

NSP TRAINING STRUCTURE FOR LARAVEL BEGINNERS

Purpose of this Structured Roadmap:

This roadmap is designed to guide the individual with minimal tech experience towards becoming a proficient Laravel developer. It outlines progressive phases and practical tasks that cover the essential skills and knowledge required to work effectively with Laravel. By following this roadmap, the individual can develop the abilities needed to build complex web applications using the Laravel framework.

Goal at the End of the Studies:

At the end of this journey, the individual should be equipped with the skills to create and maintain Laravel applications. You will have practical experience in designing databases, implementing user authentication, working with APIs, and adhering to best coding practices

Phase 1: Introduction to Laravel

Reference Materials:

- [Laravel Documentation](#)
- [Laravel Tutorials](#)

Task 1: Setting Up Laravel

1. Learn how to install and use Composer, a PHP dependency management tool, to install and manage Laravel and its packages. Click on the link to access the official website for composer <https://getcomposer.org/>
2. Set up a local development environment.
3. Create a basic Laravel project. <https://laravel.com/docs/10.x#your-first-laravel-project>

Task 2: Routing and Controller

1. **Routing:** Learn about routing in Laravel, which maps URLs to controllers. Define routes for different endpoints of your application. <https://laravel.com/docs/10.x/routing>
2. **Controllers:** Controllers handle the logic of your application. Learn how to create controllers, define methods, and manage data flow. <https://laravel.com/docs/10.x/controllers>
3. Build and display a simple web page using a controller and route.

Task 3: Eloquent ORM

1. **Eloquent ORM:** Learn about Eloquent, Laravel's Object-Relational Mapping (ORM) system. Learn how to define models, set up relationships, and perform database operations. <https://laravel.com/docs/10.x/eloquent>
2. Practice CRUD operations using Eloquent.

Task 4: Database Migrations and Seeders

1. Understand database migrations to manage your database schema changes. Seeders help populate your database with sample data for testing. <https://laravel.com/docs/10.x/database>

Task 5: Views and Blade Templating

1. Explore Blade, Laravel's templating engine. Create dynamic views using Blade's syntax, which includes directives and control structures.
2. <https://laravel.com/docs/10.x/views>
3. <https://laravel.com/docs/10.x/blade>
4. Create a Blade template and integrate it with your routes and controllers.

Task 6: Build a project

1. Create a web-based To-Do app using Laravel that allows users to manage their tasks.
 - a. Add a task
 - b. Display all tasks
 - c. Be able to mark task as completed

Do well to implement all that you have learnt so far

Phase 2: Building Advanced Applications

Task 6: Middleware and Authentication

1. **Middleware:** Learn how middleware works in Laravel. It filters HTTP requests before they reach the application's routes or controllers. <https://laravel.com/docs/10.x/middleware>
2. **Authentication and Authorization:** Study Laravel's built-in authentication and authorization features. Learn how to secure routes and control user access.

Task 7: Form Validation and Request Handling

1. Understand how to validate user input using Laravel's built-in tools, which help maintain data integrity.
2. Implement form validation for your application's forms.

Task 8: Error Handling and Logging

1. Explore error handling and logging mechanisms to ensure your application's stability and traceability.

Task 9: RESTful APIs

1. Learn how to create RESTful APIs using Laravel. This involves setting up API routes, controllers, and responses. Create an API that has endpoints to perform the following:
2. Create a new user account in the database
3. Update a user account data
4. Delete a user account

5. Authenticate the requests using an api-key which is assigned to users when their account is created.