**Learning Outcome-1**

**Activity 2**

Aim: Create an Angular application which add, multiply, divide the two input numbers from textbox.

Learning outcome: Able to understand how to create an angular application.

Duration: 3 hours

List of Hardware/Software requirements:

1. Operating System – Windows 10/11 or Linux

2. Command Prompt/Power Shell

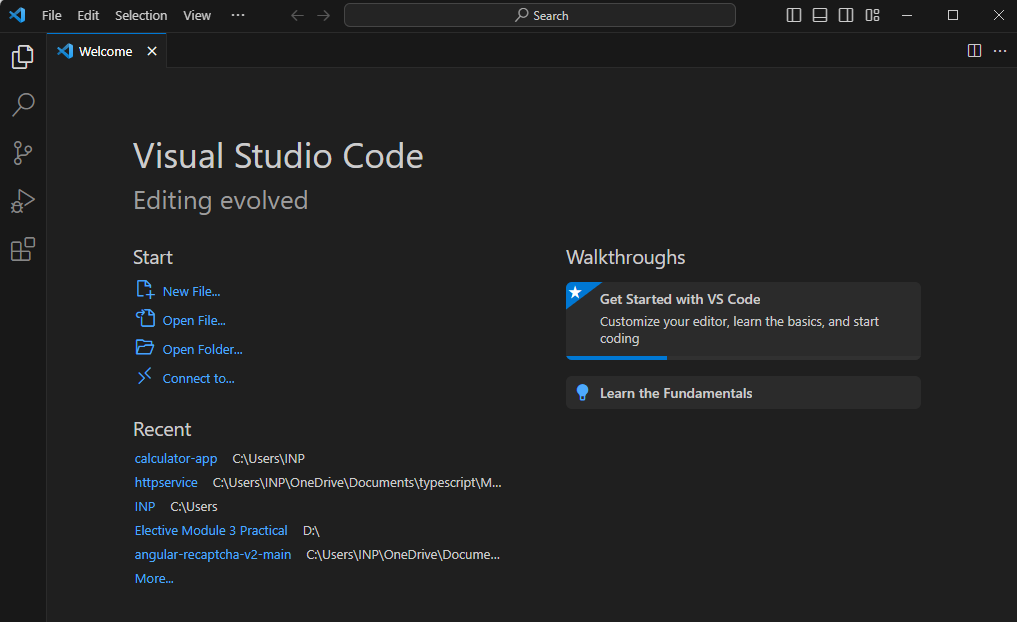
3. Internet Connectivity, Browser (Chrome or Edge)

4. Text Editor – Notepad / IDE – Visual Studio Code/Any IDE

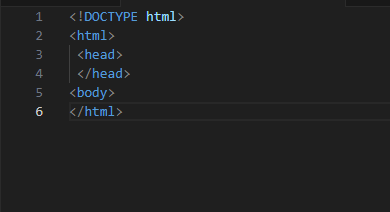
5. NodeJS, Angular CLI

Code/Program/Procedure (with comments):

1. Open Visual Studio.



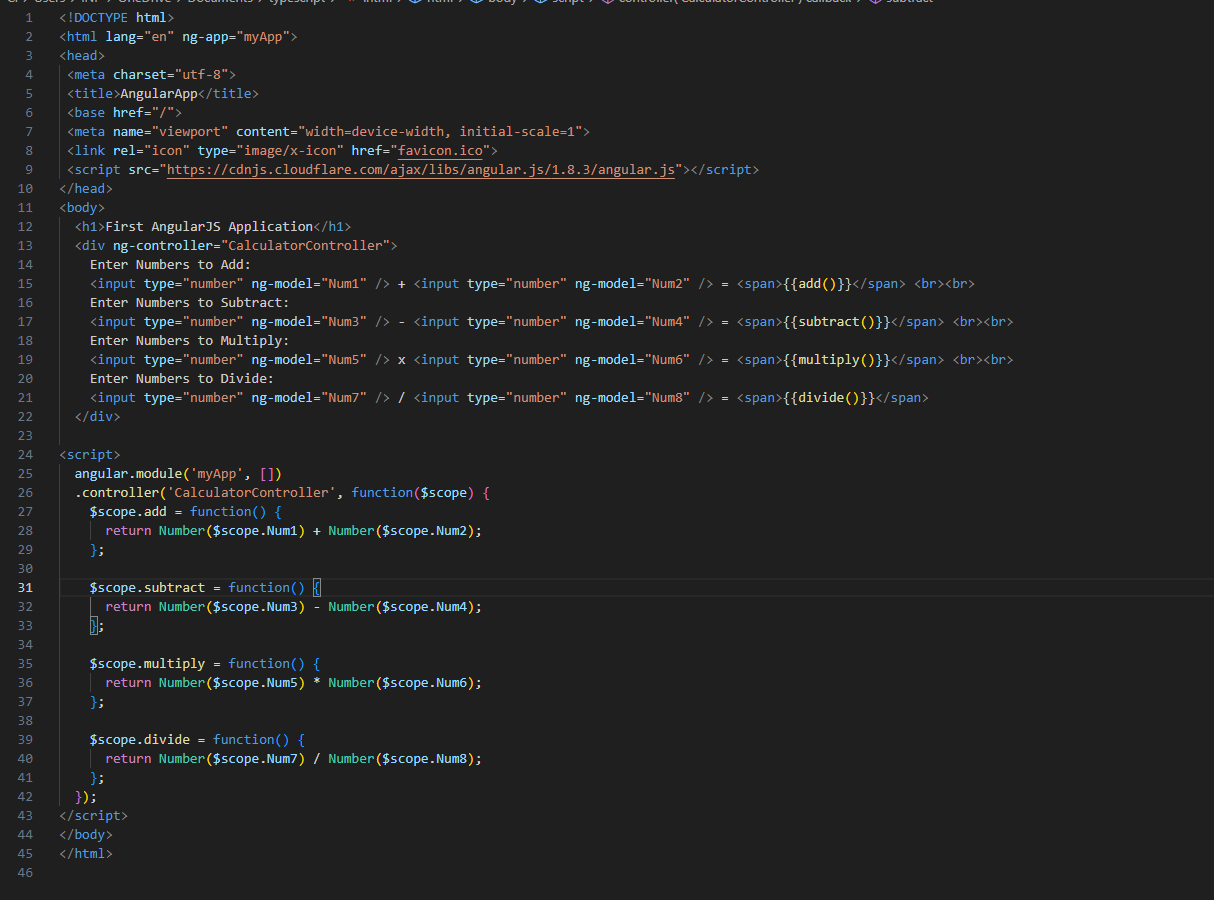
1. Let's create a simple AngularJS web application step by step and understand the basic



1. Include angular.js file in the head section

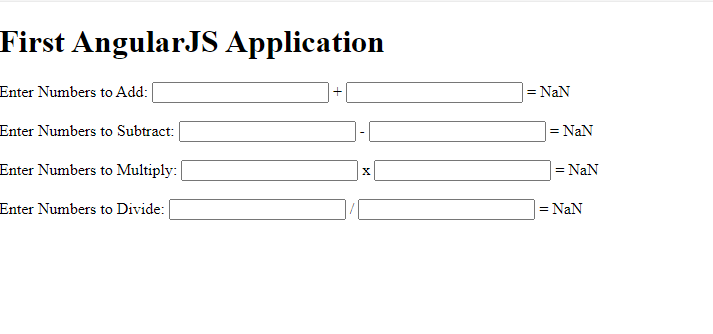


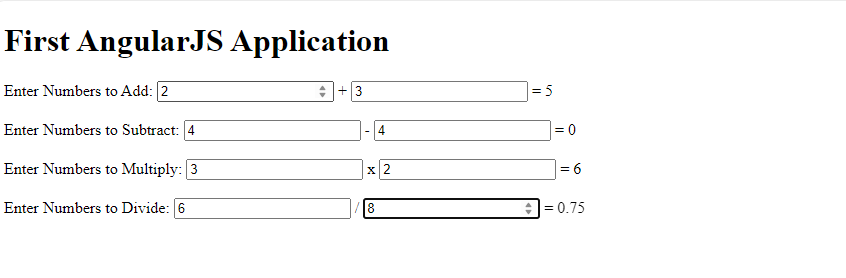
1. Here, we will be creating a simple calculator application which will add, subtract, multiply, divide two numbers.



1. Save as index.html

Output





**Learning Outcome-2**

**Activity 1**

Aim: Create an Angular application which can validate the email accepted from user

Learning outcome: Able to develop the real time scenarios based on Node JS applications.

Duration: 3 Hour

List of Hardware/Software requirements:

6. Operating System – Windows

7. Command Prompt/Power Shell

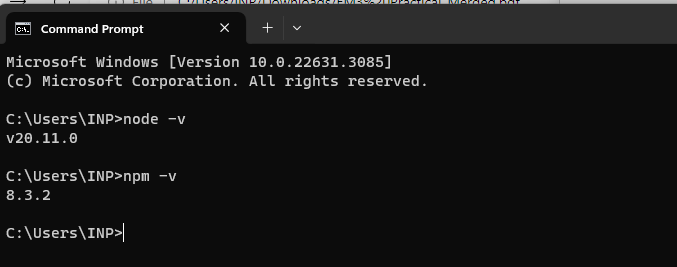
8. Internet Connectivity, Browser (Chrome or Edge)

9. Text Editor – Notepad / IDE – Visual Studio Code/Any IDE

10. NodeJS, Angular CLI

Code/Program/Procedure (with comments):

**Step 1: To verify the node, npm version, use the following command:**

****

**Step 2: Installing Angular CLI**

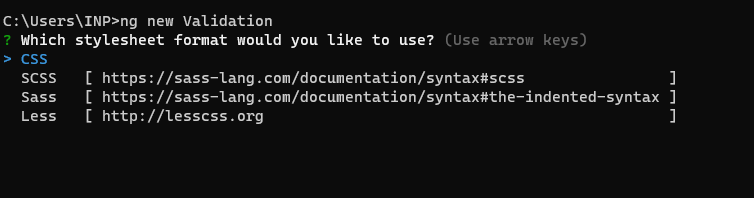
* **$ npm install -g @angular/cli**

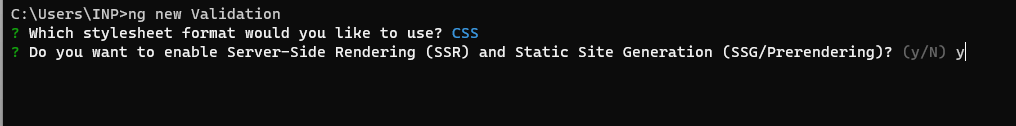
****

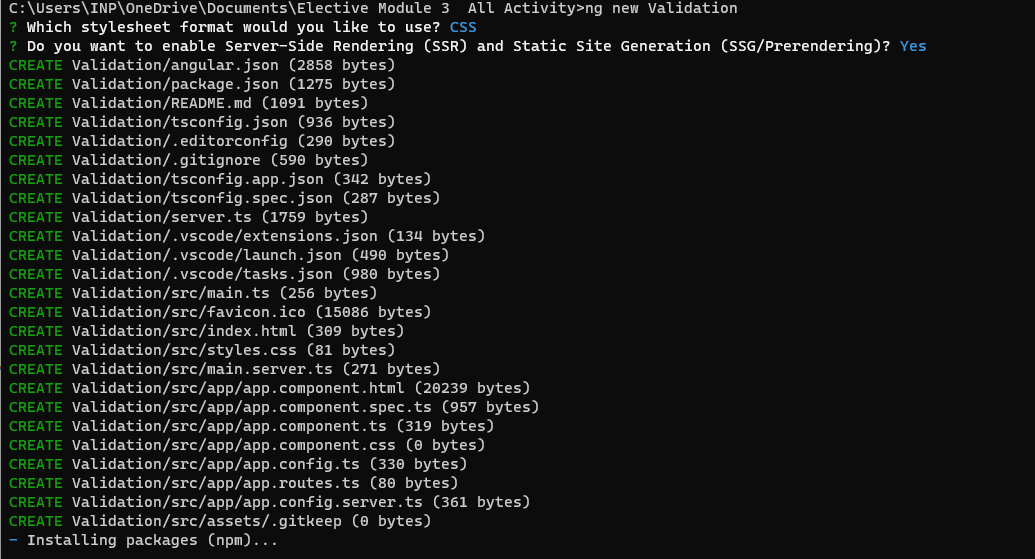
**Step 3: Creating your Angular 12 Project**

Go to CMD or Terminal and use this command:

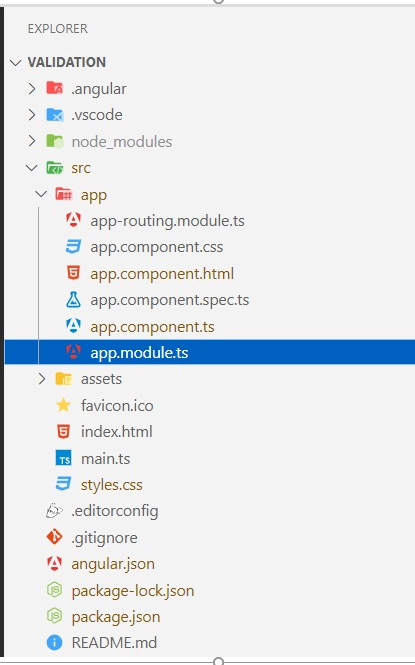
* **ng new Validation**

****

****

****

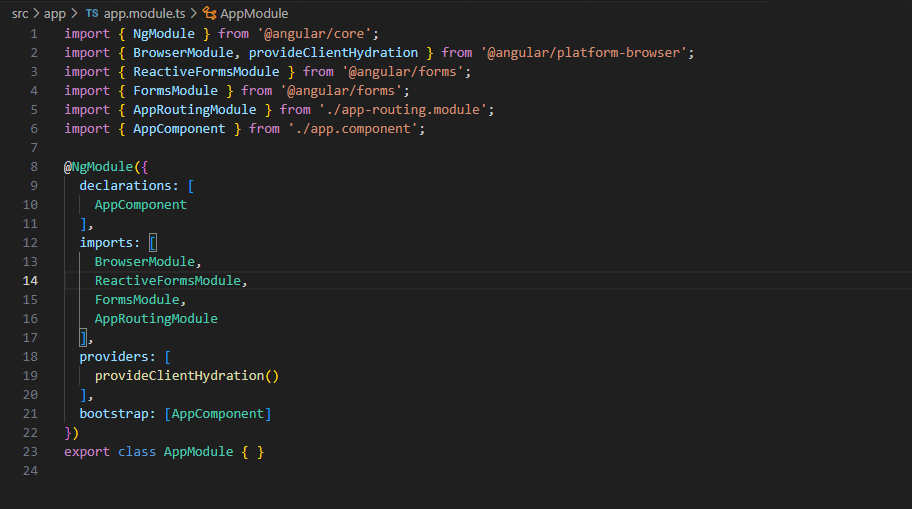
Once all packages have been added.

****

**Step 4: Import Forms Module**

Create form in angular app then you need to import Forms Module from @angular/forms library.

**src/app/app.module.ts**

****

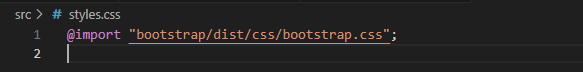
**Step 5: Form with ngModel write code of html form with ngModel.**

* **src/app/component.html**

****

You need to import your bootstrap css on style.css file

* **src/style.css**

****

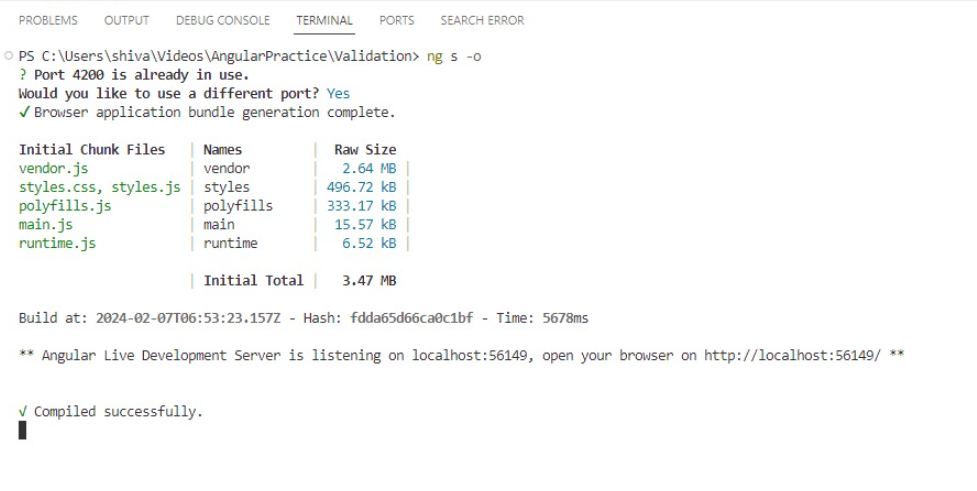
**Step 6: updated Ts File**

In ts file. we will write submit() and get all input fields values.

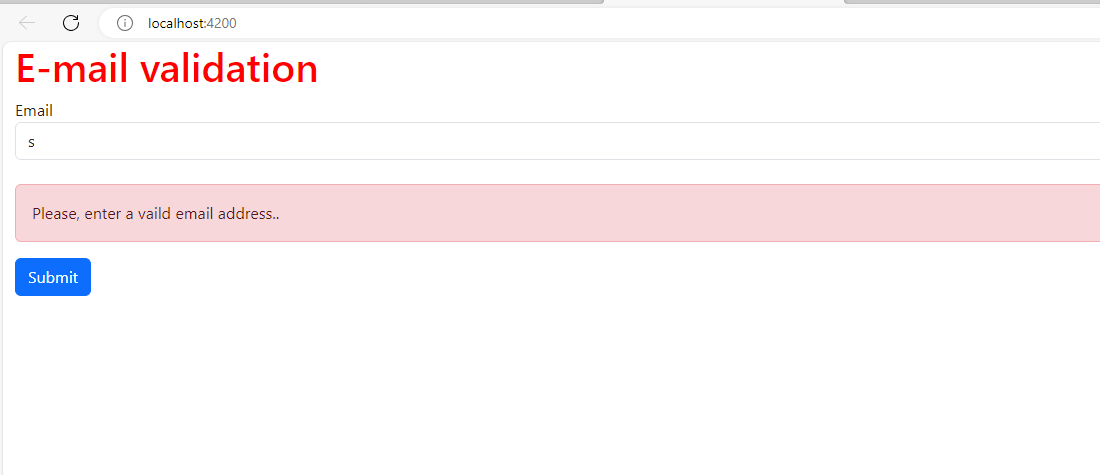
* **src/app/app.component.ts**

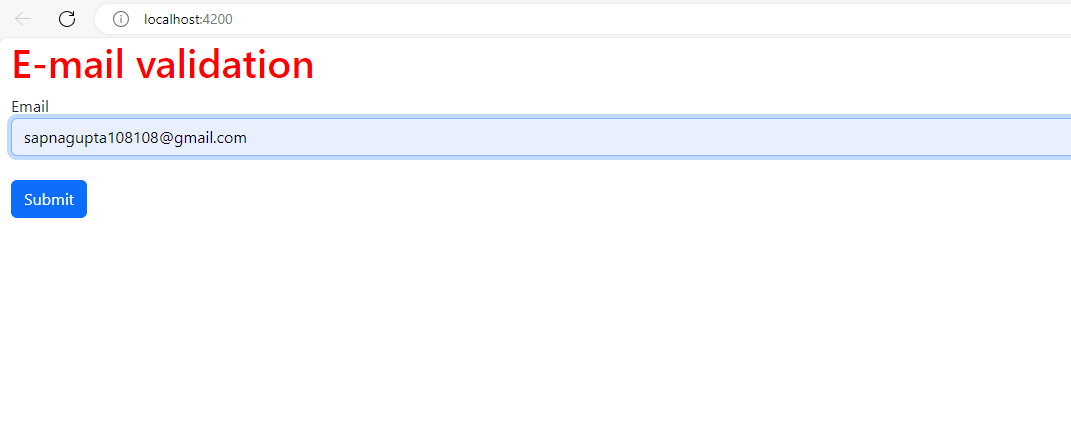
**Step 7: Run your application**

* **ng s –o**
* **ng serve**

****

**Output**

****

****

**Activity 2**

Aim: Create an Angular application which can validate the email accepted from user

Learning outcome: Able to develop the real time scenarios based on Node JS applications.

Duration: 3 Hour

List of Hardware/Software requirements:

6. Operating System – Windows

7. Command Prompt/Power Shell

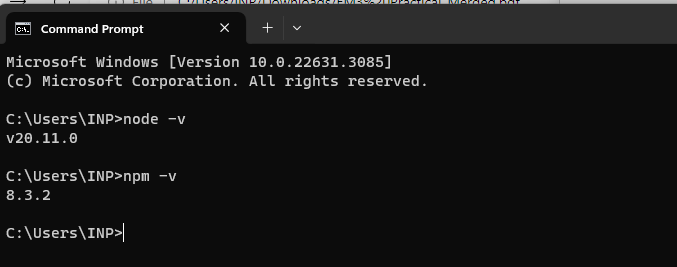
8. Internet Connectivity, Browser (Chrome or Edge)

9. Text Editor – Notepad / IDE – Visual Studio Code/Any IDE

10. NodeJS, Angular CLI

Code/Program/Procedure (with comments):

**Step 1: To verify the node, npm version, use the following command:**

****

**Step 2: Installing Angular CLI**

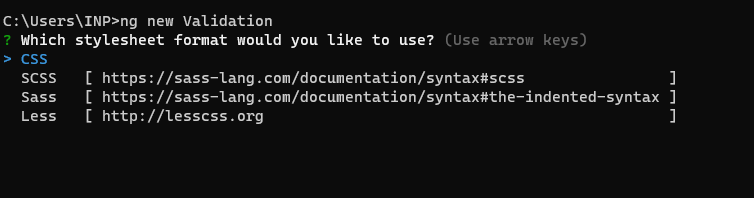
* **$ npm install -g @angular/cli**

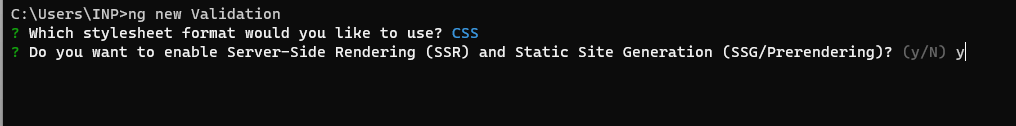
****

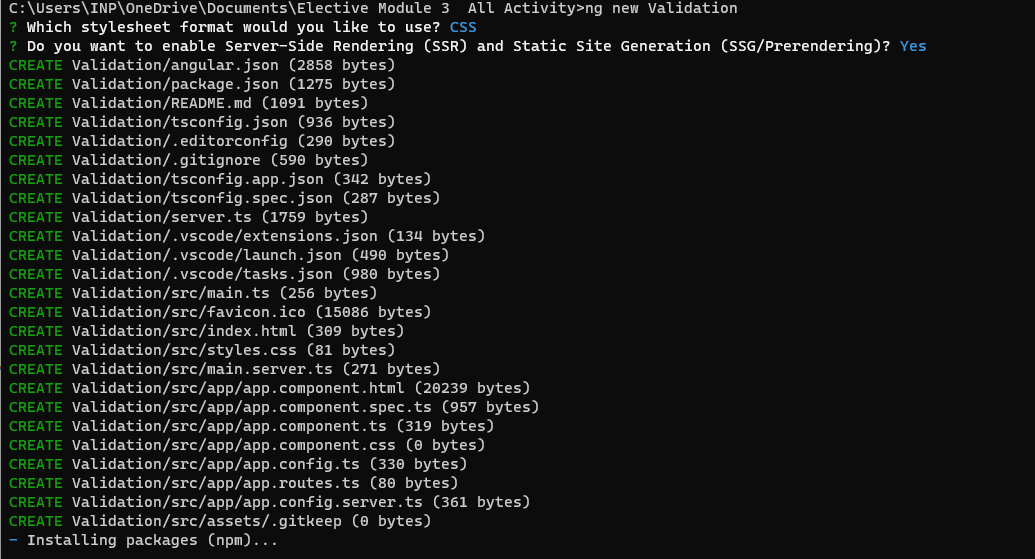
**Step 3: Creating your Angular 12 Project**

Go to CMD or Terminal and use this command:

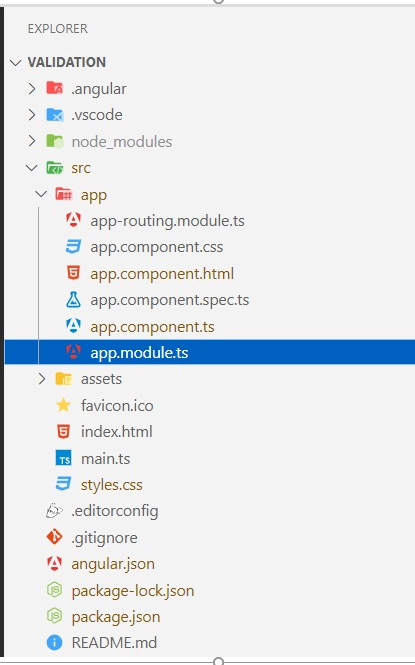
* **ng new Validation**

****

****

****

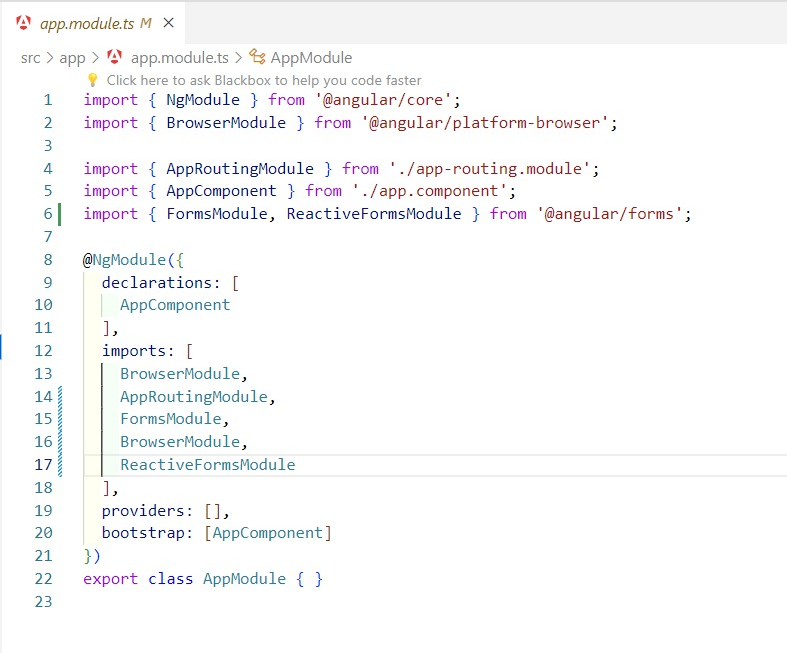
Once all packages have been added.

****

**Step 4: Import Forms Module**

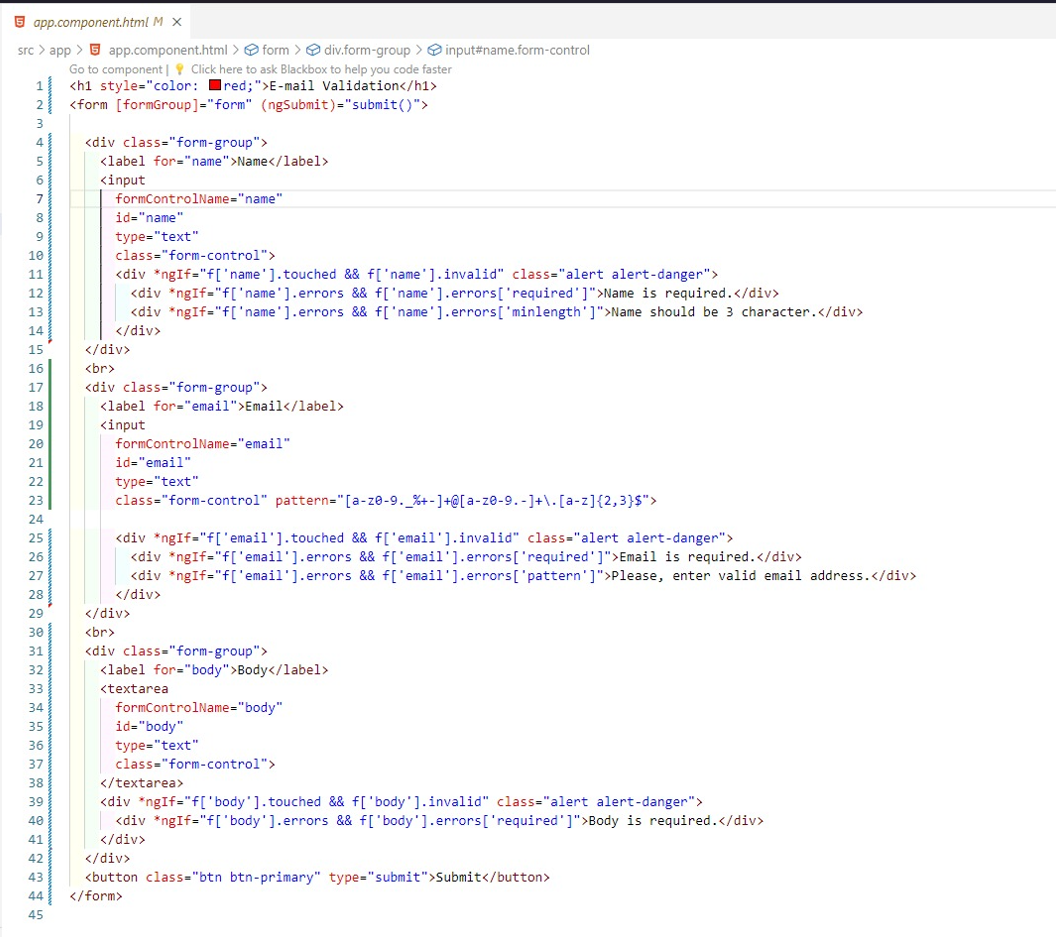
Create form in angular app then you need to import Forms Module from @angular/forms library.

**src/app/app.module.ts**

****

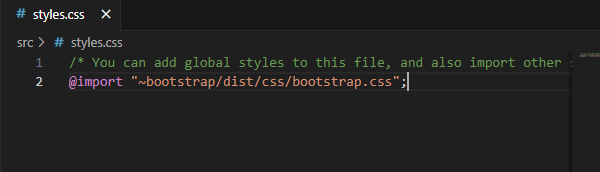
**Step 5: Form with ngModel write code of html form with ngModel.**

* **src/app/component.html**

****

You need to import your bootstrap css on style.css file

* **src/style.css**

****

**Step 6: updated Ts File**

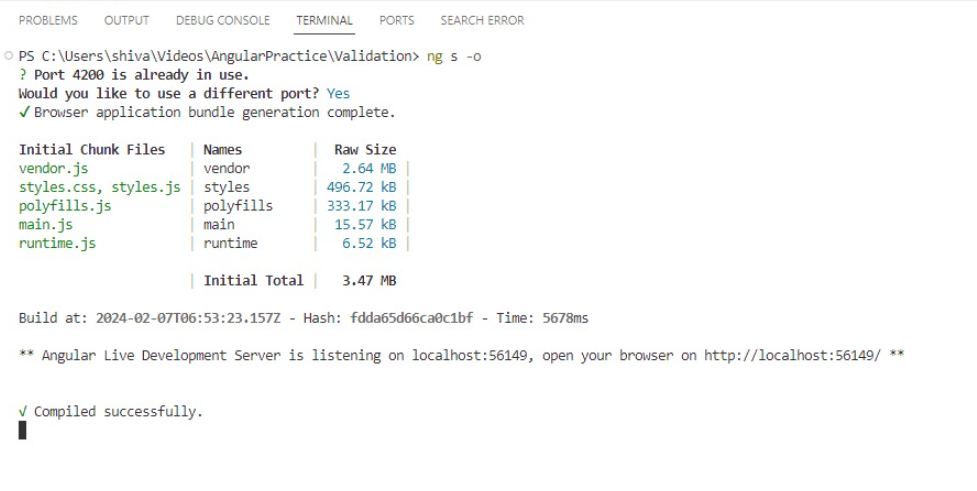
In ts file. we will write submit() and get all input fields values.

* **src/app/app.component.ts**

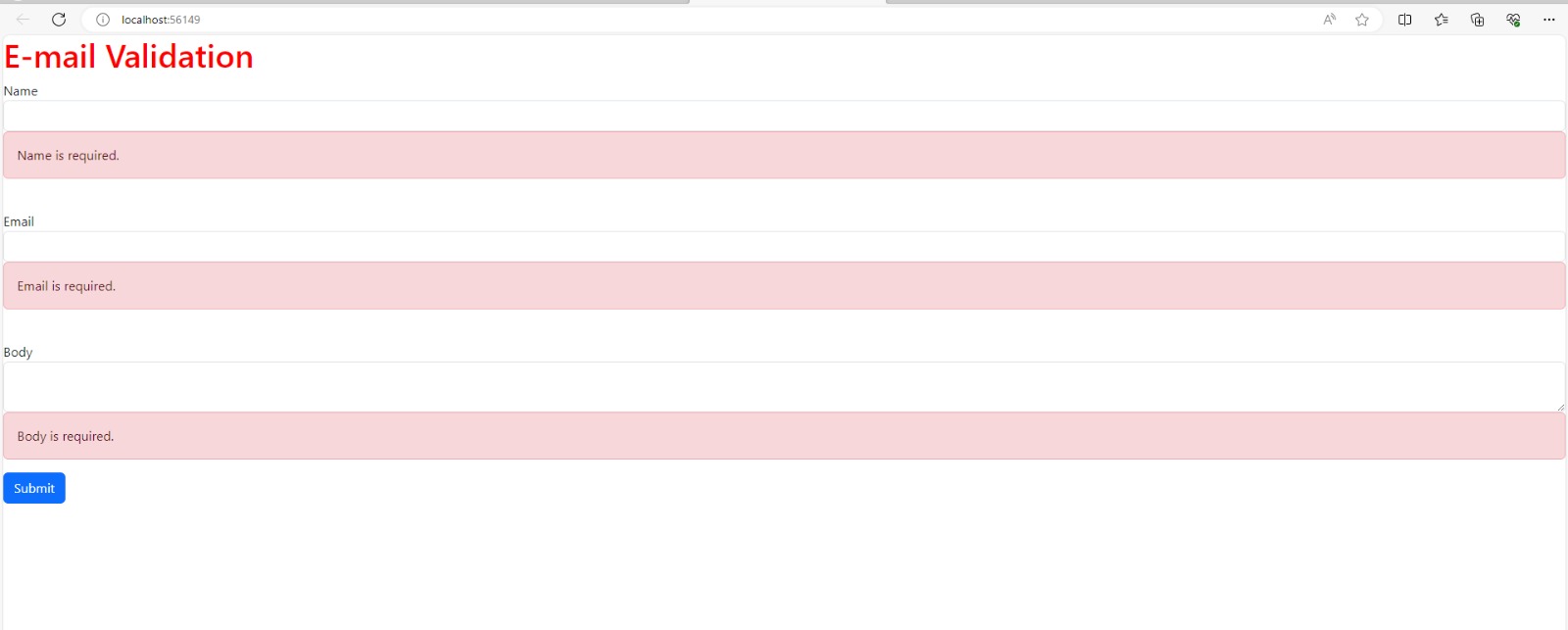
****

**Step 7: Run your application**

* **ng s –o**
* **ng serve**

****

**Output**

****

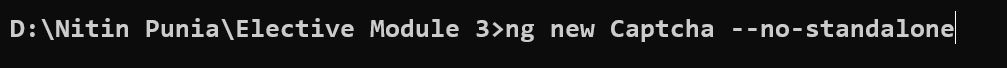
****

**Learning Outcome – 3**

**Activity1: Create an Angular application which can create Captcha.**

**Code:**

1. Create new project.



2. cd Captcha

3. Then go to the app.component.ts

**Coding:**

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

@Component({

  selector: 'app-root',

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

  styleUrl: './app.component.css'

})

export class AppComponent implements OnInit {

  captchaText: string = '';

  constructor() { }

  ngOnInit(): void {

    this.generateCaptcha();

  }

  generateCaptcha(): void {

    const chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';

    const captchaLength = 6;

    let captcha = '';

    for (let i = 0; i < captchaLength; i++) {

      const index = Math.floor(Math.random() \* chars.length);

      captcha += chars[index];

    }

    this.captchaText = captcha;

  }

}

4. After that go to the app.component.html

**Coding:**

<div>

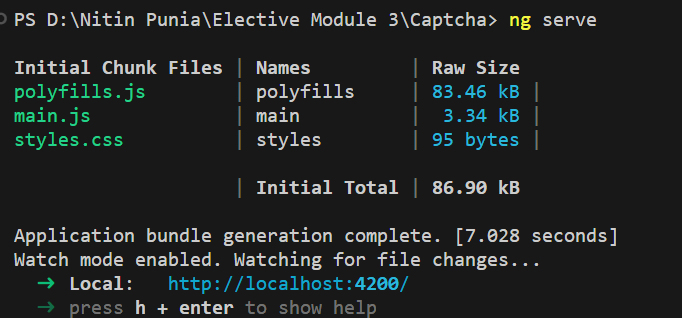
  <h2>Generated CAPTCHA</h2>

  <p>{{ captchaText }}</p>

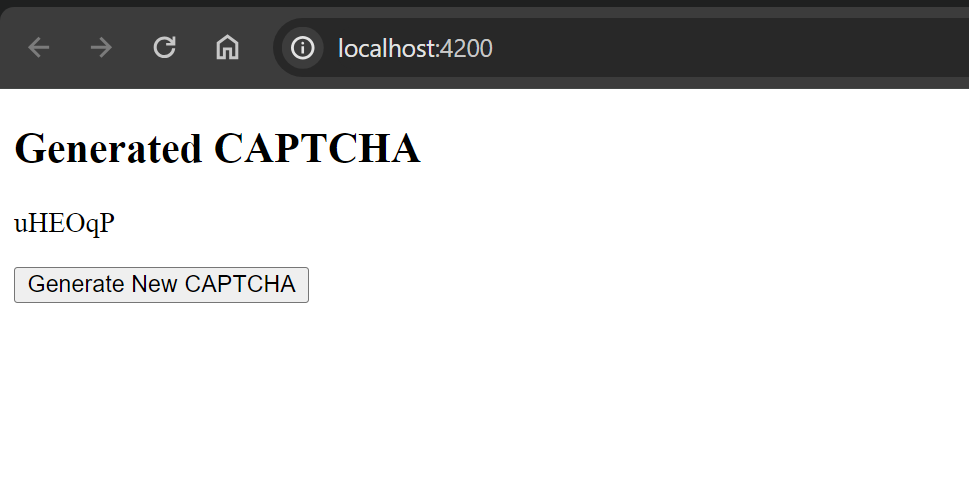
  <button (click)="generateCaptcha()">Generate New CAPTCHA</button>

</div>

5. Lastly run the application



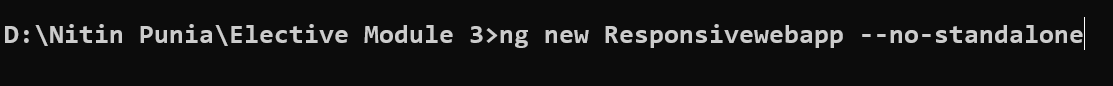
**Output:**



**Activity: -2 Create Responsive Web Application using Angular.**

**Code:**

1. Create new project.



2. cd Captcha

3. Then go to the app.component.ts

**Coding:**

<div fxLayout="column" style="height: 100vh;">

  <!-- Header -->

  <div class="header">

    <h1>Responsive Angular App</h1>

  </div>

  <!-- Main Content -->

  <div fxLayout="row" fxLayout.xs="column" fxLayoutAlign="start stretch" class="main-content">

    <!-- Sidebar -->

    <div fxFlex="20" fxFlex.xs="100" class="sidebar">

      <ul>

        <li><a href="#">Link 1</a></li>

        <li><a href="#">Link 2</a></li>

        <li><a href="#">Link 3</a></li>

      </ul>

    </div>

    <!-- Content -->

    <div fxFlex class="content">

      <p>This is the main content area. It adjusts based on the screen size.</p>

    </div>

  </div>

</div>

4. After that go to the app.component.css

**Coding:**

.header {

    background-color: #333;

    color: #fff;

    padding: 20px;

    text-align: center;

  }

  .main-content {

    padding: 20px;

  }

  .sidebar {

    background-color: #f4f4f4;

    padding: 20px;

  }

  .content {

    background-color: #fff;

    padding: 20px;

  }

5. Then go to app.module.ts

And import this library

import { FlexLayoutModule } from '@angular/flex-layout';

And also type in imports also

  imports: [

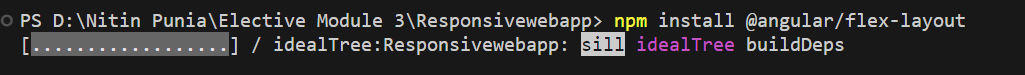
    BrowserModule,

    AppRoutingModule,

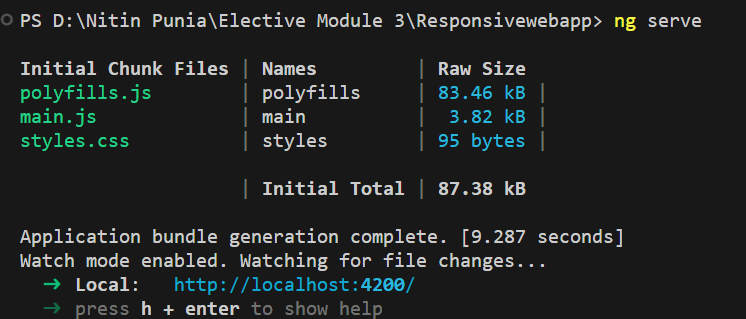
    FlexLayoutModule

  ]

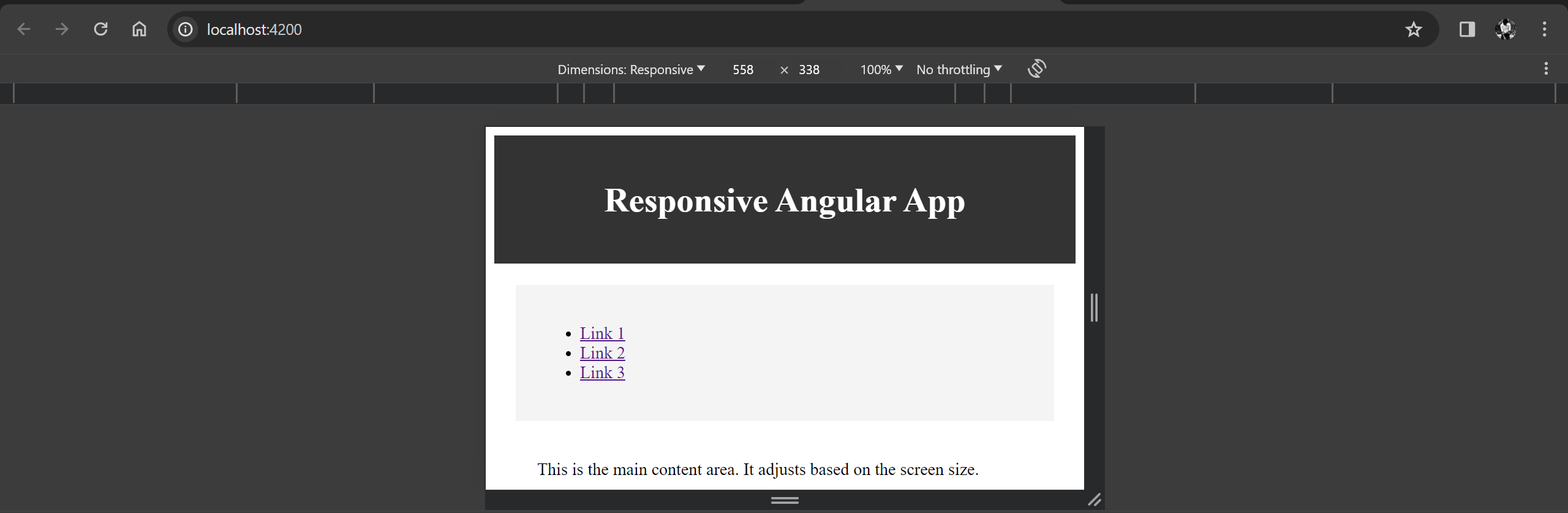
6. And after importing install that library (npm install @angular/flex-layout)



7. Run the application



**Output:**

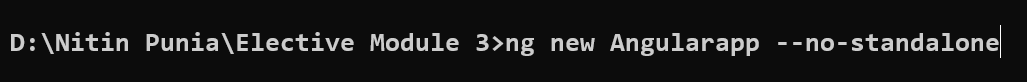
****

**Learning Outcome – 4**

**Activity 1: Create an Angular application using Node JS with complete templating system.**

**Code:**

1. Create new project.



2. cd Angularapp

3. Create a component.



4. open the project in vs code.

5. Firstly go to the greet.component.ts

**Coding:**

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

@Component({

  selector: 'app-greet',

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

  styleUrl: './greet.component.css'

})

export class GreetComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

  name: string = "Nitin";

  greet(): void {

  alert("Hello " + this.name);

  };

  }

6. Go to the greet.component.html

**Coding:**

<div>

    Enter Your Name: <input type="text" value={{name}} /> <br />

    <button (click)="greet()">Greet Me!</button>

    </div>

7. Go the app.module.ts

Import the Component

import { GreetComponent } from './greet/greet.component';

and imports in NgModule also:

@NgModule({

  declarations: [

    AppComponent,

    GreetComponent

  ],

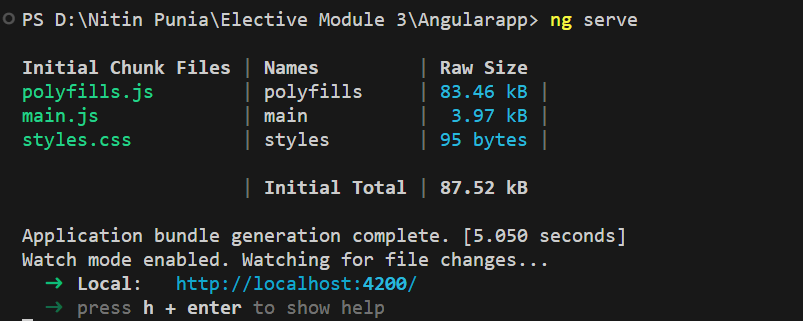
8. Go to the app.component.html

<div>

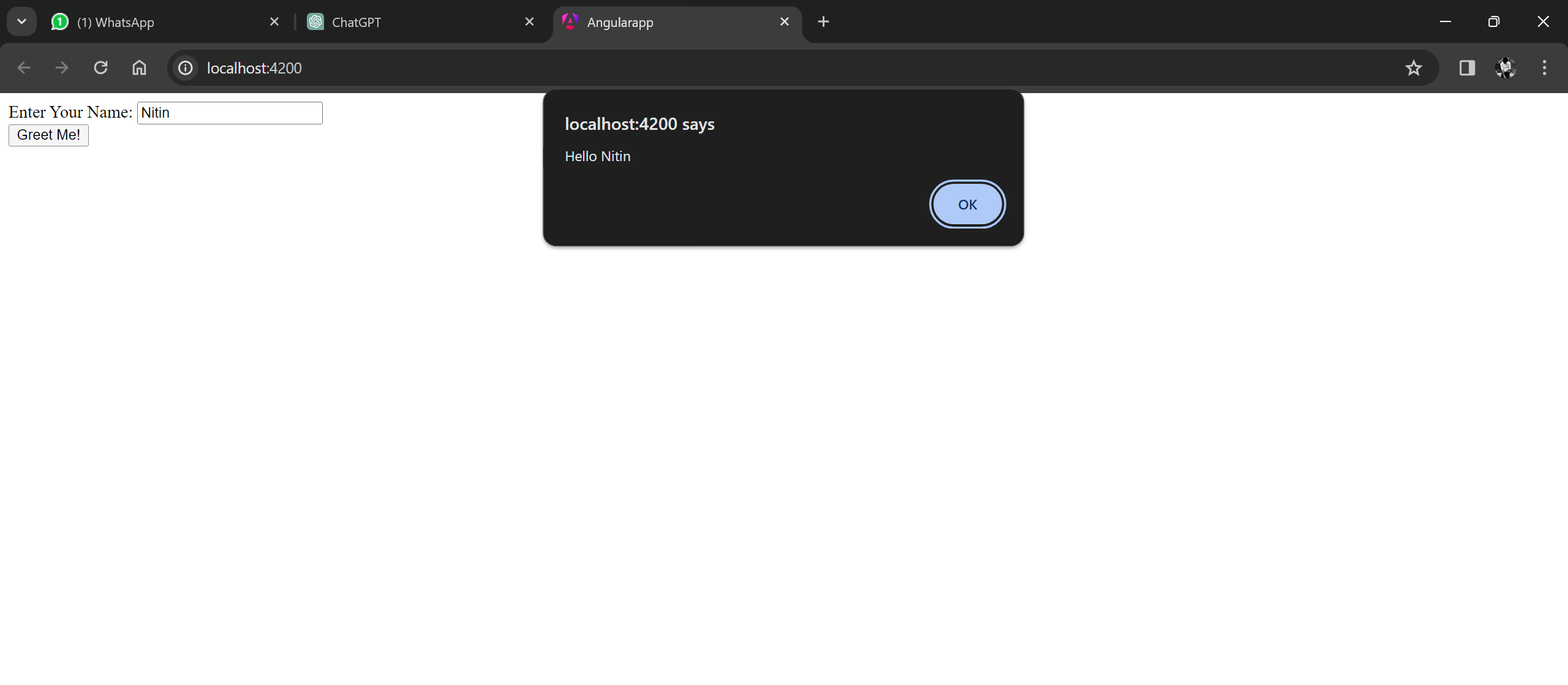
  <app-greet></app-greet>

  </div>

9. Then run the application



**Output:**



**Activity 2: Create a basic Program with Node**

**Code:**

1. Create new File (hello.js).

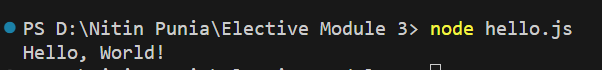
**Coding:**

console.log("Hello, World!");

2. Then run the application



**Output:**

****