**Introduction to Angular**

Typescript is a open source javascript compiler.

It’s a strongly-typed javascript

Superset of javascript

It has support by Angular

Angular 2 is written in a superset of Javascript called Typescript.

Typescript is the ES6 version of Javascript plus a few other Typescript only features which Angular 2 needs in order to work.

You can write Angular 2 applications in either *Typescript*, *ES6* or even *ES5 Javascript*.

However Angular 2 itself is written in Typescript,

Browsers don’t support Typescript. Browsers barely support ES6 Javascript. So how can we write our code in Typescript?

We use something called a transpiler which converts from one language to another.

We can write in Typescript and have a transpiler convert to ES6 or ES5.

#### Tip

To understand why is it called transpilation and not compilation in more detail see <https://www.stevefenton.co.uk/2012/11/compiling-vs-transpiling/>

Since most browsers don’t support ES6 features yet we are going to transpile our Typescript into ES5.

If we look at the tsconfig.json file in our project, we can see there are a few settings we are using to convert Typescript into Javascript.

{

"compilerOptions": {

"target": "es5", (1)

"module": "commonjs",

"moduleResolution": "node",å

"sourceMap": true,

"emitDecoratorMetadata": true,

"experimentalDecorators": true,

"removeComments": false,

"noImplicitAny": true,

"suppressImplicitAnyIndexErrors": true

}

}

* Typescript is just javascript with a few more advanced features.
* Browser can’t run Typescript so we first need to Transpile it into Javascript.
* The most common version of javascript is currently ES5 so we transpile Typescript into ES5 Javascript.

# Compiling vs Transpiling

Despite the fact that the term “transpiling” has been around since last century, there appears to be a fair bit of confusion about what it means and what the difference between transpiling and compiling is.

Firstly, transpiling is a specific kind of compiling.

**Compiling**

is the general term for taking source code written in one language and transforming into another

**Transpiling**

is a specific term for taking source code written in one language and transforming into another language that has a similar level of abstraction

When you compile C#, your method bodies are transformed by the compiler into IL. This cannot be called transpiling because the two languages are very different levels of abstraction.

When you compile TypeScript, it is transformed by the compiler into JavaScript. These are very similar levels of abstraction, so you could call this transpiling.

Both compilers and transpilers can optimise the code as part of the process.

Angular

Angular 2 is an open source JavaScript framework to build web applications in HTML and JavaScript

Angular is an open source framework built over JavaScript.

It was built by the developers at Google.

This framework was used to overcome obstacles encountered while working with Single Page applications.

Also, testing was considered as a key aspect while building the framework. It was ensured that the framework could be easily tested.

The initial release of the framework was in October 2010.

Latest version of Angular is 8

Features of Angular 2

Following are the key features of Angular 2

* **Components** − The earlier version of Angular had a focus of Controllers but now has changed the focus to having components over controllers. Components help to build the applications into many modules. This helps in better maintaining the application over a period of time.
* **TypeScript** − The newer version of Angular is based on TypeScript. This is a superset of JavaScript and is maintained by Microsoft.
* **Services** − Services are a set of code that can be shared by different components of an application. So for example if you had a data component that picked data from a database, you could have it as a shared service that could be used across multiple applications.

## Components of Angular 2

Angular 2 has the following components −

* **Modules** − This is used to break up the application into logical pieces of code. Each piece of code or module is designed to perform a single task.

**Component** − This can be used to bring the modules together.

**Templates** − This is used to define the views of an Angular JS application.

**Metadata** − This can be used to add more data to an Angular JS class.

**Service** − This is used to create components which can be shared across the entire application.

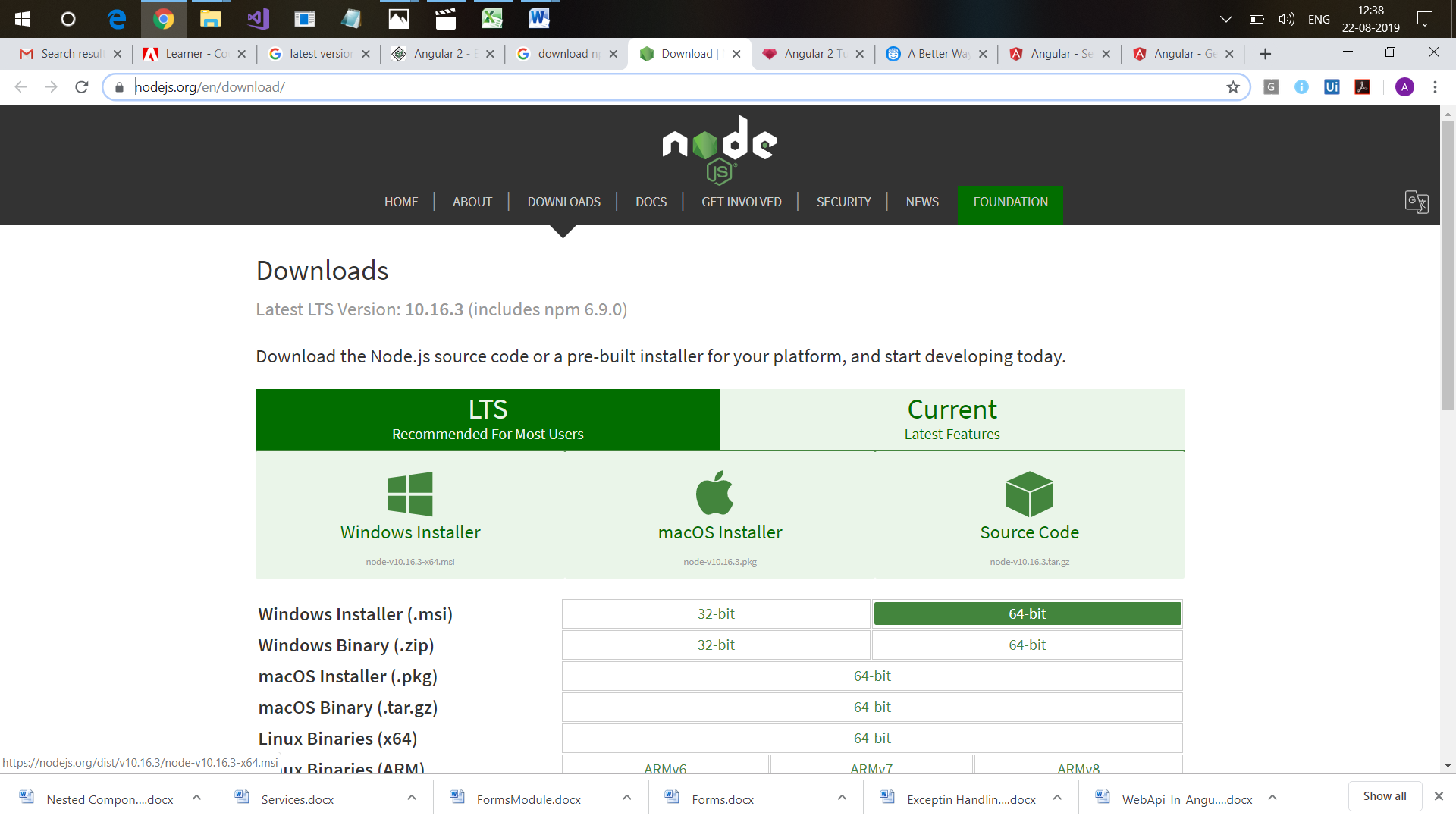
**To start working with Angular 2, you need to get the following key components installed.**

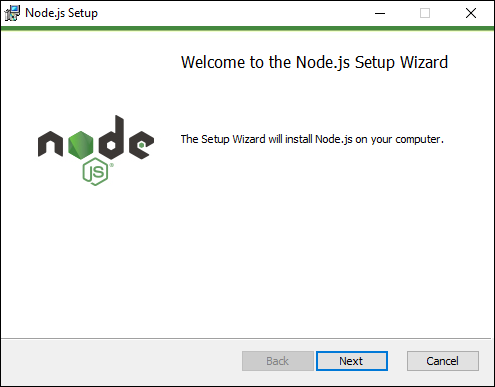
* **Npm** − This is known as the node package manager that is used to work with the open source repositories. Angular JS as a framework has dependencies on other components. And **npm** can be used to download these dependencies and attach them to your project.
* **Git** − This is the source code software that can be used to get the sample application from the **github** angular site.
* **Editor** − There are many editors that can be used for Angular JS development such as Visual Studio code and WebStorm. In our tutorial, we will use Visual Studio code which comes free of cost from Microsoft.

## npm Installation

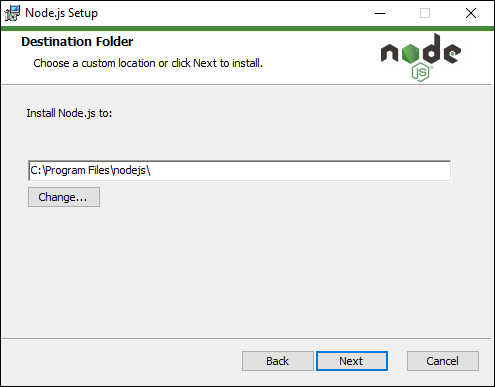
The official site for npm is <https://www.npmjs.com/>

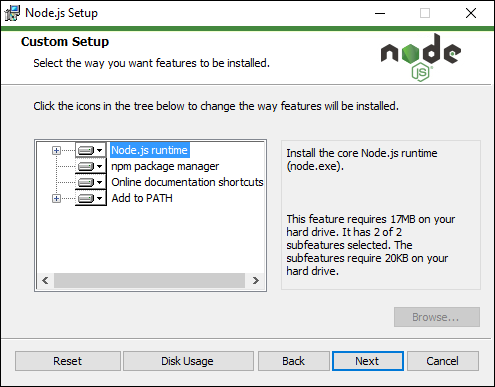
<https://nodejs.org/en/download/>

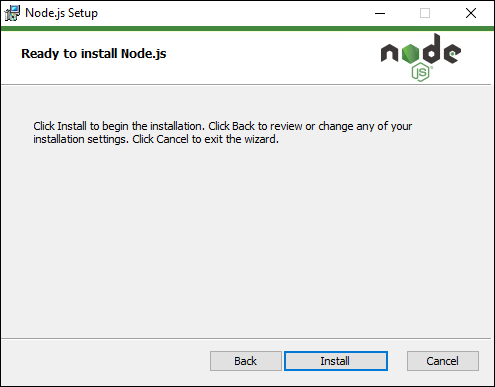


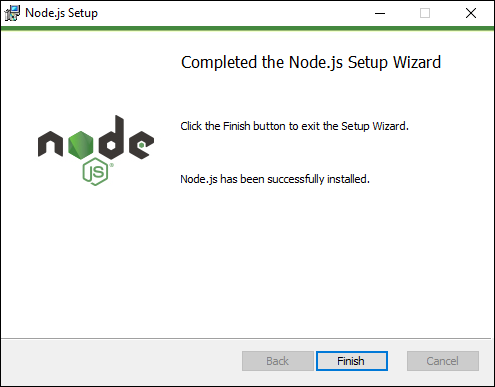






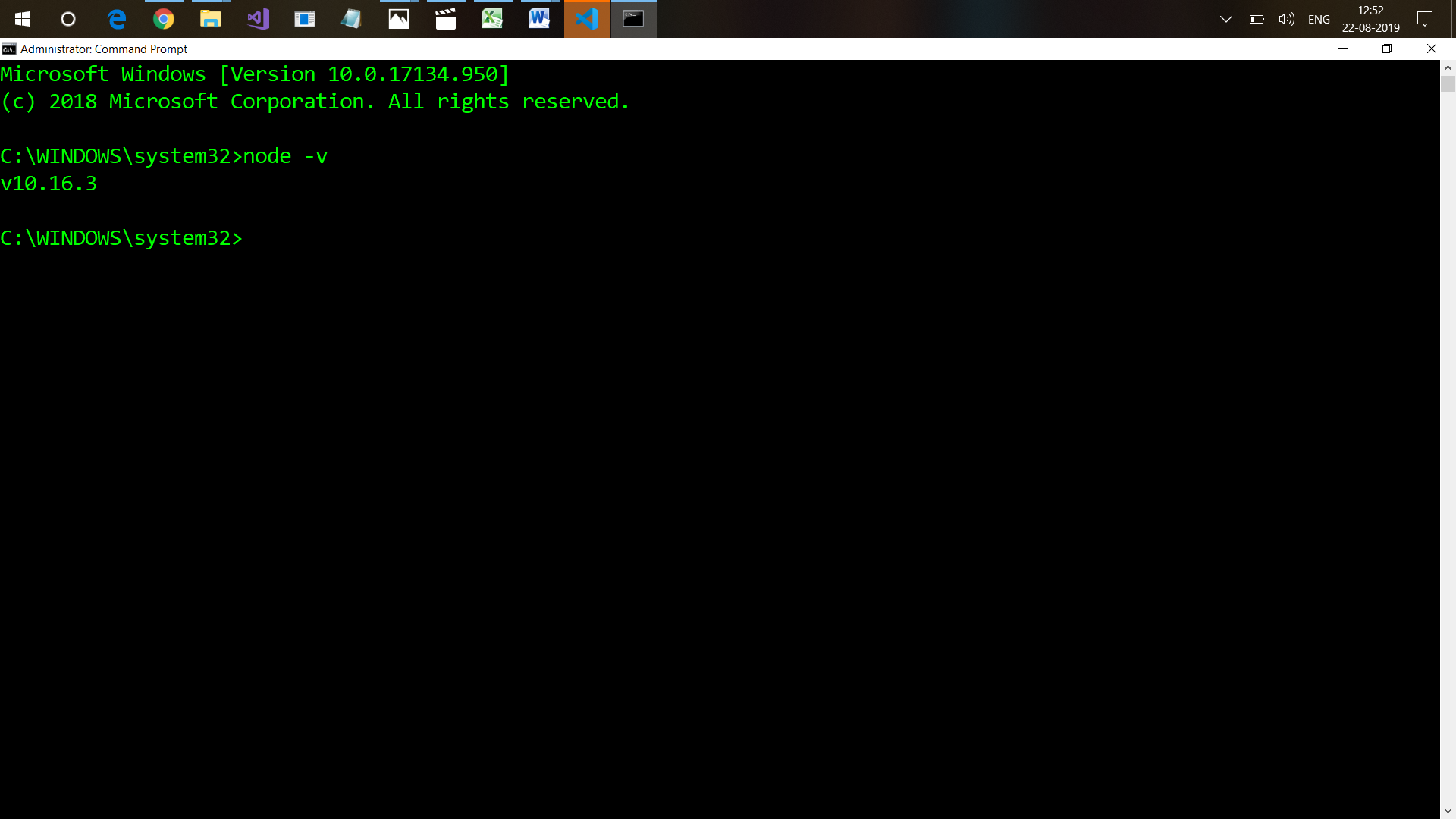






Once we get nodejs, npm also gets installed

To check node is installed or not



And to check npm,



Now we can install Angular

Check the link <https://cli.angular.io/>

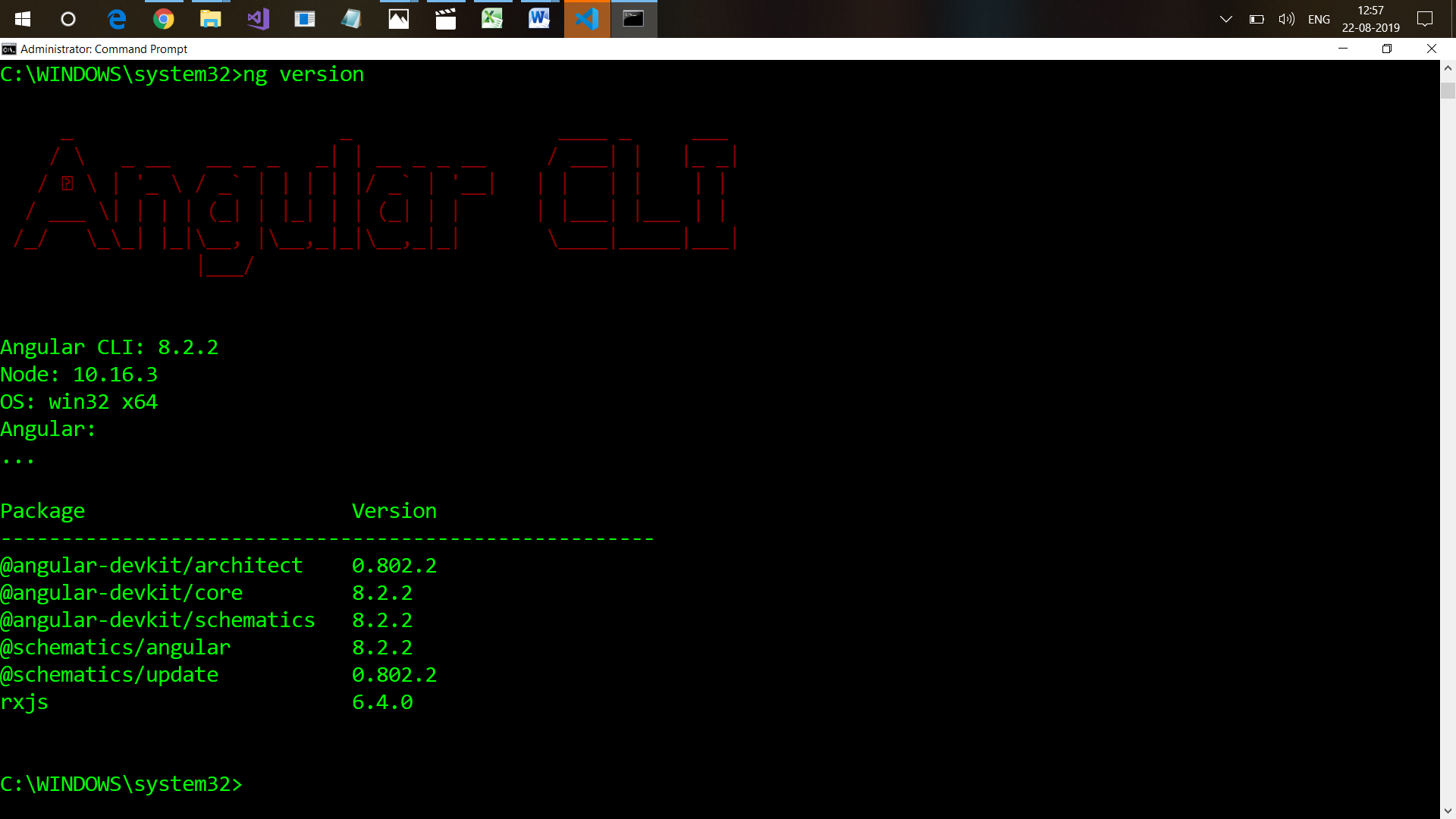
npm install –g @angular/cli -g means Angular CLI has been installed globally on the system. It means that I don’t need to install angular cli in every project

So now Angular Cli is installed on my system and every new project that we create has angular cli

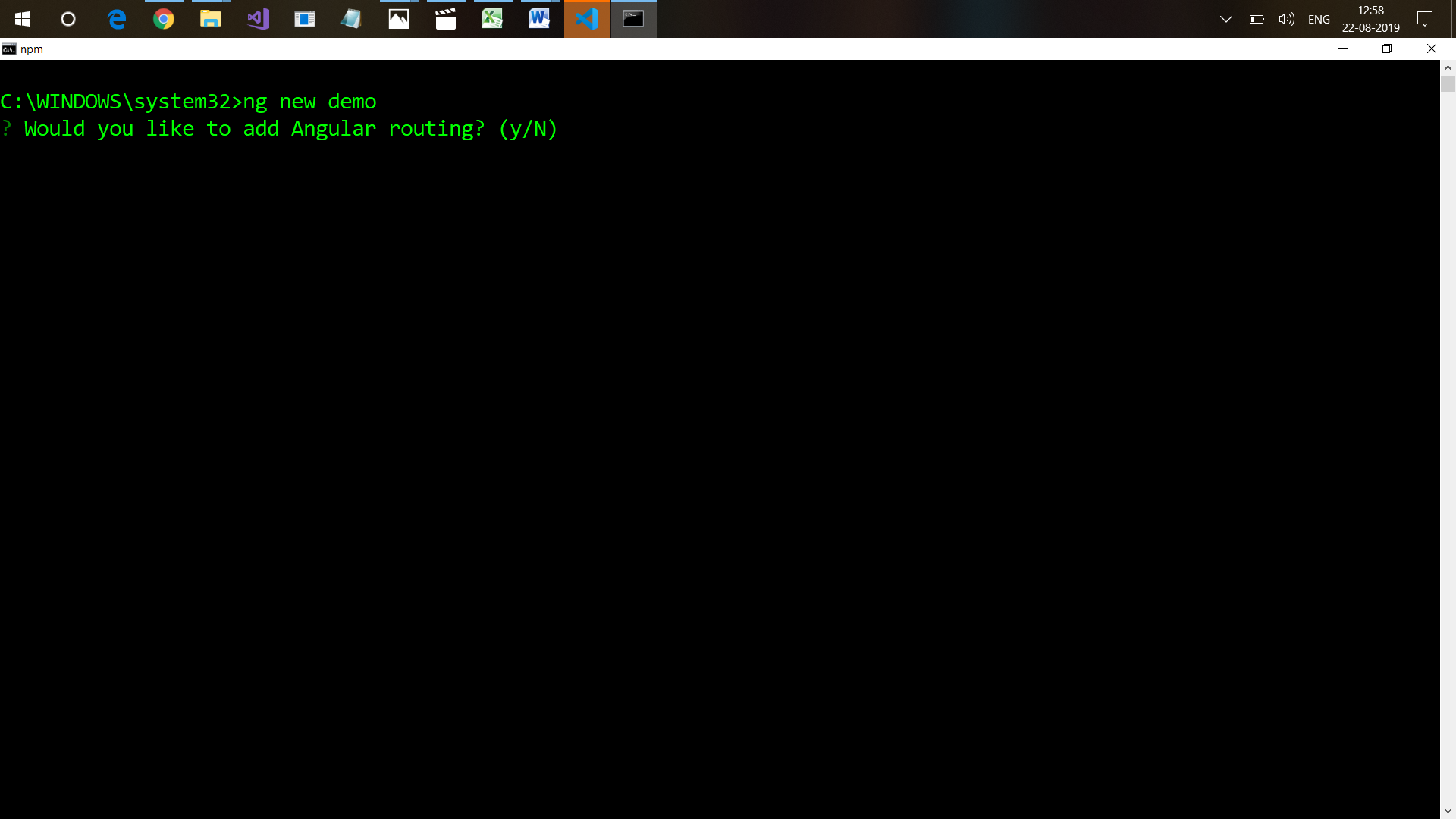
If you do not use –g, angular cli will be installed in the specific folder that you have opened in your terminal cmd.

It will take some time

To check , whether angular cli is installed in my system successfully,



Angular CLI is installed…. Now we are good to work with Angular 4



## 

## 

## 

It will create a new project

## It will create all the required packages for running the application.. It will create a project with the name demo in demo folder.

## So, we get three folders , e2e, node\_modules,src

## To open it in Visual Studio Code

## 

## And it will open project in Visual Studio Code

## 

## Understand project structure

## Node\_modules contains all the packages needed for application

Src directory : In this directory, we are going to write all our angular4 code for our application

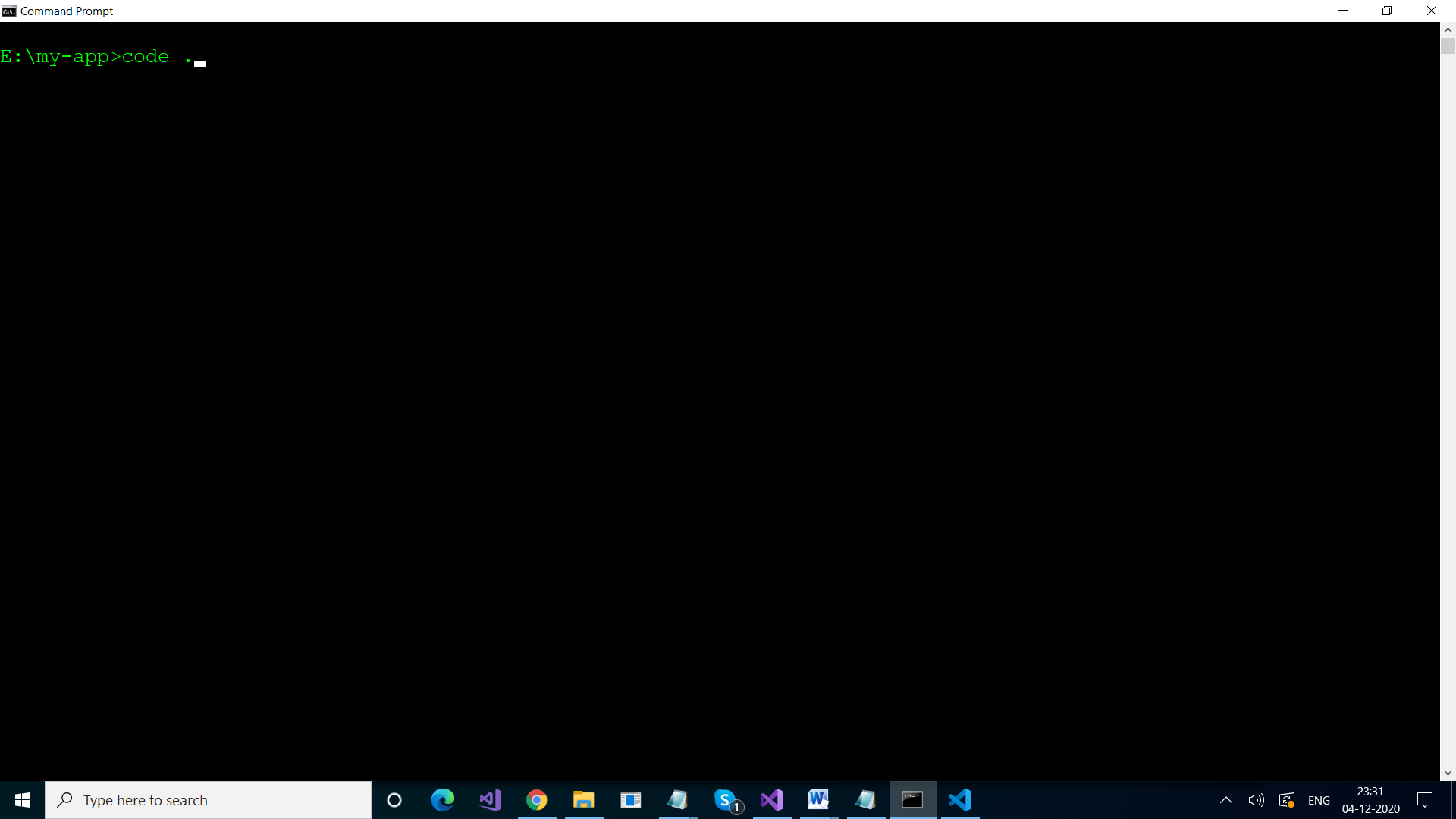
Karma.conf.js is used for writing unit test

Package.json for managing the libraries

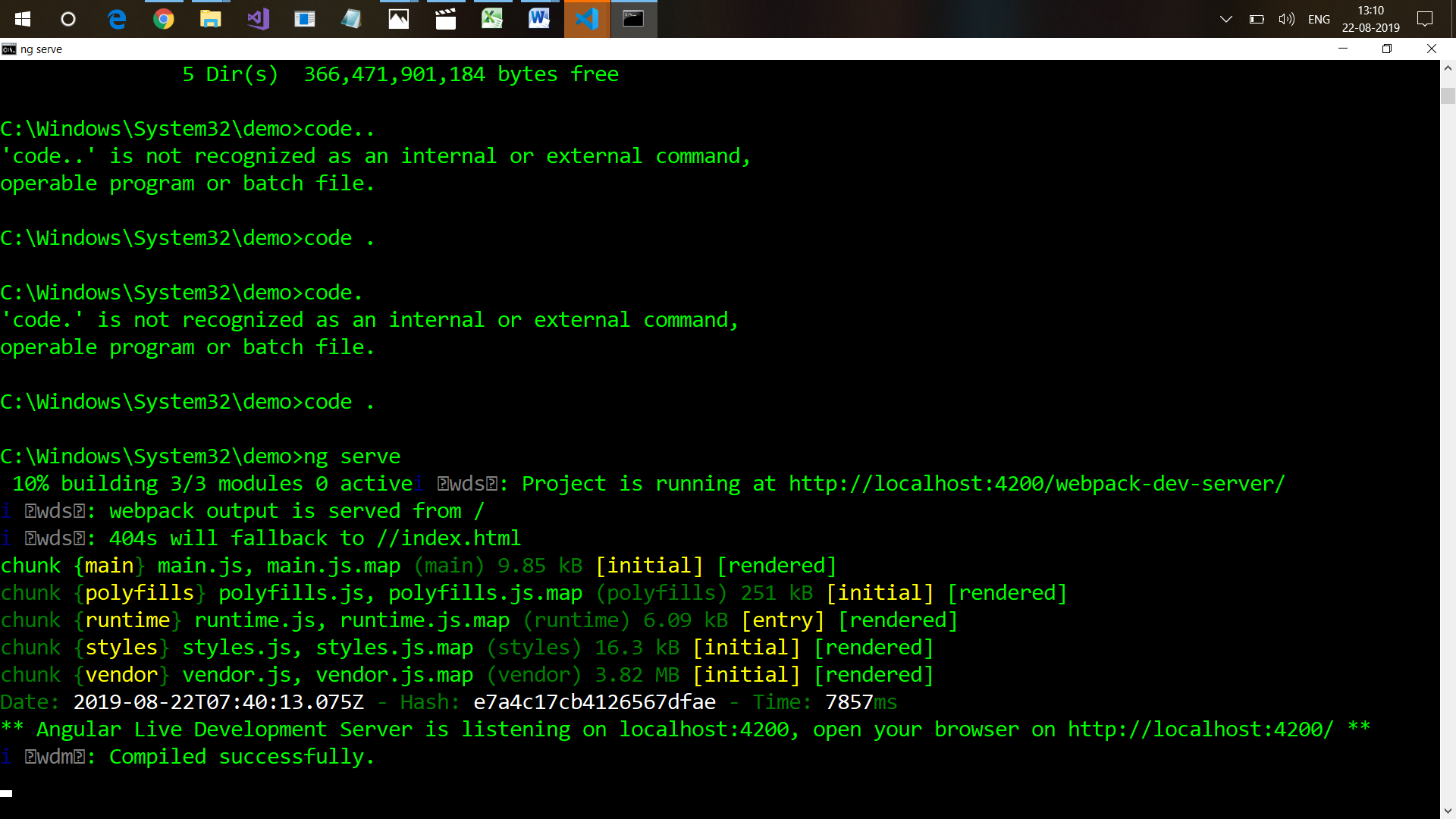
To start the application

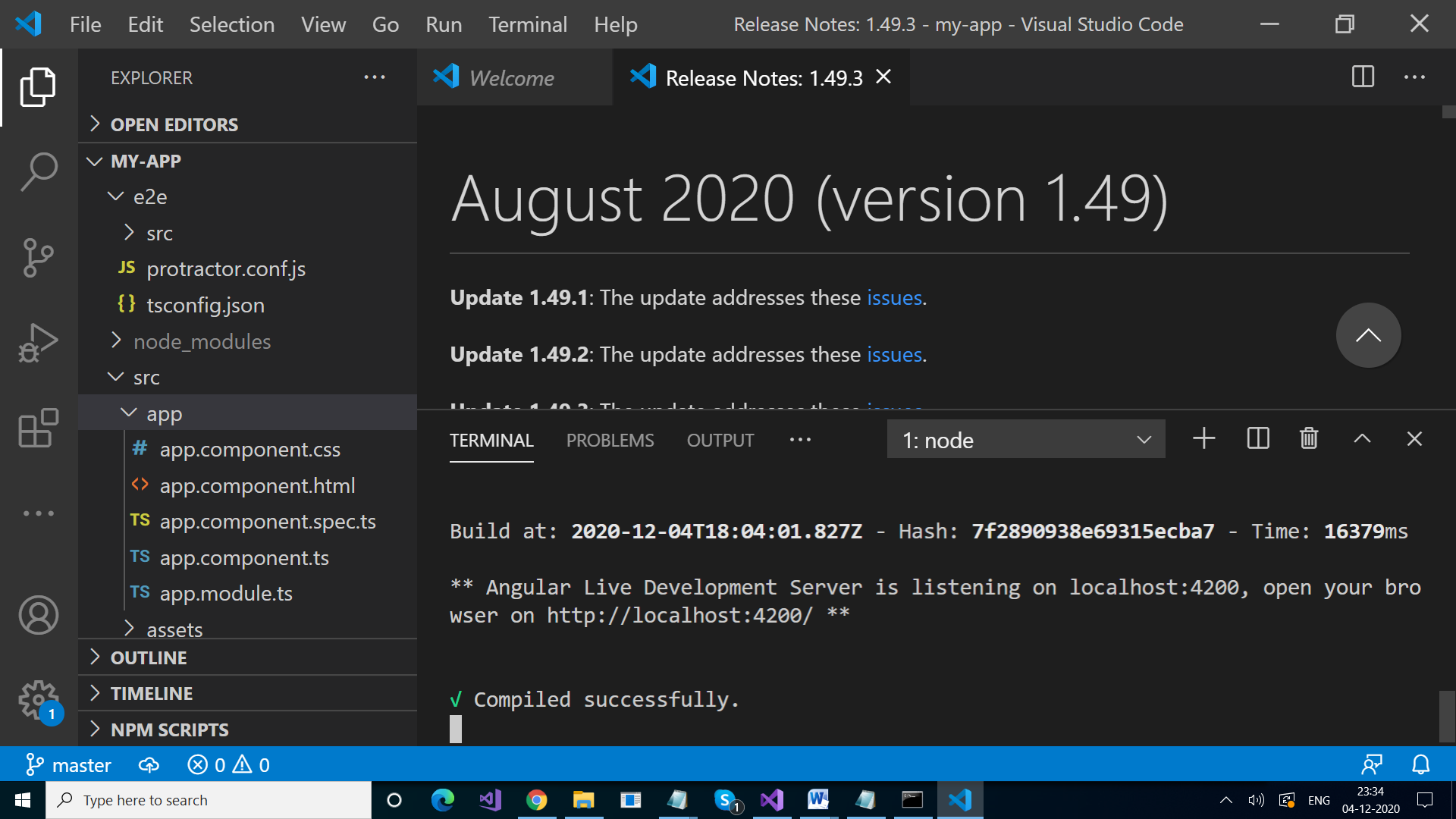
ng serve

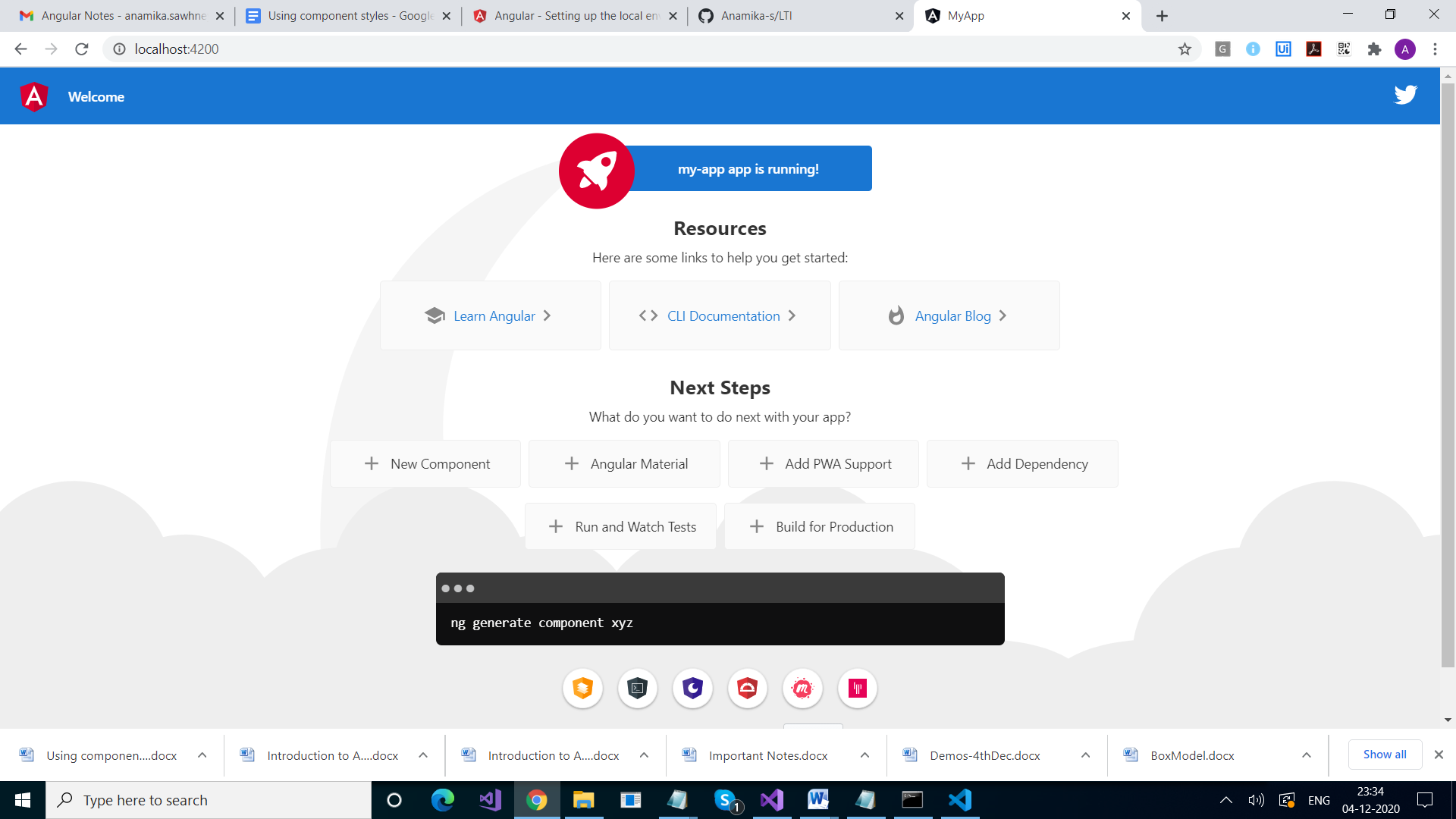
It will run my application in <http://localhost:4200>



It will take few seconds



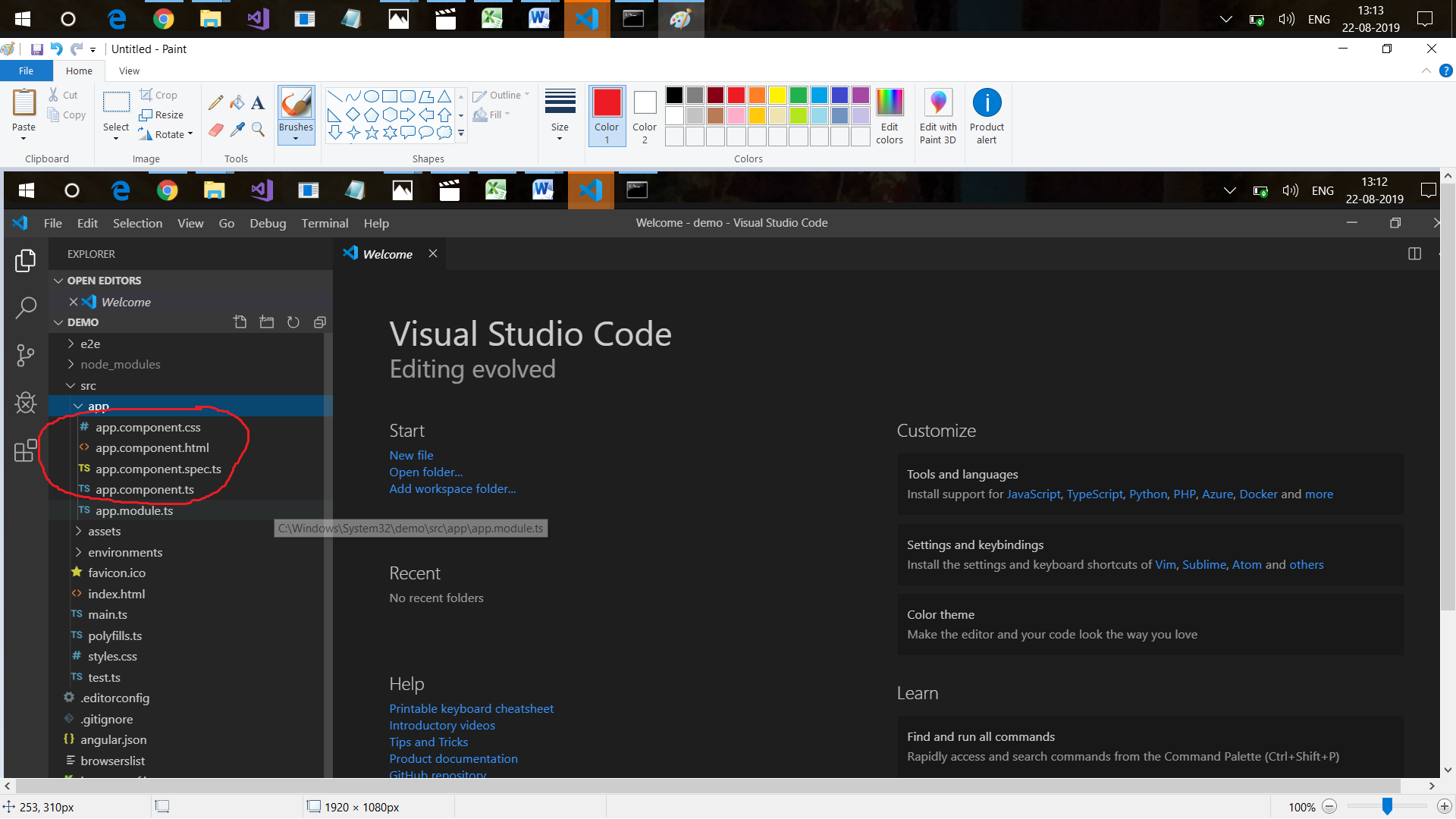




From where all this text is coming…..

Lets explore our application

Go to src folder



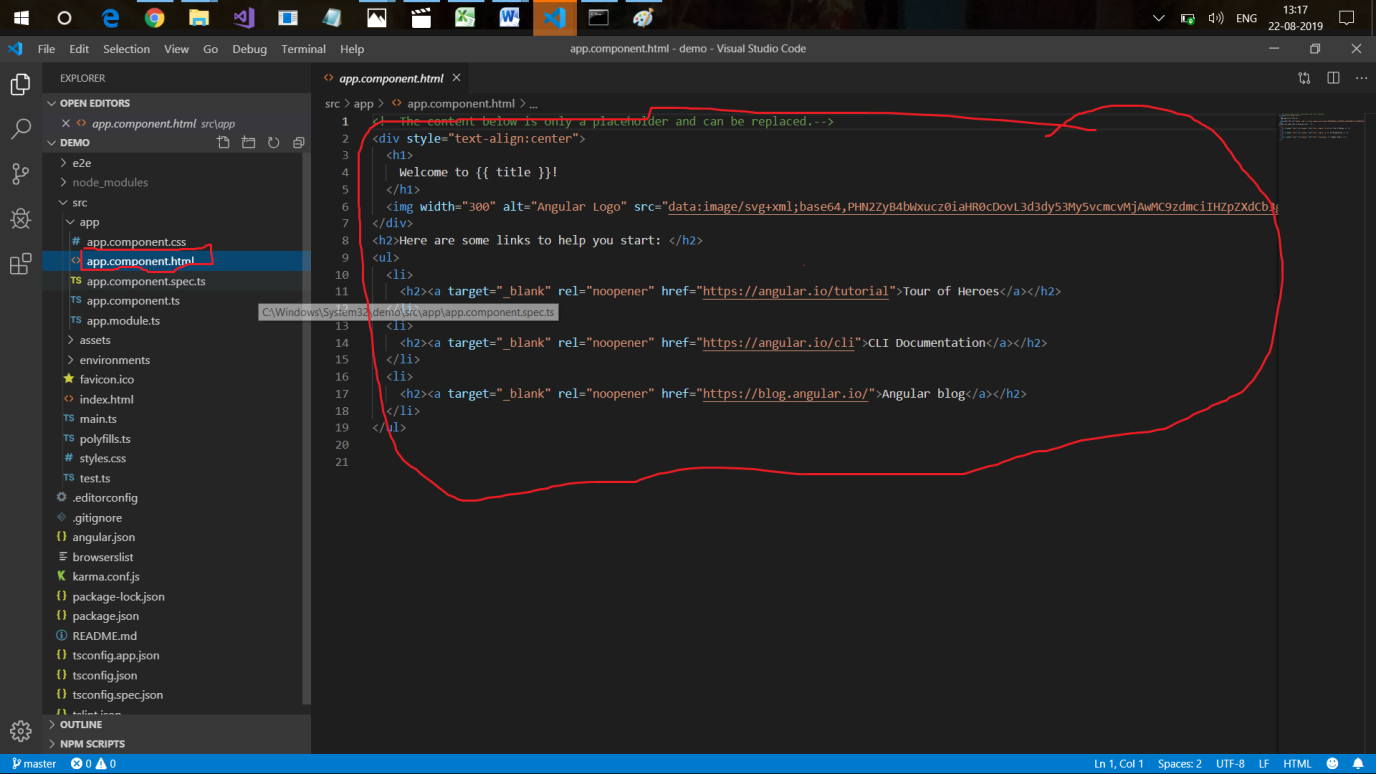
Here, this app is a component

Check its app.component.ts file



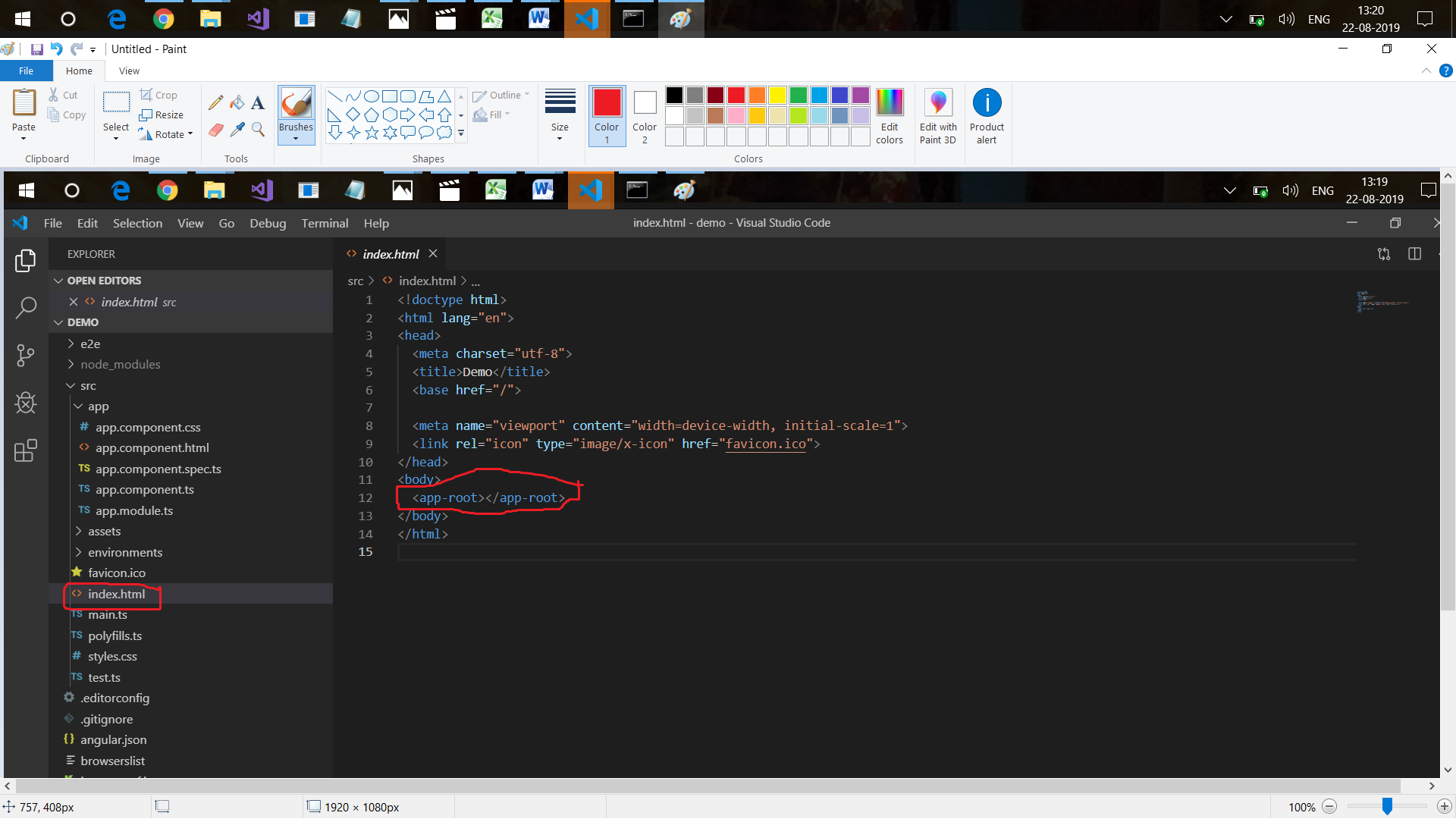
Demo written on application in browser is coming from point 2

Rest, is coming from app.component.html file



Ok, But where it is mentioned that this content should come in beginning, we have not mentioned anywhere

For this, check index.html page



This is a normal html page, Check its body element.

<app-root> </app-root> is mentioned here. Contents are being rendered here.

Again check your app.component.ts file

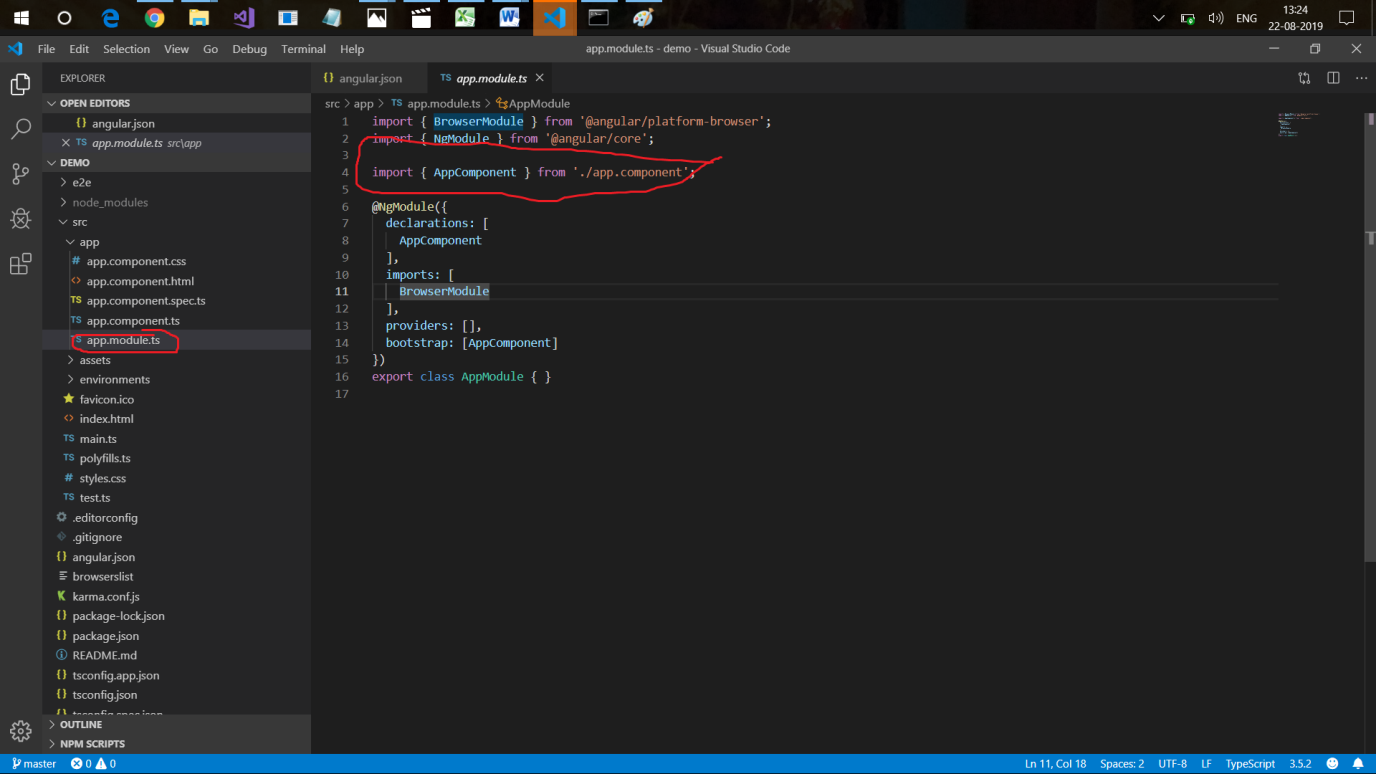


Can u see app-root in selector.

This is how these are linked.

But why only app-component.ts file is coming first.

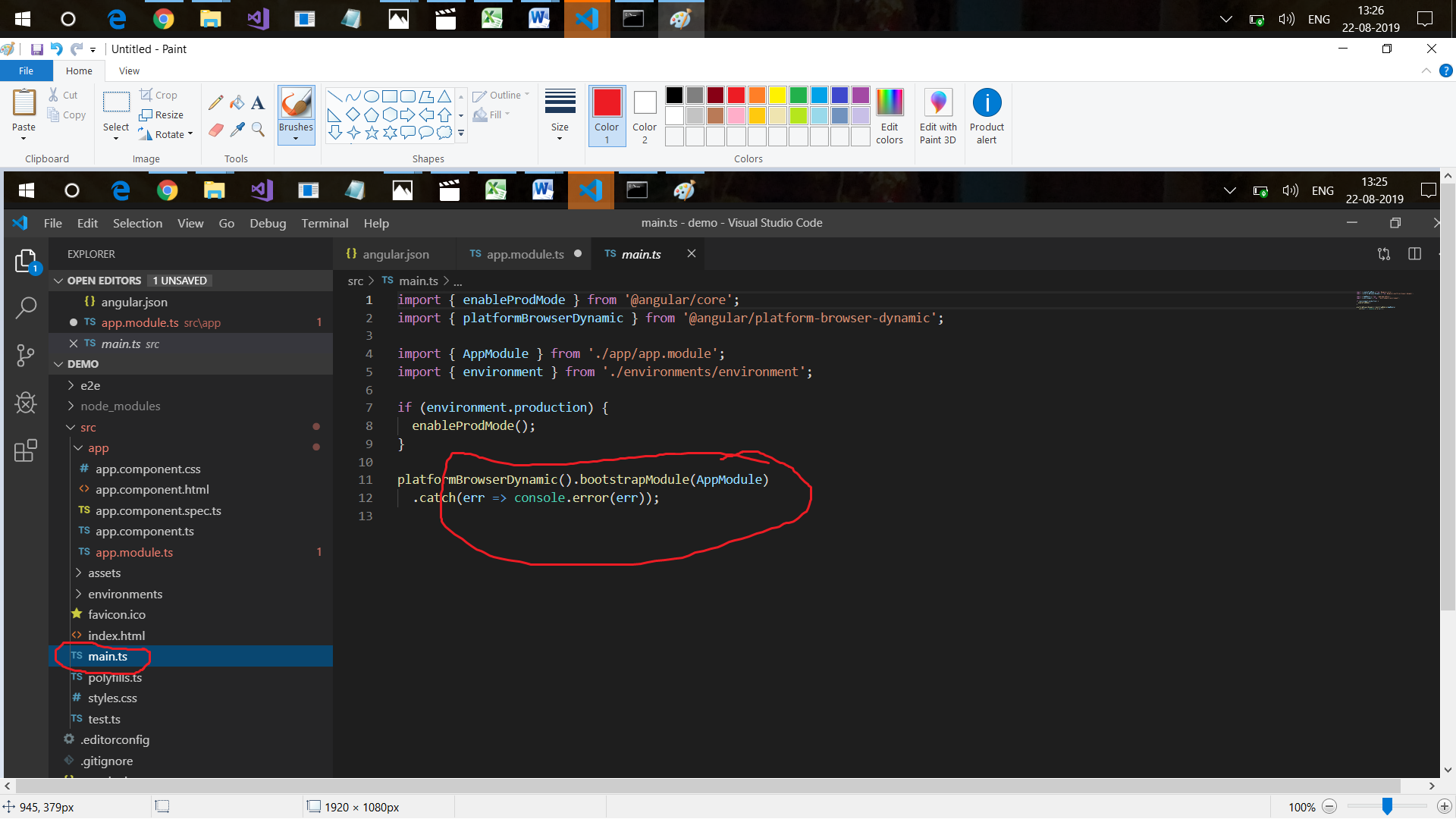
For this, check app-module.ts file



This component is called here

And how,

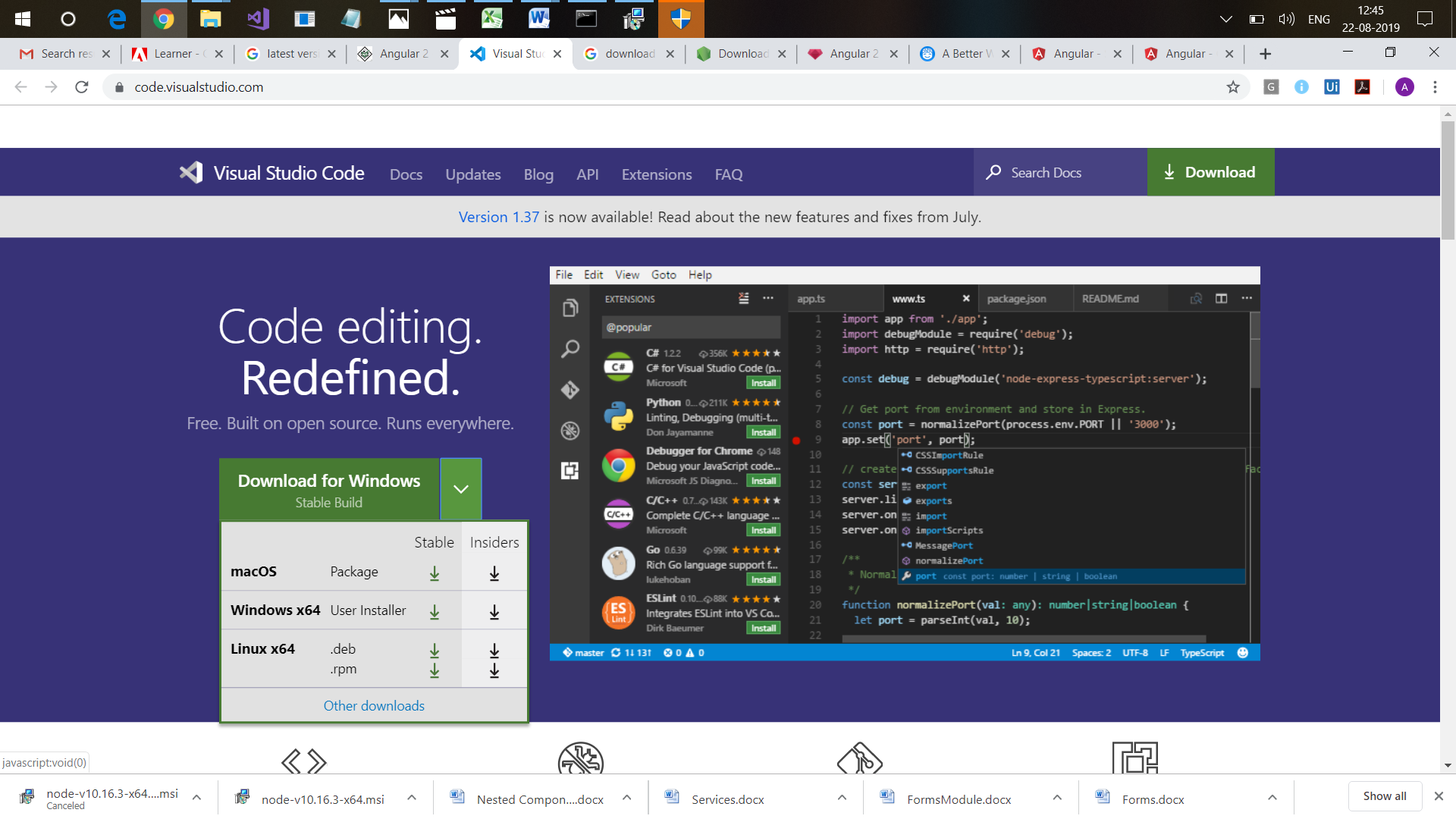
Check main.ts file



So, when we run Application in Angular, Main.ts file runs first, it checks which module to load first, which has the component that needs to be displayed.

## Installation of Visual Studio Code

[**https://code.visualstudio.com**](https://code.visualstudio.com/)



Project Structure

Different configuration files.

At the end, we have tslint.json : This file is the root for building our application that how our application shud be built.

Above that we have tsconfig.json, configuration fiel for our typescript

We have protractor.conf.js : It has the testing configuration.

Above that we have package.json, It contains all the depenedicies, all the modules that are reqiotred for our application.

Under node\_modules, we have bunch of libraries, dependencies, packages.which are added by default by CLI.

Karma is test runner and jasmine is the framework and both of these are developed by angular team for writing unit tests.

And above that we have .gitignore which conatins all the files which we want to exclude from our repository

.editorconfig : it has settings for our editor.

Then we have .angular-cli.json , it actually contains all the configuration of angular project.

In scr folder, we have app, assets, envrionment folders.

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'Demo-One';

}