Gyanmanjari Innovative University

Course Syllabus
Gyanmanjari Institute of Technology
Semester-5 (B. Tech)

Subject: Advance Web Development- BETCE15316

Type of course: Professional Core and Professionals Elective Courses

Prerequisite: Basic Knowledge of PHP, HTML, CSS and JavaScript

Rationale:

In the era of digitization, the demand for Internet-based applications is surging, driving the need for skilled developer's adept in both front-end and back-end design. This comprehensive course aims to immerse students in the intricacies of web development, empowering them to navigate the dynamic landscape of the Internet-driven world with confidence. By focusing on both front-end and back-end design principles, learners will gain a holistic understanding of the development process, from creating visually engaging user interfaces to implementing robust server-side functionalities.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
				Theory Marks		Practical Marks		CA	
CI	T	P	C	ESE	MSE	V	P	ALA	
3	0	2	4	60	30	10	20	30	150

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; ESE - End Semester Examination; MSE- Mid Semester Examination; V – Viva; CA - Continuous Assessment; ALA- Active Learning Activities.

Course Content:

Sr. No	Course content	Hrs.	Weightage %
1	**Advanced PHP concepts** **OOP Concepts** Classes, Objects, Constructors, Destructors, Inheritance and Method Overriding, Abstract Classes & Interfaces, Traits & Multiple Inheritance **Advanced Features** Namespaces & Autoloading, Magic Methods (__construct, __destruct, __call, etc). **Error Handling** Custom Exception Handling	07	15%

2	**Secure Database Connectivity (PDO & Prepared Statements)** **Introduction to PDO** Benefits of PDO over MySQLi, Setting Up Database Connection **Executing Queries** Fetching Data (fetch, fetchAll, fetchColumn), Executing Non-SELECT Queries (INSERT, UPDATE, DELETE) **Security Measures** Prepared Statements & Binding Parameters, Preventing SQL Injection, Using Transactions for Data Integrity **Error Handling in PDO** Try-Catch Blocks for Database Errors	11	25%
3	**Advanced Form Handling & Session Management** **Secure Form Validation** Validating Input Fields (Sanitization & Validation), Handling Form Errors **File Uploads with Security Measures** Restricting File Types & Sizes, Preventing Directory Traversal Attacks **Security Measures in Sessions** Preventing Session Hijacking & Fixation, Implementing Secure Cookies **CAPTCHA Integration** Google reCAPTCHA Implementation	07	15%
4	**AJAX & PHP** **Introduction to AJAX** Synchronous vs Asynchronous Requests, Setting Up an AJAX Call with JavaScript **Working with JSON in PHP** Encoding & Decoding JSON, Sending JSON Responses in PHP **Fetch API vs XMLHttpRequest** Making GET & POST Requests Using Fetch API **Handling AJAX Errors** Server-Side Error Handling, Client-Side Error Handling	07	15%
5	**Web Services & API Development** **Building RESTful APIs using PHP** Understanding RESTful Architecture, Creating API Endpoints with PHP **Working with JSON & XML** Generating JSON & XML Responses, Parsing JSON & XML Data **API Authentication** Implementing JWT (JSON Web Token) Authentication, OAuth 2.0 Basics **Consuming Third-Party APIs** Sending Requests to External APIs, Handling API Responses & Errors	09	15%

6	**Introduction to PHP Frameworks** **Overview of PHP Frameworks** Purpose and Benefits of Using Frameworks, advantages and disadvantages of framework **Popular PHP Frameworks** Laravel, CodeIgniter, Symfony, Yii, CakePHP (Brief Comparison) **Basic Laravel Introduction** Installation, Directory Structure, Routing, Controllers, and Blade Templates.	04	10%

Continuous Assessment:

Sr. No	Active Learning Activities	Marks
1	**PDO Secure CRUD Challenge** In this challenge, students will create a secure web application using PHP Data Objects (PDO) for database interaction, preventing SQL injection with prepared statements. They will implement the four CRUD operations (Create, Read, Update, Delete) while ensuring secure data handling through validation and sanitization. Once completed, students must upload the code individually in the GMIU portal for evaluation.	10
2	**AJAX & API Integration Task** In this task, students will build an app using AJAX to fetch real-time data from the server without reloading the page, enhancing user experience with dynamic updates. They will also integrate third-party APIs, such as weather or news data, to expand functionality. This task helps students learn to create interactive, real-time web applications and utilize APIs for added features. Once completed, students must upload the code individually in the GMIU portal for evaluation.	10
3	**Mini Project** In this Mini Project, students will build a secure web application using PDO for database operations (CRUD) and implement security measures like prepared statements and data validation. They will use AJAX for real-time data fetching without page reloads and integrate third-party APIs (e.g., weather, news). The project also includes session management, form validation, and CAPTCHA integration for security. Upon completion, students must upload the code individually in the GMIU portal for evaluation.	10

Suggested Specification table with Marks (Theory):60

Distribution of Theory Marks (Revised Bloom's Taxonomy)						
Level	**Remembrance (R)**	**Understanding (U)**	**Application (A)**	**Analyze (N)**	**Evaluate (E)**	**Create (C)**
Weightage	10%	20%	30%	10%	20%	10%

Course Outcome:

After learning the course, the students should be able to:	
CO1	Understand Object-Oriented PHP Concepts.
CO2	Implement Secure Database Connectivity using PDO & Prepared Statements.
CO3	Develop Secure Form Handling & Session Management Techniques.
CO4	Build Asynchronous Web Applications using AJAX & APIs.
CO5	Generate and structure API responses in JSON and XML formats.
CO6	Explore PHP frameworks & implement basic of Laravel.

List of Practical

Sr. No	Descriptions	Unit No	Hrs.
1	**Create a PHP Class and Object**: Write a PHP script to create a class with properties and methods, instantiate the class, and work with objects.	1	2
2	**Inheritance and Abstract Classes**: Create a parent class with some common properties, and a child class that extends the parent class and overrides its methods. Demonstrate abstract class usage.	1	2
3	**Error Handling with Try-Catch**: Develop a PHP script that uses try-catch blocks to handle runtime exceptions. Include custom exception handling for a specific error type.	1	1
4	**PDO Connection and Query Execution**: Write a PHP script that connects to a MySQL database using PDO, executes a SELECT query, and displays the results.	2	2
5	**Prepared Statements to Prevent SQL Injection**: Write a PHP script that uses prepared statements with PDO to securely insert user input into a database without exposing the application to SQL injection attacks.	2	2
6	**PDO Transactions**: Create a PHP script that demonstrates the use of PDO transactions to ensure that multiple database queries are executed as a single unit of work.	2	2
7	**Session Handling for Login System**: Write a PHP script to create a login form that uses sessions to store user data after successful authentication, and a logout mechanism to destroy sessions.	3	2
8	**Form Validation in PHP**: Develop a PHP script that validates form data by checking for valid input (e.g., email, password) and ensures proper sanitization to avoid security risks.	3	2
9	**Secure File Upload:** Write a PHP script that allows file uploads with file type and size restrictions, checking for security vulnerabilities like directory traversal.	3	2
10	**Integrate CAPTCHA in Form**: Implement Google reCAPTCHA or another CAPTCHA service in a PHP form to prevent bot submissions.	3	2
11	**AJAX Data Fetching**: Create a PHP script and a corresponding JavaScript file that allows asynchronous data fetching from the server without reloading the page, demonstrating how to handle AJAX requests in PHP.	4	2

12	**JSON Data Handling in PHP**: Write a PHP script that encodes an array of data into JSON format and decodes a JSON string back to a PHP array.	4	2
13	**Building a RESTful API**: Develop a simple RESTful API in PHP that allows creating, reading, updating, and deleting records from a database using GET, POST, PUT, and DELETE methods.	5	3
14	**JWT Authentication for API**: Write a PHP script to implement JSON Web Token (JWT) authentication, where users can log in and receive a token to access secured API endpoints.	5	2
15	**Consume Third-Party API**: Create a PHP script that fetches data from a third-party API (e.g., weather API) and displays the results in the application.	5	2
		Total	**30**

Instructional Method:

The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of 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 Active Learning Assignment.

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

Reference Books:

[1] PHP and MySQL Web Development - Luke Welling & Laura Thomson
[2] PHP Object-Oriented Solutions - David Powers
[3] Modern PHP: New Features and Good Practices - Josh Lockhart
[4] Laravel: Up and Running - Matt Stauffer
[5] PHP Objects, Patterns, and Practice" – Mika Schwartz