

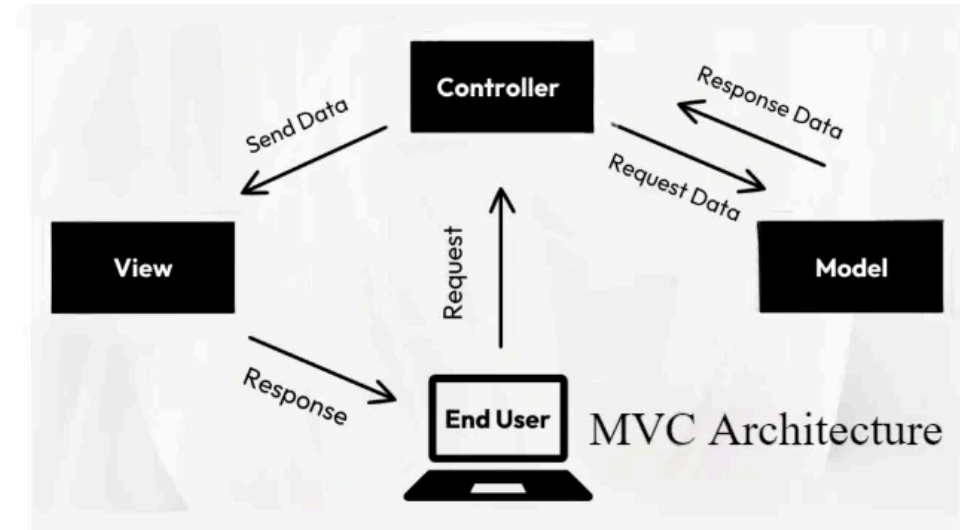
Laravel 12 for Beginners

Your Journey into Modern Web Development

What is Laravel?

Laravel is an open-source PHP framework that follows the Model-View-Controller (MVC) architecture. It provides tools and features to simplify web application development.

- ✓ Elegant syntax and powerful tools
- ✓ Built-in authentication system
- ✓ Robust routing capabilities
- ✓ Session and database management
- ✓ Designed to make development enjoyable



Why Laravel is Popular?

Easy to Learn

Clear documentation and intuitive syntax make Laravel beginner-friendly and easy to adopt.

Rich Features

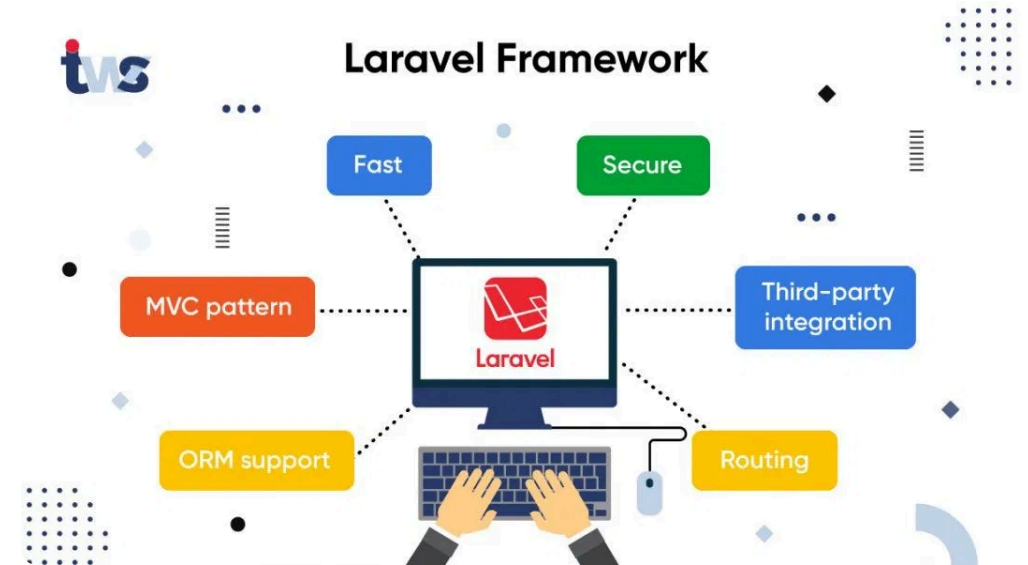
Built-in authentication, routing, and caching reduce development time and effort.

Community Support

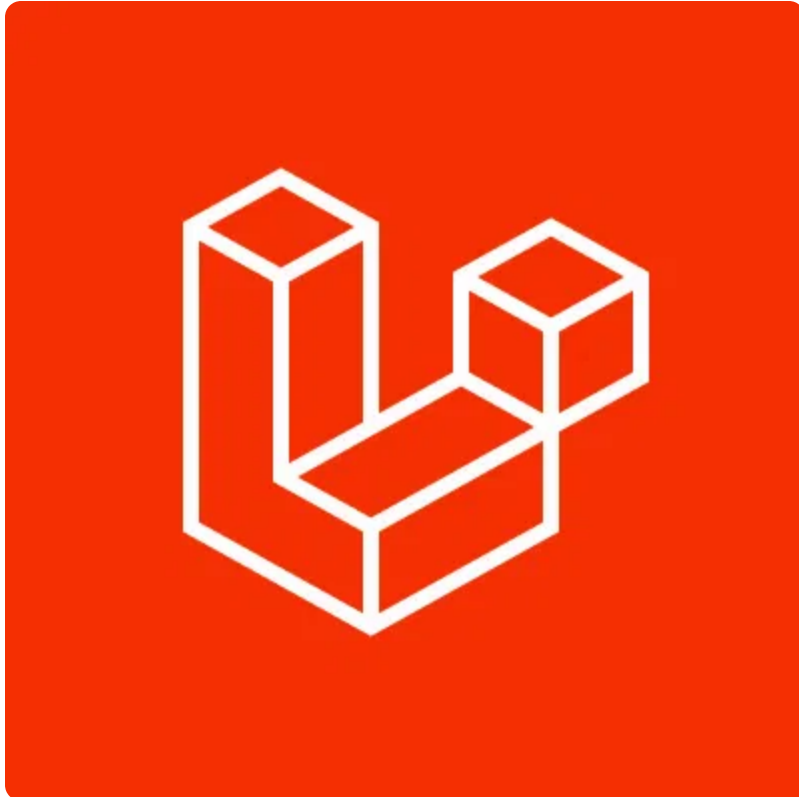
Strong community of developers continuously contribute to its growth and improvement.

Security

Built-in security features protect applications from common web vulnerabilities.



What's New in Laravel 12?



Improved Performance & Speed

Laravel 12 introduces significant performance optimizations, making applications run faster with lower memory usage.

Better Frontend Framework Support

Enhanced integration with popular frontend frameworks like React and Vue, making full-stack development smoother.

New Development Tools

New tools and utilities that streamline the development process and increase productivity.

More Efficient Database Handling

Improved database query builder and Eloquent ORM with optimized database interactions.

Installation Guide

1 Install Composer

Laravel requires Composer, a PHP dependency manager. Download and install it from getcomposer.org.

2 Create a New Laravel Project

Run the following command to create a new Laravel project:

```
composer create-project --prefer-dist  
laravel/laravel myproject
```

3 Start the Development Server

Navigate to your project folder and start the Laravel development server:

```
cd myproject  
php artisan serve
```

Your Laravel application will be available at <http://127.0.0.1:8000>



Directory Structure



app/

Contains the core application code including models, controllers, and service providers.



config/

Stores all configuration files for your application.



database/

Includes migration and seed files for database setup.



public/

Holds publicly accessible files like index.php, images, CSS, and JavaScript.



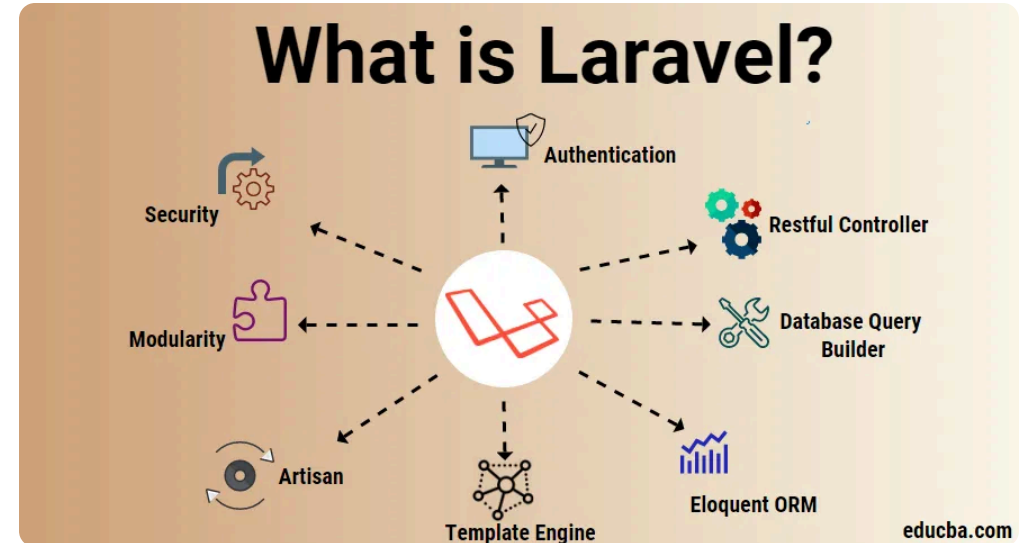
routes/

Defines the application's routes and endpoints.



resources/

Contains views, CSS, JavaScript files, and language files.



Routing in Laravel

Routing in Laravel directs web requests to the correct part of your application, defining how your application responds to specific URLs.

Routes are defined in `routes/web.php` for web applications.

Basic Route Example:

```
Route::get('/', function () {  
    return view('welcome');  
});
```

This route maps the URL `'/'` (homepage) to the `'welcome'` view. When a user visits the home page, Laravel will display the `welcome.blade.php` file from the `resources/views` directory.

Route with Parameters:

```
Route::get('/user/{id}', function ($id) {  
    return 'User ID: ' . $id;  
});
```



Blade Templating Engine

Blade is Laravel's powerful templating engine that makes writing HTML easier by using simple syntax instead of complex PHP code.

View Example (Blade Template):

```
<!-- resources/views/home.blade.php -->
<!DOCTYPE html>
<html>
<head>
  <title>My Laravel App</title>
</head>
<body>
  <h1>Welcome, {{ $name }}!</h1>
</body>
</html>
```

Controller Example:

```
return view('home', ['name' => 'John']);
```

In this example, the controller passes the variable **'name'** with the value **'John'** to the view. The Blade template then displays "Welcome, John!" by using the **{{ \$name }}** syntax.

Laravel Blade Template Engine



The diagram illustrates the Blade template engine syntax. It features a central code block with three labels and arrows pointing to specific parts of the code:

- Section Keyword**: Points to the `@section('body')` line.
- Extends Keyword**: Points to the `@extends('layouts.layout_1')` line.
- Foreach Loop**: Points to the `@foreach($texts as $text)` line.

```
@extends('layouts.layout_1')
@section('body')
  @foreach($texts as $text)
    {{$text}}
  @endforeach
@endsection
```

#13

Database Migrations & Eloquent ORM

Database Migrations

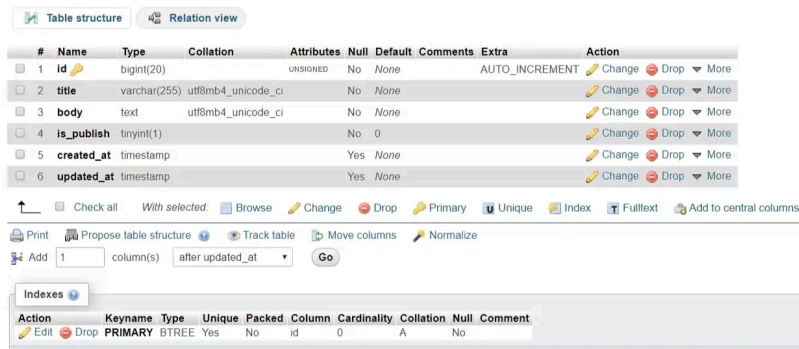
Migrations allow you to manage database tables using PHP instead of writing raw SQL.

Create a migration:

```
php artisan make:migration create_users_table
```

Define table structure:

```
Schema::create('users', function ($table) {  
    $table->id();  
    $table->string('name');  
    $table->string('email')->unique();  
    $table->timestamps();  
});
```



The screenshot shows a database management interface with a 'Table structure' tab selected. It displays the schema for a table named 'users'. The columns are: 'id' (bigint(20), UNSIGNED, No, None, AUTO_INCREMENT), 'title' (varchar(255), utf8mb4_unicode_ci, No, None), 'body' (text, utf8mb4_unicode_ci, No, None), 'is_publish' (tinyint(1), No, 0), 'created_at' (timestamp, Yes, None), and 'updated_at' (timestamp, Yes, None). Below the table structure, there are options to check all, browse, change, drop, primary, unique, index, fulltext, and add to central columns. There is also a 'Print' button and a 'Propose table structure' button. At the bottom, there is an 'Indexes' section showing a primary index on the 'id' column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	title	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	body	text	utf8mb4_unicode_ci		No	None			Change Drop More
4	is_publish	tinyint(1)			No	0			Change Drop More
5	created_at	timestamp			Yes	None			Change Drop More
6	updated_at	timestamp			Yes	None			Change Drop More

Eloquent ORM

Eloquent is Laravel's built-in ORM (Object-Relational Mapping) that simplifies database interactions.

Using Eloquent to save data:

```
$user = new User();  
$user->name = 'John Doe';  
$user->email = 'john@example.com';
```

Retrieving data:

```
$users = User::all();  
$user = User::find(1);
```



Conclusion & Next Steps

Laravel 12 is a powerful, efficient, and beginner-friendly PHP framework for building modern web applications. Its MVC architecture, elegant syntax, and rich features make development enjoyable and productive.

Key Takeaways

- ✓ MVC architecture provides clean separation of concerns
- ✓ Rich ecosystem with built-in tools and packages
- ✓ Strong community support and extensive documentation

Next Steps

- Practice installation and create your first Laravel project
- Explore the official Laravel documentation at laravel.com
- Build small projects to reinforce your learning
- Join Laravel communities and forums for support

