Part 1:

To install the latest version of Laravel, we can choose between using Composer or the Laravel installer. Here are the steps that I followed:

Using Composer:

1. I have installed composer on my system from the official website (https://getcomposer.org/).

2. Opened my terminal or command prompt.

3. Navigated to the directory where I want to install Laravel.

4. Run the following command to create a new Laravel project:

**composer create-project laravel/laravel Assignment13**

This command will download and install the latest version of Laravel and its dependencies.

5. Once it's done, It will create a new Laravel project set up in the specified directory.

That's it! Now the installation of the latest version of Laravel using Composer done. Now I can start working on Laravel project by navigating to its directory and running appropriate Laravel commands like `php artisan serve` to launch the development server.

Part 2: Laravel Folder Structure

Describing the purpose of each of the following folders in a Laravel project:

In a Laravel project, the folders serve different purposes and contain specific types of files. Here's a brief description of each of the folders you mentioned:

**1. App:** This folder contains the core files of your application. It typically includes the models, controllers, and other classes that define your application's business logic.

**2. Bootstrap:** The bootstrap folder contains files that are responsible for bootstrapping and initializing your Laravel application. It includes the app.php file, which sets up the basic environment and loads the necessary components.

**3. Config:** The config folder contains various configuration files for your Laravel application. These files allow you to define settings related to database connections, services, application behavior, and more. You can modify these files to customize the behavior of your application.

**4. Database:** The database folder is used to store database-related files. It includes the migrations directory where you can define database schema changes, as well as the seeds directory for defining initial data. The factories directory is used for generating fake data during development and testing.

**5. Public:** The public folder contains the entry point for your application, which is the index.php file. This is the only publicly accessible folder, and it typically contains assets like CSS, JavaScript, and image files.

**6. Resources:** The resources folder stores files that are used during the development process. It includes directories like views, which contain Blade templates for generating HTML, and assets, which can include raw CSS, JavaScript, or other front-end files.

**7. Routes:** The routes folder contains route definitions for your application. It includes files that define the URL endpoints and map them to appropriate controllers or closures. You can define web routes, API routes, or console routes depending on the type of application you are building.

**8. Storage:** The storage folder is used to store files generated or used by your application at runtime. It includes directories like app, framework, and logs. The app directory is commonly used for storing user-generated files, while the framework directory contains cache, sessions, and other framework-related files.

**9. Tests:** The tests folder is where you can write automated tests for your application. It includes directories for unit tests, feature tests, and browser tests. These tests help ensure the quality and correctness of your codebase.

**10. Vendor:** The vendor folder is not created by default but is commonly present in a Laravel project. It is where Composer installs the dependencies for your project. It includes all the third-party packages and libraries that your application relies on.



