Part 1: Laravel Installation Steps:

**Ans:**

1. First, make sure that you have Composer installed on your system. You can download and install Composer from <https://getcomposer.org/>.
2. Open a terminal or command prompt and navigate to the directory where you want to install your Laravel project.
3. Run the following command to create a new Laravel project:

composer create-project laravel/laravel Laravel\_ostad.

1. Composer will download and install the necessary dependencies for Laravel, which may take a few minutes depending on your internet connection speed.
2. Once the installation is complete, navigate into the newly created project directory:

cd Laravel\_ostad

1. You can then start a development server by running the following command:

php artisan serve

This will start a server at [http://localhost:8000](http://localhost:8000/) where you can view your Laravel application in your web browser.

That's it! You now have a new Laravel project up and running.

A screen shot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Part 2: Laravel Folder Structure

**Ans:**

**app:** This folder contains most of the application's logic. It includes subfolders for various types of application components, such as controllers, models, and views. The app folder is where you'll spend most of your time developing the core functionality of your application.

**bootstrap:** This folder contains the files needed to bootstrap the Laravel application. It includes the app.php file that loads the application and its dependencies, as well as various configuration files and other setup scripts.

**config:** This folder contains configuration files for the application, including database connections, service providers, and various other settings. You can modify these files to customize the behavior of your application.

**database:** This folder contains database migrations, seeders, and other database-related files. You can use Laravel's built-in database tools to create and manage your database schema, and to interact with the database from within your application.

**public:** This folder contains the front controller and other files that are publicly accessible from the web. It includes the index.php file, which is the entry point to your application, as well as other assets such as images, JavaScript, and CSS files.

**resources:** This folder contains resources that are used by the application, such as views, language files, and assets like images and CSS. This folder is organized by subfolders for each type of resource, such as views and lang.

**routes:** This folder contains the routes for your application, which define how incoming requests should be handled. You can define routes for different HTTP methods, such as GET, POST, PUT, and DELETE, and you can specify how parameters should be passed to your controllers.

**storage:** This folder contains files that are generated by the application, such as log files, cache files, and session data. You can also use this folder to store user-generated files, such as uploaded images.

**tests:** This folder contains automated tests for your application. You can use Laravel's built-in testing tools to write and run tests for your application's functionality.

**vendor:** This folder contains the third-party packages and libraries that your application depends on. These packages are installed using Composer and are managed by Laravel's dependency injection system.

new route in your Laravel project that displays a simple "Hello, World!" message:

**Ans:**

A screenshot of a computer program

Description automatically generated with medium confidence

