# Learning Outcome

# After completing this module, the student should be able to understand Laravel Framework.

To meet the learning outcome, a student has to complete the following activities

1. Installation of the Laravel in the System
2. First Program by edit the View as Hello World
3. Create a Web application of minimum 4 blades demonstrating the blade template in laravel
4. Create a web Application which demonstrate routing through controller

# Activity 1

## Aim: Installation of the Laravel in the System.

**Learning outcome:** Able to understand Laravel Framework.

**Duration:** 5 hour

**List of Hardware/Software requirements:**

1. …
2. …...
3. …….

**Code/Program/Procedure (with comments):**

Install Laravel framework 8 for windows 10

Install composer on Windows, Follow below given steps to install composer on windows 10:

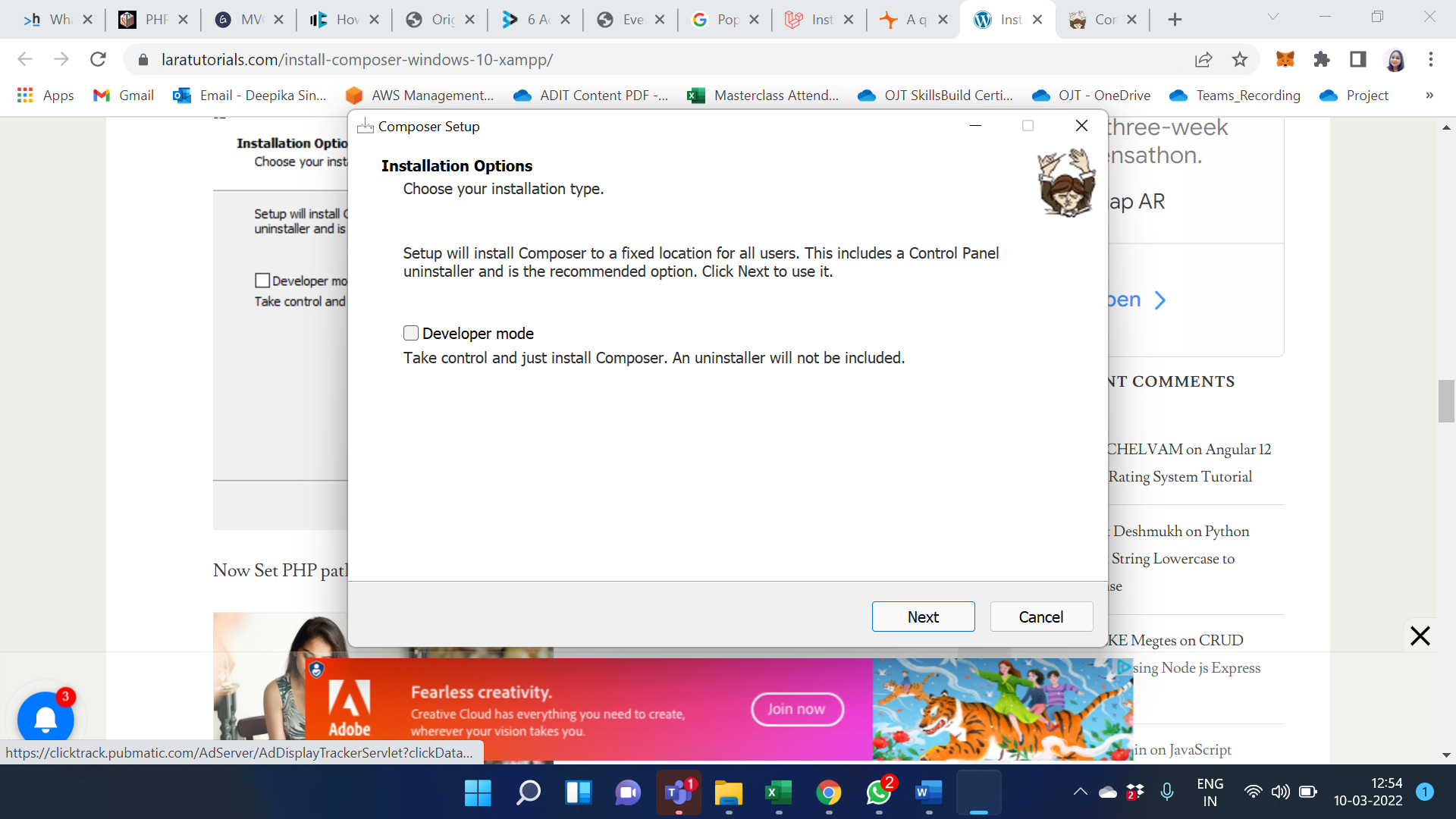
Step 1: Download Composer .exe File

In step 1, you need to open your browser and type getcomposer. Then visit on <www.getcomposer.org>. And download composer. If you already composer download / installed Composer. So, you need to move on next step.

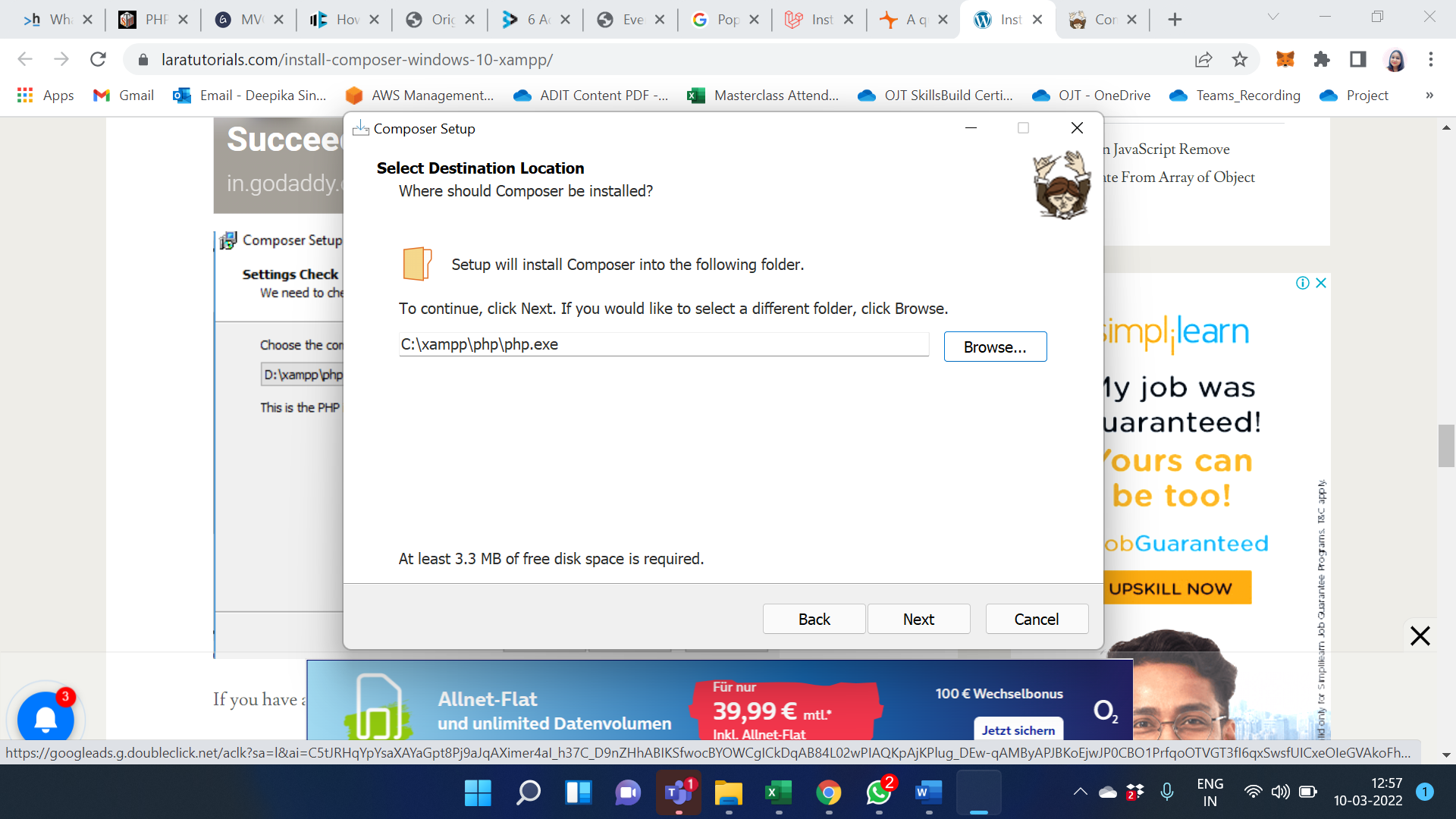
Click Download Composer from <www.getcomposer.org>.

Step 2 – Run the setup and Install Composer

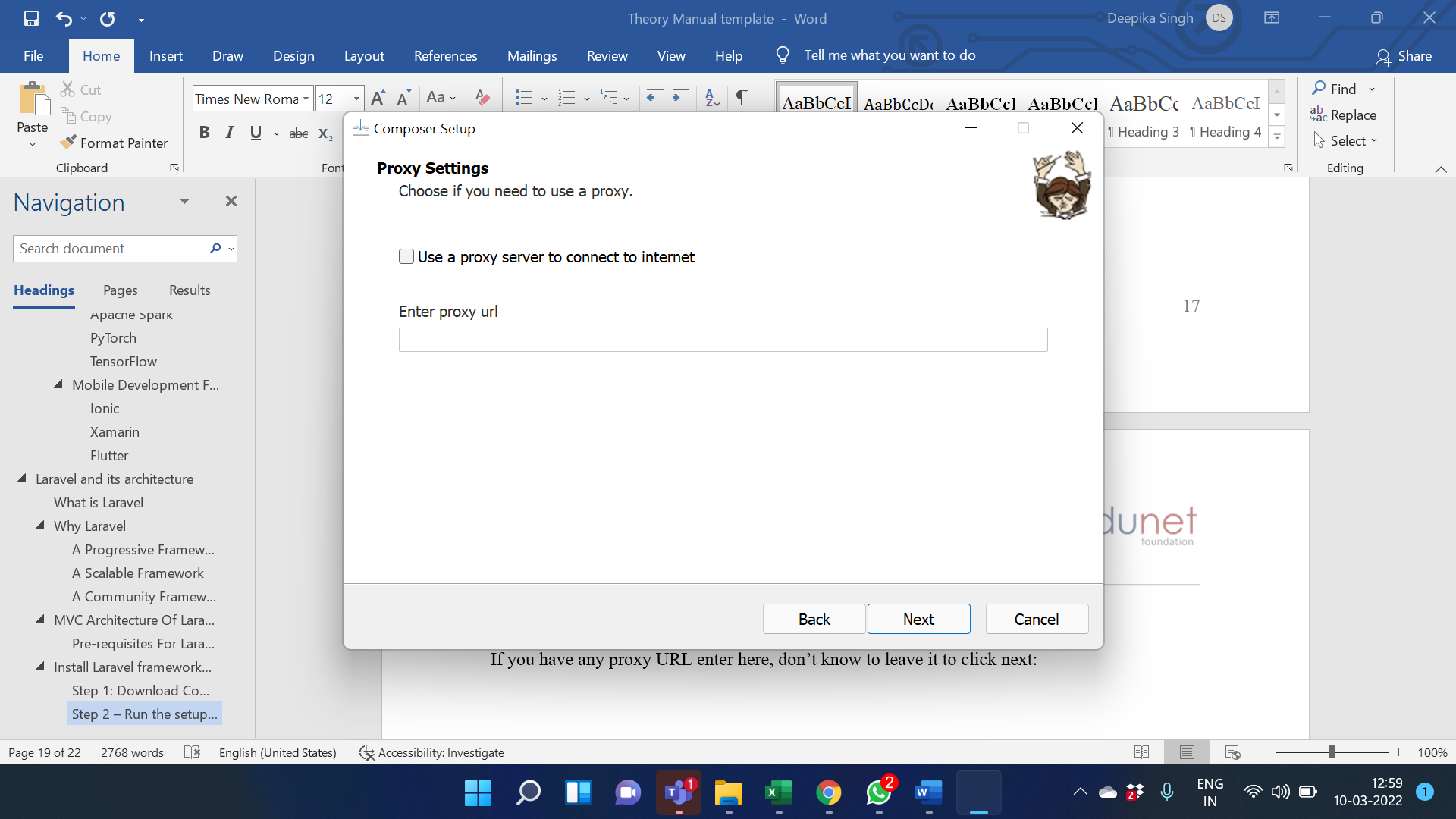
In step 2, click on download composer.exe file. Then open one prompt box :



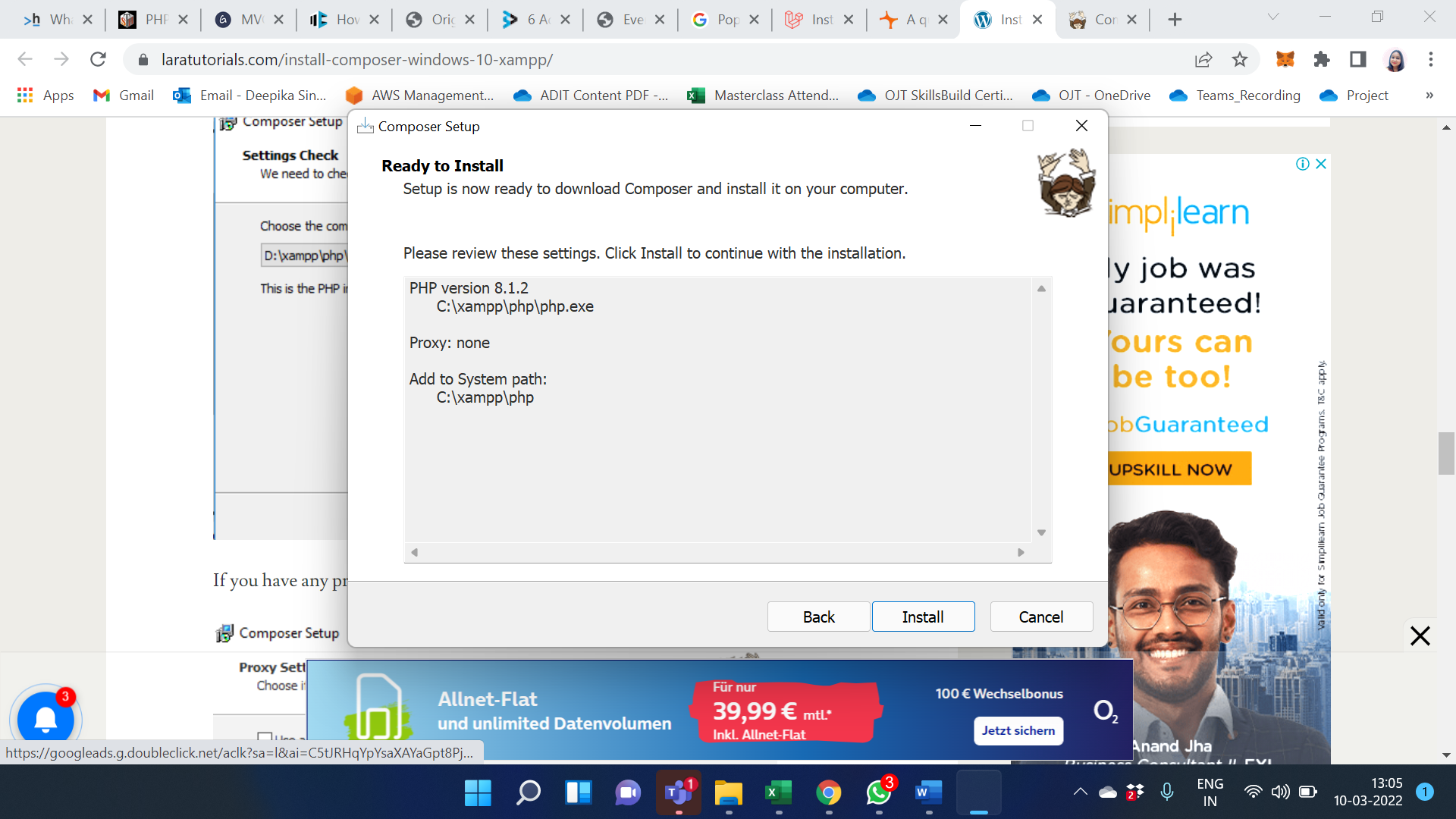
Now Set PHP path and click on next:



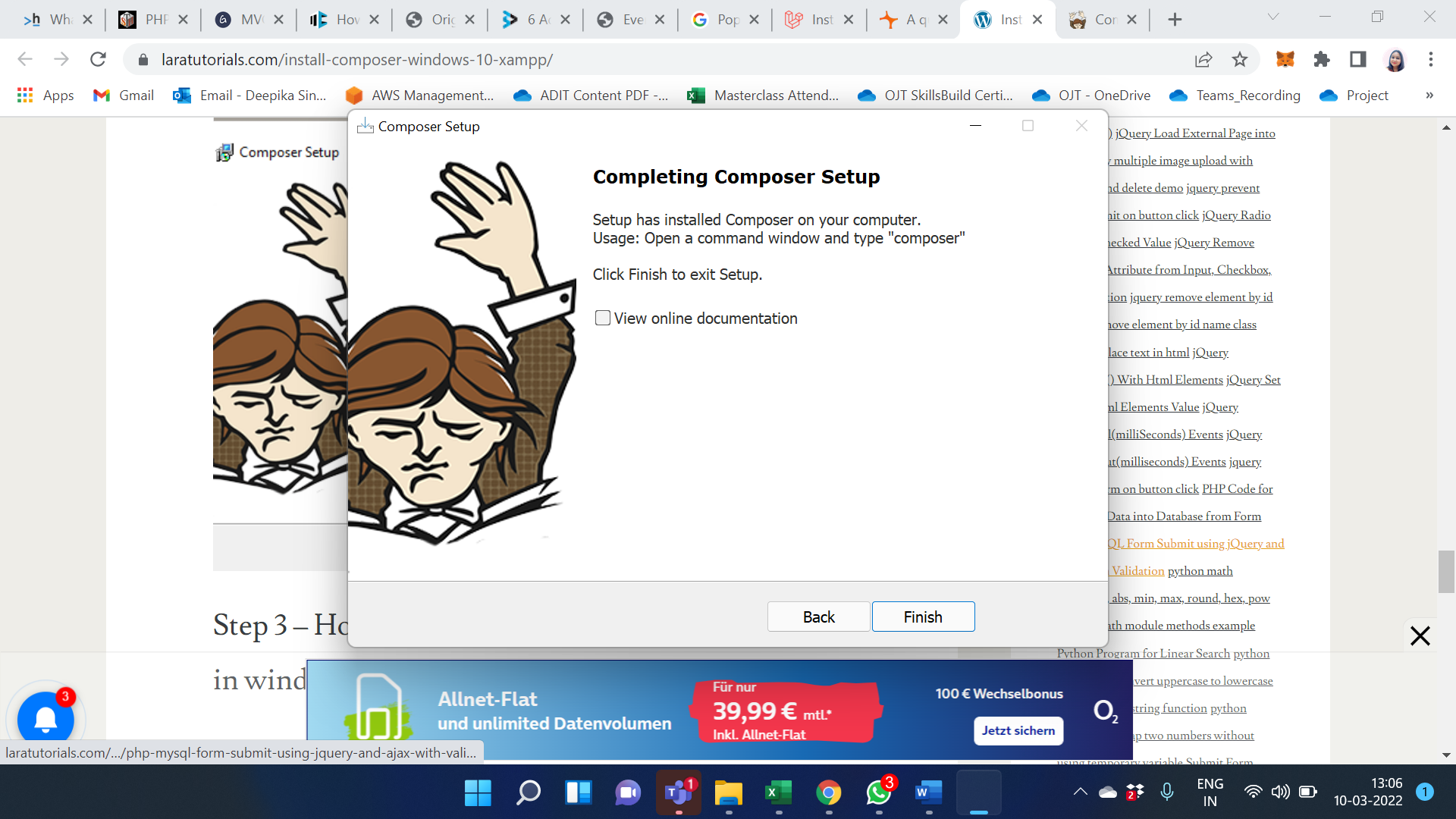
If you have any proxy URL enter here, don’t know to leave it to click next:



Now, review the composer setup wizard. And click on install.

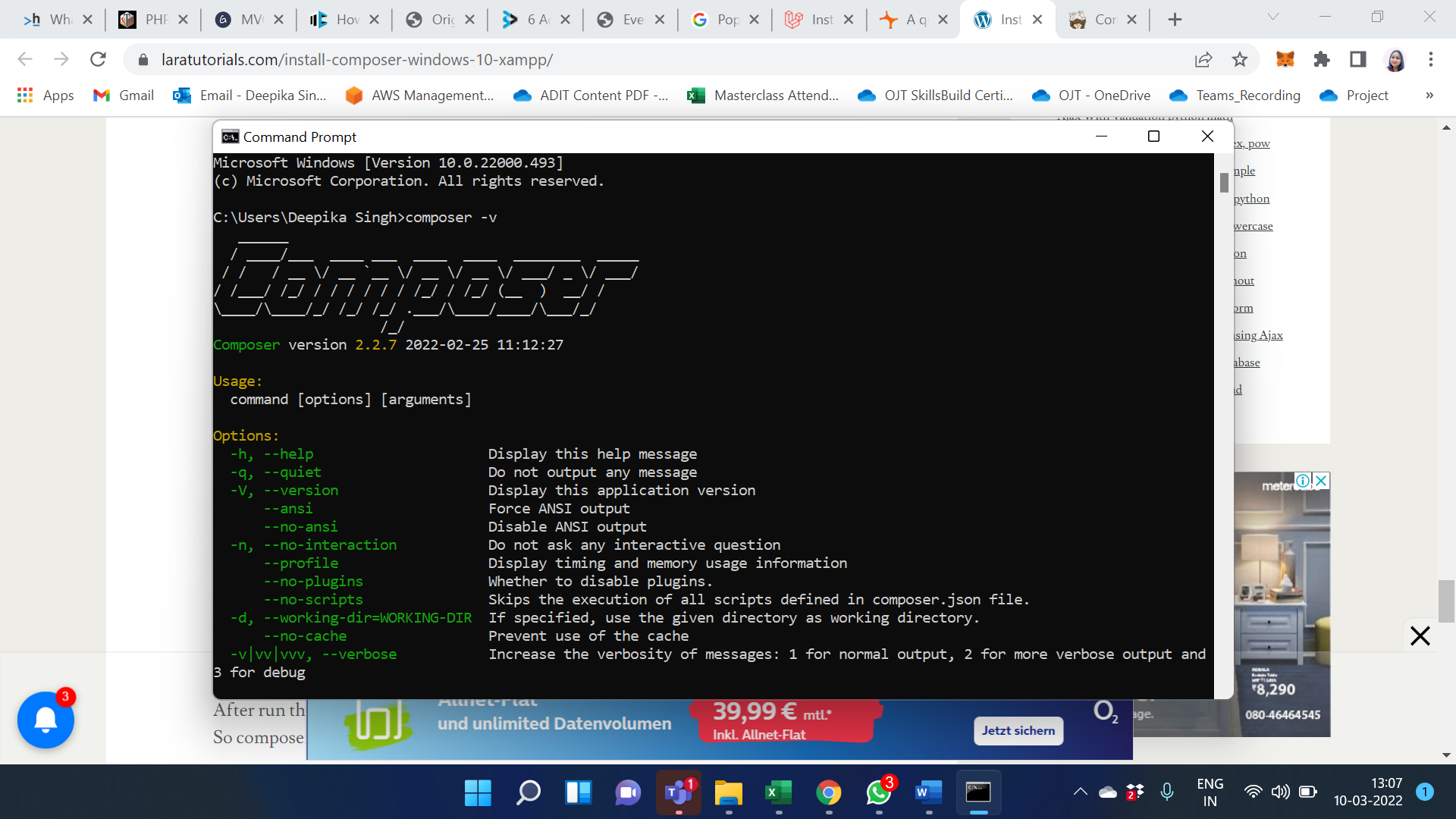


Composer has been installed successfully. Click to finish button.



Step 3 – How to check composer is installed or not in windows

Now, open your terminal and type the following command on command prompt:



After run the above command on cmd. If You will look like the image given below. So composer successfully installed on your windows 10 xampp.

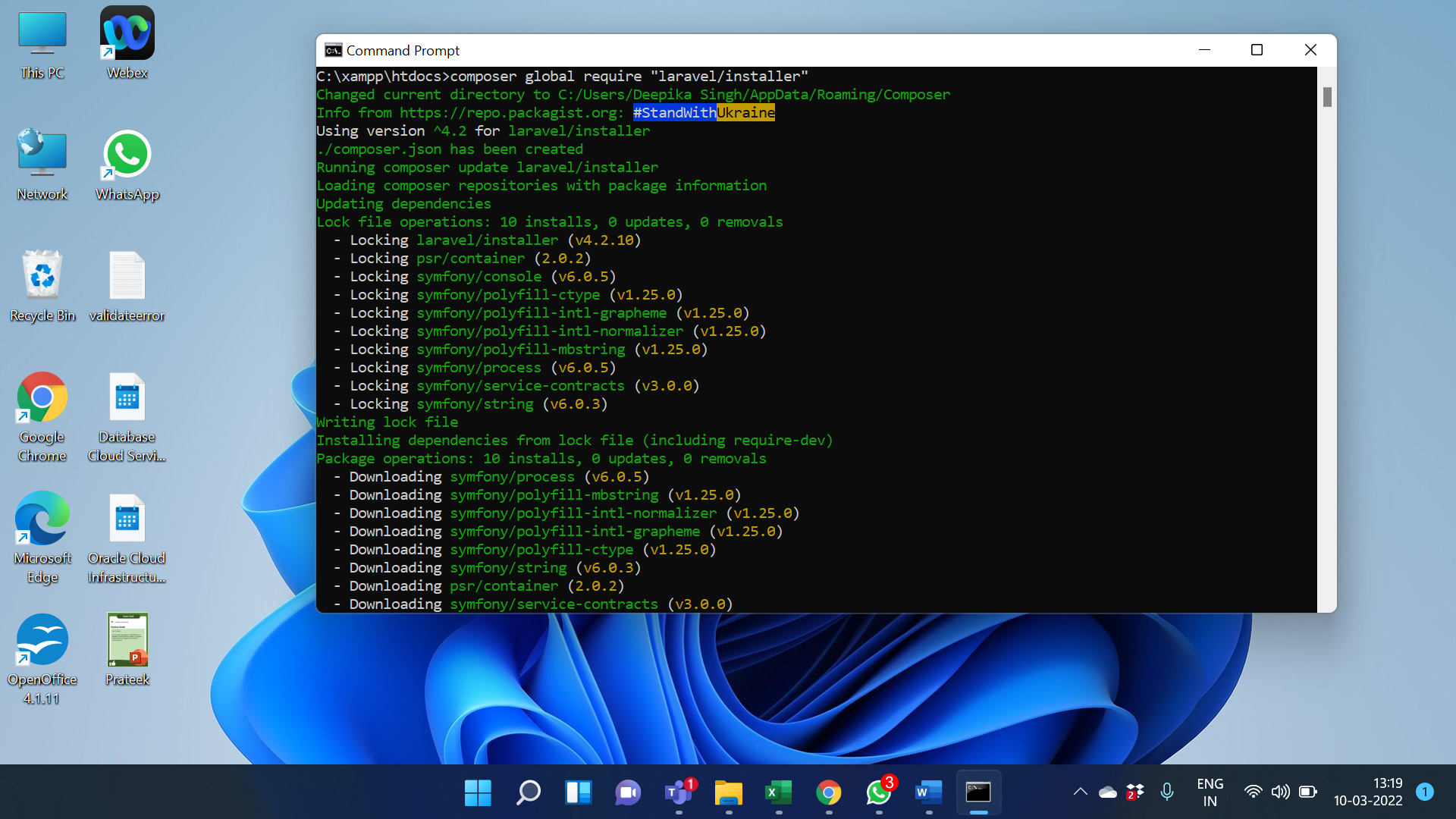
Check the server requirement for the setup:

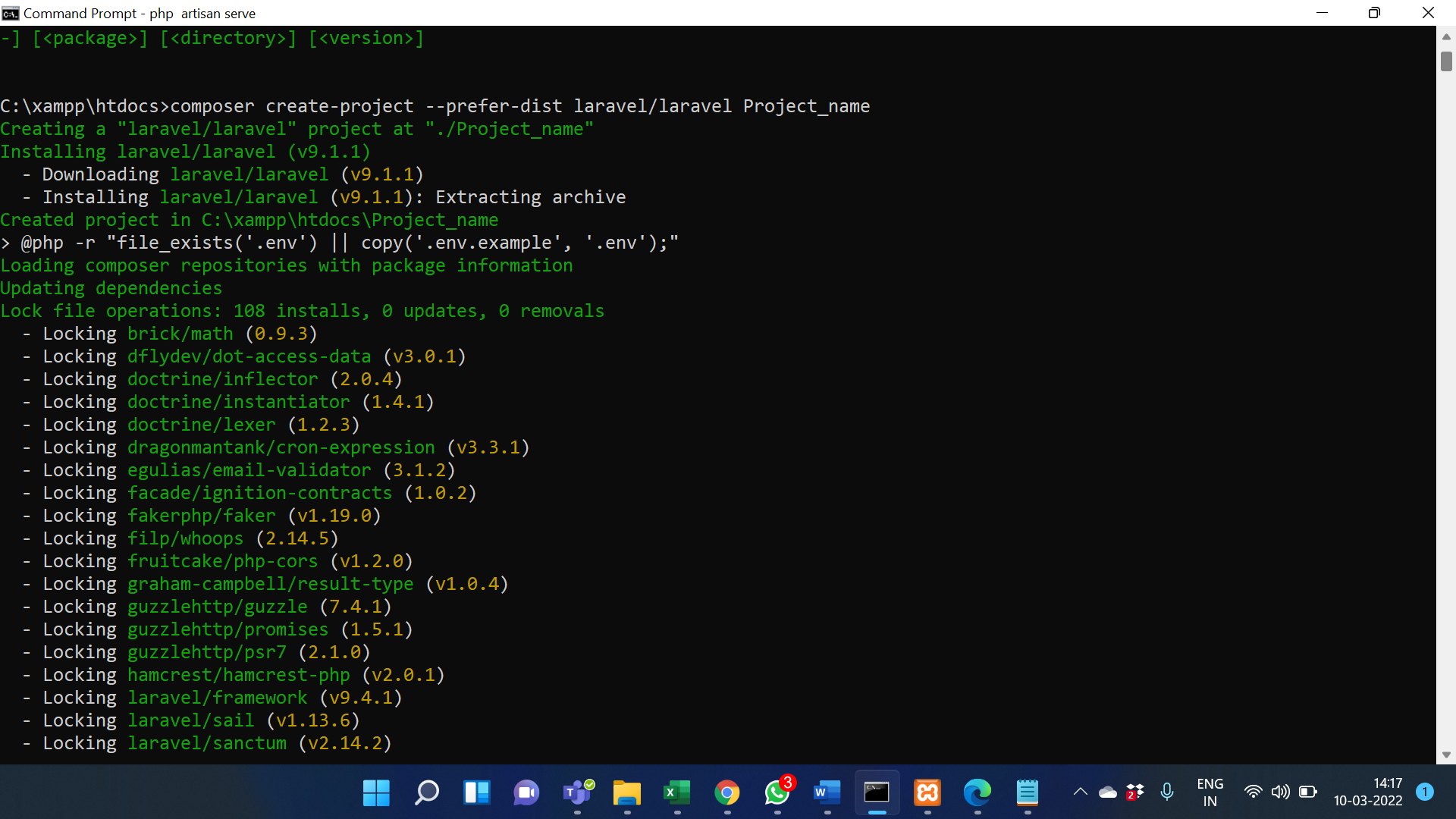
* PHP >= 7.3
* BCMath PHP Extension
* Ctype PHP Extension
* Fileinfo PHP extension
* JSON PHP Extension
* Mbstring PHP Extension
* OpenSSL PHP Extension
* PDO PHP Extension
* Tokenizer PHP Extension
* XML PHP Extension.

Installing Laravel:

Type the following command in your command prompt window:

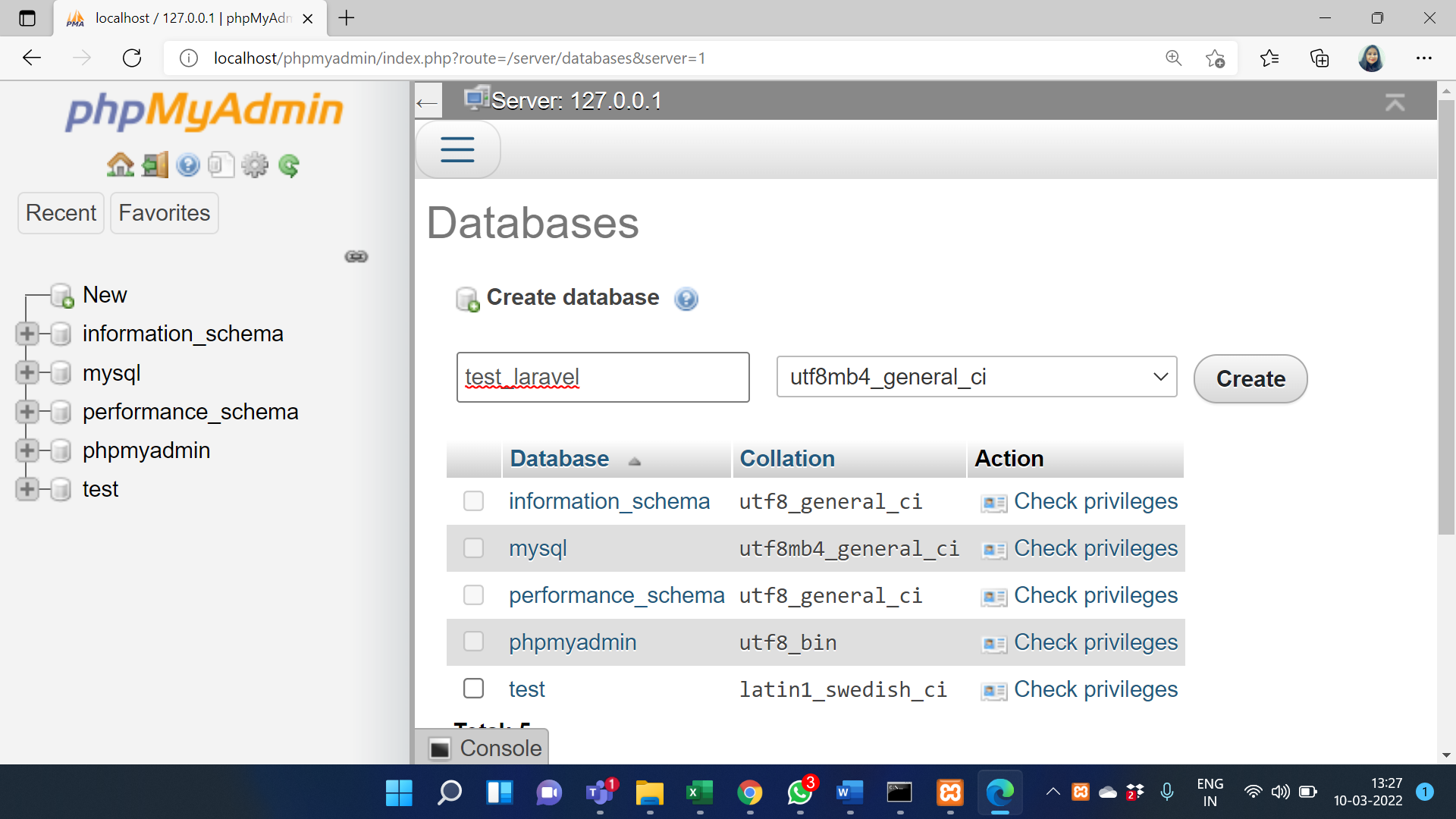
* composer global require “laravel/installer”.
* composer create-project –prefer-dist laravel/laravel Project\_name: this command will install Laravel and other dependencies with it also generate the ANSI key.





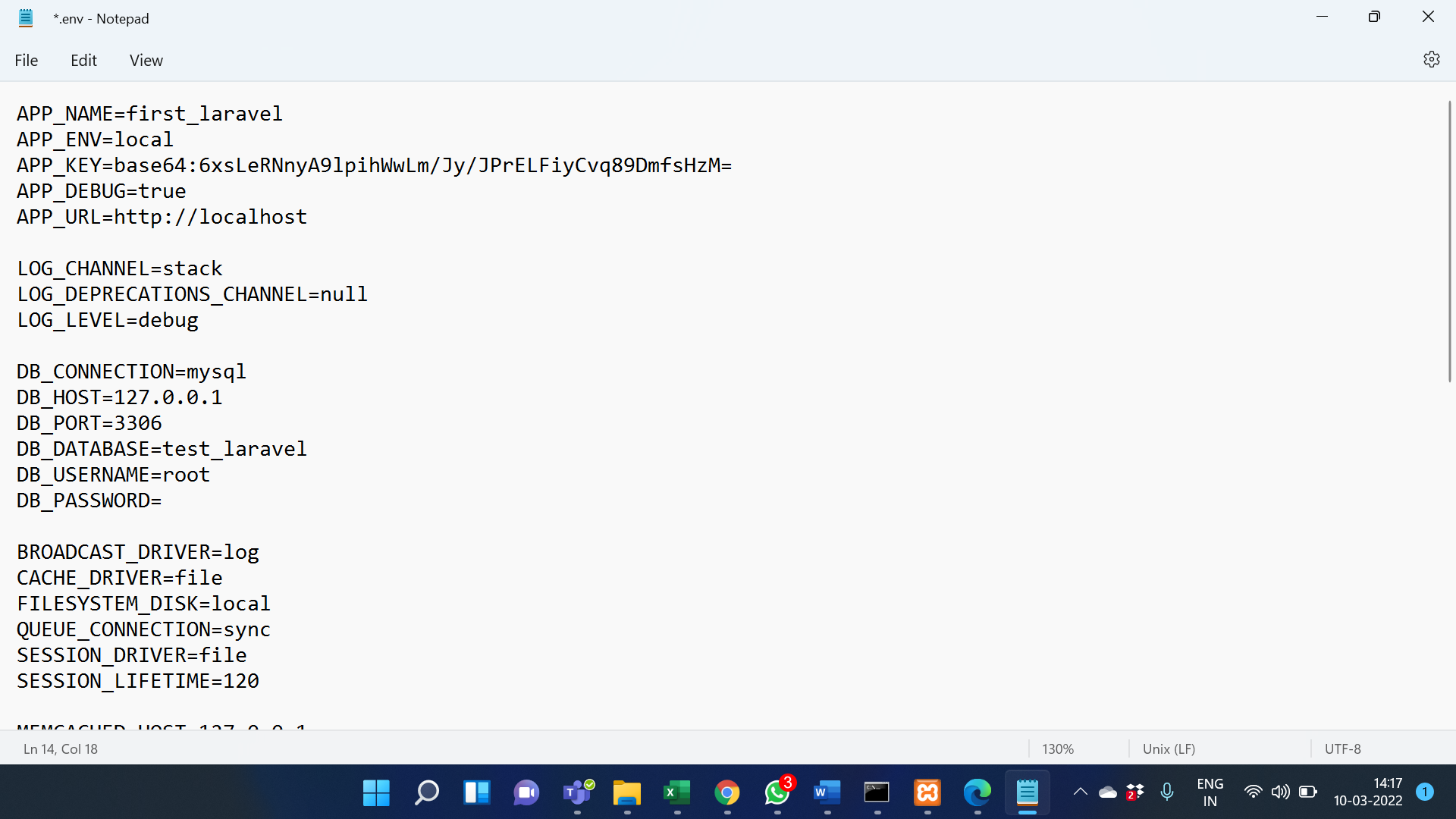
Create Database for Project

* Go to phpMyAdmin click on create a new tab.
* Name the database.
* Press create button.



Update .Env file:

* APP\_NAME=first\_laravel
* APP\_ENV=local
* APP\_KEY=base64:TJ9Sob7KFPhL5XkqT+TyQux3x7UbW08QLb0xtirLWSs=
* APP\_DEBUG=true
* APP\_URL=http://127.0.0.1:8000
* LOG\_CHANNEL=stack
* LOG\_LEVEL=debug
* DB\_CONNECTION=mysql
* DB\_HOST=127.0.0.1
* DB\_PORT=3306
* DB\_DATABASE=test\_laravel
* DB\_USERNAME=root
* DB\_PASSWORD=



Migrate database:

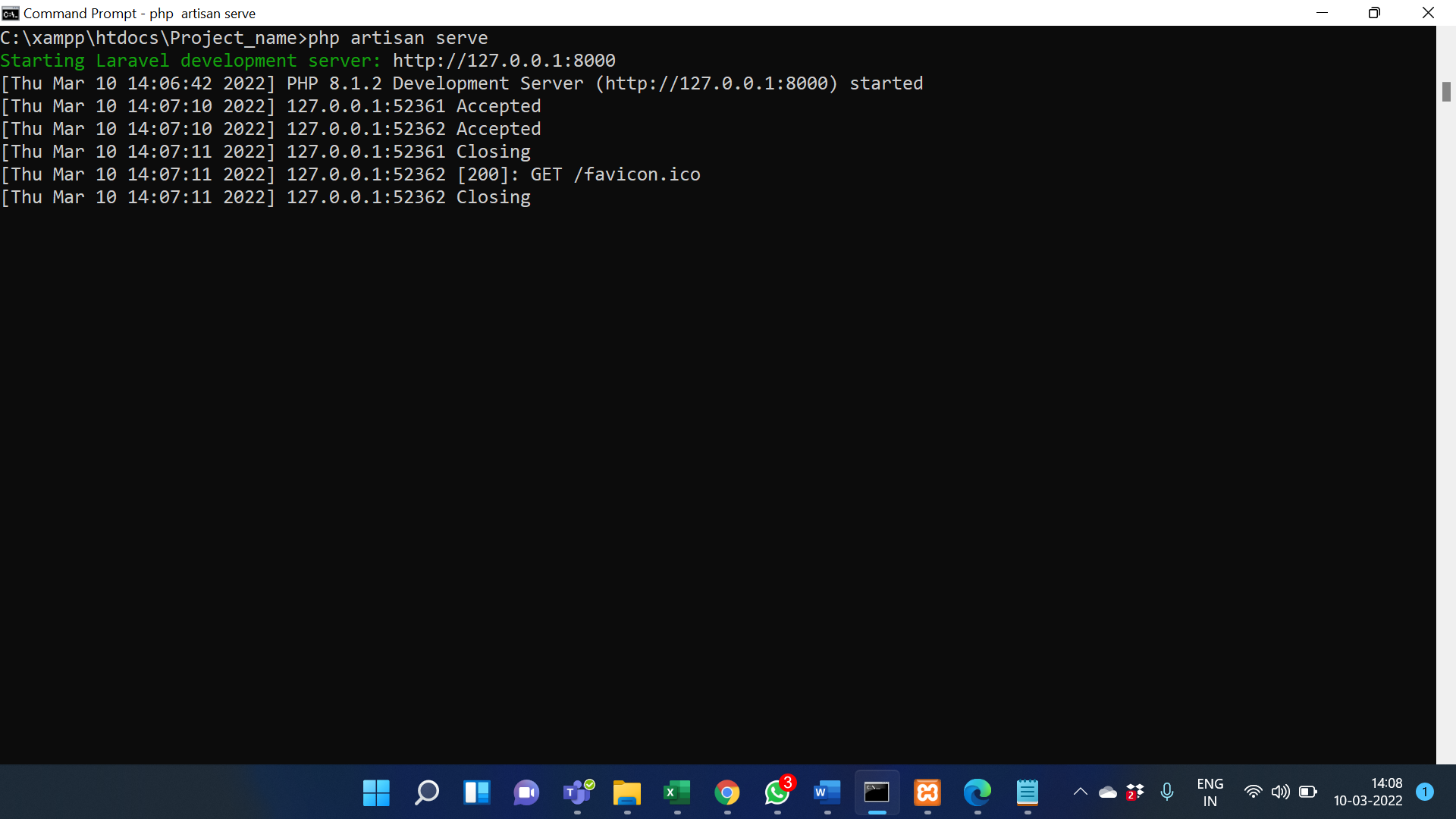
Create tables in the database for Laravel access, also helps in database version control following are the commands for migration.

PHP artisan make migration create\_databse\_table: this command is used to create the DB migration file in your ‘database/migration’ folder.

PHP artisan migrate: used to run the pending migration changes to the database

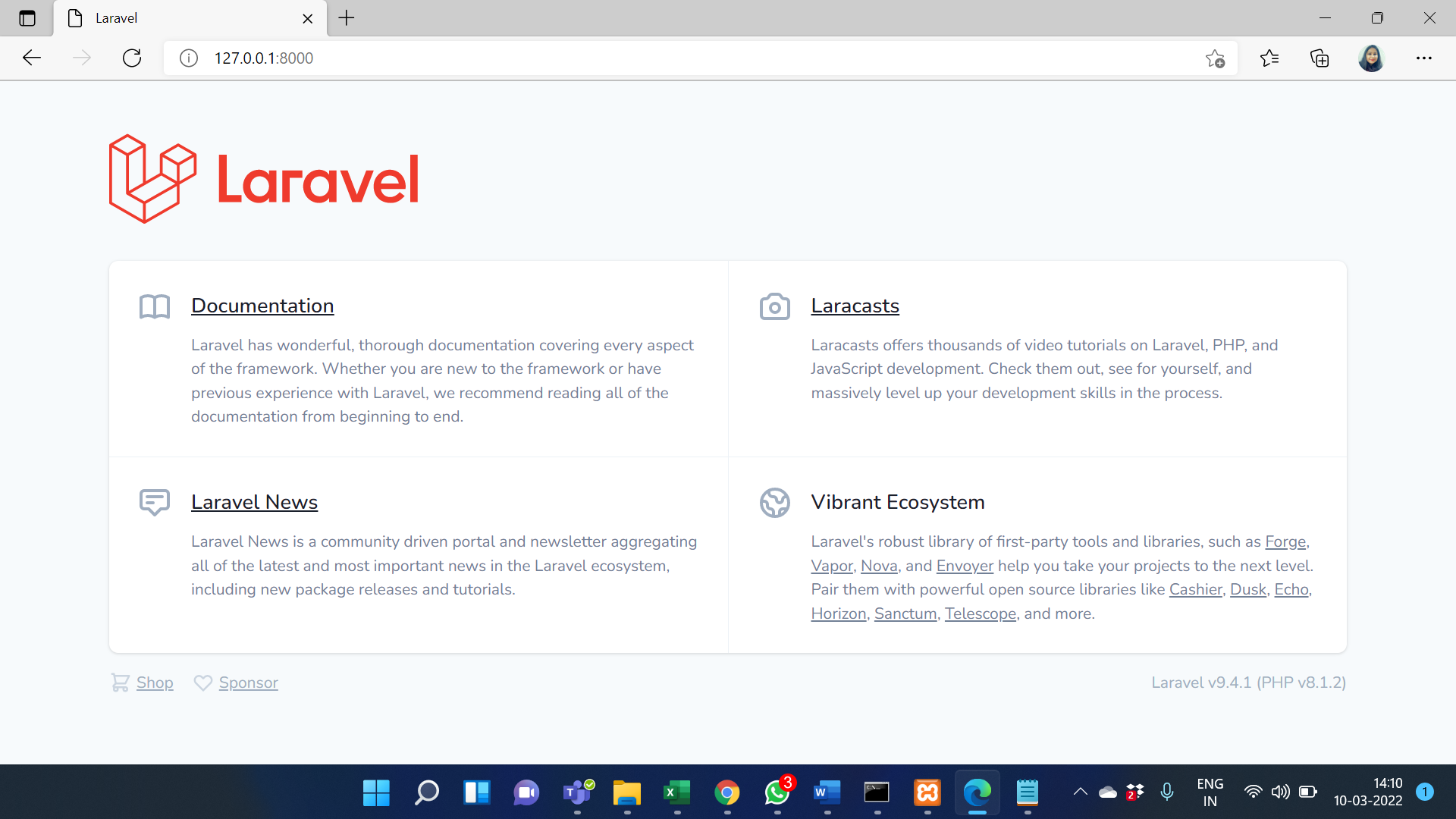
Start development server:

PHP artisan serve: this command starts your development server.



**Output/Results snippet:**

Go to the IP URL that you see on your CMD screen.



**References:**

**<**Include at least 1 reference per activity where students can go and explore. Remove this part from your final document.**>**

* …….
* …….

# 

# Activity 2

## Aim: First program by edit the view as Hello World.

**Learning outcome:** Able to understand Laravel Framework.

**Duration:** 5 hour

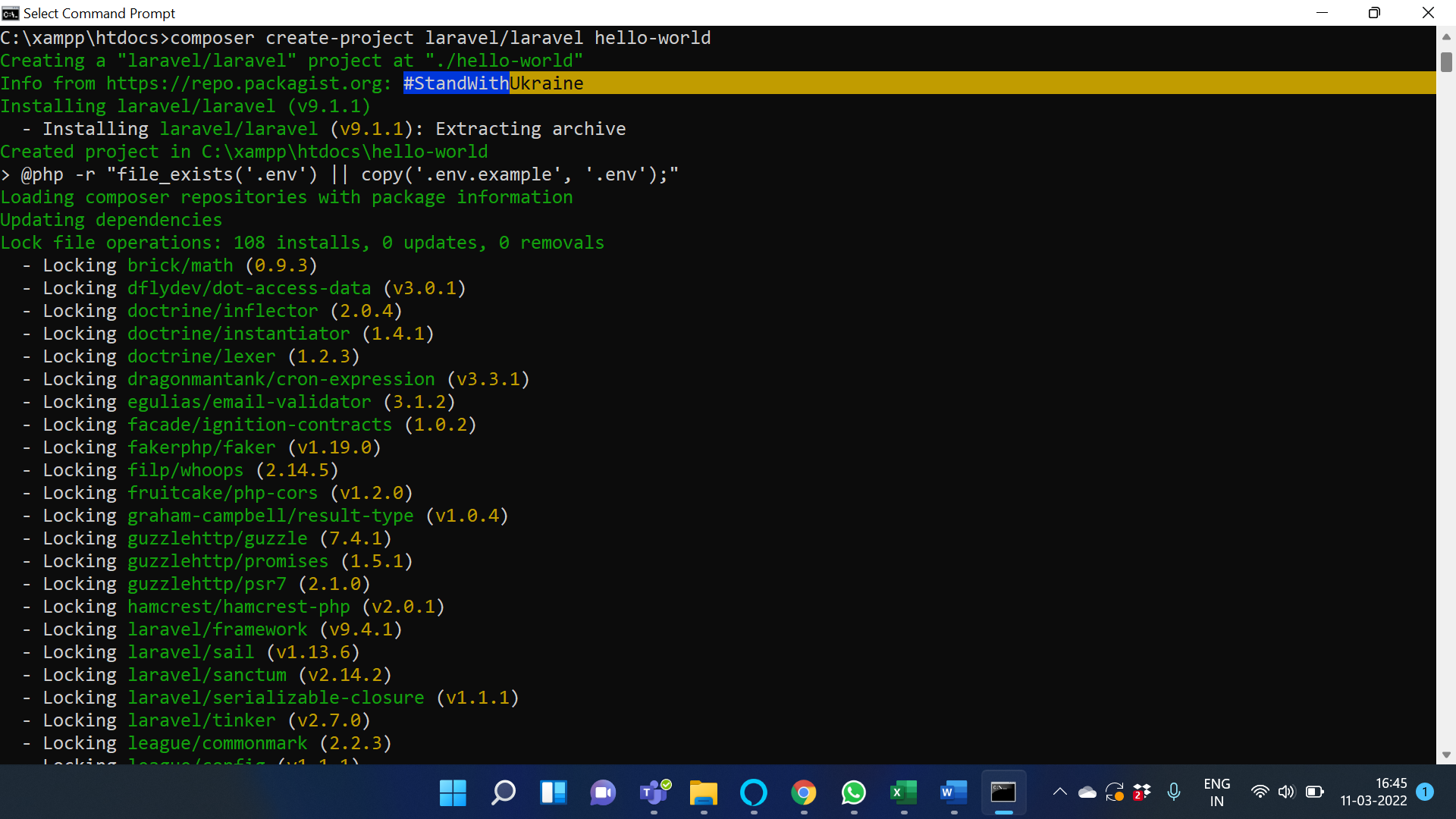
**List of Hardware/Software requirements:**

1. …
2. …...
3. …….

**Code/Program/Procedure (with comments):**

Create a Laravel application:

composer create-project laravel/laravel hello-world

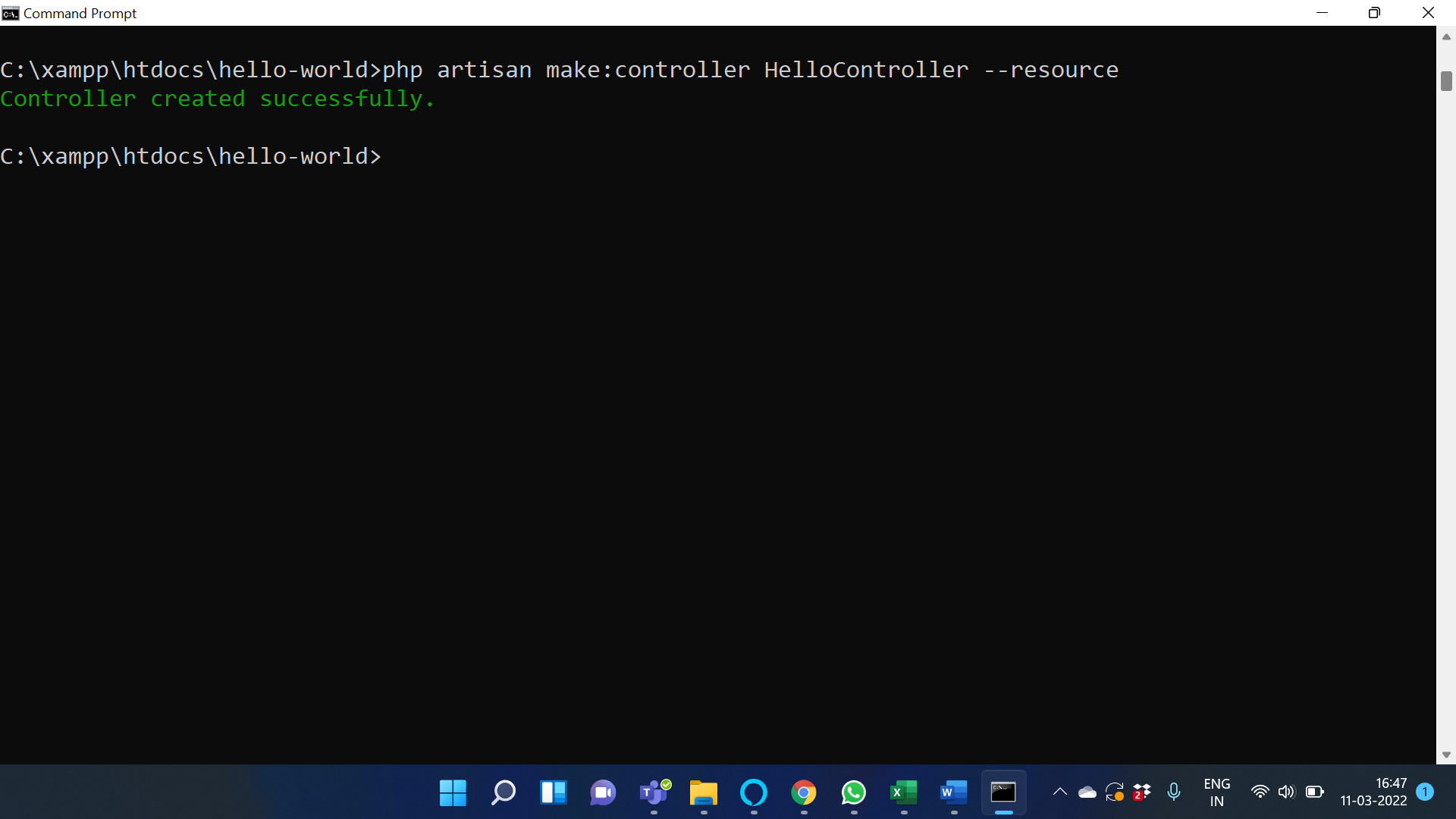


Navigate to the project folder, e.g.

$ cd C:\xampp\htdocs\hello-world

Create a controller:

$ php artisan make:controller HelloController –resource



Step 1: Create a view named index.php in the resources/views directory and save the following code.

<!DOCTYPE html>

<html lang="en">

<body>

<h1>Hello World!</h1>

</body>

</html>

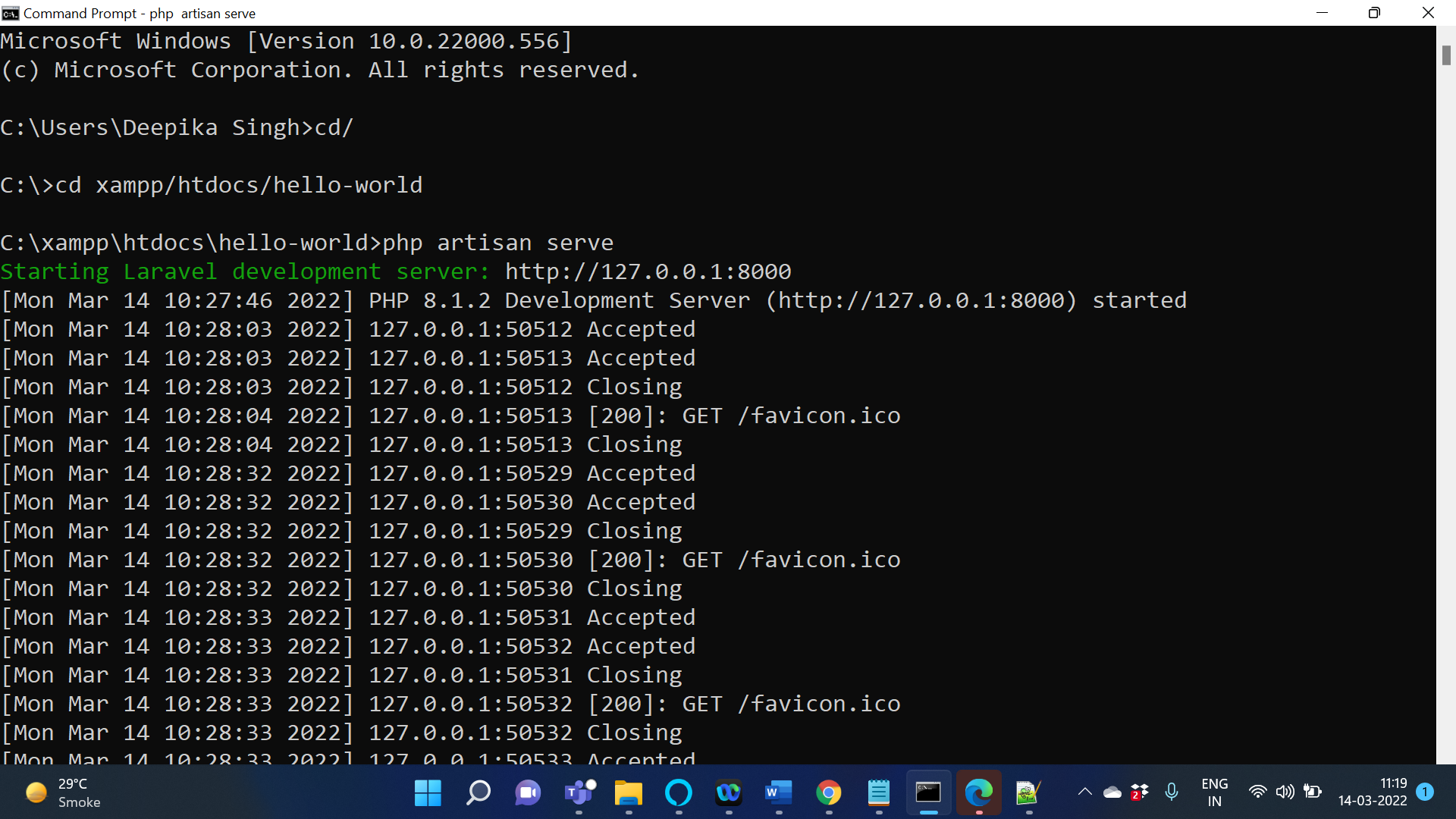
Step 2: Add the following code segment in the routes/web.php file to register a new route.

Route::get('/index', function () {

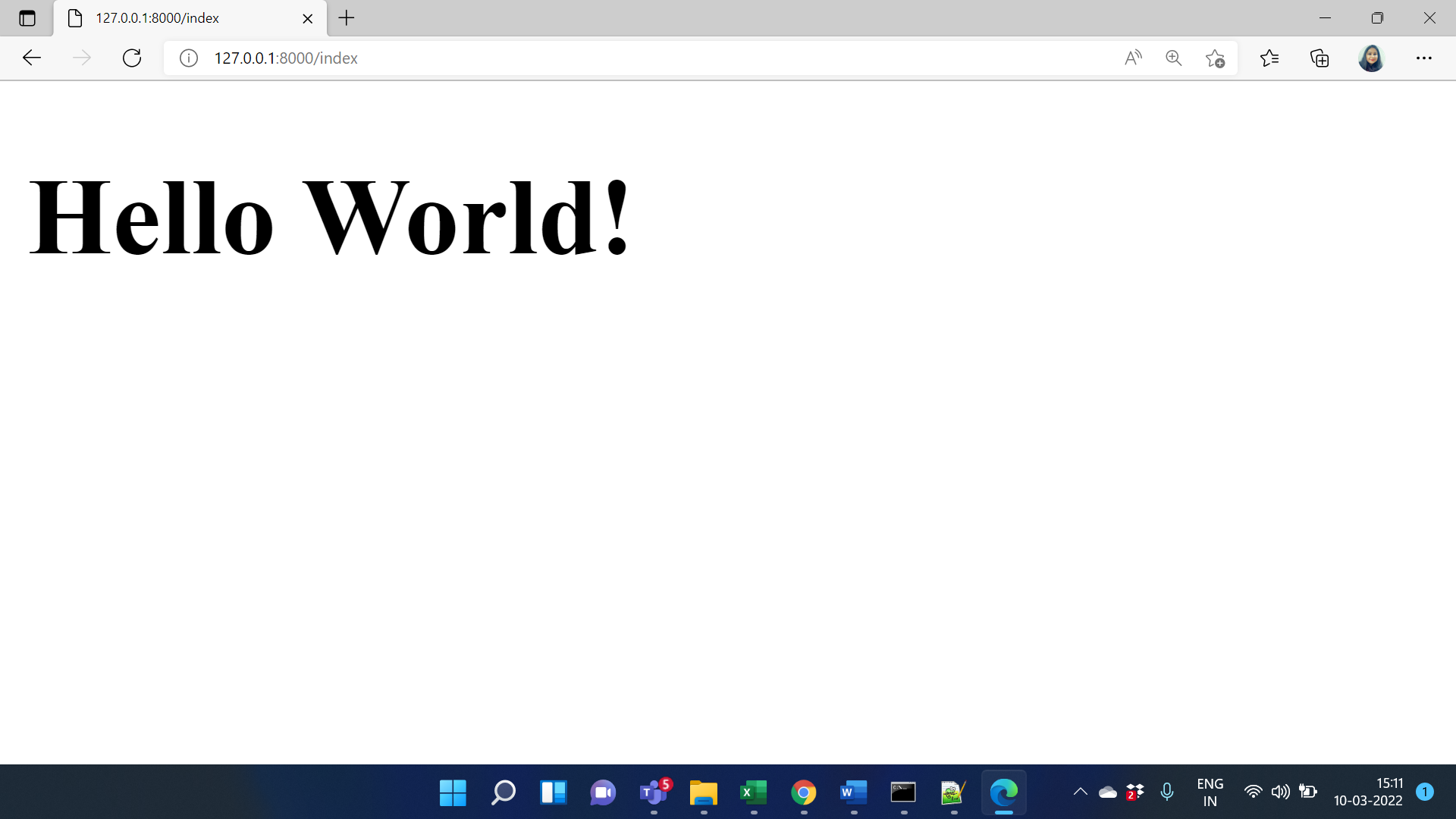
return view('index');

});

Step 3: Visit the URL: http://127.0.0.1:8000/index to see the output.



Step 4: The following screenshot shows the output.



# Activity 3

## Aim: Create a web application of minimum 4 blades demonstrating the blade template in Laravel.

**Learning outcome:** Able to understand Laravel Framework.

**Duration:** 5 hour

**List of Hardware/Software requirements:**

1. …
2. …...
3. …….

**Code/Program/Procedure (with comments):**

First, I will create the routes to gain access to my page. Open routes/web.php and add the following lines in it.

Route::get('/', function()

{

return View::make('pages.home');

});

Route::get('/about', function()

{

return View::make('pages.contact');

});

We are only working on views only, and thus don’t need to access the controllers. To learn more about controllers, take a look at: Working with Controllers and Middleware

**Create Views Structure**

Now that the routes are ready, let’s create the Views structure by creating the following folders and files.

- resources

-- views

--- layouts

------- default.blade.php

--- pages

------- home.blade.php

------- contact.blade.php

--- includes

------- head.blade.php

------- header.blade.php

------- footer.blade.php

Create Includes

Create the following includes, with the following code:

**head.blade.php**

<meta charset="utf-8">

<meta name="description" content="">

<meta name="Saquib" content="Blade">

<title>Checkout our layout</title>

<!-- load bootstrap from a cdn -->

<link rel="stylesheet" href="//netdna.bootstrapcdn.com/twitter-bootstrap/3.0.3/css/bootstrap-combined.min.css">

**Header.blade.php**

<div class="navbar">

<div class="navbar-inner">

<a id="logo" href="/">Single Malt</a>

<ul class="nav">

<li><a href="/">Home</a></li>

<li><a href="/contact">Contact</a></li>

</ul>

</div>

</div>

**footer.blade.php**

<div id="copyright text-right">© Copyright 2017 Saquib Rizwan </div>

**Create the Layout**

I will use @include to bring in tiny parts of the code that I have created in includes folders, and @yield to bring in content from the individual pages I will be using.

<!doctype html>

<html>

<head>

@include('includes.head')

</head>

<body>

<div class="container">

<header class="row">

@include('includes.header')

</header>

<div id="main" class="row">

@yield('content')

</div>

<footer class="row">

@include('includes.footer')

</footer>

</div>

</body>

</html>

Blade allows the use of the layout that I just created by using @extends. By creating @section, I will create a section that will be used in the layout. Here I will use @section(‘content’) and in the layout, everything that I will type here will be injected in @yield in the layout.

Go to resources/views/pages and put the following code in these files.

**pages/home.blade.php**

@extends('layouts.default')

@section('content')

i am the home page

@stop

**pages/contact.blade.php**

@extends('layouts.default')

@section('content')

i am the contact page

@stop

# Activity 4

## Aim: Create a Web Application which demonstrate routing through controller.

**Learning outcome:** Able to understand Laravel Framework.

**Duration:** 5 hour

**List of Hardware/Software requirements:**

1. …
2. …...
3. …….

**Code/Program/Procedure (with comments):**

Here are some of the steps for creating laravel routing controllers which are explained below:

**Step 1**: The very first step would be to create a controller. If you are not familiar with creating a controller, then go through the below points of creating a controller otherwise move directly to step 2 for Routing Controllers.

Use the below artisan command to create the controller.

**Code:**

Php artisan make: Controller MyController

MyController.php file will be created whose default code is as below.

**Code:**

<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;

class MyController extends Controller

{

public function show($id)

{

//

}

}

**Step 2**: Now you have to write this below route in web.php file.

**Code:**

Route::get('/post','MyController@show');

Here the first parameter is URL which you want access to, and MyController is pretty obviously our controller name. The ‘show’ as you can see in MyController.php file, is the method. So, @show here indicates that the show() method would be called when we hit the URL ‘/post’.

**Step 3**: You can now add coding lines as shown below.

**Code:**

<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;

class MyController extends Controller

{

/\*\*

\*Display resource

\*

\*/

public function show($id)

{

//

}

/\*\*

\*Display resource listing

\*

\*/

public function index()

{

//

}

/\*\*

\*Editing resource

\*

\*/

public function edit($id)

{

//

}

}

Step 4: Now it’s time to hit the URL. You will get a specified output after entering the URL. Hopefully, we have covered enough insights of controllers that you will be able to access your Controller now.