Laravel 12

This 3-day accelerated Laravel 12 program delivers focused coverage of the essential Laravel development workflow, combining strong PHP foundations, core framework concepts, database integration, APIs, security, and deployment practices in a compact format.

Overview

This 3-day accelerated Laravel 12 program is designed for PHP developers who want broad and practical exposure to modern Laravel development in a shorter timeframe. The course preserves the essential foundation topics needed to understand Laravel properly, including modern PHP concepts, framework structure, configuration, routing, controllers, Blade, and database integration, while streamlining advanced areas such as APIs, security, deployment, and integration for efficient delivery. By the end of the program, participants will gain solid working knowledge of Laravel 12 and understand how its major building blocks fit together in real-world web application development.

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 routing, controllers, Blade templating, database access, APIs, authentication, middleware, deployment workflows, and application integration. This 3-day program is structured to help developers who already know basic PHP and web development quickly understand how Laravel applications are organized and how to use the framework effectively in practical project scenarios.

Topics Covered

- Modern PHP Foundations for Laravel
- Laravel Overview
- Installation and Environment Setup
- Application Structure and Configuration
- Routing and Controllers
- Views, Blade, Forms, and Validation
- Working with Database and Migrations
- Request, Response, and REST APIs
- Authentication, Middleware, and Mobile Integration

- Security, Sessions, and File Handling
- Events, Jobs, Notifications, and Performance
- Version Control, Deployment, and Troubleshooting
- Integration with Other Frameworks
- Hands-On Project

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 and APIs using Laravel.

Duration: 3 Days

Course Outline

Day 1 - Foundations, Environment Setup, and Core Laravel Concepts

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 and Configuration
   4.1. App, Routes, Config, Database, Resources
   4.2. Understanding request lifecycle
   4.3. .env deep dive
   4.4. Environment-based configuration
   4.5. Debug mode and production mode
   4.6. Config caching

5. Routing and Controllers
   5.1. Basic routing
   5.2. Route parameters
   5.3. API routing vs Web routing
   5.4. Creating controllers
   5.5. Resource controllers
   5.6. Dependency injection

6. Views, Blade, Forms, and Validation
   6.1. Blade basics
   6.2. Layouts
   6.3. Passing data
   6.4. Form handling
   6.5. Validation rules
   6.6. Error handling

Day 2 - Database, CRUD Workflow, and API Development

7. Working with Database and Migrations
   7.1. Connecting Laravel to MariaDB
   7.2. Eloquent ORM basics
   7.3. Query Builder
   7.4. Migrating existing database to Laravel
   7.5. Reverse engineering existing DB
   7.6. Creating migrations from existing schema
   7.7. Handling legacy database structures

8. Request, Response, and REST APIs
    - 8.1. Handling input
    - 8.2. JSON responses
    - 8.3. API validation
    - 8.4. REST fundamentals (for beginners)
    - 8.5. Building API endpoints
    - 8.6. API versioning

9. Authentication, Middleware, and Mobile Integration
    - 9.1. Token-based authentication (Laravel Sanctum)
    - 9.2. Login API
    - 9.3. Securing endpoints
    - 9.4. API middleware
    - 9.5. Authentication middleware
    - 9.6. How mobile apps connect to Laravel APIs
    - 9.7. JSON data flow
    - 9.8. CORS handling
    - 9.9. Testing APIs using Postman

Day 3 - Security, Deployment, Integration, and Production Readiness

10. Security, Sessions, and File Handling
    - 10.1. CSRF protection
    - 10.2. Authentication vs Authorization
    - 10.3. API security
    - 10.4. Session handling
    - 10.5. Stateless APIs vs session apps
    - 10.6. Upload files
    - 10.7. Cloud storage concepts

11. Events, Jobs, Notifications, and Performance
    - 11.1. Event-driven architecture
    - 11.2. Background jobs (intro)
    - 11.3. Sending emails
    - 11.4. Queue basics
    - 11.5. Whatsapp
    - 11.6. Caching
    - 11.7. Query optimization
    - 11.8. Config caching
    - 11.9. Route caching

12. Version Control, Deployment, and Troubleshooting
    - 12.1. Git basics (for beginners)
    - 12.2. GitLab workflow: Clone, commit, push
    - 12.3. Branching strategy
    - 12.4. Managing Laravel project in GitLab
    - 12.5. Local to server deployment
    - 12.6. Environment differences
    - 12.7. Build steps: composer install npm build and config cache
    - 12.8. .env for production
    - 12.9. Security best practices
    - 12.10. File permissions
    - 12.11. Why it works locally but fails in server
    - 12.12. Case sensitivity
    - 12.13. Environment configs
    - 12.14. Missing extensions
    - 12.15. Cache issues
    - 12.16. Debugging strategies

13. Integration with Other Frameworks
    - 13.1. Monolithic vs Microservices
    - 13.2. Laravel as API provider / consumer
    - 13.3. Django = AI/ML service and Laravel = Business API
    - 13.4. API-to-API Communication
    - 13.5. Consuming Django APIs in Laravel
    - 13.6. Authentication Between Systems

14. Hands-On Project
    - 14.1. Build CRUD system
    - 14.2. Connect to MariaDB
    - 14.3. Create APIs
    - 14.4. Integrate with Postman (simulate mobile)
    - 14.5. Push to GitLab
    - 14.6. Deploy to server