PHP Lectureflow

Module 12)PHP - HTML, CSS, SQLite	10

- HTML-(1 session)Basic HTML like What is Tags attribute property Form Tag HTML-HTML5 tags
- CSS (1) • Selectors • Margin-Padding-display property • Create login-registration page
- Bootstrap (2) Grid • Rows-Cols • Forms-Cards
- SQL-(MySQLi)(4)
- SQL Queries
- Normalization
- Joins
- Indexer
- Practical Examples 1) MySQL data types 2)Create Multiple Tables and perform the queries using Select, Update, Delete, Insert Where, Like, Group By, Having, Limit, Offset, Sub Query and Or, Not, In 3) Function and procedure 4) Trigger 5) Views 6) Primary and auto increment

Module 13) PHP - Core PHP	10

- HTTP Protocol
- PHP syntax
- PHP variable
- Super Global Variables
- Practical Example: 1) Create Multiple Tables and perform the queries using Select,Update,Delete,Insert Where,Like,Group By,Having,Limit,Offset,Sub QueryAnd ,Or,Not,In
- Conditions Events and flows
- If condition
- If else if
- Practical Example : 1)Make Simple Calculator 2) Find the current day and if it is Sunday than print happy Sunday
- Practical Example: 1) Restaurant food Category Program using Switch case Display Which Category and dish user selected. 2) Using Ternary display age if it is greater than 18 3)Display Name Of the color which user selected
- Do While Loop
- For each Loop
- For loop
- Practical Example : 1) Create a script that displays 1-2-3-4-5-6-7-8-9-10 on one line. 2) Create a script using a for loop to add all the integers between 0 and 30 and display the total. 3)Write a PHP script using nested for loop that creates a chess board 4)All Patterns
- Array Function
- Date-time function
- Header function
- Include require

- Practical Example: 1) Perform all the function. 2)In user define function: 3)Create calculator 4)Find factorial using recursion 5)Reverse string without function 6) Download file on button click
- PHP array
- PHP expression
- PHP function
- PHP operations
- String Function
- PHP String
- Practical Example: 1)Display the value of the array 2) To Find the number of odd and even element in array. 3) Create associative array for user detail and display the user detail. 4)shifted all the zero in bottom up of the array

Module 14) PHP- Advance PHP	**25**

- Day 1 Introduction to Object-Oriented Programming (OOPs). Core OOPs principles: Encapsulation, Inheritance, Polymorphism, Abstraction. Advantages of OOPs. Practical: Write a program in PHP to create and use a simple class and object.
- Day 2: Classes, Objects, and Extends Theory: Detailed explanation of Classes and Objects. The concept of inheritance using the extends keyword. Practical: Implement a parent class Animal and a child class Dog using inheritance.
- Day 3: Overloading and Abstraction Interface Theory: Function Overloading in PHP. Abstract Classes and Interfaces in PHP. Practical: Create an abstract class Shape with an abstract method calculateArea(). Implement it in child classes like Circle and Rectangle.
- Day 4: Constructor, Destructor, and Magic Methods Theory: Constructors and destructors in PHP. Key magic methods like __construct, __destruct, __toString, __get, __set. Practical: Create a program to demonstrate the lifecycle of objects using constructors and destructors.
- Day 5: Scope Resolution and Traits Theory: Scope resolution operator (::) and its use in accessing static properties and methods. Traits and their advantages in PHP. Practical: Create a PHP program using a trait for common methods like logging.
- Day 6: Visibility and Final Keyword Theory: Understanding public, private, and protected visibility in PHP. The final keyword for classes and methods. Practical: Implement visibility modifiers to demonstrate access control.
- Day 7: Email Security Functions and File Handling Theory: Sending emails in PHP. Basics of file handling: reading, writing, appending. Practical: Create a program to send an email using mail() and save email logs in a file.
- Day 8: MVC Architecture Theory: Introduction to MVC (Model-View-Controller) architecture. Importance of separating concerns. Practical: Set up a basic MVC structure in PHP.
- Day 9: MySQL Connection and SQL Injection Theory: Connecting PHP with MySQL database. Understanding and preventing SQL injection. Practical: Write a script to fetch and display user data from a database.
- Day 10: Server-Side Validation and Registration Theory: Importance of server-side validation. Using regular expressions for input validation. Practical: Develop a registration form with validations.
- Day 11: Session and Cookies Theory: Basics of session management and cookies in PHP. Practical: Create a program to maintain user login sessions.

- Day 12: File Uploads and Downloads Theory: Handling file uploads securely in PHP. Practical: Build a script to upload and download files.
- Day 13: JavaScript and jQuery Basics Theory: Introduction to JavaScript and jQuery. Basics of jQuery selectors and events. Practical: Create a basic calculator using JavaScript.
- Day 14: Validation Using JavaScript Theory: Client-side form validation. Practical: Build a program with real-time form validation using JavaScript.
- Day 15: PHP and AJAX Theory: Basics of AJAX and its importance in PHP. Practical: Create a dynamic search feature using AJAX.
- Day 16-18: Project Implementation Focus: Start working on a mini-project combining OOPs, MVC, JavaScript, and AJAX concepts. Assign tasks such as: Create a registration module. Implement CRUD operations using MVC and AJAX. Add session and cookie handling.in the lab for the given topics. This plan assumes a 20-day training program. Day 1: Introduction to OOPs Concepts Theory: Introduction to Object-Oriented Programming (OOPs). Core OOPs principles: Encapsulation, Inheritance, Polymorphism, Abstra

Module 15) PHP - Web services, API, Extensions - Industry	**4**

- Payment Gateway Integration
- -> Create API With Header -> API with Image Uploading
- Practical Example: Payment Gateway Implement on MVC Project
- -> SOAP and REST AP, Create API for insert, update and delete
- Product Catalog
- Shopping Cart
- Web Services
- Practical Example: Create Web Services for MVC Project
- Integration of API in Project
- RESTful principles, such as resource identification, statelessness, and uniform interfaces, is important for designing and consuming RESTful APIs.
- OpenWeatherMap API: This API provides weather data for various locations worldwide. You can retrieve current weather conditions, forecasts, and historical weather data.
- Google Maps Geocoding API: This API allows you to convert addresses into geographic coordinates (latitude and longitude) and vice versa. You can use it to retrieve location data, calculate distances between points, and display maps.
- GitHub API: GitHub provides an API that enables you to interact with repositories, issues, pull requests, and more. You can perform actions like retrieving repository information, creating issues, and accessing user data.
- Twitter API: Twitter offers an API that allows you to integrate Twitter functionality into your applications. You can fetch tweets, post tweets, retrieve user information, and perform searches.
- REST Countries API: This API provides information about countries, including details like population, languages spoken, currencies, and more. You can retrieve country-specific data and use it for various applications.
- SendGrid provides an API for sending transactional and marketing emails. You can integrate it into your applications to send emails, manage templates, and track delivery statistics.
- Social authentication (For eg; Login with Google, Login with Facebook...etc)

- Email sending APIs (For eg; Mailchimp, Mailgun...etc)
- SMS sending APIs (For eg; Twilio)
- Normal payments (For eg; Paypal, Stripe)
- - Google Maps API

Module - 17) PHP - Introduction of Laravel PHP Framework - Industry	**7**

- Installing Laravel
- Artisan CLI (command-line interface)
- Laravel Directory Structure
- Configuring a new Laravel project
- Artisan command to generate boilerplate code for a controller
- Basics of laravel:----> Routing
- Controller
- Middleware
- Request
- Response
- HTML Template to Laravel Blade Template
- HTML Template to Laravel Blade Template:-----> Template inheritance
- Master layout
- Extending the master layout
- Displaying variables
- Blade conditional statements
- Blade Loops
- Executing PHP functions in blade
- Displaying Your Views
- Creating and using a basic views
- Loading a view into another view/nested views
- Adding assets
- Integrating with Bootstrap
- Laravel Component
- LaravelHelper and Custom Helper

Module - 19) PHP - Laravel – Database , Laravel Query Builder -	**5**

- Http Client and Reques
- Http Request Method
- Laravel Flash session
- Laravel Database & SQL Queries
- Select, Where Clauses,
- Aggregate Function
- Join , Union

- Ordering, Grouping,
- Limit, & Offset etc.
- Laravel Query Builder Insert, Update and Delete Statements

Module - 20) PHP - Laravel - Migrations , Eloquent ORM	8

- Migration Introduction, Requirements for running migrations, Artisan migration command, Migration structure
- How to create a table using a migration, Laravel migration rollback, • Database Seeding, Migrations for our project database,
- • Laravel Eloquent ORM, Laravel Eloquent Model, Generating Model Class, Eloquent ORM naming conventions.
- Table names, primary keys, and timestamps, Database Operations with Eloquent, Laravel Eloquent Crud (Create, delete, Update, Retrieve)
- Practice : Eloquent ORM Basics Theory: Eloquent ORM naming conventions. Table names, primary keys, and timestamps. Practical: Creating and interacting with Eloquent models. Performing basic CRUD operations: INSERT, READ, UPDATE, DELETE.
- Laravel File Uploading, Laravel Pagination, Laravel Eloquent Relationships, One-to-one , one-to-many, many-to-many ,
- Has Many Through, Polymorphic Relations ,Many-to-many Polymorphic, Practice : Relationships and Models in Controllers Theory: Types of relationships (one-to-one, oneto-many, many-to-many). Using models in controllers. Practical: Implementing Eloquent relationships. Fetching and displaying related data in views.
- Accessors & Mutators
- Laravel Route Model Binding
- Implicit Binding,
- Explicit Binding

Module - 21) PHP – Laravel Advance Laravel, APIs, Project	8

- • Laravel Localization • Laravel Validation • Laravel Mail
- Laravel Queues and Jobs ?Practice :Queue and Task Scheduling Theory: Understanding the need for queues. Task scheduling and its use cases. Practical: Implementing a queue for delayed tasks. Scheduling tasks using the scheduler.
- Laravel Events & Listeners ?Practice :Event and Mail Theory: Introduction to events and listeners. Overview of the Mail component for sending emails. Practical: Creating and triggering events. Sending emails using prebuilt templates.
- Laravel - Artisan Console
- Laravel REST API ?Practice :Introduction and Basics Theory: Introduction to the framework and its features. Basics of MVC architecture. Overview of routing and controllers. Practical: Setting up the environment. Creating a basic controller. Creating a route using a closure. ?Practice :RESTful Controllers and Route Groups Theory: Making controllers RESTful. Understanding route groups and middleware. Practical: Creating RESTful controllers. Using route groups to organize routes. ?Practice :Building

- Laravel Sanctum API
- Laravel Ajax CRUD ?Practice : Using Ajax and jQuery Theory: Introduction to AJAX and jQuery. How to get data from another page dynamically. Practical: Setting up a controller to return JSON data. Creating an AJAX-based comment system. • Composer Packages Theory: Introduction to Composer and package management. Exploring prebuilt packages. Practical: Installing and configuring a shopping cart package. Integrating the package into a project.
- • Laravel E-Commerce Project