

## **Part 2: Laravel Folder Structure**

In a Laravel project, the purpose of each of the following folders is as follows:

**app:** The "app" folder contains the core application files. It includes the application's models, views, controllers, and other classes that define the business logic and functionality of the Laravel application.

**bootstrap:** The "bootstrap" folder contains files that are responsible for bootstrapping the Laravel framework and setting up the initial configuration. It includes the "app.php" file that initializes the application and registers service providers.

**config:** The "config" folder contains configuration files for various aspects of the Laravel application. It includes files like "app.php" for general application configuration, "database.php" for database connection settings, and "mail.php" for email-related configuration.

**database:** The "database" folder is used for database-related files. It includes database migrations, which are files that define the structure of database tables, as well as seeders for populating the database with test or dummy data.

public: The "public" folder is the web server's document root and serves as the entry point for the application. It contains the "index.php" file, which is responsible for handling incoming requests and bootstrapping the Laravel application.

resources: The "resources" folder contains non-PHP resources like views, language files, and assets such as CSS and JavaScript files. Views are typically stored in the "views" subfolder, while assets are stored in the "assets" or "public" subfolders.

routes: The "routes" folder contains the route definitions for the application. It includes files like "web.php" for defining web routes and "api.php" for defining API routes. Routes determine how incoming requests are handled and which controllers or actions are responsible for processing them.

storage: The "storage" folder is used for storing various files generated by the application, such as logs, cached views, session files, and uploaded files. It includes subfolders like "app" for application-specific files, "logs" for log files, and "framework" for framework-generated files.

tests: The "tests" folder contains test files for the application. Laravel provides a testing framework that allows developers to

write automated tests to ensure the correctness of their code. Test files typically follow the naming convention of "Feature" and "Unit" tests.

vendor: The "vendor" folder is created by Composer, the PHP package manager used by Laravel. It contains all the third-party dependencies installed via Composer, including the Laravel framework itself and any additional packages or libraries used in the project. It should not be modified manually, as Composer manages its content.

These folders play crucial roles in organizing and structuring a Laravel project, separating concerns, and facilitating various aspects of development, such as configuration, routing, database management, and testing.