

# PHP and Laravel Developer Internship Roadmap

---

## Introduction

This roadmap is designed to guide new interns through a structured learning path to become proficient PHP and Laravel developers. The program is divided into three progressive levels—Basic, Medium, and High—each focusing on essential concepts, core framework features, and advanced integration techniques.

The goal is to move from foundational knowledge to building complete, modern web applications.

## Resources

The following resources are highly recommended for covering the topics outlined in this roadmap.

Resource	Description	Link
<a href="#">W3Schools PHP Tutorial</a>	Excellent resource for foundational PHP concepts.	[1]
<a href="#">Official Laravel Documentation</a>	The definitive guide for all Laravel features and best practices.	[2]
<a href="#">Yahoo Baba - Laravel Playlist</a>	Comprehensive video tutorial series covering Laravel topics.	[3]
<a href="#">Yahoo Baba - PHP OOP Playlist</a>	Video tutorials dedicated to Object-Oriented Programming in PHP.	[4]
<a href="#">Yahoo Baba - PHP Tutorial</a>	General PHP video tutorials.	[5]

# Level 1: Basic - The Foundation

This level focuses on establishing a strong foundation in core PHP and understanding the basic structure of a Laravel application.

Topic Category	Topics to Cover	Resource Links
Core PHP	Variables, Data Types, Operators, Conditions ( <code>if / else / switch</code> ), Loops ( <code>for</code> , <code>while</code> , <code>foreach</code> ), Arrays (indexed, associative, multidimensional), Functions (with return values).	[1], [5]
PHP OOP	Classes & Objects, Constructors, Properties & Methods, Access Modifiers ( <code>public</code> , <code>protected</code> , <code>private</code> ), Inheritance.	[4]
Laravel Fundamentals	Installing Laravel, Understanding the MVC Architecture, Folders and Files Structure, Basic Routing, Blades Structure (Views).	[2], [3]

## Level 1 Assignment: Simple Static Website

**Goal:** Build a simple, multi-page static website using Laravel to demonstrate understanding of installation, routing, and views.

### Requirements:

- 1. Setup:** Successfully install Laravel and configure the development environment.
- 2. Routing:** Define at least three routes: `/`, `/about`, and `/contact`.
- 3. Views (Blades):** Create corresponding Blade templates for each route.
- 4. Layout:** Use a master layout file (`app.blade.php`) to ensure consistent navigation and footer across all pages.
- 5. PHP Logic:** Use a simple PHP function within a Blade file to display the current year in the footer.

## Level 2: Medium - Core Application Development

This level transitions from foundational concepts to building functional, data-driven applications using Laravel's core features.

Topic Category	Topics to Cover	Resource Links
Data Management	Migrations (creating and modifying tables), Seeders (populating the database), Models (connecting to tables).	[2], [3]
Application Flow	Controllers (handling requests and logic), Form Validation (server-side), Eloquent ORM (Basic CRUD operations: Create, Read, Update, Delete).	[2], [3]
Web State & Security	Sessions and Cookies (how Laravel manages state), Middleware (basic usage, e.g., authentication check).	[2], [3]

### Level 2 Assignment: Task Manager CRUD Application

**Goal:** Develop a complete **Task Manager** application that allows users to manage a list of tasks.

#### Requirements:

1. **Database:** Create a `tasks` table using a migration (fields: `title`, `description`, `is_completed`).
2. **CRUD:** Implement all four CRUD operations:
  - **Create:** A form to add new tasks with server-side validation.
  - **Read:** A page to list all tasks, showing their status.
  - **Update:** A form to edit an existing task's title and description.
  - **Delete:** A button to remove a task.
3. **Middleware:** Apply a simple custom middleware to one of the routes (e.g., to log the request time).

## Level 3: High - Advanced Features and Integration

This level focuses on complex data relationships, file handling, external communication, and preparing for modern frontend integration.

Topic Category	Topics to Cover	Resource Links
Advanced Eloquent	Eloquent Relationships: One-to-One, One-to-Many, Many-to-Many, Pivot Tables.	[2], [3]
External Communication	Sending Emails (using Mailables and a service like Mailtrap), API Basics (creating simple RESTful endpoints for data).	[2], [3]
Modern Web	File and Image Uploads (storing files on disk/S3), JavaScript Fetch API Method with Laravel (consuming a Laravel API endpoint from a Blade view using JavaScript).	[2], [3], [6]

### Level 3 Assignment: Product Catalog with API

**Goal:** Enhance the Task Manager (or create a new **Product Catalog**) application to include advanced features and API integration.

#### Requirements:

- 1. Relationships:** Implement a **Many-to-Many** relationship (e.g., between `Products` and `Categories` or `Tasks` and `Tags` ).
- 2. File Upload:** Allow the user to upload an image for a product/task and store it on the local disk.
- 3. Email:** Configure and send a welcome email to a user (or a notification email when a task is completed).
- 4. API:** Create a simple API route (`/api/products`) that returns a JSON list of all products.
- 5. Frontend Integration:** On the main product listing page, use **JavaScript's Fetch API** to asynchronously load and display the product data from the API endpoint you created, demonstrating modern data fetching.

## References

---

- [1] W3Schools PHP Tutorial. <https://www.w3schools.com/php/> [2] Laravel 12.x Documentation. <https://laravel.com/docs/12.x/installation> [3] Yahoo Baba - Laravel 11 Tutorial Course in Hindi / Urdu. [https://www.youtube.com/playlist?list=PL0b60zIxLPbz7JK\\_YYrRJ1Kx1GG4dizHJ](https://www.youtube.com/playlist?list=PL0b60zIxLPbz7JK_YYrRJ1Kx1GG4dizHJ) [4] Yahoo Baba - PHP OOP Tutorial in Hindi / Urdu. <https://www.youtube.com/playlist?list=PL0b60zIxLPbzGtrDFaF6uoC33cPNHDmeV> [5] Yahoo Baba - PHP Tutorial in Hindi / Urdu. <https://www.youtube.com/playlist?list=PL0b60zIxLPbzvz4j1N4J8zCY8mu3l29MG> [6] Laravel 12.x Frontend Documentation (Fetch API). <https://laravel.com/docs/12.x/frontend#using-fetch-api>