

Course Description:

This comprehensive **Laravel + React Full-Stack Developer** course is designed to equip you with the skills and knowledge necessary to build dynamic, data-driven web applications from scratch. The course covers both front-end and back-end development, utilizing PHP as the core server-side scripting language and React JavaScript as front-end. By the end of this course, you will be proficient in creating robust and scalable web applications, handling databases, and implementing efficient user interfaces.

Course Objectives:

- Gain solid understanding of Web Development Essentials (HTML, CSS, JavaScript).
- Understand and use Git commands for versioning and collaboration.
- Learn and create simple web application using React.
- Explore modern PHP frameworks like Laravel to streamline development.
- Understand and implement database operations using SQL and MySQL.
- Develop full-stack web applications, integrating front-end and back-end technologies.
- Implement best practices in web development, including version control, testing, and deployment.

Software and technologies to be needed:

- Visual Studio Code - <https://code.visualstudio.com/download>
- Node.js - <https://nodejs.org/en/>
- XAMPP ver. 8.0 or above - <https://www.apachefriends.org/download.html>
- MYSQL Workbench - <https://dev.mysql.com/downloads/workbench/>
- Git CLI - <https://git-scm.com/downloads>
- Gitlab account - https://gitlab.com/users/sign_in/

Week	Training Content	Learning Objectives
1	<p>Jump Start: Build Portfolio</p> <ul style="list-style-type: none">- Create a simple single page web portfolio using HTML and CSS- Gitlab for version control- Files structure and linking of HTML and CSS	<p>HTML and CSS</p> <ul style="list-style-type: none">- Immersive, hands-on introduction on building a web page using HTML and CSS <p>Gitlab</p> <ul style="list-style-type: none">- Creating Gitlab account for pushing project and repository to Gitlab.

		<p>VS code</p> <ul style="list-style-type: none"> - Understanding basic folder structure, live server and some useful keys to speed up coding <p>Activity:</p> <ul style="list-style-type: none"> - Create a simple single web app portfolio - Push code to Gitlab
2 - 3	<p>Deep Dive:</p> <ul style="list-style-type: none"> - Git Fundamentals <ol style="list-style-type: none"> a. Version Control b. Repositories c. Branches d. Local vs Remote e. Cloning f. Committing g. Pushing h. Pulling - HTML Fundamentals <ol style="list-style-type: none"> a. HTML Basics b. Forms c. Tables - CSS Fundamentals <ol style="list-style-type: none"> a. Selectors (Element, Class, ID) b. Properties c. Values d. Declaration and Ruleset e. External and Inline styling f. Box Model g. Media Queries - Bootstrap 	<p>Git</p> <ul style="list-style-type: none"> - Understanding usage of git - Learning git most use commands <ol style="list-style-type: none"> a. git clone b. git commit -m "commit description" c. git push -u origin <branch name> d. git checkout -b (-b tag for creating new branch) <branch name > <p>HTML Fundamentals</p> <ul style="list-style-type: none"> - Introduction to web design and HTML – 60 - Understanding the document structure (DOCTYPE, html, head, body, footer) - Essential tags for headings, paragraphs, line breaks and images - Adding links to web pages <p>Forms and User Interaction</p> <ul style="list-style-type: none"> - Introduction to form and their purposes - Creating different input fields (text, email, checkbox, radio) - Using labels for better user experience - Submitting form data <p>Working with tables</p> <ul style="list-style-type: none"> - Understanding the purpose and structure of tables

		<ul style="list-style-type: none"> - Creating table elements (rows, cells, and headers) - Organizing data effectively in tubular format <p>Activities:</p> <ol style="list-style-type: none"> 1. Develop contact form with name, email, and message fields. 2. Develop Weekly Class Schedule using html table with days and times 3. Develop a single page website with header navigation, hero banner, about us page, service page, contact page, footer. Use bootstrap for enhancing UI.
4 - 5	<p>JavaScript Fundamentals</p> <ul style="list-style-type: none"> - Variable and Data Types - Operators - Control Flow (Conditional and Loops) - Functions - Scope - Objects and Prototypes - Arrays - Asynchronous JavaScript - Event handling 	<p>JavaScript Fundamentals</p> <ul style="list-style-type: none"> - Learn how JavaScript works on web pages and interactive web pages - Understand how to declare variables and work with different data types in JavaScript. - Write conditional statements and loops to control the flow of program execution. - Define and invoke functions to organize and reuse blocks of code. - Explore how scope affects variable accessibility in different parts of a JavaScript program. - Create and manipulate objects and understand how prototype-based inheritance works. - Use arrays to store and manage collections of data,

		<p>and apply common array methods.</p> <ul style="list-style-type: none"> - Understand and implement asynchronous code using callbacks, promises, and async/await. <p>Activity:</p> <ul style="list-style-type: none"> - Interactive Portfolio Pop Up
6-7	<p>React Fundamentals</p> <ul style="list-style-type: none"> - Creating a React App - JSX - Components and Templates - Props - State - Hooks (useState and useEffect) - Event Handling - Basic Routing - Fetching Data - Object Mapping and Displaying it on UI 	<p>React Fundamentals</p> <ul style="list-style-type: none"> - Initialize a new React App and understand its file structure. - Understand and use JSX to write HTML syntax inside JavaScript. - Create reusable function components and structure their UI using component-based design. - Learn to pass data between components using props, deconstruct props to be used on other components. - Learn and understand how to store and update dynamic values inside a component. - Manage component state using useState and useEffect. - Navigate between pages using react-router-dom and set up routes with URL paths. - Learn how to retrieve data from external APIs using fetch and display it within React component - Use “.map()” and render list of objects dynamically as part of UI. <p>Activity:</p> <ul style="list-style-type: none"> - Simple Lead Management App (Basic CRUD)
8-9	<p>Intro PHP</p> <ul style="list-style-type: none"> - Variable and Data Types - Operators 	<p>PHP Fundamentals</p> <ul style="list-style-type: none"> - Be able to declare variables and use appropriate PHP

	<ul style="list-style-type: none"> - Conditional Statement - Loops - Functions - Arrays - Superglobals - Forms and User Inputs - OOP 	<p>data types such as strings, integers, booleans, and arrays.</p> <ul style="list-style-type: none"> - Perform operations using arithmetic, comparison, logical, and assignment operators in PHP. - Control the flow of execution using if, else, and switch statements based on conditions. - Repeat blocks of code using for, while, and foreach loops to process data efficiently. - Define reusable functions, pass arguments, and return values to structure code logically. - Create and manipulate indexed and associative arrays to store and access multiple values. - Access and use PHP superglobals like \$_GET, \$_POST, and \$_SESSION to handle external data and sessions. - Learn about PHP Form Handling and validation. - Understand and apply OOP concepts such as classes, objects, properties, and methods to structure more modular PHP applications. <p>Activity:</p> <ul style="list-style-type: none"> - Simple Interest Calculator App
10 - 11	<p>MYSQL</p> <ul style="list-style-type: none"> - Workbench Installation - ERD - Database Creation - Data Types - Table and Column Creation - Reading Data 	<p>MYSQL Fundamentals</p> <ul style="list-style-type: none"> - Know and understand how to use Workbench. - Understand what MySQL is and how it manages relational data using structured queries.

	<ul style="list-style-type: none"> - Data Manipulation (Create, Update, Delete) - SQL Joins - Primary Keys and Foreign Keys - Aggregate Functions (SUM, COUNT, AVG) - Basic Constraints 	<ul style="list-style-type: none"> - Create databases and tables using SQL CREATE statements with appropriate schema design. - Choose and apply the correct MySQL data types (INT, VARCHAR, DATE, etc.) based on the kind of data being stored. - Add new records to tables using the INSERT INTO statement. - SELECT queries to fetch data from tables, including specific columns and rows. - Filter query results using the WHERE clause to match specific criteria. - Sort query results and limit the number of rows returned using ORDER BY and LIMIT. - Modify existing records in a table using the UPDATE statement with conditions. - Safely remove specific records from tables using the DELETE statement. - Retrieve related data from multiple tables using join operations like INNER JOIN, LEFT JOIN, and RIGHT JOIN. - Define primary and foreign keys to maintain data integrity and create table relationships. - Use functions like COUNT(), SUM(), and AVG() to perform calculations on query results. - Enforce data rules by applying constraints to table columns.
12 - 19	Laravel and React (Project Development)	Full Stack Fundamentals <ul style="list-style-type: none"> - Learn how to create a Laravel + React Starter Kit Template.

	<ul style="list-style-type: none"> - Initialize Laravel + React Starter Kit - Database Configuration - User authorization and authentication - Migrations and Seedings - Modal, View, Controller - Basic CRUD 	<ul style="list-style-type: none"> - Configure database set up - Create and run migrations to define database schema and use seeders to populate initial data. - Learn the concept of Model-View-Controller and the use of PHP frameworks like Laravel. - Implement user registration, login, and protected routes using Laravel Sanctum and React. - Build a full Create, Read, Update, Delete feature with Laravel API endpoints and React forms/components. <p>Activity:</p> <ul style="list-style-type: none"> - E-commerce Site. - Inventory Management App - Other Student Ideas for an App
20	Project Presentation	<ul style="list-style-type: none"> - Full functional App