine whether cookies are enabled in the user's browser and display an appropriate message.</p> <p>57. Write PHP syntax to create a new file (e.g., example.txt) and open it in write mode to store content.</p> <p>58. Create PHP code to open a file and read its contents, then display them on the web page.</p> <p>59. Write PHP code to open an existing file, e.g., "newfile.txt", and append or write new data into it.</p> <p>60. Create a PHP script that attempts to open a file and displays a message to the user if the file cannot be opened. Example: If data.txt does not exist, display "Unable to open the file."</p>	<p>18</p>	<p>20%</p>
----------	---	-----------	------------



61. Create a file upload form where users can upload any file type, display the file name, type, and size, and store images in a specific folder.
62. Write a PHP script that allows the user to upload an image file to a server folder and checks whether the upload was successful.
63. Write a PHP script to throw a custom exception if a number is negative, using try-catch blocks for handling.

Evaluation Method:

Sr. No.	Evaluation Methods	SEE	CCE
1	PHP Session - Based Item Management System: Students will create a PHP shopping cart using sessions to add, update, remove items, and display totals dynamically.	20	
2	Interactive PHP Image Upload and Gallery System Build an interactive PHP app for validated image uploads, dynamic gallery display, and session-based upload management. Save your work as a PDF and upload it to the GMIU portal.		10
Total		20	10

Examination Style:

PHP Session - Based Item Management System (20 Marks)

Students will create a PHP application that uses sessions to manage a list of selectable items. The system will allow users to add items, update quantities, remove entries, and view a dynamically calculated summary. Session data should reset when the user completes the process or logs out. This project reinforces concepts such as sessions, arrays, calculations, session handling, and dynamic page updates.

Interactive PHP Image Upload and Gallery System (10 Marks)

Students will build an interactive PHP application that enables users to upload images with proper validations and display them in a dynamic gallery. The project involves creating a user-friendly form, handling file uploads using PHP's file handling functions, and managing sessions to track and organize uploads. Each group will showcase their gallery and explain the techniques they used for file handling and session management. Save your work as a PDF and upload it to the GMIU portal.



5	<p>Database Connectivity</p> <p><u>Theory Topics:</u></p> <p>Using PHP and MySQL, MySQL architecture, MySQL Syntax and Commands, PHP with MySQL, Connecting to the MySQL Server, Functions Accessing MYSQL from PHP, Mysqli_Connect(), Mysqli_Select_db(), Mysqli_Query(), mysqli_fetch_array(), Extract(), mysqli_close(), mysqli_numrows(), Database interaction: creating and connecting database, Creating Table in PHP, Getting Information from the User, Executing Commands – Selecting, Inserting, Updating, Deleting, Searching Records from the Table, Small Application development page redirection in PHP, Website Hosting, Domain Name and Web Space, Types of Hosting: Free web hosting, Shared hosting, Dedicated server hosting, Website Hosting Companies Name</p> <p><u>Practical:</u></p> <ol style="list-style-type: none"> 64. Write a PHP script to connect to a MySQL server and select a database (e.g., school_db). 65. Create a PHP script to generate a table (e.g., users table with fields id, name, and email) in a MySQL database, and dynamically insert data using an HTML form. 66. Create a PHP web page that retrieves and displays all student records from a students table. 67. Develop a PHP program to read customer information (cust_no, cust_name, item_purchase, mob_no) from the customer table and display the data in a table format on the output screen. 68. Develop a PHP page that prompts the user to add new student records into the students table. 69. Design a user registration form with required fields like name, email, password, etc., and save the submitted data into a users table. 70. Create a web form to input mobile phone details including model, price, brand, and specifications, and store the submitted data into a mobile_products table. 71. Build a feedback page with fields like username, email, comments, etc., and save the feedback into a feedback table. 72. Develop a Contact Us page with fields like name, email, message, and store the submitted information in a contacts table. 73. Create a web page that displays all orders received and allows the user to update order status or quantity, saving changes in an orders table. 74. Design a web page following a specific layout and implement Insert, Update, Delete, and Display operations for the employees table. 	18	20%
---	---	----	-----



- 75. Create a PHP web page that allows the user to upload a profile picture, store it in the database, and display the image on the page.
- 76. Create a PHP web page where the user can select one cuisine option (Thai, Japanese, Mexican, or Italian) using radio buttons. Display the selected cuisine in a message and store the choice in a user_preferences table.
- 77. Create a login form in PHP that authenticates users by checking their email and password against records stored in the database.
- 78. Develop a small registration application using PHP and MySQL where users can register with their name, email, and password. Implement full CRUD (Create, Read, Update, Delete) and display a success message.

Evaluation Method:

Sr. No.	Evaluation Methods	SEE	CCE
1	Web-Based PHP-MySQL Data Management: Create a PHP & MySQL application to track, store, and display website visitor activity, including visit details, search, and navigation, practicing database operations, queries, and dynamic page display.	20	
2	Mini Project: Dynamic PHP Task Management System with Report: Students will develop a dynamic PHP-MySQL web application to add, update, and delete tasks. The project will involve PHP CRUD operations, form handling, and session management. Completed projects report in pdf form will be submitted via the GMIU portal.		10
Total		20	10



<p>Examination Style:</p> <p>Web-Based PHP-MySQL Data Management (20 Marks) Create a simple PHP web application connected to a MySQL database that enables users to add, view, update, and delete records from a single table. The application should connect to the MySQL database using <code>mysqli_connect()</code> and select the appropriate database, creating a table via PHP if it does not already exist. Users should be able to insert new records through a form, and all records should be displayed in a table with options to update or delete individual entries. The application must make proper use of PHP-MySQL functions such as <code>mysqli_query()</code>, <code>mysqli_fetch_array()</code>, <code>mysqli_num_rows()</code>, <code>extract()</code>, and <code>mysqli_close()</code>, and include simple page redirection after performing insert, update, or delete actions. Demonstrates fundamental PHP-MySQL CRUD operations along with dynamic page handling. During the examination, students must complete the CRUD operations specified by the faculty.</p> <p>Mini Project: Dynamic PHP Task Management System with Report (10 Marks) Students will work in group of two to develop a dynamic web application using PHP and MySQL. The application will allow users to add, update, and delete tasks, with all data securely stored and managed in a database. Students will implement PHP CRUD operations, handle forms efficiently, and manage sessions for a seamless user experience. The completed project will be submitted and uploaded to the GMIU portal.</p>	
--	--

Suggested Specification Table:

Distribution of Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage %	10%	10%	25%	15%	15%	25%

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from the above table.



Course Outcome:

After learning the course, the students should be able to:	
CO1	Design and develop interactive, user-friendly, and dynamic web applications.
CO2	Apply PHP techniques for string manipulation, date and time processing, and form handling.
CO3	Implement and evaluate methods for transferring data between web pages using PHP, including the GET and POST mechanisms.
CO4	Create and manage PHP scripts that utilize Sessions and Cookies for state management.
CO5	Build PHP applications that connect to MySQL databases to perform data access and manipulation operations

Instructional Method:

The course delivery method will depend upon the requirement of content and needs of students. The teacher, in addition to conventional teaching methods by black board, may also use any tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

From the content 10% topics are suggested for flipped mode instruction. Students will use supplementary resources such as online videos, NPTEL/SWAYAM videos, e-courses, Virtual Laboratory.

The internal evaluation will be done on the basis of the Active Learning Assignment.

Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in the laboratory.

Reference Books:

- [1] The Complete Reference PHP by Steven Holzner, McGraw-Hill.
- [2] PHP6 and MySQL Bible by Steve Suehring, Tim Converse, Joyce Park, WILEY
- [3] Beginning PHP6, Apache, MySQL by Timothy Boronczyk, Elizabeth, Naramore, Jason Gerner, Wiley Publishing Inc
- [4] SamsTeachYourself PHP, MySQL & JavaScript by Julie C. Meloni, Pearson.
- [5] PHP: A BEGINNER'S GUIDE by Vikram Vaswani, McGraw-Hill.



Suggested Assessment Guidelines

Module-1: Introduction to PHP

Build Your Own Dynamic Webpage (20 Marks)		
Criteria	Description	Marks
Problem Understanding	Student clearly identified the task: creating an interactive webpage using PHP, HTML, CSS with a form (name, department, interests) and generating a personalized welcome message using PHP. Understanding of constraints (e.g., No AI usage) is evident.	5
Design	Logical planning of webpage structure, layout, form design, styling approach, and PHP processing flow. Proper explanation of how data flows from form → PHP → output.	5
Methodology	Use of appropriate tags, styling methods, and structured methodology.	5
Implementation	Webpage is functional: form inputs work, PHP processes user data, and personalized message (e.g., "Welcome, Ram from CSE!") is displayed correctly. Proper folder structure, error-free code, and working demonstration/preview.	5
Total		20

Module-2: Conditional Statements, Looping, PHP Functions and Arrays

Dynamic E-Commerce Style Product Display in PHP (20 Marks)		
Criteria	Description	Marks
Product Data Structure (Arrays)	Proper use of numeric, associative, and/or multidimensional arrays to store product details. Correct use of array functions.	5
Looping & Conditional Logic	Correct use of loops (for/foreach/while) to display products and conditions for category filtering.	5
Functions Usage & Category Filter Functionality	Use of functions with parameters, default values, and return values to organize code. Category-based product display works correctly and dynamically updates results.	5
Use of PHP Server Variables, Output Formatting, Code Quality & Organization	Meaningful use of \$_SERVER variables (e.g., script name, request method) to improve interaction. Clean layout, readable product display, proper HTML-PHP integration. Proper indentation, comments, meaningful variable names, modular structure.	5
Total		20



Module-3: Built-in Functions and Working with Forms

PHP Input Validation and Statistical Processing Application (20 Marks)		
Criteria	Description	Marks
Form Design & Input Validation	Proper use of form fields (text, email, date, hidden, file upload). Correct GET/POST/REQUEST handling. Includes required field checks, email validation, regex (preg_match), file validation, and error handling.	5
Logic, Calculations & Data Processing	Accurate calculation of totals, percentages, and identifying entries below threshold (e.g., 75%). Effective use of arrays, string/date/math functions, and structured processing logic.	5
File Handling & Code Structure	Correct use of PHP file functions for storing and retrieving data. Includes modular code using include/require, proper redirection, and clean organization of logic.	5
Output Display & UI	Clear, user-friendly display of summary and status. Clean, organized, commented code with correct execution.	5
Total		20

Module-4: Sessions, Cookies, Images, Files and Error Handling

PHP Session - Based Item Management System		
Criteria	Description	Marks
Session management	Correct use of sessions to add, update, remove items; session ID handling; session destruction on checkout.	5
Item operations and calculations	Accurate adding, updating quantities, removing entries, and displaying a dynamically calculated summary. Totals and any derived values update correctly after each operation. Functionality matches the stated workflow of the system.	5
Error/Exception Handling and Dynamic Page Display	Effective use of try-catch-throw, die(), and proper handling of runtime errors. Correct display of cart contents, totals, product list, and messages in a user-friendly layout.	5
Code Quality & Structure	Clean coding style, comments, meaningful variable names, well-structured PHP + HTML integration.	5
Total		20



Module-5: Database Connectivity

Web-Based PHP-MySQL Data Management (20 Marks)		
Criteria	Description	Marks
Database Connectivity & Table Setup	Correct use of <code>mysqli_connect()</code> , selection of database, connection error handling, and creation of table via PHP if it does not exist. Proper use of MySQL connection functions such as <code>mysqli_query()</code> , <code>mysqli_num_rows()</code> , and <code>mysqli_close()</code> .	5
CRUD Functionality (Insert, View, Update, Delete)	Application correctly implements all CRUD operations. Form input successfully inserts new records, displayed table shows all records, and update/delete actions function correctly with appropriate page redirection using <code>header()</code> .	5
Data Retrieval & Dynamic Display	Effective use of functions like <code>mysqli_fetch_array()</code> , <code>extract()</code> , and loops to dynamically display database records in an HTML table. Clean, user-friendly interface showing action buttons for update/delete.	5
Code Quality, Structure & Execution	Code is well-structured, readable, and properly commented. Logical separation of PHP and HTML, secure handling of queries, and smooth execution without runtime errors. Includes proper redirection after CRUD actions and overall consistent workflow.	5
Total		20

