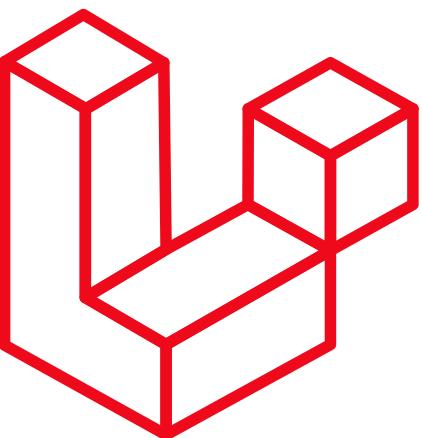


MY LARAVEL JOURNEY

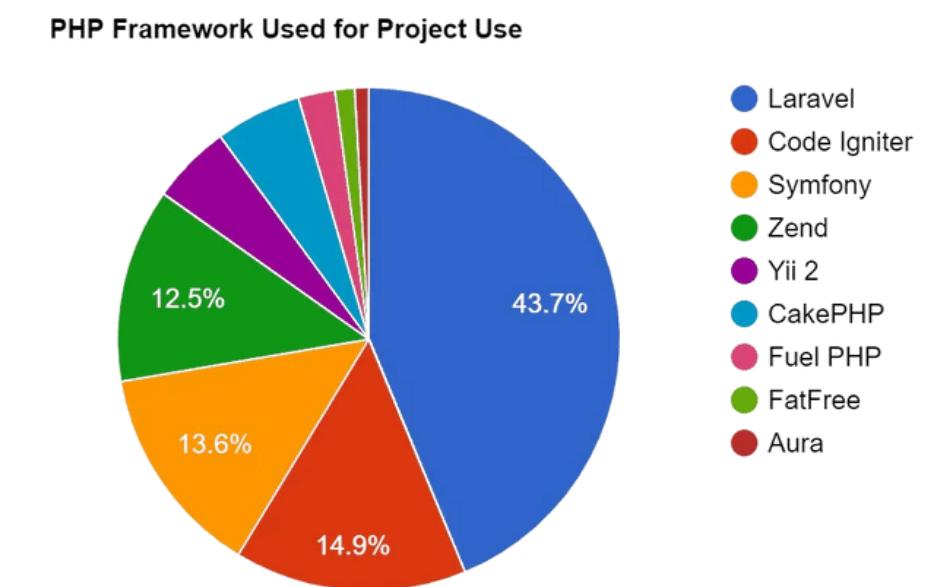


Hi, I'm Aya, a passionate web developer.
Recently completed an internship, sharpening my Laravel skills.
This carousel highlights Laravel basics I've mastered—perfect for
beginners!
Excited to explore MVC, routing?

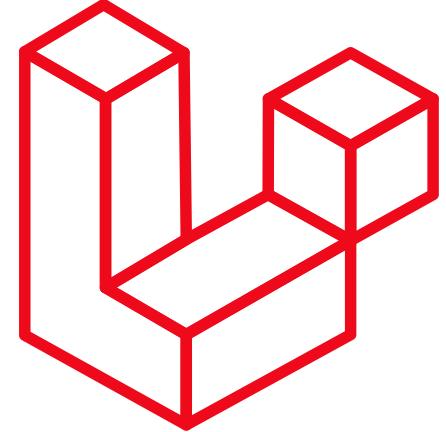
Laravel: The PHP Framework for Modern Web Apps



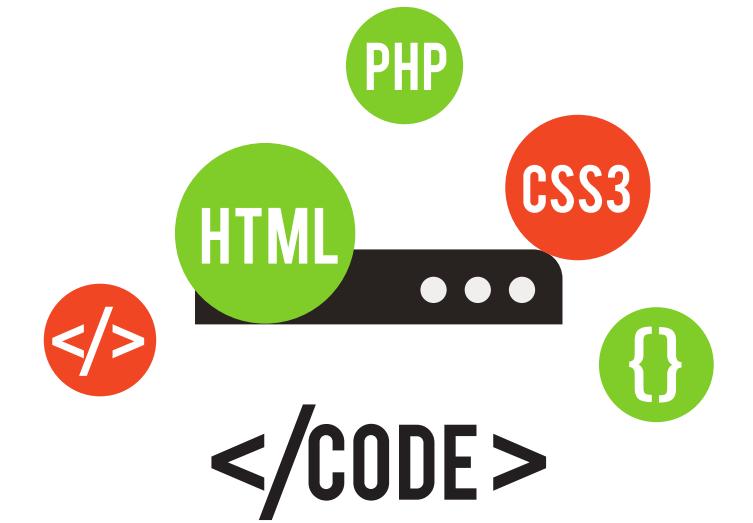
- Open-source PHP framework for building modern web apps efficiently.
- Created by Taylor Otwell in 2011; follows MVC pattern for structured code.
- Emphasizes "developer happiness" with expressive syntax and tools.
- Used by giants like BBC and Pfizer for scalable, secure applications.



Laravel Features: Powering Modern Development

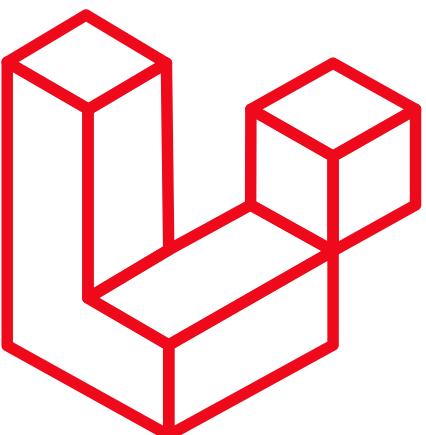
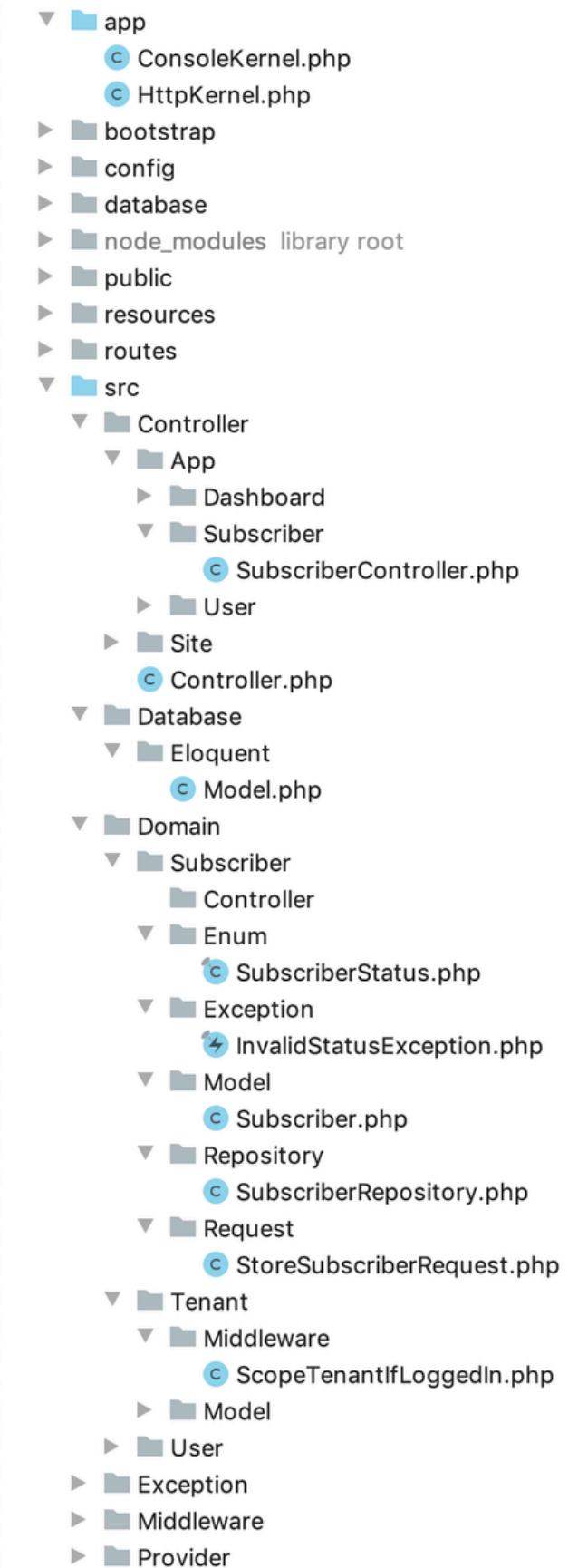


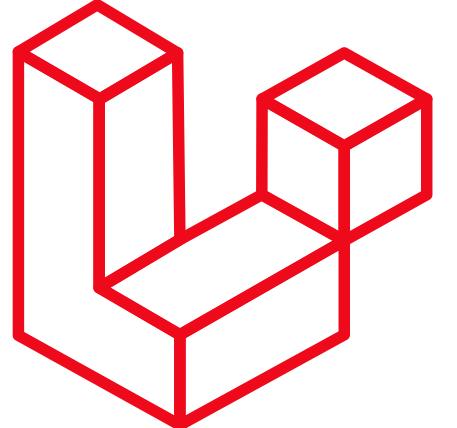
- **Routing:** Defines URL paths and maps them to app logic.
- **Middleware:** Filters HTTP requests for security and logic.
- **Blade Template Engine:** Simplifies dynamic HTML creation.
- **Eloquent ORM:** Streamlines database interactions with ActiveRecord.
- **Artisan Command Line Interface:** Automates tasks like migrations.
- **Security Features:** Includes CSRF protection and encryption.



Laravel Project Structure: Organized & Scalable

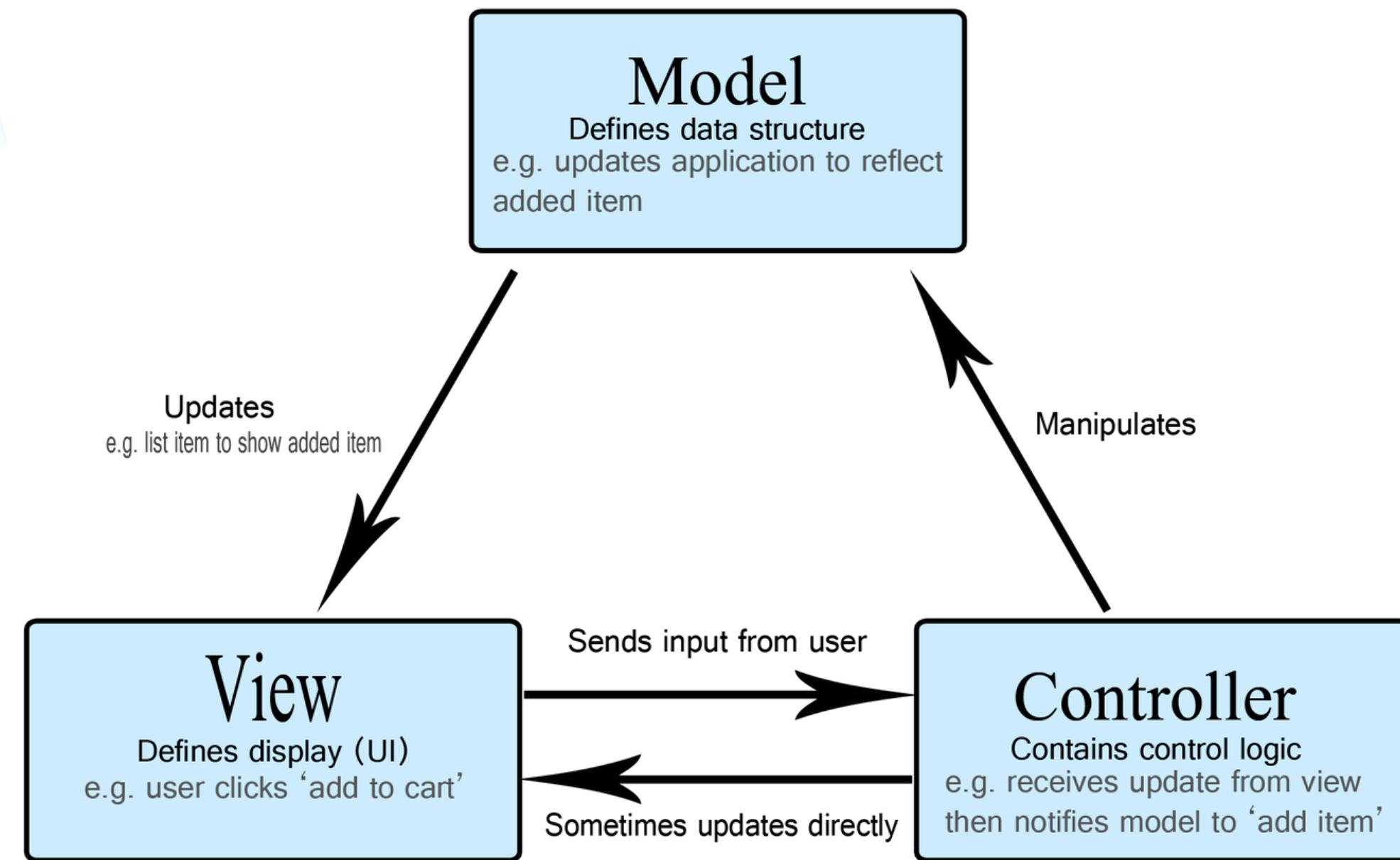
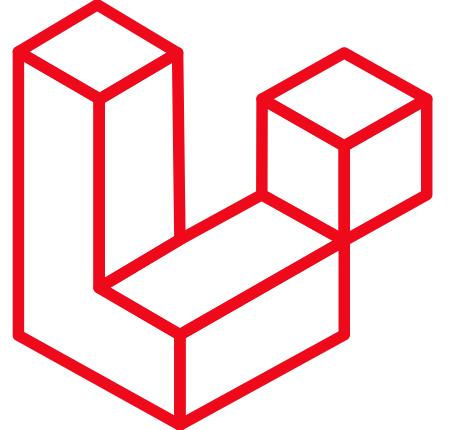
- **app/**: Core logic (Controllers, Models, Providers).
- **bootstrap/**: Initializes the framework (e.g., app.php).
- **config/**: Stores configuration files (e.g., database.php).
- **database/**: Manages migrations and seeds.
- **public/**: Entry point with index.php and assets.
- **resources/**: Holds views, CSS, and JS.
- **routes/**: Defines web and API routes.

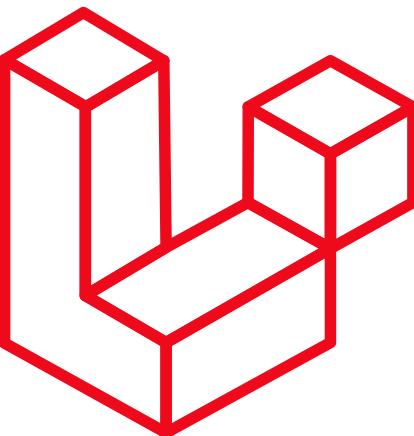




MVC: The Heart of Laravel Architecture

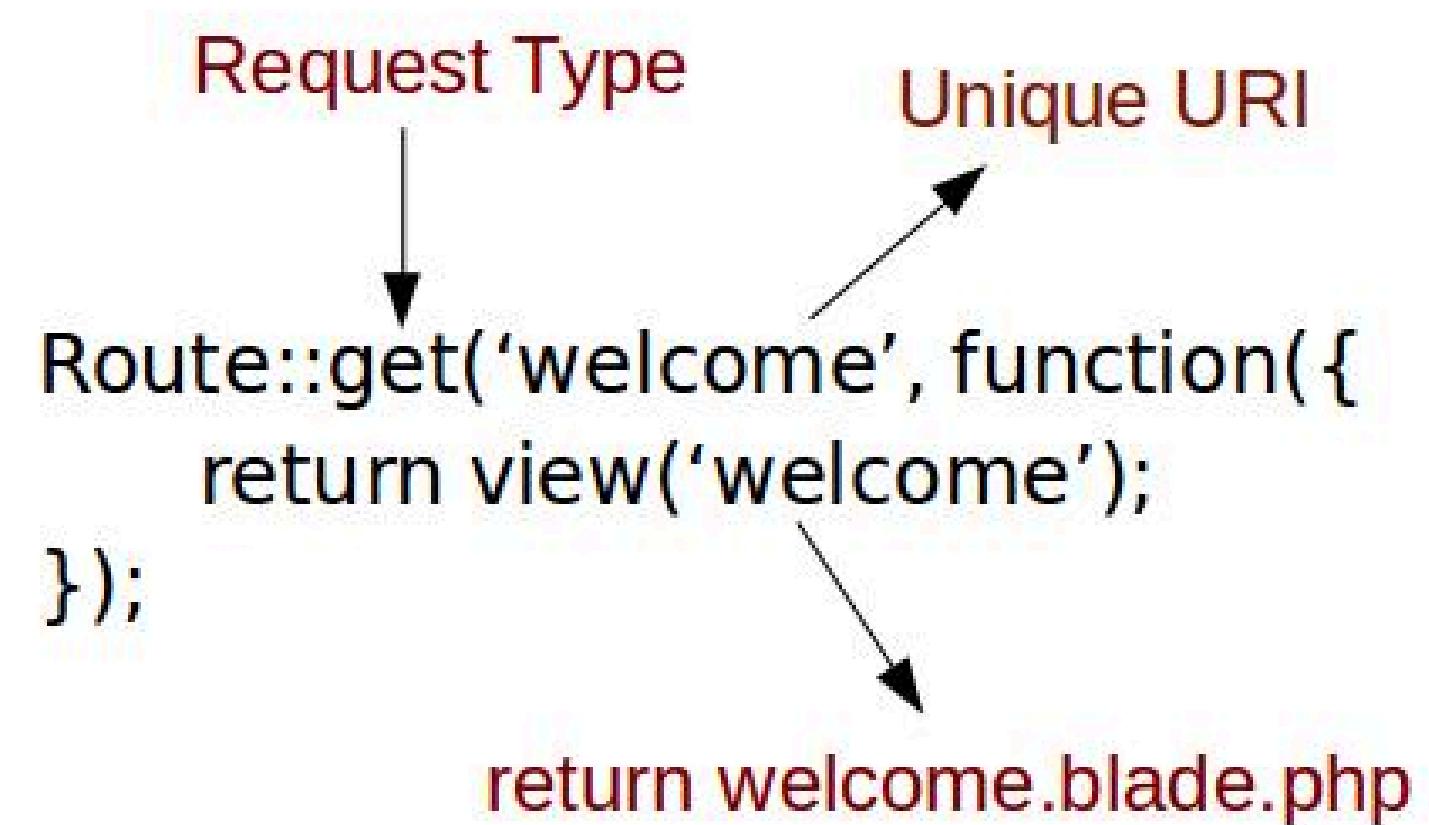
- **Model:** Manages data and database logic (e.g., Eloquent ORM).
- **View:** Handles UI rendering using Blade templates.
- **Controller:** Processes requests, interacts with Models, and returns Views.
- Laravel follows MVC to separate concerns for scalability and maintainability.



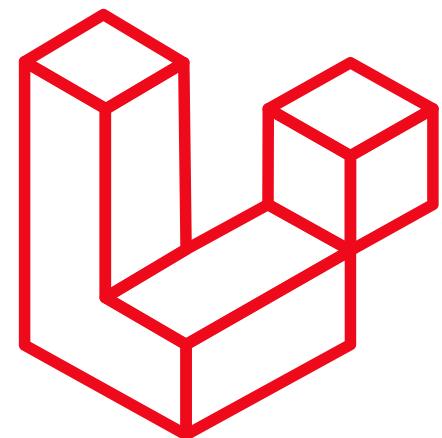


Routes: The Gateway to Your App

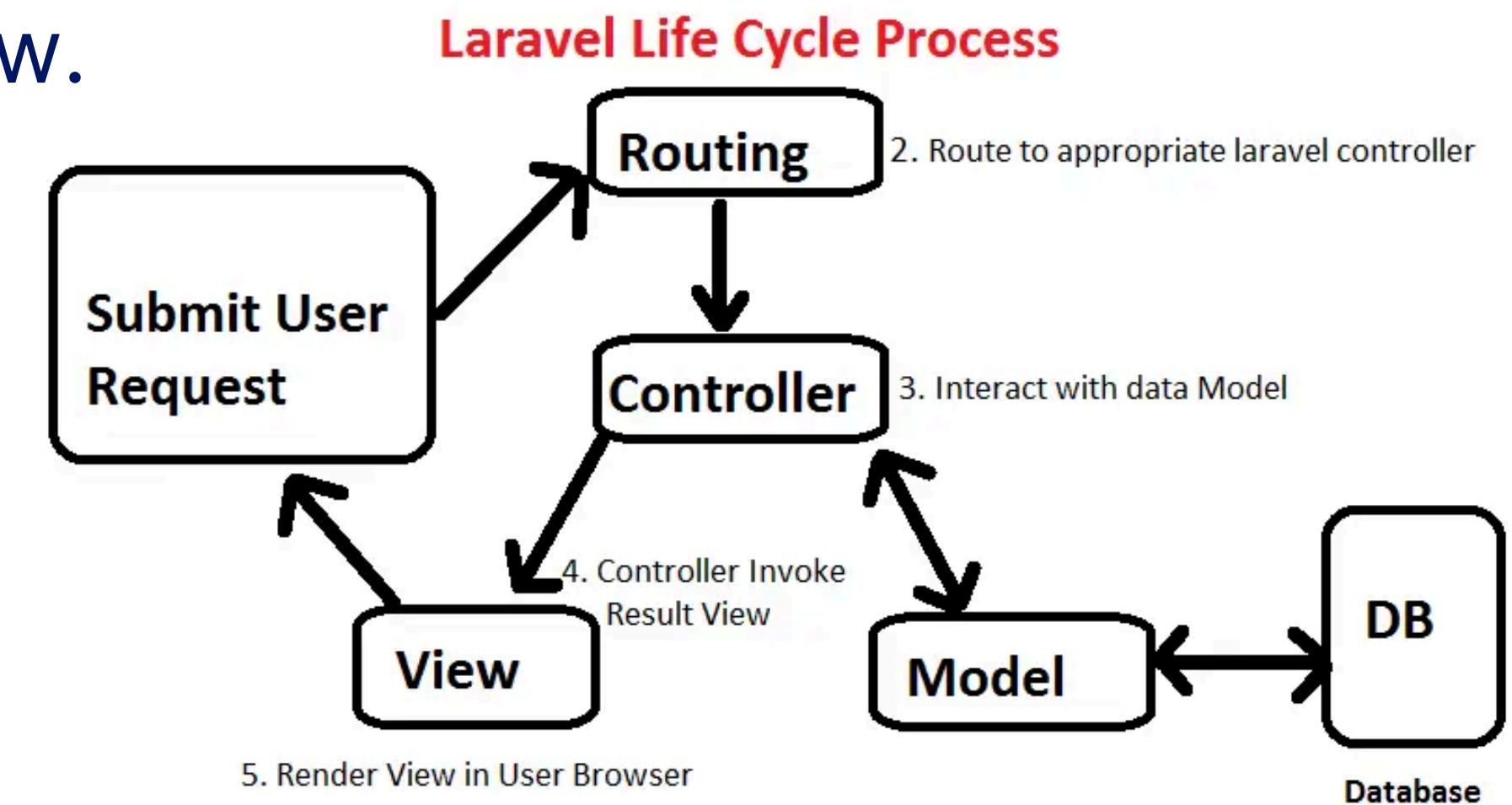
- Routes define URLs and map them to actions (e.g., controllers or closures).
- For web: Created in routes/web.php; for APIs: routes/api.php.
- Handles request types (GET, POST) and unique URIs.

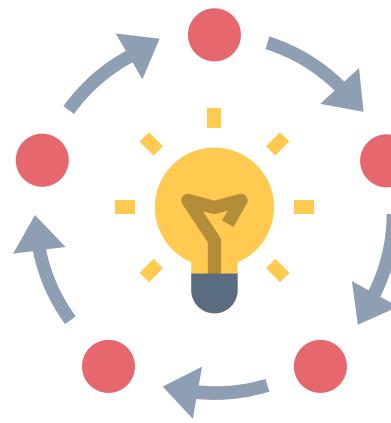


How Requests Flow Through Laravel

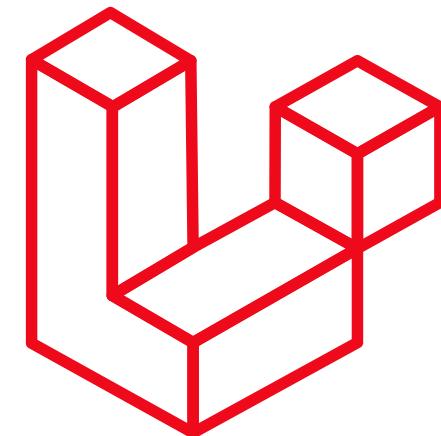


1. User submits request via browser (e.g., URL entry).
2. Route directs to appropriate Controller.
3. Controller interacts with Model/Database.
4. Controller invokes results in View.
5. View renders in user's browser.



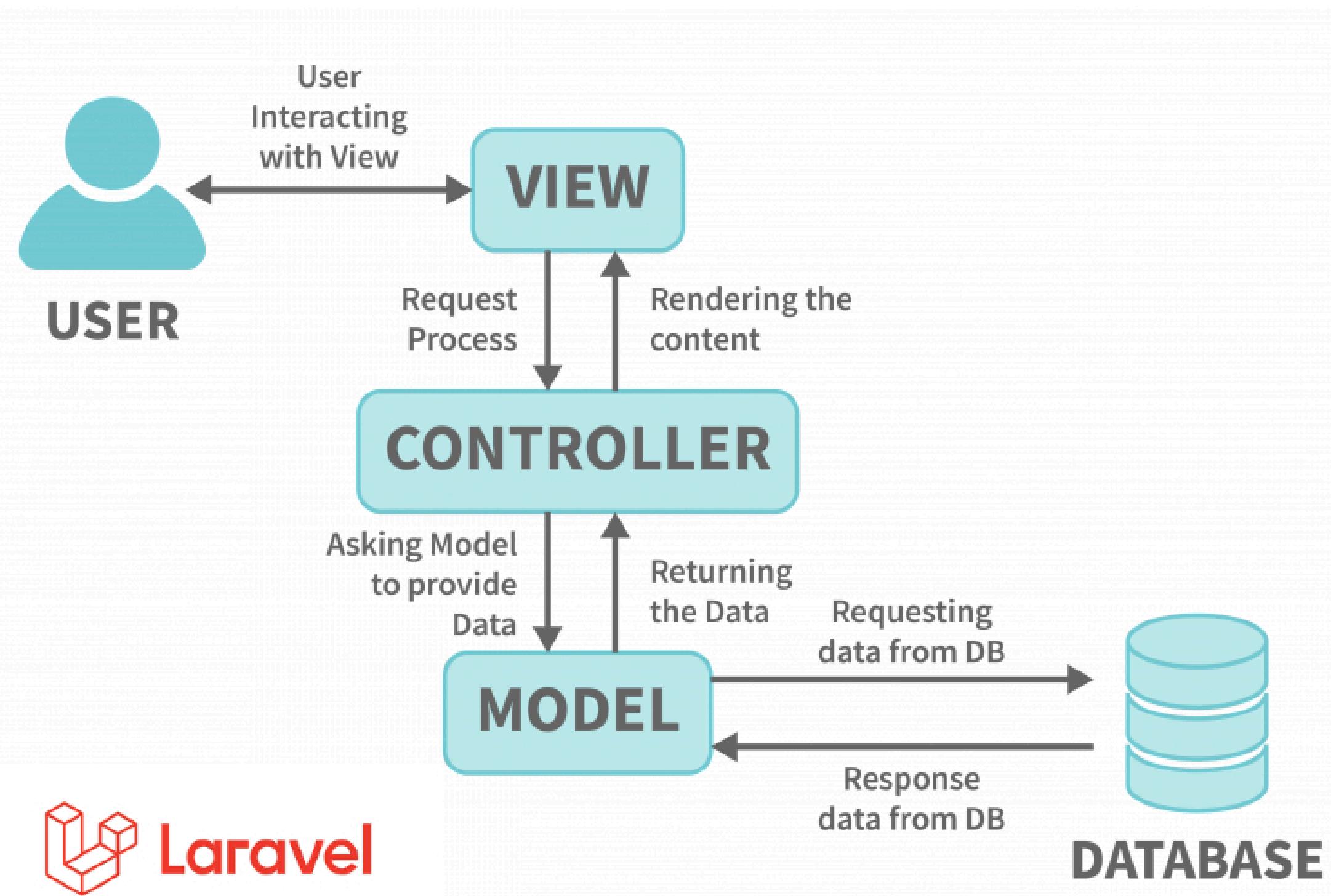


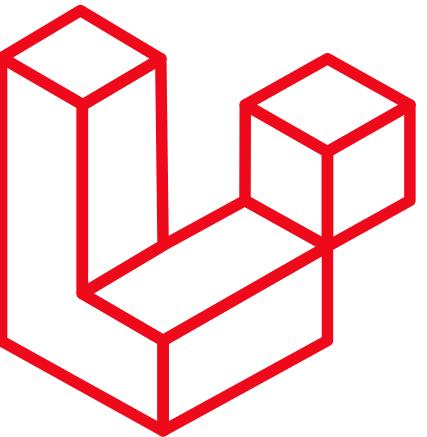
Controllers: The Logic Hub



- Centralize business logic; handle requests and return responses.
- Created via Artisan: **php artisan make:controller UserController**.
- Methods for CRUD: `index()`, `store()`, `show()`,
`update()`, `destroy()`.
- Inject dependencies like Request or Models for clean code.

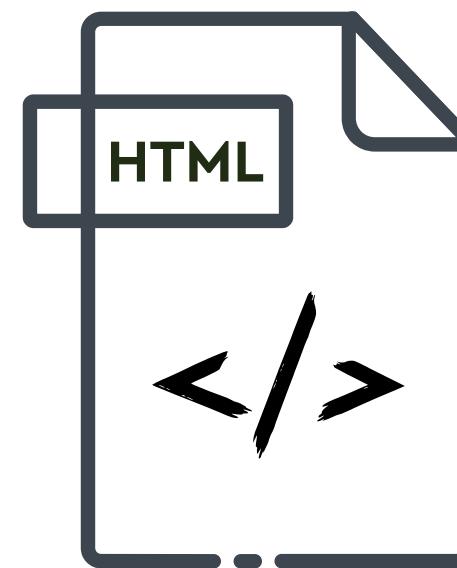
```
... Code  
1 class TaskController extends Controller  
2 {  
3     use ValidatesRequests;  
4     public function store(Request $request)  
5     {  
6         $this->validate($request, [  
7             'name' => 'required|max:255',  
8             'description' => 'required',  
9         ]);  
10        // Code to store the task in the database  
11    }  
12 }
```



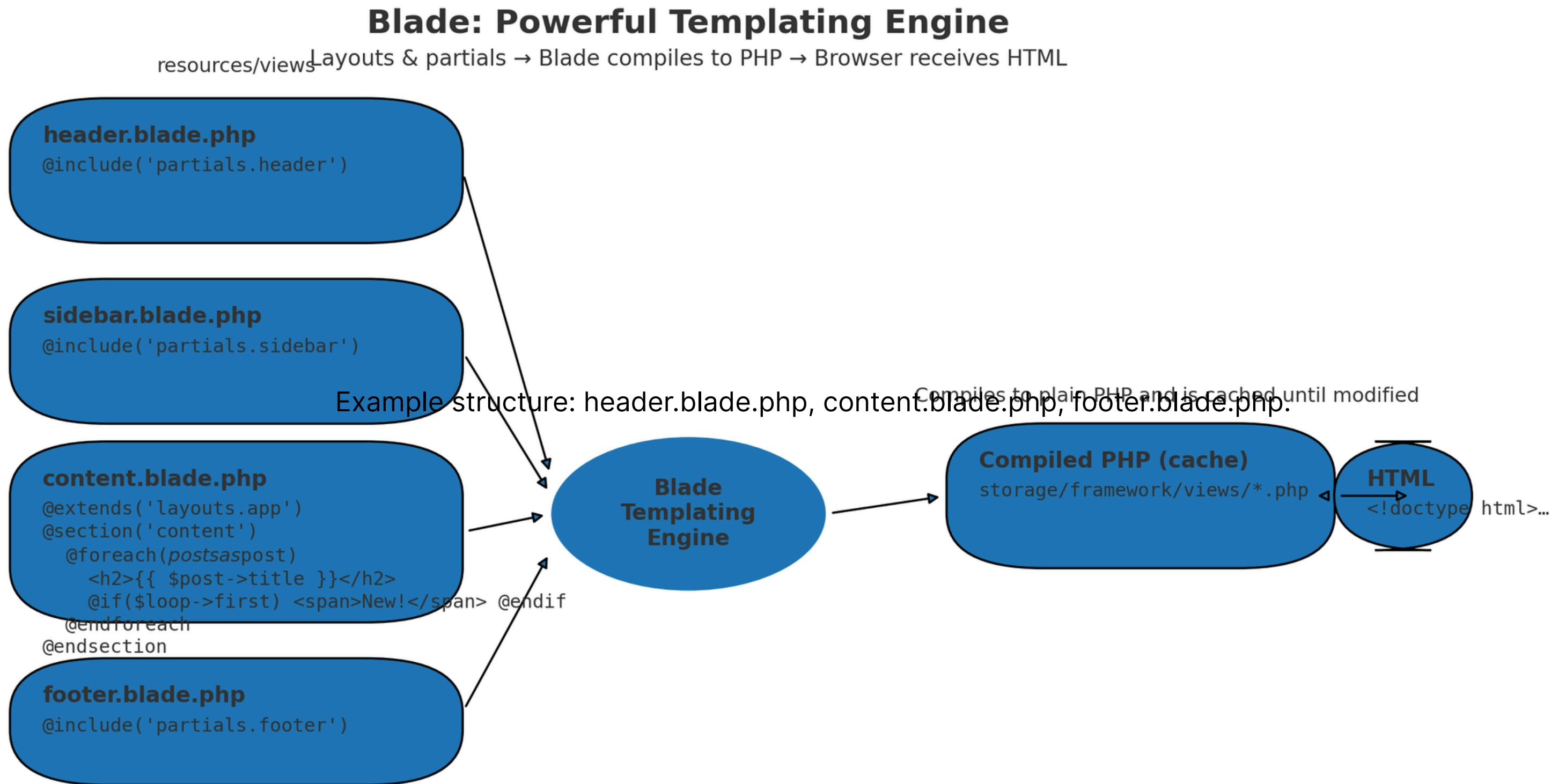


Powerful Templating Engine

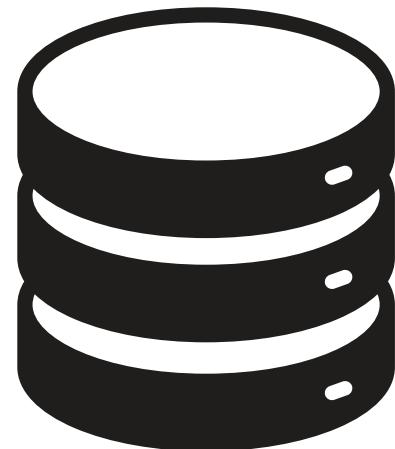
- Laravel's templating system for dynamic HTML.
- Files in resources/views; extend layouts with @extends, @section.
- Compiles to plain PHP/HTML; supports loops (@foreach), conditionals (@if).
- Example structure: header.blade.php, content.blade.php, footer.blade.php.



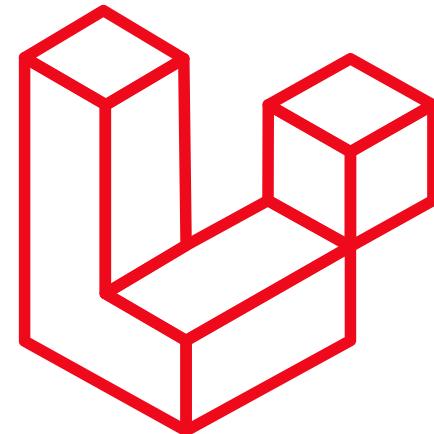
- Example structure: header.blade.php, content.blade.php, footer.blade.php.



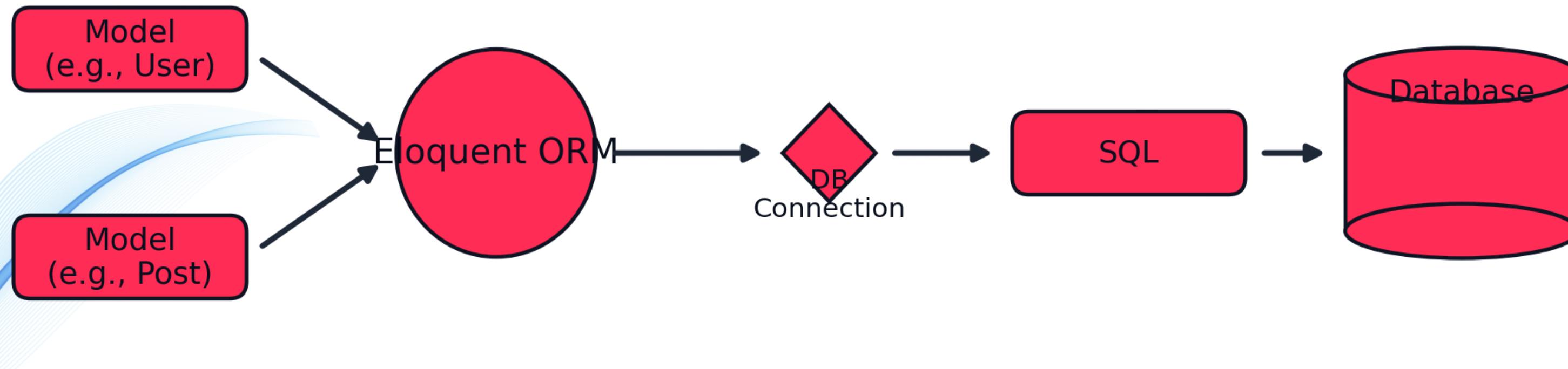
Blade directives shown: @extends, @section, @foreach, @if

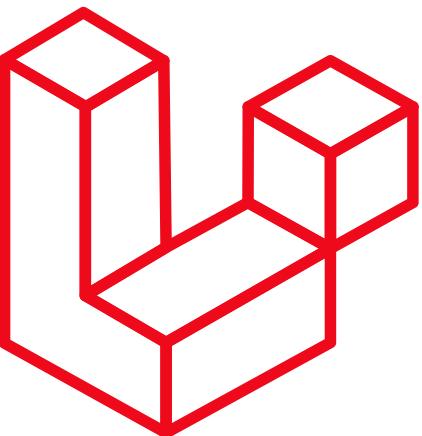


Seamless Database Interaction



- **Eloquent ORM** provides ActiveRecord for database work.
- Each table has a **Model** (e.g., User model for 'users' table).
- Simple queries: `User::find(1);` or `User::where('age', '>', 18)->get();`
- Handles relationships: **hasOne**, **hasMany**, **belongsToMany**.





Migrations, Seeds, and Queries

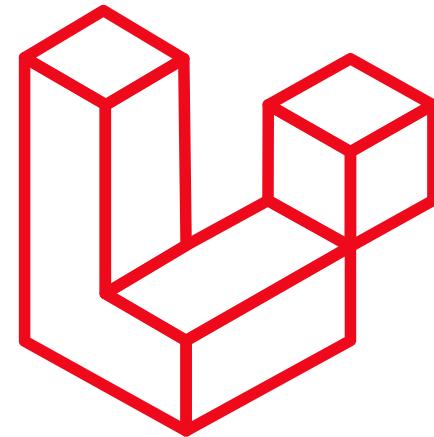
- **Migrations:** Version control for schema (php artisan make:migration).
- **Seeds:** Populate data (php artisan db:seed).
- **Query Builder:** DB::table('users')->select('name')->get();
- **Advanced:** Eager loading to avoid N+1 issues (with('relation')).

```
E:\laragon\www\laravel10>php artisan migrate:fresh
Dropping all tables ...
INFO Preparing database.

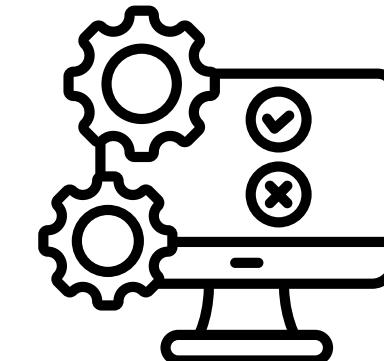
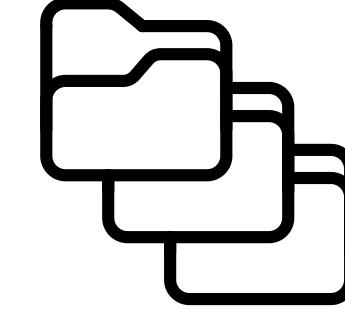
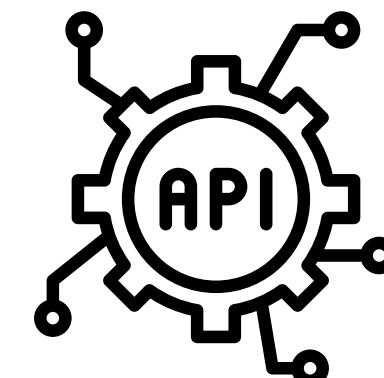
Creating migration table ...
INFO Running migrations.

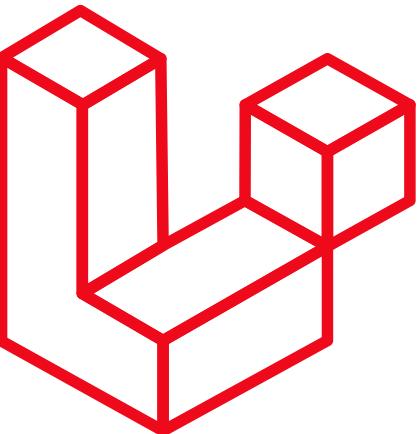
2014_10_12_000000_create_users_table ...
2014_10_12_100000_create_password_reset_tokens_table ...
2019_08_19_000000_create_failed_jobs_table ...
2019_12_14_000001_create_personal_access_tokens_table ...
```

What Makes Laravel Shine



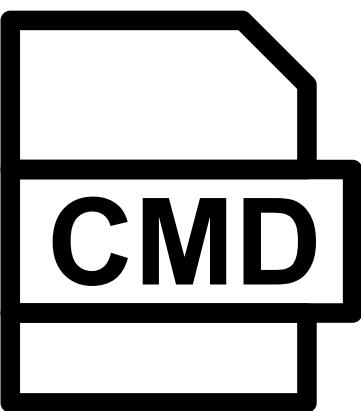
- **Authentication:** Quick setup with Laravel Breeze or Jetstream.
- **APIs:** Built-in support for RESTful routes and Sanctum tokens.
- **Queues & Jobs:** Handle background tasks with Horizon.
- **Testing:** PHPUnit integration for units and features.
- **Security:** Built-in CSRF, encryption, and validation.



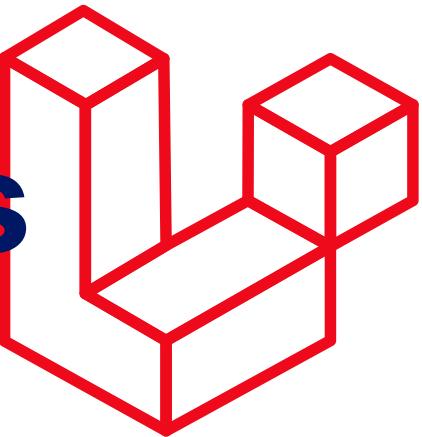


Must-Know Artisan Commands

- `php artisan make:model ModelName`: Creates a new model.
- `php artisan make:controller ControllerName`: Generates a controller.
- `php artisan migrate`: Runs database migrations.
- `php artisan db:seed`: Populates database with seed data.
- `php artisan serve`: Starts the local development server.
- `php artisan route:list`: Displays all registered routes.



Laravel Mastery: My Takeaways & Next Steps



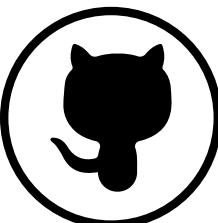
Scalable, community-driven, and fun to use—perfect for modern dev.

I've applied these in projects; excited to build more!

What's your go-to Laravel feature? Share below!

Connect: <https://www.linkedin.com/in/ayamekni/> | Open to Laravel chats or

roles.



Check out my Git repo: [https://github.com/ayamekni/Eventify_events_Management_app]

Thank You

