

Laravel Project Structure Explanation

This document explains every folder in a Laravel project and how this specific authentication project uses them.

Root-Level Directories

/app - Application Core

What it is: Contains all your business logic - models, controllers, services, and other custom classes.

What typically goes in it:

- Models (database tables as PHP classes)
- Controllers (handle HTTP requests)
- Services (reusable business logic)
- Middleware (request filtering)

How this project uses it:

- `app/Models/User.php` - Represents customers
- `app/Models/Admin.php` - Represents staff/admins
- `app/Http/Controllers/AuthController.php` - Handles user login/signup
- `app/Http/Controllers/AdminAuthController.php` - Handles admin login/signup
- `app/Services/AuthManager.php` - Shared authentication logic

MVC Role: This is primarily the **M (Model)** and **C (Controller)** in MVC.

/app/Models - Database Tables as Classes

What it is: Each file represents a database table. For example, `User.php` represents the `user` table.

What typically goes in it:

- Eloquent models (Laravel's ORM)
- Relationships between tables
- Data validation rules
- Accessor/mutator methods

How this project uses it:

- `User.php` : Customer accounts with `register()`, `login()`, `logout()` methods

- `Admin.php` : Staff accounts (completely separate from users)

MVC Role: The **M (Model)** - represents data and business rules.

/app/Http/Controllers - Request Handlers

What it is: Controllers receive HTTP requests, process them, and return responses (views or redirects).

What typically goes in it:

- Methods for each page/action (`showLogin`, `login`, `register`)
- Form validation
- Calls to models and services
- Return views or JSON responses

How this project uses it:

- `AuthController.php` :
 - `show_login()` - Display login page
 - `login_user()` - Process login form
 - `show_register()` - Display signup page
 - `register_user()` - Process signup form
 - `logout_user()` - Log user out
- `AdminAuthController.php` : Same methods but for admins

MVC Role: The **C (Controller)** - handles user input and coordinates between models and views.

/routes - URL Definitions

What it is: Defines which URLs map to which controller methods.

What typically goes in it:

- `web.php` : Routes for browser-based pages (HTML responses)
- `api.php` : Routes for API endpoints (JSON responses)

How this project uses it:

- `/login` → `AuthController@show_login`
- `/register` → `AuthController@show_register`
- `/admin/login` → `AdminAuthController@showLogin`

MVC Role: Routing layer - determines which controller handles which request.

/resources - Frontend Assets

What it is: Everything related to the user interface.

What typically goes in it:

- `views/` : Blade templates (HTML with PHP)
- `css/` : Stylesheets
- `js/` : JavaScript files

How this project uses it:

- `resources/views/auth/login.blade.php` : User login page
- `resources/views/auth/register.blade.php` : User signup page
- `resources/views/admin/login.blade.php` : Admin login page
- `resources/views/admin/dashboard.blade.php` : Admin dashboard

MVC Role: The **V (View)** - presentation layer.

/public - Web Server Root

What it is: The only folder accessible directly via browser.

How this project uses it:

- `public/csm-logo.png` : Logo image
- `public/build/` : Compiled frontend assets

MVC Role: Entry point and static asset storage.

/config - Configuration Files

What it is: Application settings.

How this project uses it:

- `config/auth.php` : Defines `web` and `admin` guards
- `config/session.php` : Session lifetime and security
- `config/database.php` : MySQL connection settings

/database - Database Files

What it is: Everything related to the database schema.

What typically goes in it:

- `migrations/` : Database table definitions
- `seeders/` : Sample data
- `factories/` : Test data generators

How this project uses it:

- `migrations/2025_11_19_170953_create_user_table.php` : Creates `user` table
 - `migrations/2025_12_03_230000_create_admins_table.php` : Creates `admins` table
 - `migrations/2025_12_03_222000_add_remember_token_to_user_table.php` : Adds "Remember Me" feature
-

/tests - Automated Tests

What it is: Code that automatically verifies your application works correctly.

What typically goes in it:

- `Unit/` : Tests for individual classes (models, services)
- `Feature/` : Tests for entire features (login flow, signup flow)

How this project uses it:

- `tests/Unit/UserTest.php` : Tests `User` model methods
- `tests/Unit/AuthManagerTest.php` : Tests authentication service
- `tests/Feature/LoginTest.php` : Tests entire login process
- `tests/Feature/AdminAuthTest.php` : Tests admin authentication

MVC Role: Quality assurance layer.

/storage - Application Storage

What it is: Files generated by the application.

What typically goes in it:

- `logs/` : Error logs
- `framework/sessions/` : Session files
- `framework/views/` : Compiled Blade templates

Do NOT commit to Git: This folder's contents change constantly.

/vendor - Third-Party Code

What it is: Laravel framework and all installed packages (via Composer).

Do NOT commit to Git: Generated by running `composer install`.