

# Laravel

This 5-day accelerated Laravel 12 program provides broad coverage of modern Laravel development, combining essential PHP foundations, core framework features, integration topics, and selected hands-on implementation to help participants build practical working knowledge in a short time.

## Overview

This 5-day accelerated Laravel 12 program is designed for PHP developers who want to move from basic website development into modern application development using Laravel. The course begins with a focused PHP bridge covering the language features commonly used in Laravel, then progresses into routing, controllers, Blade, database integration, APIs, security, deployment, integration, and production practices. With practical demonstrations and guided implementation throughout the program, participants gain broad working knowledge of Laravel 12 and understand how it is used to build structured web applications, APIs, and business systems in real-world environments.

## Introduction

Laravel is a powerful MVC PHP framework created by Taylor Otwell and designed to give developers a clean, elegant, and productive toolkit for building full-featured web applications. Laravel 12 continues this modern development approach with strong support for structured application design, routing, controllers, Blade templating, database access, APIs, authentication, middleware, queues, deployment workflows, and system integration. This program is especially valuable for developers who already use basic PHP in their daily work but want to understand the modern PHP concepts and framework patterns that Laravel uses to deliver scalable, maintainable, and production-ready applications.

## Topics Covered

- Modern PHP Foundations for Laravel
- Laravel Overview
- Installation and Environment Setup
- Application Structure
- Configuration
- Server Setup for Laravel
- Routing
- Controllers

- Views and Blade
- Working with Database (MariaDB Focus)
- Database Migration and Conversion
- Forms and Validation
- Request and Response
- REST API Development
- Authentication (API Focus)
- Mobile App Integration
- Middleware
- Error Handling
- Security
- Sessions and Cookies
- File Upload and Storage
- Events and Jobs
- Email and Notifications
- Performance Optimization
- GitLab and Version Control
- Deployment Workflow
- Production Setup
- Troubleshooting (Real-World)
- Integrate with Other Frameworks

## Prerequisites

Participants should have prior knowledge of basic PHP, HTML, CSS, SQL, and general website development concepts. Prior Laravel experience is not required.

## Audience:

PHP developers, web developers, and technical learners who already use basic PHP and want to build structured modern web applications, APIs, and business systems using Laravel.

**Duration: 5 Days**

# Course Outline

## Day 1 - Foundations, Environment Setup, and Modern PHP for Laravel

1. Modern PHP Foundations for Laravel
  - 1.1. Modern PHP overview
  - 1.2. Variables, arrays, and associative arrays
  - 1.3. Functions
  - 1.4. Passing functions as arguments
  - 1.5. Anonymous functions and closures
  - 1.6. OOP fundamentals
  - 1.7. Classes and objects
  - 1.8. Inheritance
  - 1.9. Interfaces and abstract classes
  - 1.10. Traits
  - 1.11. Namespaces
  - 1.12. Type declarations and return types
  - 1.13. Composer and autoloading basics
  - 1.14. Practical PHP patterns commonly used in Laravel
  
2. Laravel Overview
  - 2.1. What is Laravel (Modern Laravel 12 perspective)
  - 2.2. MVC Architecture (for beginners)
  - 2.3. Laravel vs Native PHP (client context)
  - 2.4. How Laravel fits into enterprise and API ecosystems
  
3. Installation and Environment Setup
  - 3.1. Installing PHP, Composer, Node
  - 3.2. Laravel 12 installation
  - 3.3. Project structure walkthrough
  - 3.4. Local development setup (XAMPP / Docker optional)
  
4. Application Structure
  - 4.1. Deep dive:
  - 4.2. App, Routes, Config, Database, Resources
  - 4.3. Understanding request lifecycle

## Day 2 - Core Development and Server Foundations

5. Configuration
  - 5.1. .env deep dive
  - 5.2. Environment-based configuration
  - 5.3. Debug mode
  - 5.4. Production mode
  - 5.5. Config caching
  
6. Server Setup for Laravel
  - 6.1. Basic Linux server setup (Ubuntu)
  - 6.2. Installing: PHP, Nginx/Apache, Composer
  - 6.3. Folder permissions
  - 6.4. Running Laravel on server
  
7. Routing
  - 7.1. Basic routing
  - 7.2. Route parameters
  - 7.3. API routing vs Web routing
  
8. Controllers
  - 8.1. Creating controllers
  - 8.2. Resource controllers
  - 8.3. Dependency injection
  
9. Views and Blade
  - 9.1. Blade basics
  - 9.2. Layouts
  - 9.3. Passing data

## Day 3 - CRUD, Database, and API Foundations

10. Working with Database (MariaDB Focus)
  - 10.1. Connecting Laravel to MariaDB
  - 10.2. Eloquent ORM basics
  - 10.3. Query Builder
  
11. Database Migration and Conversion
  - 11.1. Migrating existing database to Laravel
  - 11.2. Reverse engineering existing DB
  - 11.3. Creating migrations from existing schema
  - 11.4. Handling legacy database structures

- 12. Forms and Validation
  - 12.1. Form handling
  - 12.2. Validation rules
  - 12.3. Error handling
  
- 13. Request and Response
  - 13.1. Handling input
  - 13.2. JSON responses
  - 13.3. API Validation
  
- 14. REST API Development
  - 14.1. REST fundamentals (for beginners)
  - 14.2. Building API endpoints
  - 14.3. API versioning

## **Day 4 - API Development, Security, and Integration**

- 15. Authentication (API Focus)
  - 15.1. Token-based authentication (Laravel Sanctum)
  - 15.2. Login API
  - 15.3. Securing endpoints
  
- 16. Mobile App Integration
  - 16.1. How mobile apps connect to Laravel APIs
  - 16.2. JSON data flow
  - 16.3. CORS handling
  - 16.4. Testing APIs using Postman
  
- 17. Middleware
  - 17.1. API middleware
  - 17.2. Authentication middleware
  
- 18. Error Handling
  - 18.1. API error responses
  - 18.2. Logging
  
- 19. Security
  - 19.1. CSRF protection
  - 19.2. Authentication vs Authorization
  - 19.3. API security

- 20. Sessions and Cookies
  - 20.1. Session handling
  - 20.2. Stateless APIs vs session apps
- 21. File Upload and Storage
  - 21.1. Upload files
  - 21.2. Cloud storage concepts

## Day 5 - Advanced Features, Deployment, and Production Practices

- 22. Events and Jobs
  - 22.1. Event-driven architecture
  - 22.2. Background jobs (intro)
- 23. Email and Notifications
  - 23.1. Sending emails
  - 23.2. Queue basics
  - 23.3. Whatsapp
- 24. Performance Optimization
  - 24.1. Caching
  - 24.2. Query optimization
  - 24.3. Config caching
  - 24.4. Route caching
- 25. GitLab and Version Control
  - 25.1. Git basics (for beginners)
  - 25.2. GitLab workflow: Clone, commit, push
  - 25.3. Branching strategy
  - 25.4. Managing Laravel project in GitLab
- 26. Deployment Workflow
  - 26.1. Local to Server deployment
  - 26.2. Environment differences
  - 26.3. Build steps: composer install npm build and config cache
- 27. Production Setup
  - 27.1. .env for production
  - 27.2. Security best practices
  - 27.3. File permissions

- 28. Troubleshooting (Real-World)
  - 28.1. Why it works locally but fails in server
  - 28.2. Case sensitivity
  - 28.3. Environment configs
  - 28.4. Missing extensions
  - 28.5. Cache issues
  - 28.6. Debugging strategies
  
- 29. Integrate with Other Frameworks
  - 29.1. Monolithic vs Microservices
  - 29.2. Laravel as API provider / consumer
  - 29.3. Django = AI/ML service and Laravel = Business API
  - 29.4. API-to-API Communication
  - 29.5. Consuming Django APIs in Laravel
  - 29.6. Authentication Between Systems
  
- 30. Hands-On Project
  - 30.1. Build CRUD system
  - 30.2. Connect to MariaDB
  - 30.3. Create APIs
  - 30.4. Integrate with Postman (simulate mobile)
  - 30.5. Push to GitLab
  - 30.6. Deploy to server