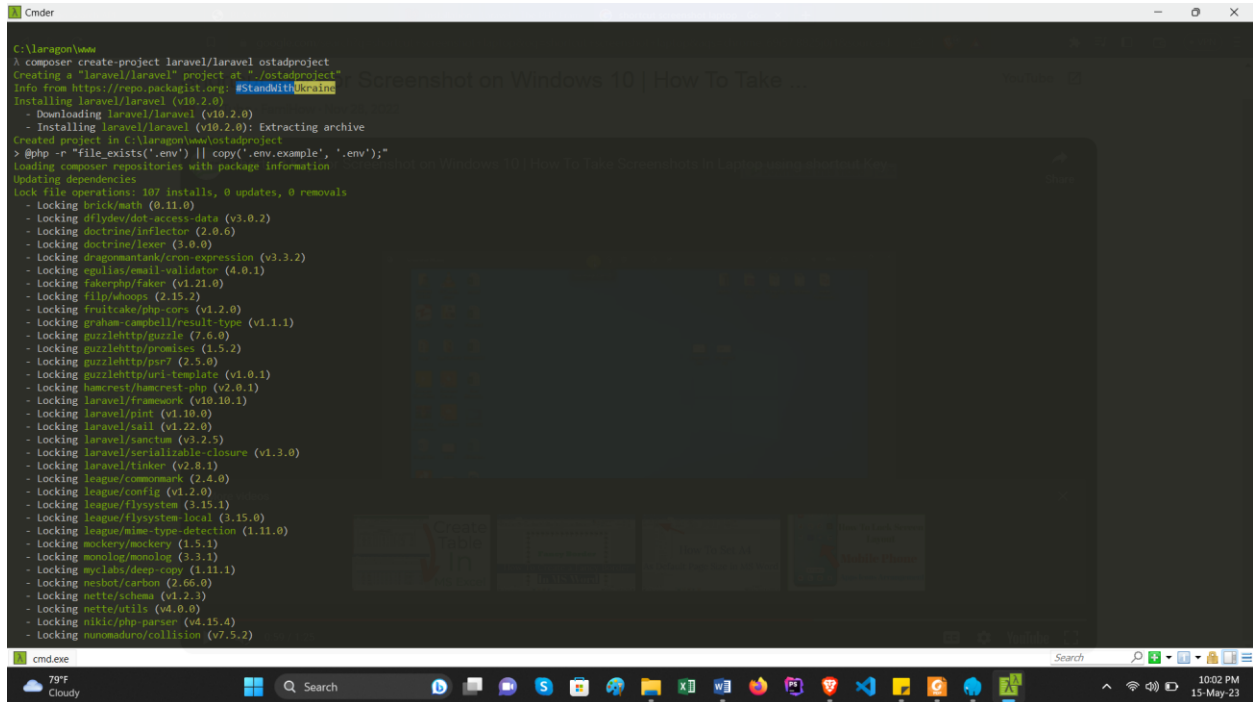


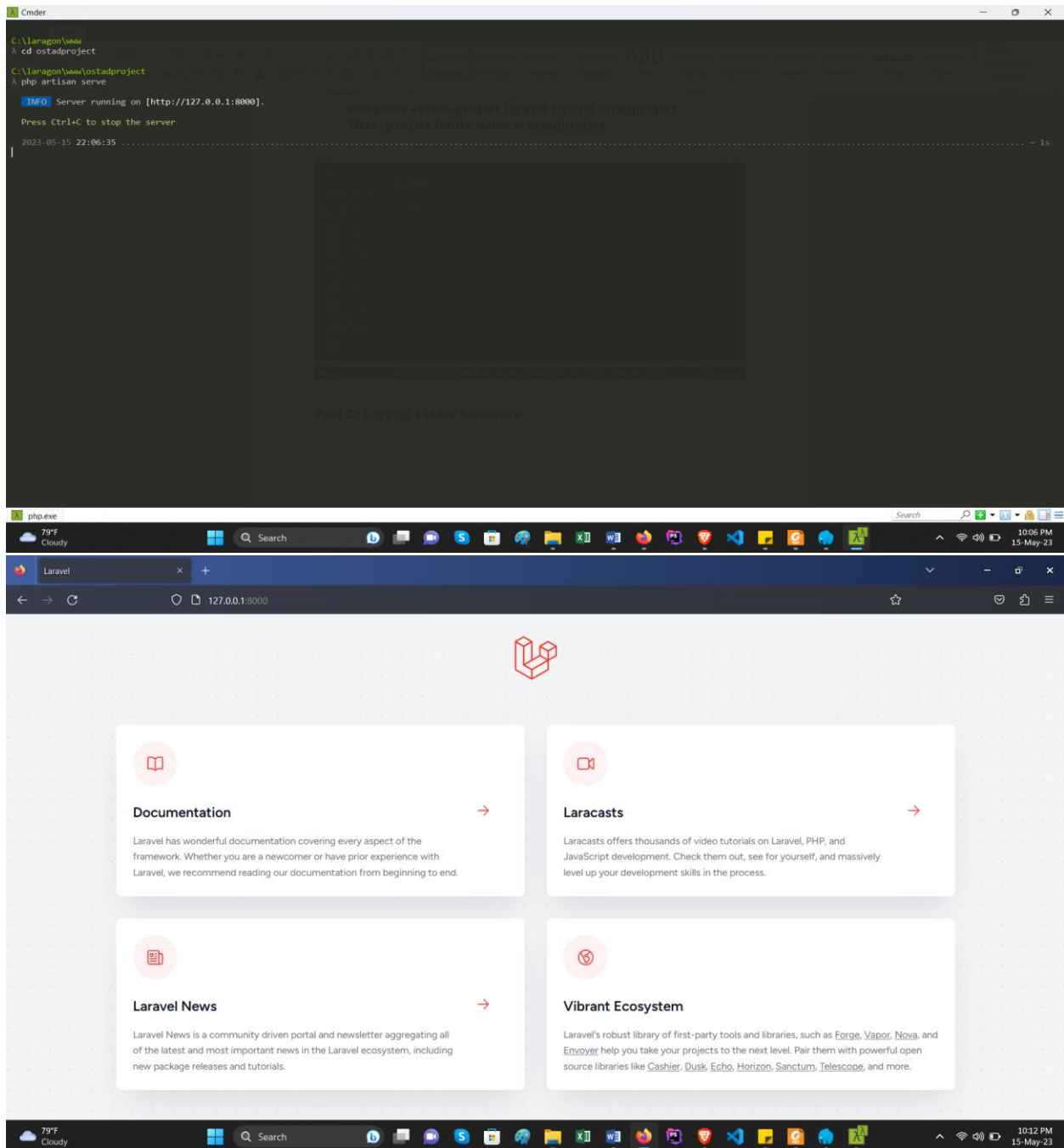
Part 1: Laravel Installation

- Installed Composer via Laragon Terminal as command
Composer install
- Create a new project using command
composer create-project laravel/laravel ostadproject
Here, project folder name is ostadproject

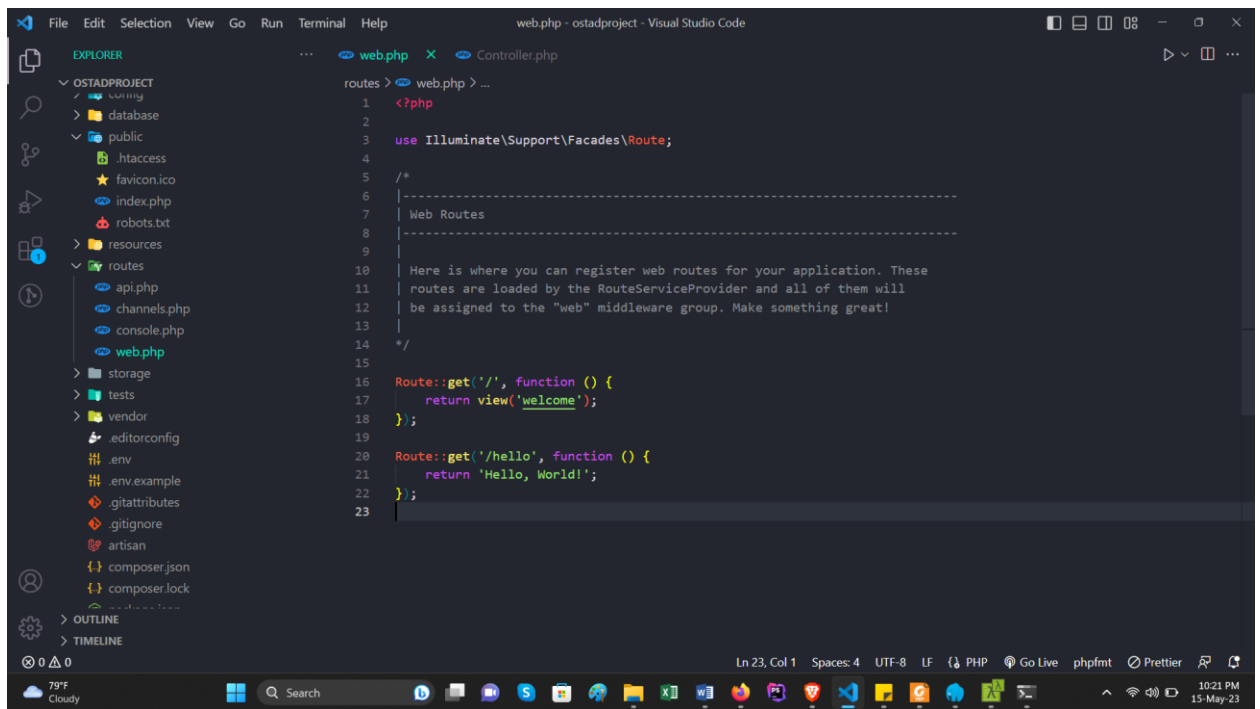


```
C:\laragon\www
λ composer create-project laravel/laravel ostadproject
Creating a "laravel/laravel" project at ".\ostadproject"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v10.2.0)
- Downloading laravel/laravel (v10.2.0)
- Installing laravel/laravel (v10.2.0): Extracting archive
Created project in C:\laragon\www\ostadproject
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 107 installs, 0 updates, 0 removals
- Locking brick/math (0.11.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inferno (2.0.6)
- Locking doctrine/lexer (3.0.0)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (4.0.1)
- Locking fakerphp/faker (v1.21.0)
- Locking filp/whoops (2.15.2)
- Locking fruitcake/php-cors (v1.2.0)
- Locking graham-campbell/result-type (v1.1.1)
- Locking guzzlehttp/guzzle (7.6.0)
- Locking guzzlehttp/promises (1.5.2)
- Locking guzzlehttp/psr7 (2.5.0)
- Locking guzzlehttp/uri-template (v1.0.1)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v10.10.1)
- Locking laravel/pint (v1.10.0)
- Locking laravel/sail (v1.22.0)
- Locking laravel/sanctum (v3.2.5)
- Locking laravel/serializable-closure (v1.3.0)
- Locking laravel/tinker (v2.8.1)
- Locking league/commonmark (2.4.0)
- Locking league/config (v1.2.0)
- Locking league/flysystem (3.15.1)
- Locking league/flysystem-local (3.15.0)
- Locking league/mime-type-detection (1.11.0)
- Locking mockery/mockery (1.5.1)
- Locking monolog/monolog (3.3.1)
- Locking myclabs/deep-copy (1.11.1)
- Locking nesbot/carbon (2.66.0)
- Locking nette/schema (v1.2.3)
- Locking nette/utils (v4.0.0)
- Locking nikic/php-parser (v4.15.4)
- Locking nunomaduro/collision (v7.5.2)
```

- Now enter into the project directory using command
cd ostadproject
- Then run the command for serving local host as
php artisan serve.
- Copy the server url from composer & paste into the browser as
<http://127.0.0.1:8000>
- After that we can successfully run the laravel project



- I have created a new route called hello & display



The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays the project structure for 'OSTADPROJECT', including folders like 'database', 'public', 'resources', 'routes', 'storage', 'tests', and 'vendor'. The 'routes' folder is expanded, showing files like 'api.php', 'channels.php', 'console.php', and 'web.php'. The main editor window is open to 'web.php', which contains the following PHP code:

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 /*
6 |-----
7 | Web Routes
8 |-----
9 |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider and all of them will
12 | be assigned to the "web" middleware group. Make something great!
13 |
14 */
15
16 Route::get('/', function () {
17     return view('welcome');
18 });
19
20 Route::get('/hello', function () {
21     return 'Hello, World!';
22 });
23
```

The status bar at the bottom indicates the current file is 'Ln 23, Col 1' with 'Spaces: 4', 'UTF-8', 'LF' line endings, and the PHP language mode is active. The Windows taskbar is visible at the bottom of the screen.



Hello, World!



Part 2: Laravel Folder Structure

Here is a brief description of each of the folders in a Laravel project:

- **app:** contains the core code of the application, including controllers, models, views, and other supporting files.
- **bootstrap:** contains the files needed to bootstrap the framework and configure autoloading.
- **config:** contains configuration files for the application, such as database connection details and environment variables.
- **database:** contains database-related files, including migrations and seeders.
- **public:** contains the front-facing assets of the application, such as CSS, JavaScript, and images.
- **resources:** contains the non-compiled resources, such as blade templates and JavaScript files, that will be compiled by Laravel's asset compilation system.
- **routes:** contains the route definitions for the application.
- **storage:** contains files that need to be stored for the application, such as logs, cache files, and user-generated content.
- **tests:** contains the automated tests for the application.
- **vendor:** contains the application's dependencies installed via Composer