

GANPAT UNIVERSITY				
FACULTY OF DIPLOMA ENGINEERING				
Programme	Diploma in Computer Engineering/ Information Technology			
Semester	IV	Version	1.0.0.0	
Effective from Academic Year	2026-27	Effective for the batch Admitted in	JULY 2025	
Course Code	ICEIT4105	Course Name	Full-Stack Web Development	

I. TEACHING-LEARNING AND ASSESSMENT SCHEME																				
Course Type	Course code	Course Title	Teaching & Learning Scheme									Examination Scheme								
			Credit				Actual Contact Hrs/week			SLH	Total Learning Hrs/Week	TH			PR			SLA	Total	
			CL	TL	LL	Total	CL	TL	LL			CE	SEE-	Total	CE	SEE	Total			
DSC	ICEIT4105	Full-Stack Web Development	3	-	2	5	3	-	4	2	9	40	60	100	60	40	100	20	220	

Abbreviation:	CL- Classroom Learning	TL- Tutorial Learning	LL-Laboratory Learning
	SLH-Self Learning Hours	SLA-Self Learning Assessment	CE - Continuous Evaluation
	SEE – Semester End Examination		

II. PRE-REQUISITES

A foundational knowledge of HTML5/CSS3 for UI structure, core problem-solving abilities, familiarity with basic database concepts, and experience navigating a computer's file system.

III. INDUSTRY /EMPLOYER EXPECTED OUTCOMES

- Writing clean, modular, and Type-Strict functions.
- Mandatory use of Prepared Statements for all DB queries.
- Designing Normalized schemas that prevent data redundancy.
- Efficient Memory Management (Garbage collection/unset).

IV. COURSE LEARNING OUTCOMES

At the end of the course, students will be able to achieve the following course learning outcomes:

- CO1.** Analyse & Configure Web Architectures.
CO2. Construct Logic & Dynamic Templates.
CO3. Engineer Modular & Scalable Data Structures.
CO4. Implement Secure Web Interactions & State Management.
CO5. Develop Integrated Database-Driven Applications.

V. THEORY LEARNING OUTCOMES AND ALIGNED COURSE CONTENT:

Name of Unit	Theory Learning outcomes (TLO's) aligned to CO's	Learning Content mapped with Theory Learning outcomes (TLO's)&CO's	Marks	Hours
Unit-1 Full-Stack Foundations & PHP Environment	<p>TLO 1.1 Categorize PHP within the programming ecosystem and justify its application in modern full-stack development based on its evolutionary milestones.</p> <p>TLO 1.2 Diagram the Request-Response cycle and trace the interaction between Client-side components and the Server-side stack (Apache, MySQL, PHP).</p> <p>TLO 1.3 Demonstrate the installation and configuration of a local server</p>	<p>1.1 Overview of Programming Types, PHP's role as a Server-Side language, Evolution, version milestones, and the modern job market for Full-Stack developers.</p> <p>1.2 relationship between the Client and the Server-side stack (Apache, MySQL and PHP).</p>	10	8

	<p>environment and manage the internal directory structure for web deployment.</p> <p>TLO 1.4 Integrate PHP scripts within HTML5 documents using correct syntax, tags, and professional documentation practices.</p> <p>TLO 1.5 Implement dynamic variables and fixed constants using industry-standard naming conventions and scope rules.</p> <p>TLO 1.6 Execute type-specific data handling and apply type casting to manage scalars and special data types within a backend logic.</p> <p>TLO 1.7 Analyse the PHP memory lifecycle and apply manual memory cleanup using unset() to develop resource-optimized applications.</p>	<p>1.3 Installation and configuration of XAMPP/WAMP directory structure and the internal workflow of a local development server.</p> <p>1.4 PHP script structure, tags and embedding PHP logic within HTML5 structures.</p> <p>1.5 variables vs. constants, naming conventions, case sensitivity.</p> <p>1.6 Data Type, Type Casting, operators.</p> <p>1.7 Introduction to PHP's memory lifecycle, Garbage Value myth in PHP vs. C++; memory management (unset()).</p>		
<p>Unit-2</p> <p>Logic Flow and Dynamic Content Rendering</p>	<p>TLO 2.1 Implement conditional structures and logical operators to manage multi-condition authentication and decision-making in web applications.</p> <p>TLO 2.2 Utilize nested decision-making and switch-case statements to execute complex business logic and role-based user routing.</p> <p>TLO 2.3 Apply shorthand operators, including Ternary and Null Coalescing, to handle default values and manage undefined data from web forms efficiently.</p> <p>TLO 2.4 Execute entry-controlled iteration using while and for loops to automate repetitive tasks and perform mathematical computations.</p> <p>TLO 2.5 Design exit-controlled loops and incorporate interruptive logic (break/continue) to optimize execution flow and loop performance.</p> <p>TLO 2.6 Construct clean and readable templates by applying alternative colon syntax for embedding logic within HTML5 structures.</p> <p>TLO 2.7 Develop dynamic user interfaces by using Boolean flags and conditional logic to toggle visibility of UI components based on user state.</p> <p>TLO 2.8 Formulate defensive coding</p>	<p>2.1 if, if...else, and elseif structures.</p> <p>2.2 Nested If statements, Switch statement.</p> <p>2.3 shorthand logic (? : and ??)</p> <p>2.4 while and for loops</p> <p>2.5 do...while loops, break and continue.</p> <p>2.6 colon syntax (if...endif;)</p> <p>2.7 flags and conditional logic to toggle UI elements.</p> <p>2.8 Defensive Code, prevent script crashes.</p>	8	6

	strategies and error suppression techniques to ensure script stability and prevent crashes during decision-making.			
Unit-3 Advanced Data Structures and Functional Logic	<p>TLO 3.1 Define and initialize Indexed and Associative arrays to manage key-value data mappings within a script.</p> <p>TLO 3.2 Utilize the foreach iteration pattern to dynamically render backend array data into front-end HTML elements like lists, tables, and cards.</p> <p>TLO 3.3 Design and traverse multidimensional data trees to represent complex structures such as product catalogs or database result sets.</p> <p>TLO 3.4 Create reusable logic blocks using functions that adhere to the "Single Responsibility Principle" to minimize code redundancy.</p> <p>TLO 3.5 Implement function parameters with mandatory/optional values and apply strict type hinting to ensure data integrity.</p> <p>TLO 3.6 Execute data returning strategies and differentiate between Call by Value and Call by Reference to optimize script performance.</p> <p>TLO 3.7 Manage global and local scopes while utilizing static variables and recursion to maintain state within functional logic.</p> <p>TLO 3.8 Apply functional utilities and JSON interoperability tools to transform data for API readiness and external system communication.</p>	<p>3.1 Indexed and Associative arrays.</p> <p>3.2 foreach loop to iterate through arrays.</p> <p>3.3 Nested array structures.</p> <p>3.4 User-Defined Function.</p> <p>3.5 functions with mandatory/optional parameters and Strict Typing.</p> <p>3.6 return statement, Call by Value vs. Call by Reference.</p> <p>3.7 Global vs. Local scope, Static Variables and recursion.</p> <p>3.8 Anonymous Functions (Closures), explode, implode, and json_encode.</p>	14	9
Unit-4 Web Interaction, State & API Integration	<p>TLO 4.1 Design semantic HTML5 form architectures and establish the structural relationship between UI input attributes and PHP backend handlers.</p> <p>TLO 4.2 Generate dynamic forms using PHP and execute secure data transmission using the \$_GET and \$_POST superglobal arrays.</p> <p>TLO 4.3 Implement dual-tier validation strategies using both client-side attributes and server-side logic (isset/empty) to ensure data completeness.</p> <p>TLO 4.4 Apply security sanitization techniques, including htmlspecialchars(), to mitigate Cross-</p>	<p>4.1 HTML5 forms, input types, attributes and relationship between the UI layer and the PHP backend handlers.</p> <p>4.2 \$_GET, \$_POST.</p> <p>4.3 Dual-tier validation</p> <p>4.4 Web Security & Sanitization.</p> <p>4.5 File Handling & Uploads.</p> <p>4.6 JSON format intro, JSON communication with JavaScript or external APIs.</p> <p>4.7 Page Redirection.</p> <p>4.8 Cookies and Sessions.</p>	14	11

	<p>Site Scripting (XSS) and ensure web safety.</p> <p>TLO 4.5 Manage multi-part data transmissions and construct secure file upload systems for documents and images using the \$_FILES superglobal.</p> <p>TLO 4.6 Utilize JSON format and execute data transformation using json_encode() and json_decode() to facilitate interoperability with JavaScript and external APIs.</p> <p>TLO 4.7 Implement the Post-Redirect-Get (PRG) pattern using header() functions to manage page flow and prevent duplicate form submissions.</p> <p>TLO 4.8 Develop secure state management systems, including login and logout functionality, by utilizing persistent Cookies and server-side Sessions.</p>			
<p>Unit-5</p> <p>Relational Database Management and PHP Integration</p>	<p>TLO 5.1 Identify the architectural role of MySQL within the RDBMS framework and its integration as a persistent storage layer in the LAMP stack.</p> <p>TLO 5.2 Design normalized database schemas and specify appropriate SQL data types for efficient table structures.</p> <p>TLO 5.3 Execute database administration tasks, including creating, altering, and dropping tables, using both Command Line Interfaces (CLI) and phpMyAdmin.</p> <p>TLO 5.4 Formulate and execute standard SQL Data Manipulation Language (DML) queries to perform CRUD operations (Create, Read, Update, Delete) on relational data.</p> <p>TLO 5.5 Implement professional database connectivity by transitioning from procedural MySQLi to the PDO (PHP Data Objects) standard to ensure portability and security.</p> <p>TLO 5.6 Construct secure database interactions by implementing the Prepare-Bind-Execute workflow to prevent SQL Injection attacks.</p>	<p>5.1 Architecture and role of MySQL in the LAMP stack.</p> <p>5.2 Data Types & Schema Design.</p> <p>5.3 Operations for Creating, Altering, and Dropping databases and tables using CLI and phpMyAdmin.</p> <p>5.4 SQL Data Manipulation (CRUD).</p> <p>5.5 Database Connectivity Standards.</p> <p>5.6 Secure Querying with Prepared Statements.</p> <p>5.7 Full-Stack Data Integration.</p>	14	11

	TLO 5.7 Develop integrated full-stack applications by bridging web forms with databases, managing transaction success, and implementing robust error handling in PHP.			
--	--	--	--	--

VI. LABORATORY LEARNING OUTCOME AND ALIGNED PRACTICAL			
SR. NO	PRACTICAL/LABORATORY LEARNING OUTCOME(LLO)	PRACTICAL TITLES	RELEVANT COs
1	LLO 1.1 Configure and troubleshoot a local development environment (XAMPP/WAMP) and the request-response cycle.	Server Configuration & Environment Setup (XAMPP/WAMP).	CO 1
2	LLO 1.2 Develop dynamic web components by embedding PHP logic within HTML5 structures using proper naming conventions.	Create a CV using HTML5 and PHP. Use Variables for your name and Constants for your Date of Birth. Perform Type Casting on age calculations.	CO 1
3	LLO 1.3 Analyse the PHP memory lifecycle and demonstrate the use of unset() to manage server resources efficiently.	Create a large array, use memory_get_usage() to track RAM, then use unset() to free the memory and observe the difference.	CO 1
4	LLO 2.1 Implement complex conditional logic and defensive coding	Calculate student grades or employee tax brackets based on input values.	CO 2
5	LLO 2.2 Use loop iterations and control flow modifiers.	Use a for loop to generate a multiplication table and a while loop to process a countdown. Use break and continue to skip specific numbers.	CO 2
6	LLO 3.1 Organize and manipulate complex data sets using arrays.	Define a multidimensional Associative Array of products. Use a foreach loop to display them in a styled HTML table.	CO 3
7	LLO 3.2 Create reusable and robust code modules using user-defined functions with strict type hints and reference passing.	Create a script with User-Defined Functions. Implement Strict Typing for a function that validates passwords and uses Call by Reference to update a login attempt counter.	CO 3
8	LLO 3.3 Transform data between internal PHP structures and external communication formats like JSON and delimited strings.	Take a comma-separated string of skills, convert it to an array using explode(), then convert that array into a JSON object using json_encode().	CO 3
9	LLO 4.1 Understand the relationship between UI and Backend handlers.	Build an HTML5 form using POST. Implement Dual-tier validation.	CO 4
10	LLO 4.2 Mitigate security risks (XSS/Injection).	Create a search bar that demonstrates Sanitization using htmlspecialchars() to prevent script injection.	CO 4
11	LLO 4.3 Build a server-side handler for validated file uploads	Create a file upload form that validates file size, extension (e.g., .jpg only), and moves the file to a permanent directory.	CO 4

	and directory management.		
12	LLO 4.4 Manage state in a stateless protocol (HTTP).	Create a multi-page site. Implement a login that sets a Session for the user and a "Remember Me" Cookie. Use header() for redirection.	CO 4
13	LLO 5.1 Design and implement a normalized relational database schema using both CLI and GUI (phpMyAdmin).	Design a "Library Management" database. Create tables using SQL queries in the CLI, then use phpMyAdmin to add indexes and constraints.	CO 5
14	LLO 5.2 Execute secure CRUD operations by bridging PHP and MySQL through modern connectivity standards.	Build a system to Create, Read, Update, and Delete tasks. Requirement: All queries must use Prepared Statements (PDO or MySQLi).	CO 5
15	LLO 5.3 Demonstrate total competency in the LAMP stack and modern job-market readiness.	Combine all previous modules to create a "Mini-Blog" or "E-commerce Dashboard" featuring user login, dynamic data fetching from MySQL, and JSON-based API output.	CO 5

VII. SUGGESTED MICRO PROJECT/ASSIGNMENTS/ACTIVITIES FOR SELF LEARNING/SKILL DEVELOPMENT (SELF LEARNING)

Micro Projects (Mini Applications / Use Cases)

- Secure "Student Result Portal" (The PDO CRUD)
Skills: Implementing PDO Prepared Statements to handle sensitive data retrieval and preventing SQL Injection through parameter binding.
- Role-Based "Access Control System"
Skills: Mastering Server-Side State Persistence and Session Hijacking prevention.
- Dynamic "E-Commerce Catalog Engine"
Skills: Developing Data-to-UI Mapping logic and mastering Template Syntax for clean separation of backend logic and HTML5.
- Asynchronous "Lead Generation Form"
Skills: Architecting JSON-based API Interoperability and implementing Dual-Tier Validation (Server + Client) with XSS sanitization.
- Persistent "User-Preference Dashboard"
Skills: Managing Client-Side Persistence via HTTP Cookies and executing Relational CRUD operations using a normalized schema.

Self-Learning / Skill Building Activities

1. Take a basic PHP script that uses \$_GET variables directly in a query and rewrite it using PDO Prepared Statements. [Skill Gained: Mastery of Defensive Programming; learning to identify and patch SQL Injection and XSS vulnerabilities before they reach production.]
2. Build a form that uses the Post-Redirect-Get (PRG) workflow to handle data submissions. [Skill Gained: Understanding of HTTP State Management; specifically, how to use header() redirection to prevent the common "Confirm Form Resubmission" browser error.]
3. Create a PHP script that fetches records from a database and converts the output into a JSON format using json_encode(), then test it using an API client like Postman. [Skill Gained: API Interoperability; shifting from rendering HTML to building "Headless" backends that can serve data to Mobile Apps or React frontends.]

4. Purposefully trigger errors (like undefined indexes or PDO exceptions) and use `var_dump()` and `error_log()` to trace the data flow. [Skill Gained: Full-Stack Troubleshooting; developing the ability to diagnose "silent failures" in the server-side lifecycle without relying on visual UI cues.]

VIII. LIST OF INSTRUMENTS / EQUIPMENT / TRAINER BOARD

1	Editor: Visual Studio Code/ Sublime Text / Notepad++/ PHPStorm
2	Software Stacks (Solution Bundles): LAMP/ XAMPP/ WAMP
3	Online LAMP Stack Simulators: PHPSandbox.io / OneCompiler / Replit / OnlineGDB:
4	Collaboration / Documentation Tools: GitHub / Git: Version control and collaboration Google Docs / Word: Report and assignment writing Draw.io / Lucidchart: Flowcharts, memory maps, logic design

IX. LIST OF REFERENCE BOOKS

Sr.No	Title	Author	Publication
1	PHP & MySQL: Server-side Web Development	Jon Duckett	John Wiley & Sons Inc
2	Learning PHP, MySQL & JavaScript	Robin Nixon	O'Reilly, 6th Edition
3	PHP & MySQL: Novice to Ninja	Kevin Yank & Tom Butler	SitePoint, 7th Edition

X. LINK OF LEARNING WEB RESOURCE

1	https://www.php.net/manual/en/
2	https://dev.mysql.com/doc/
3	https://httpd.apache.org/docs/
4	https://www.w3schools.com/php/
5	https://onlinecourses.swayam2.ac.in/aic20_sp32/preview

XI. SUGGESTED WEIGHTAGE TO LEARNING EFFORTS & ASSESSMENT PURPOSE

Unit	Unit Title	Aligned COs	Learning Hours	R-Level	U-Level	A-Level	Total Marks
1	Full-Stack Foundations & PHP Environment	CO1	8	4	6	0	10
2	Logic Flow and Dynamic Content Rendering	CO2	6	0	6	6	12
3	Advanced Data Structures and Functional Logic	CO3	9	0	5	7	12
4	Web Interaction, State & API Integration	CO4	11	0	6	8	14
5	Relational Database Management and PHP Integration	CO5	11	0	2	10	12
Grand Total			45	4	25	31	60

XIII. COs AND POs AND PSOs MAPPING

Course Outcome (Cos)	Programme Outcomes (POs)							Programme Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	2	1	0	3	0	0	0	1	1
CO2	3	3	3	2	2	0	0	0	1	0
CO3	3	3	3	2	2	0	0	1	2	1
CO4	2	3	3	2	3	2	0	3	2	2
CO5	3	3	3	3	3	1	1	2	3	2

Legends: - 3-High; 2-Moderate/Medium; 1-Slight/Low; 0-None