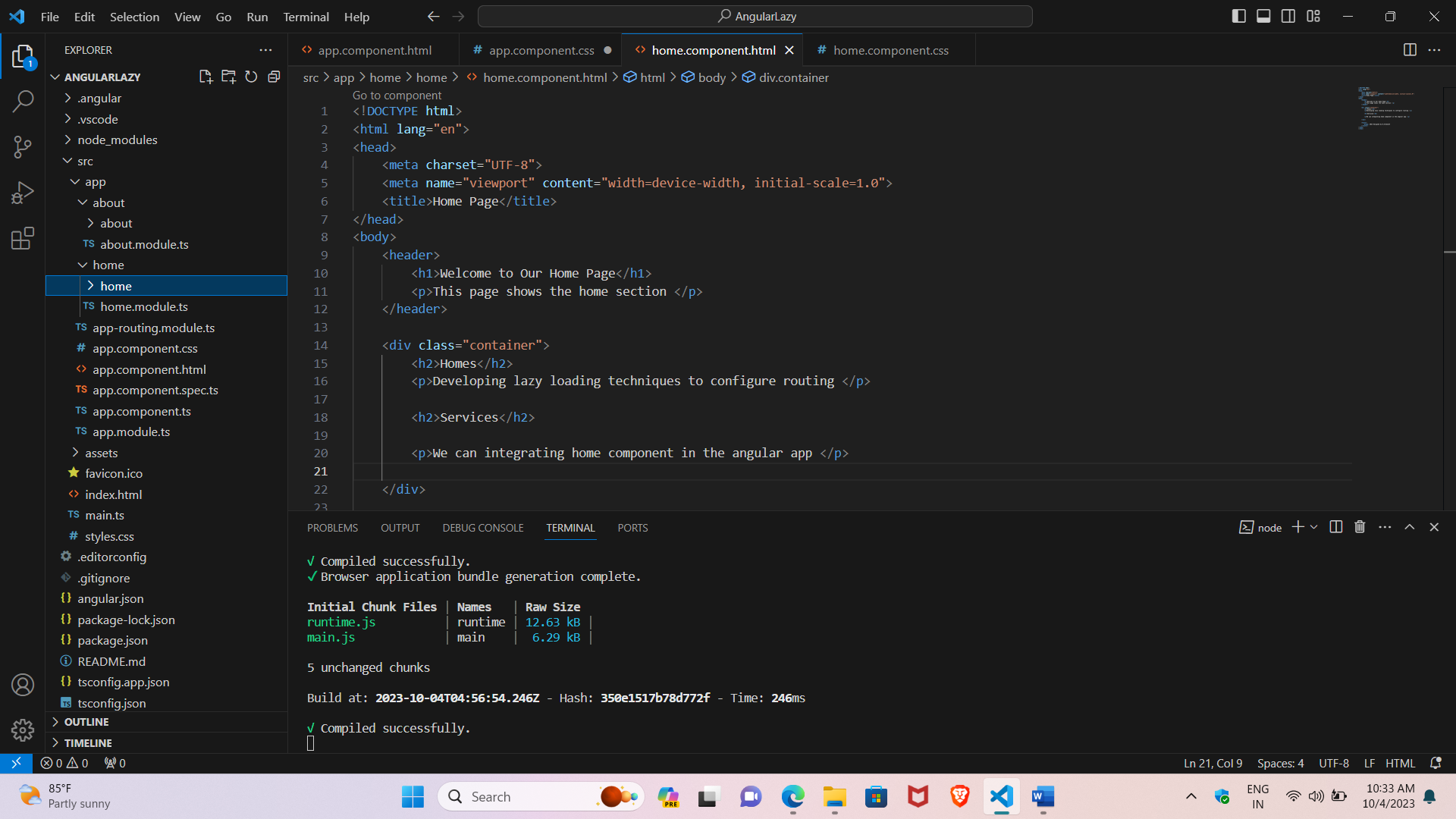
**Angular Assignments**

1) Write a program to configure routing with lazy loading technique?

**Source :**



**About.component.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>About Us</title>

</head>

<body>

    <header>

        <h1>About Us</h1>

        <p>This page shows the About section</p>

    </header>

    <div class="container1">

        <h2>About Us</h2>

        <p>Developing lazy loading techniques to configure routing </p>

        <h2>Services</h2>

        <p>We can integrating about component in the angular app </p>

    </div>

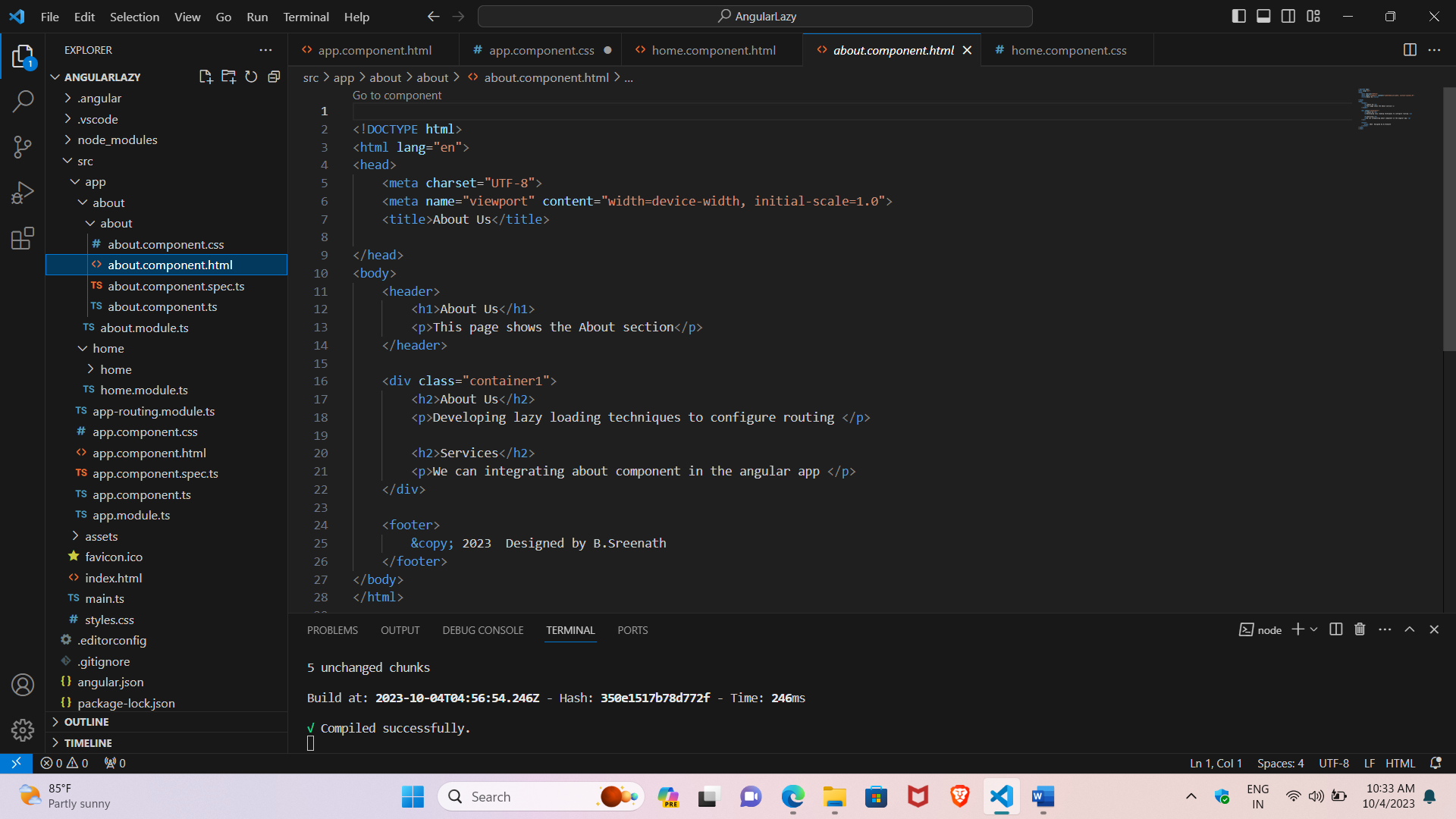
    <footer>

        &copy; 2023  Designed by B.Sreenath

    </footer>

</body>

</html>



**About module.ts**

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

import { RouterModule, Routes } from '@angular/router';

import { AboutComponent } from './about/about.component';

const routes: Routes = [

  { path: '', component: AboutComponent }

];

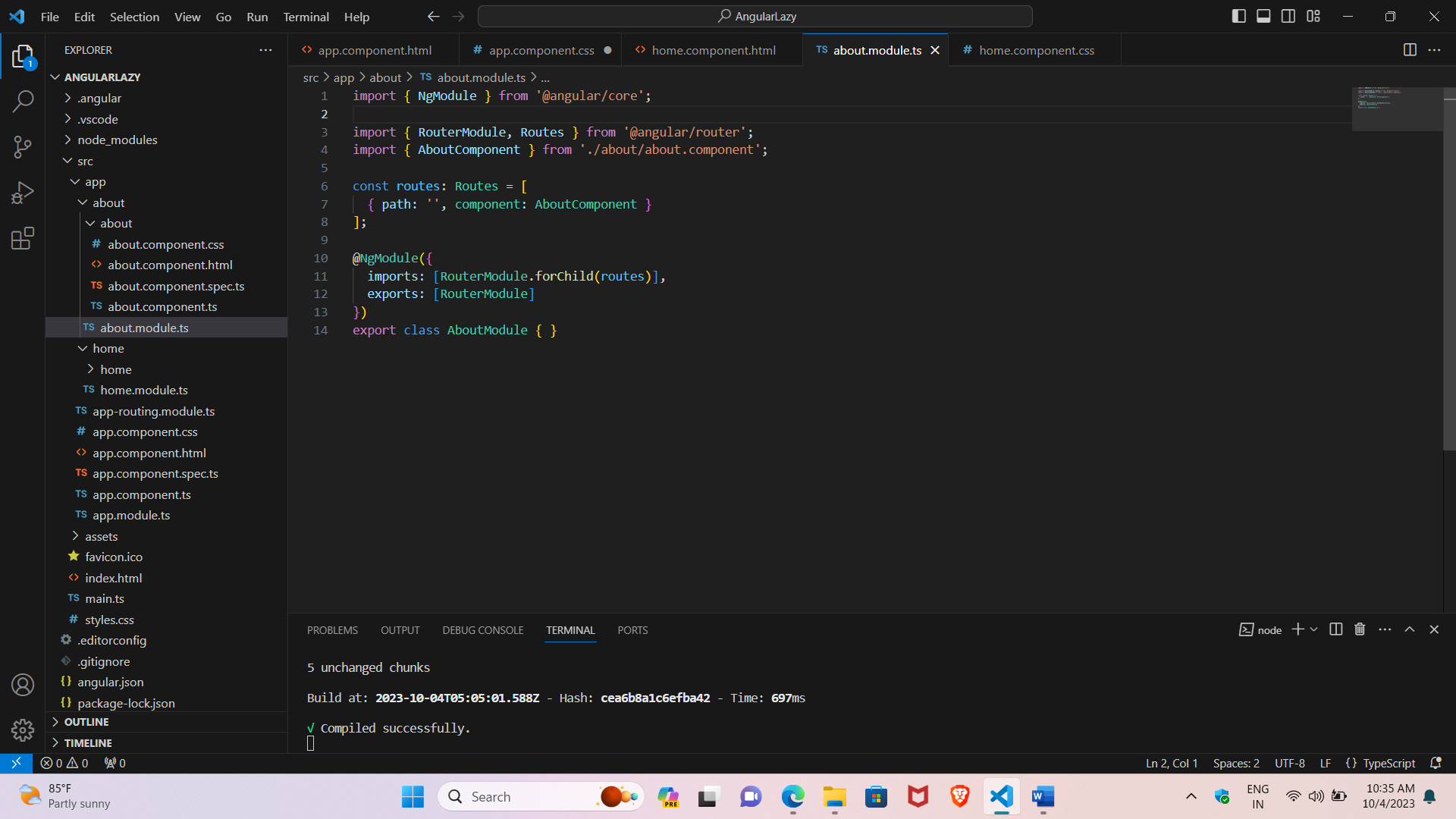
@NgModule({

  imports: [RouterModule.forChild(routes)],

  exports: [RouterModule]

})

export class AboutModule { }



[**Home.component.html**](http://Home.component.html)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Home Page</title>

</head>

<body>

    <header>

        <h1>Welcome to Our Home Page</h1>

        <p>This page shows the home section </p>

    </header>

    <div class="container">

        <h2>Homes</h2>

        <p>Developing lazy loading techniques to configure routing </p>

        <h2>Services</h2>

        <p>We can integrating home component in the angular app </p>

    </div>

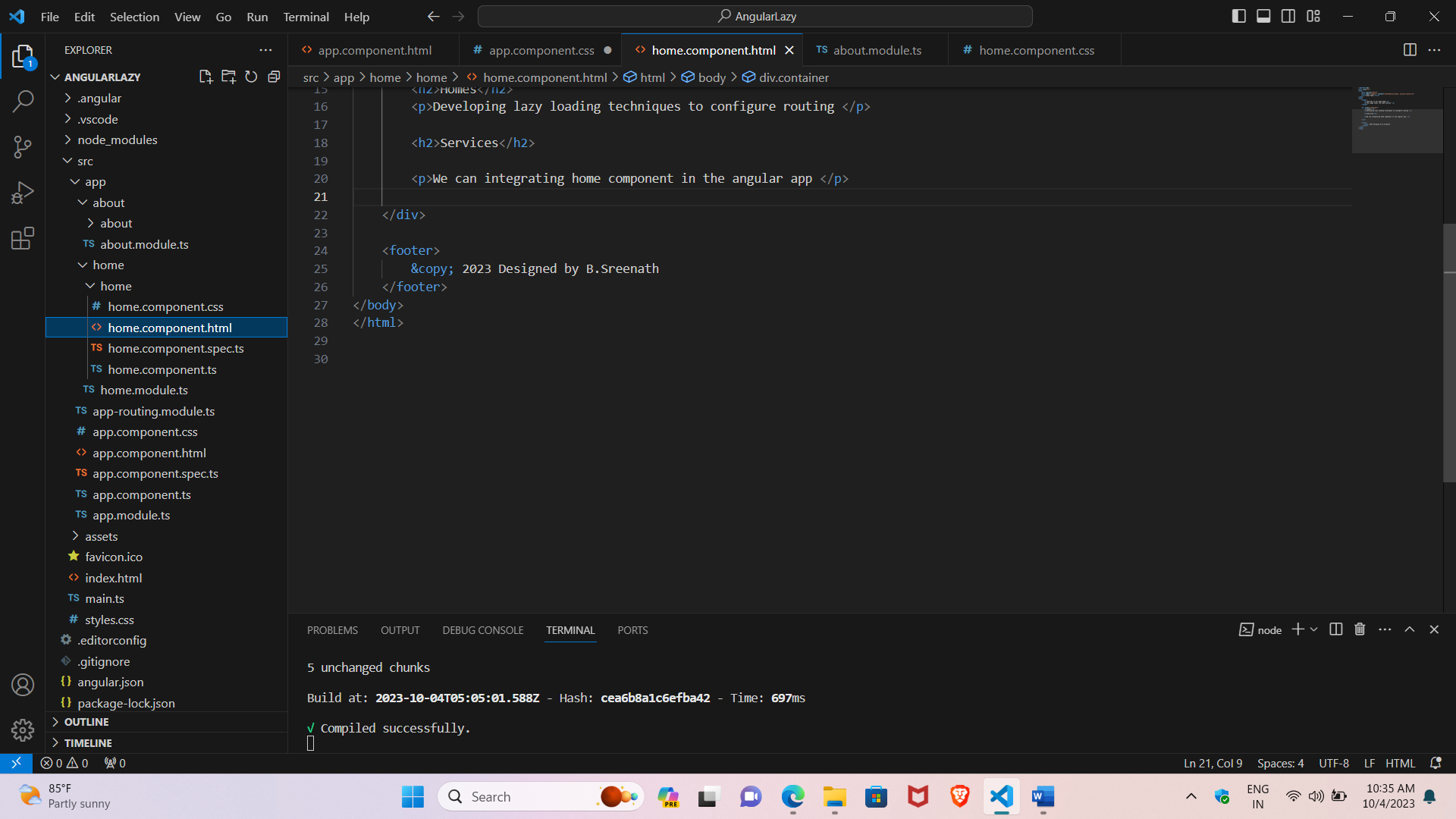
    <footer>

        &copy; 2023 Designed by B.Sreenath

    </footer>

</body>

</html>



**Home.module.ts**

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

import { RouterModule, Routes } from '@angular/router';

import { HomeComponent } from './home/home.component';

const routes: Routes = [

  { path: '', component: HomeComponent }

];

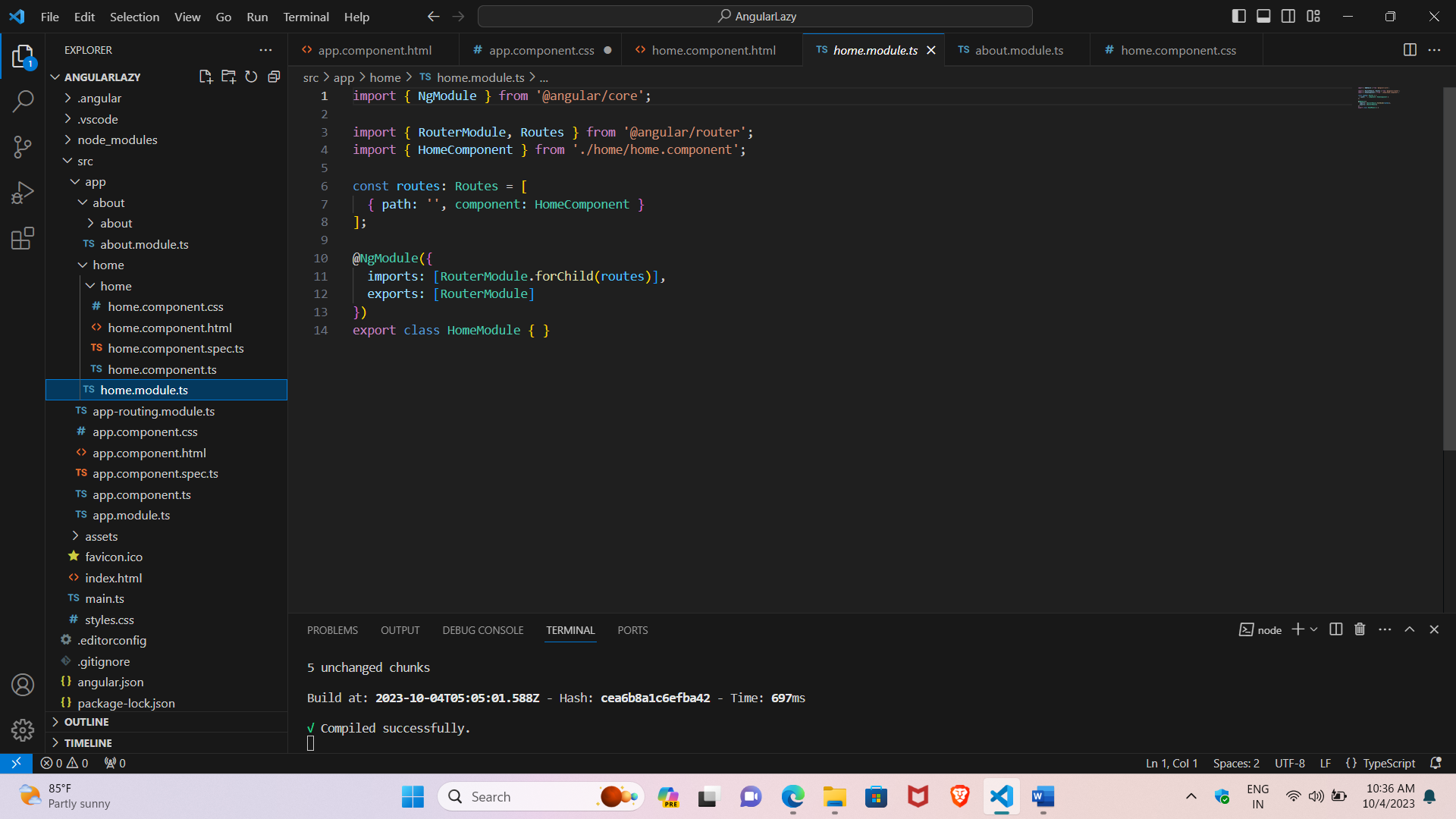
@NgModule({

  imports: [RouterModule.forChild(routes)],

  exports: [RouterModule]

})

export class HomeModule { }



**App.component.html**

<header>

<nav class="container">

  <ul>

   <h1><li><a routerLink="/home">Home</a></li></h1>

   <h1><li><a routerLink="/about">About</a></li></h1>

  </ul>

</nav>

<router-outlet></router-outlet>

</header>

**App module.ts**

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

import { BrowserModule } from '@angular/platform-browser';

import { RouterModule, Routes } from '@angular/router';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

const appRoutes: Routes = [

  { path: 'home', loadChildren: () => import('./home/home.module').then(m => m.HomeModule) },

  { path: 'about', loadChildren: () => import('./about/about.module').then(m => m.AboutModule) },

  { path: '', redirectTo: 'home', pathMatch: 'full' },

  { path: '\*\*', redirectTo: 'home' }

];

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule,

    RouterModule.forRoot(appRoutes)

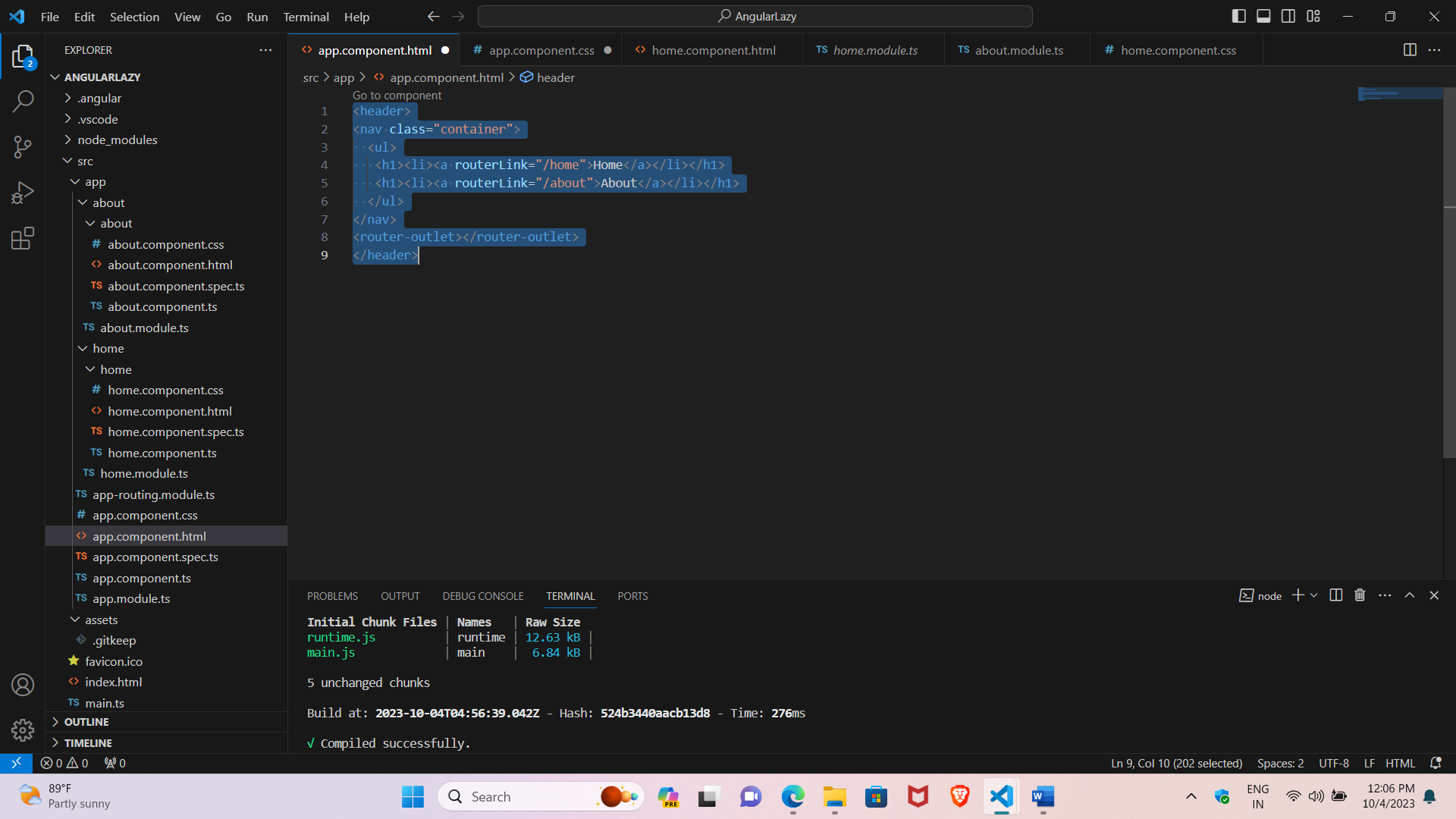
  ],

  providers: [],

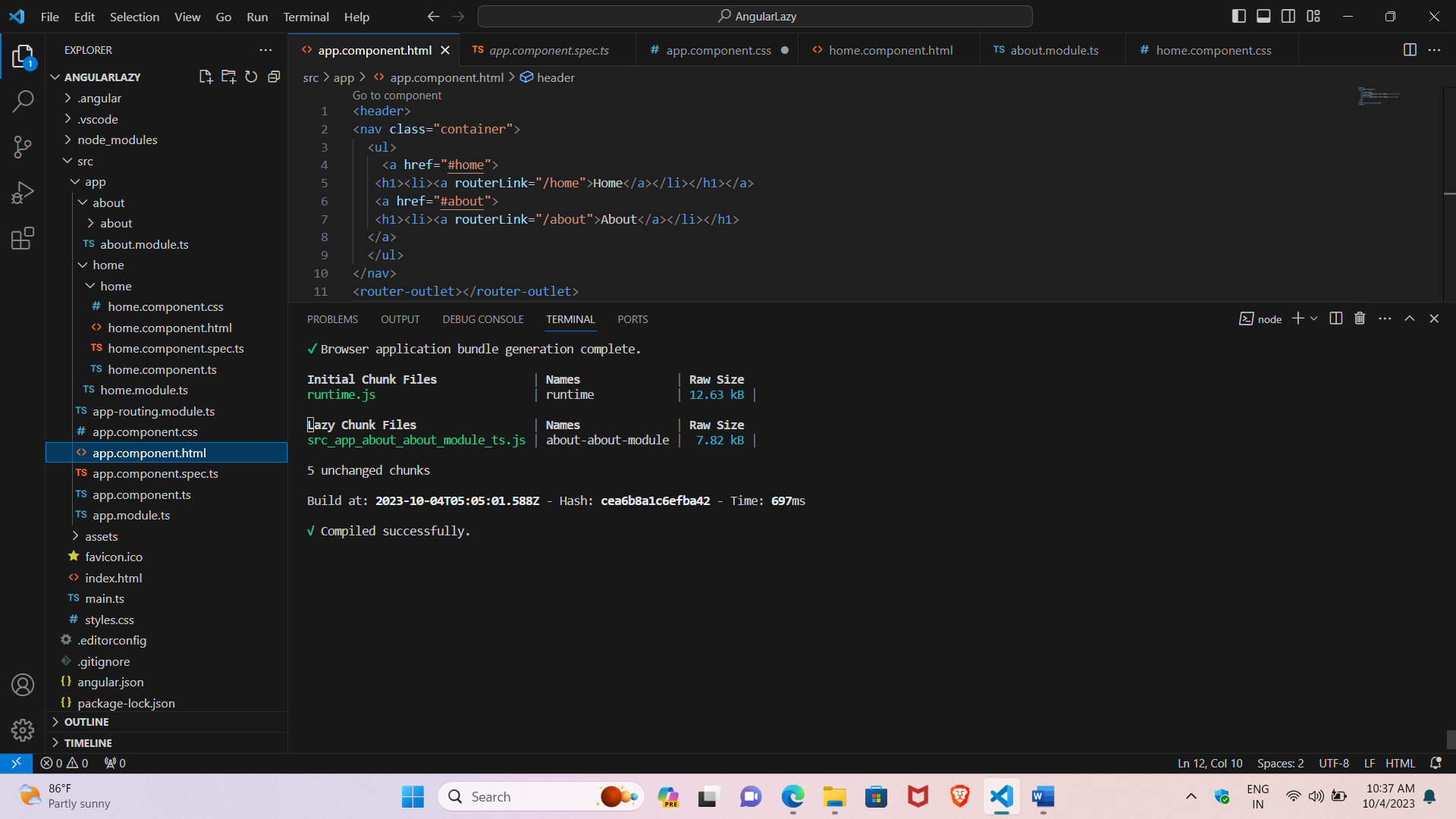
  bootstrap: [AppComponent]

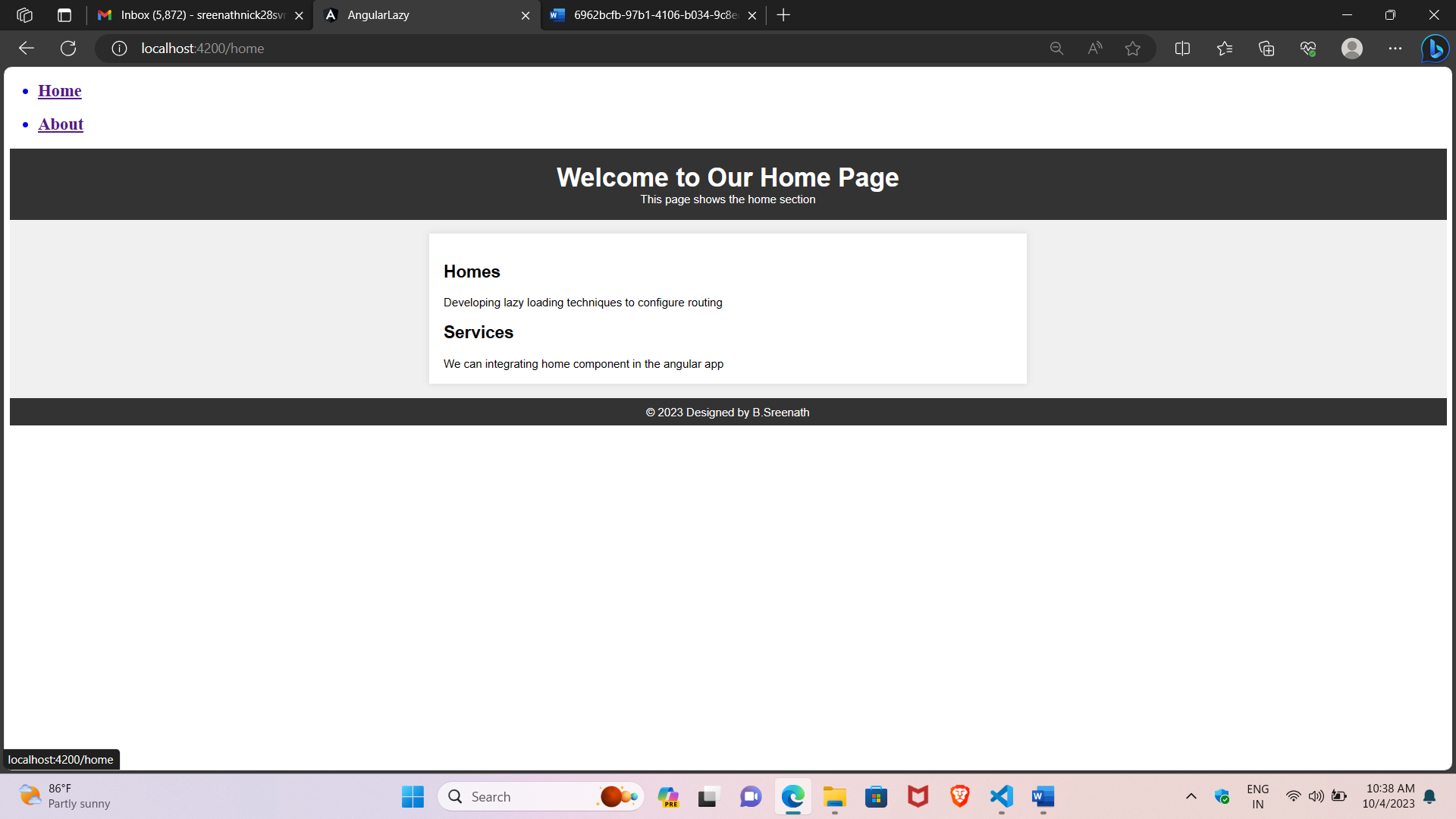
})

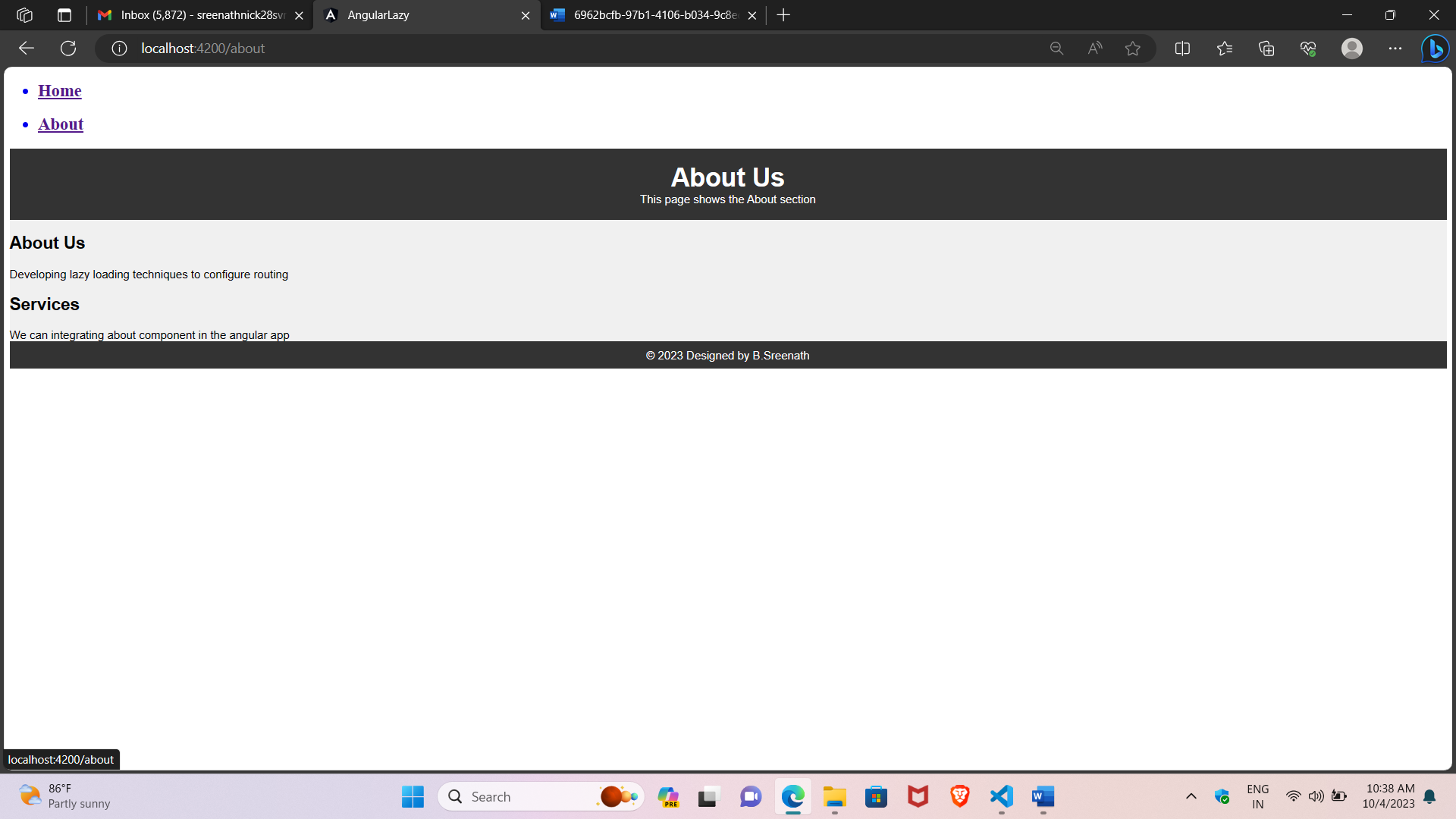
export class AppModule { }



**Output:**

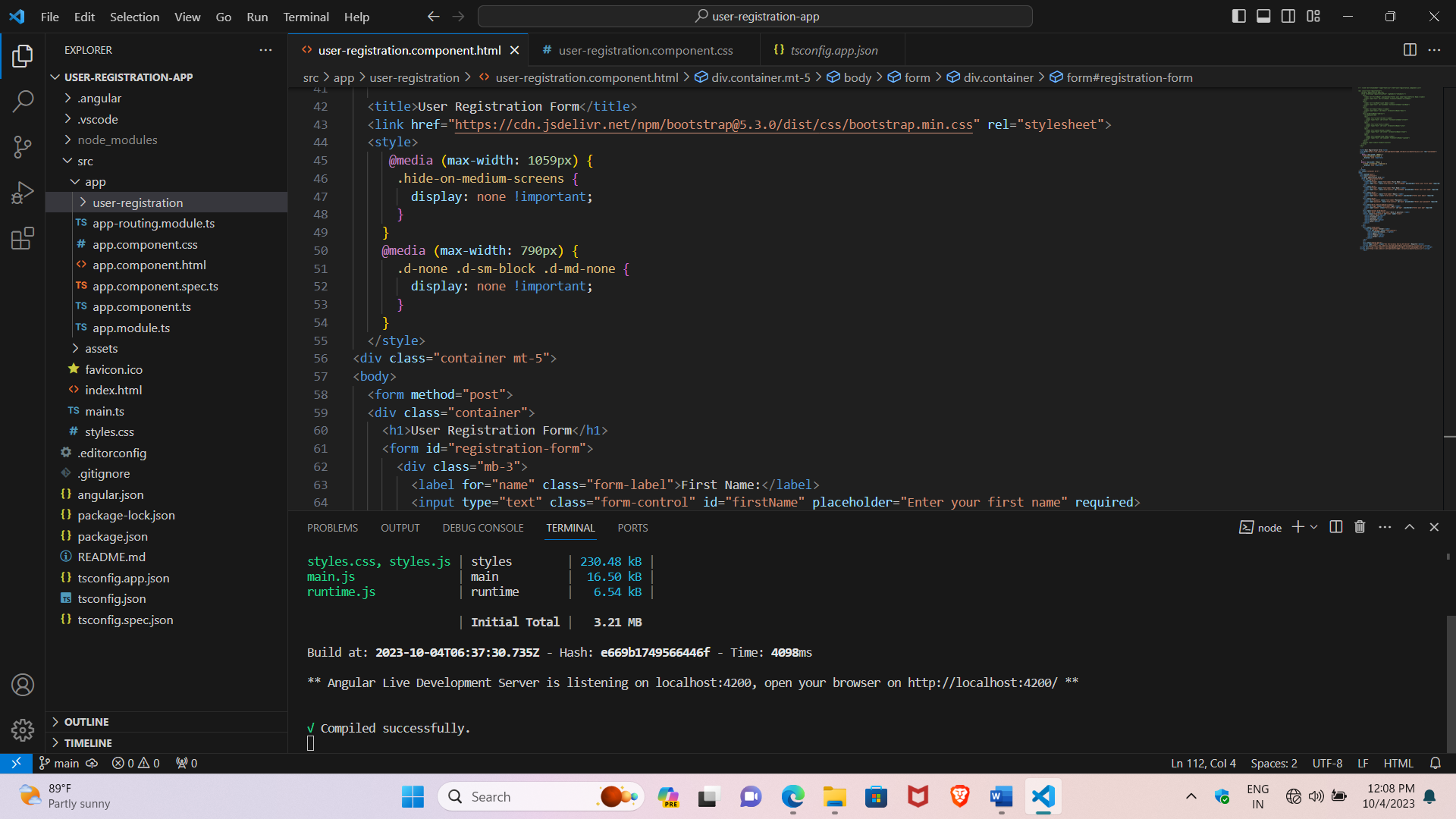






2) Write a program to create user with address fields with validation using reactive forms?

Source:



User registration.component.html

<title>User Registration Form</title>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

  <style>

     @media (max-width: 1059px) {

      .hide-on-medium-screens {

        display: none !important;

      }

    }

    @media (max-width: 790px) {

      .d-none .d-sm-block .d-md-none {

        display: none !important;

      }

    }

  </style>

<div class="container mt-5">

<body>

  <form method="post">

  <div class="container">

    <h1>User Registration Form</h1>

    <form id="registration-form">

      <div class="mb-3">

        <label for="name" class="form-label">First Name:</label>

        <input type="text" class="form-control" id="firstName" placeholder="Enter your first name" required>

      </div>

      <div class="mb-3">

        <label for="name" class="form-label">Full Name:</label>

        <input type="text" class="form-control" id="lastName" placeholder="Enter your last name" required>

      </div>

      <div class="mb-3">

        <label for="email" class="form-label">Email:</label>

        <input type="email" class="form-control" id="email" placeholder="Enter your email" required>

      </div>

      <div class="mb-3">

        <label for="Password" class="form-label">Password:</label>

        <input type="password" class="form-control" id="pass" placeholder="Enter your password" required>

      </div>

      <div class="mb-3 hide-on-medium-screens">

        <label for="age" class="form-label">Age:</label>

        <input type="number" class="form-control" id="age"  placeholder="Enter your age" required>

      </div>

      <div class="d-none d-md-block">

      <label for="Role" class="form-label">Role of position:</label>

      <select class="form-select" id="role1" name="role1">

        <option>select Role</option>

        <option>frontend</option>

        <option>backend</option>

        <option>fullstack</option>

        <option>None</option>

      </select>

      <br>

    </div>

      <div class="d-md-none">

        <label for="gender">Gender</label>

          <select id="gender" class="form-control">

            <option selected>Choose...</option>

            <option>Male</option>

            <option>Female</option>

            <option>Other</option>

          </select>

      </div>

      <br>

      <div class="d-grid gap-2">

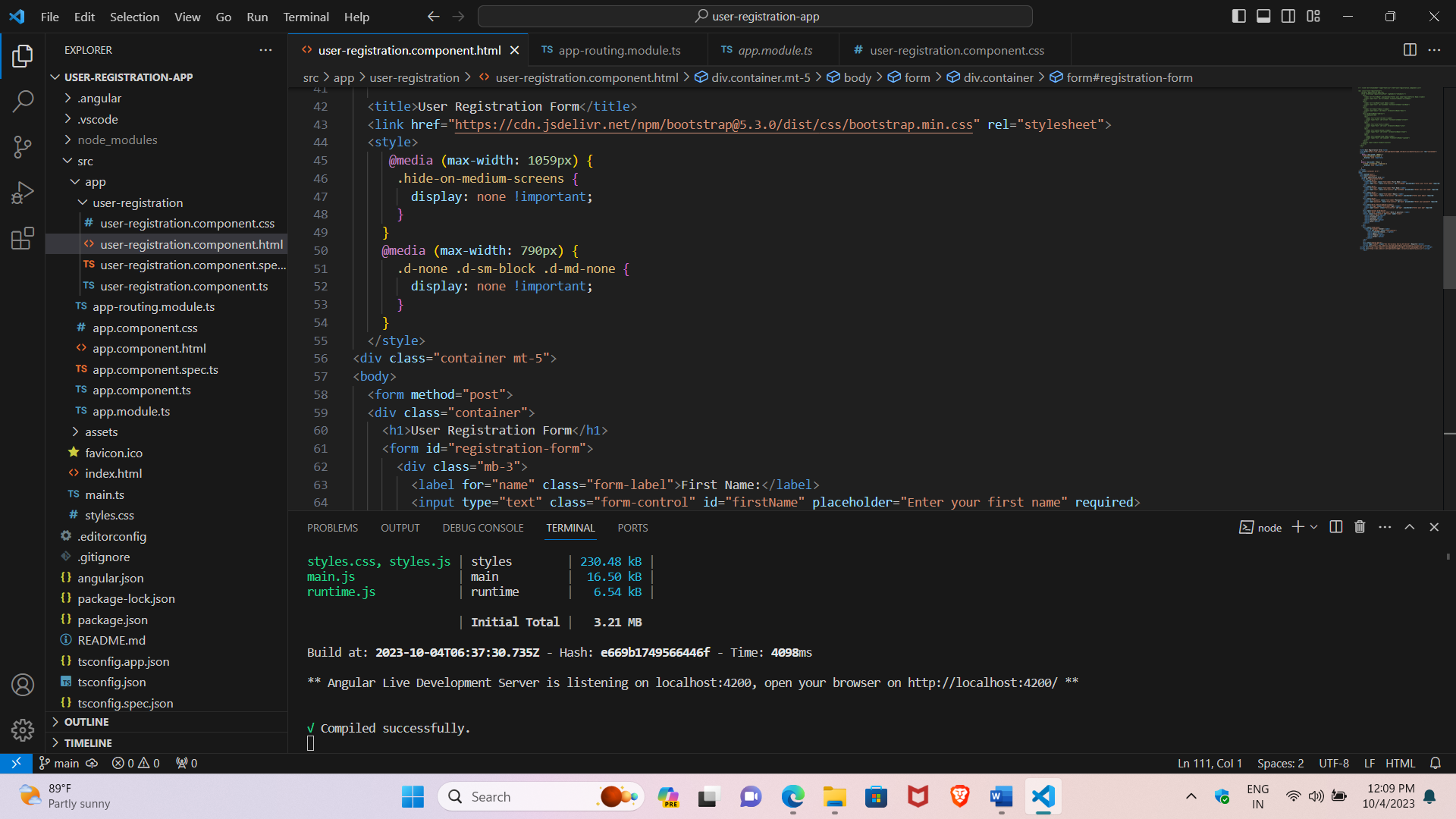
      <button type="submit" class="btn btn-primary btn-lg btn-block" >Register</button>

      <script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.min.js"></script>

      </div>



User-registration.component.ts

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

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

@Component({

  selector: 'app-user-registration',

  templateUrl: './user-registration.component.html',

  styleUrls: ['./user-registration.component.css']

})

export class UserRegistrationComponent {

  registrationForm: FormGroup;

  constructor(private fb: FormBuilder) {

    this.registrationForm = this.fb.group({

      firstName: ['', [Validators.required, Validators.minLength(2)]],

      lastName: ['', [Validators.required, Validators.minLength(2)]],

      email: ['', [Validators.required, Validators.email]],

      address: this.fb.group({

        street: ['', Validators.required],

        city: ['', Validators.required],

        state: ['', Validators.required],

        zipCode: ['', [Validators.required, Validators.pattern(/^\d{5}$/)]]

      })

    });

  }

  onSubmit() {

    if (this.registrationForm.valid) {

      // Handle form submission here

      console.log('Form submitted:', this.registrationForm.value);

    } else {

      // Form is invalid, display validation errors

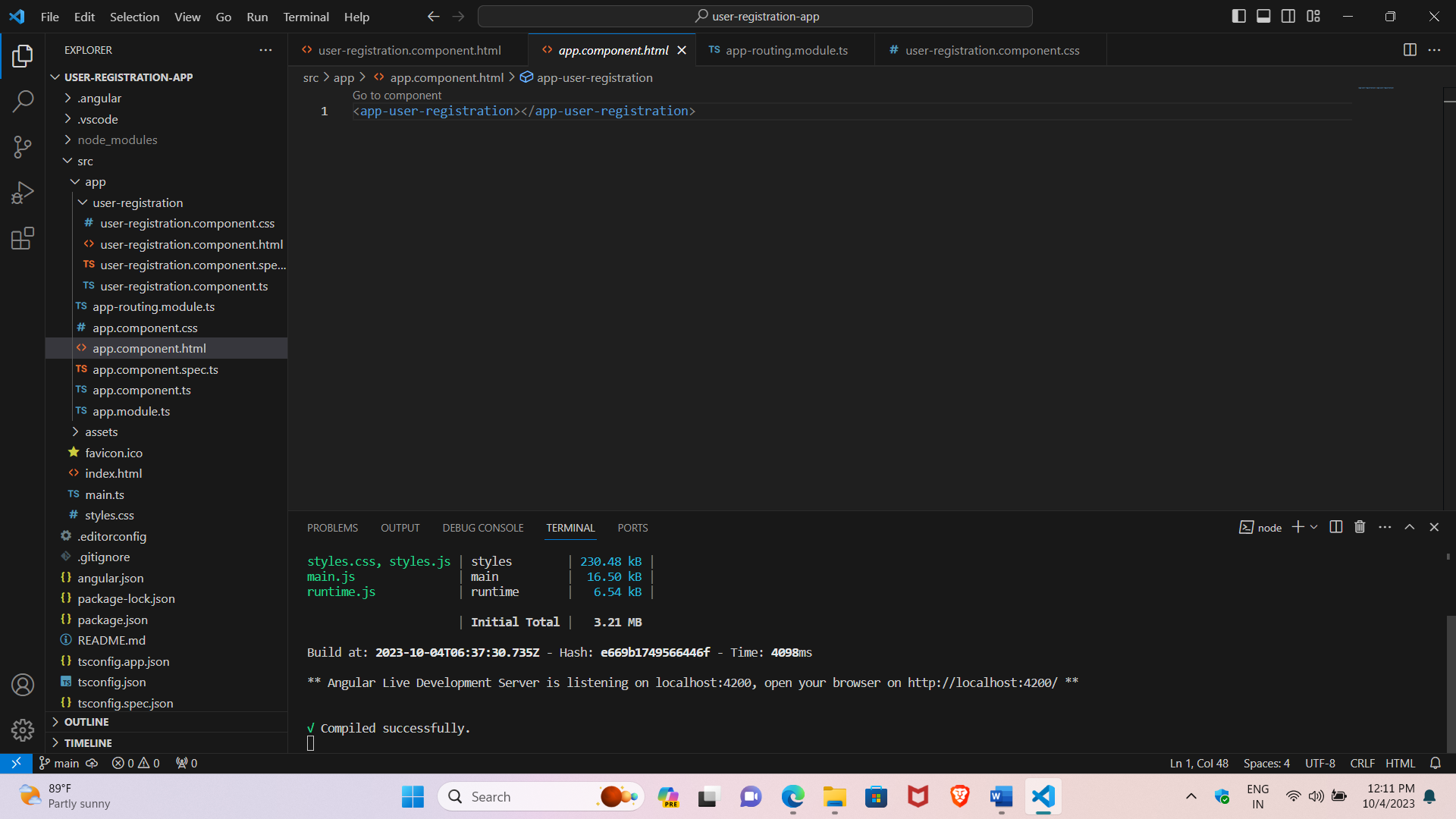
      alert('Please fill out the form correctly.');

    }

  }

}

App cmponent.html



**App component.ts**

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'user-registration-app';

}

**App.module.ts**

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

import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { UserRegistrationComponent } from './user-registration/user-registration.component';

@NgModule({

  declarations: [

    AppComponent,

    UserRegistrationComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule

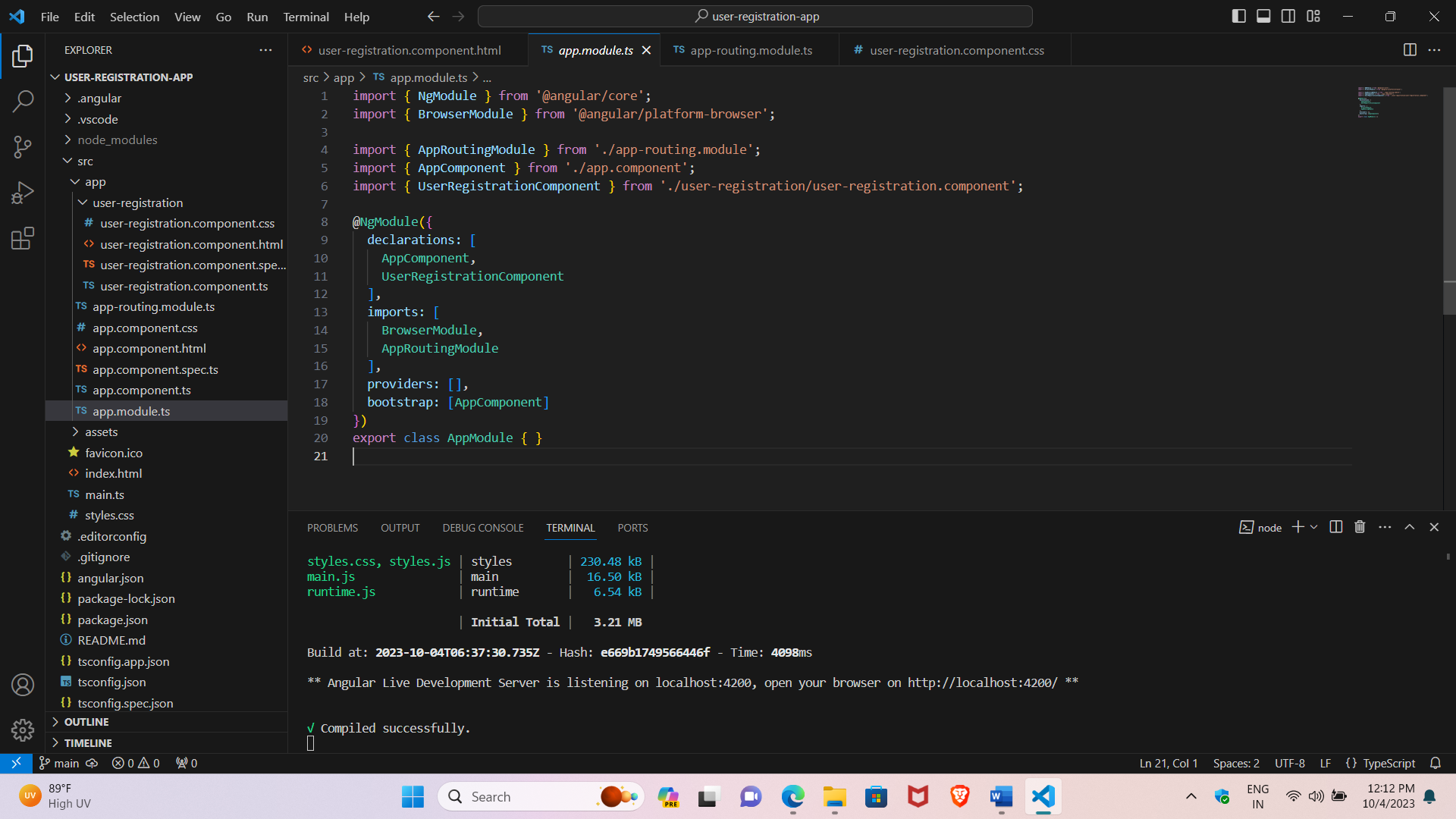
  ],

  providers: [],

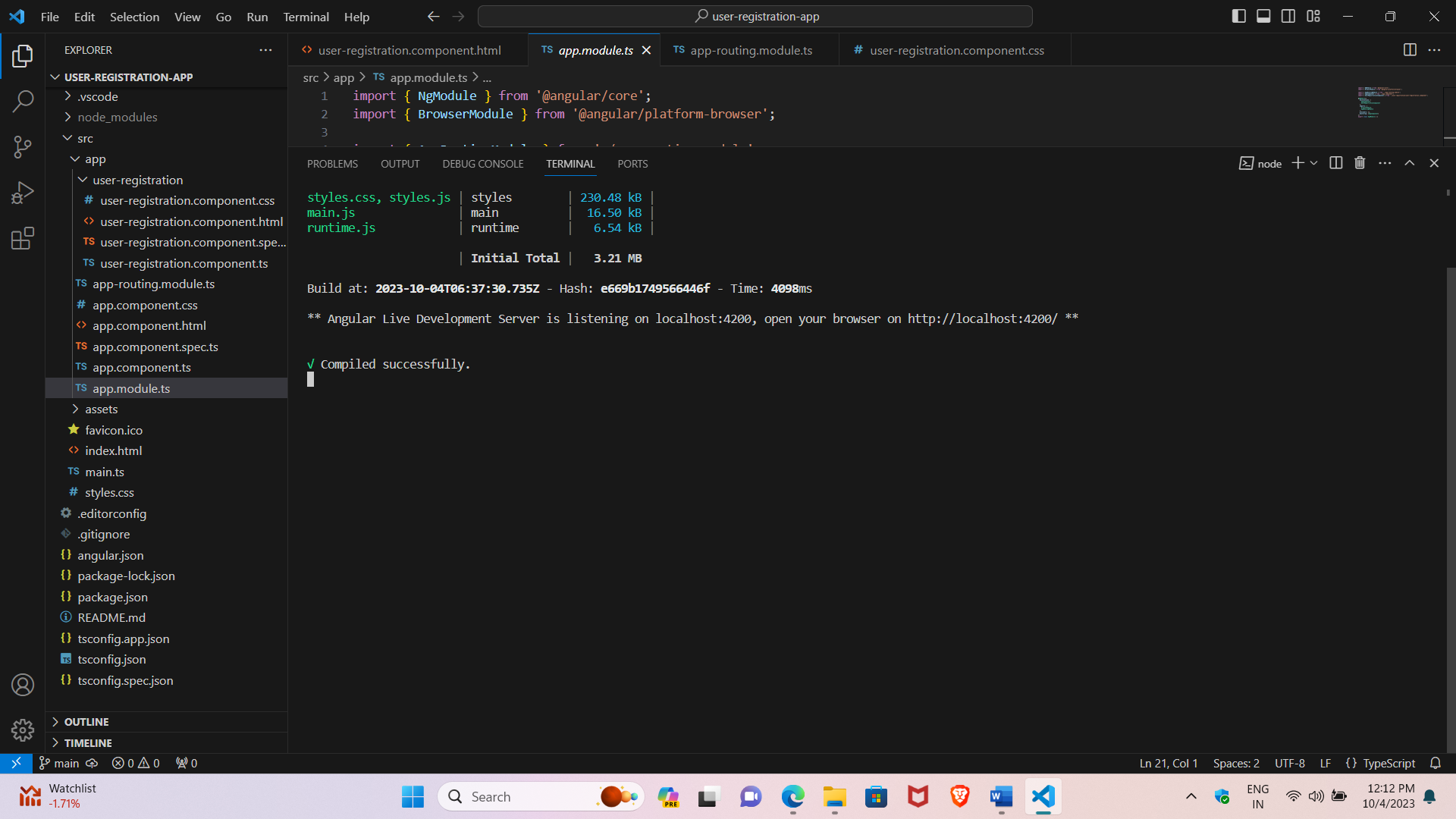
  bootstrap: [AppComponent]

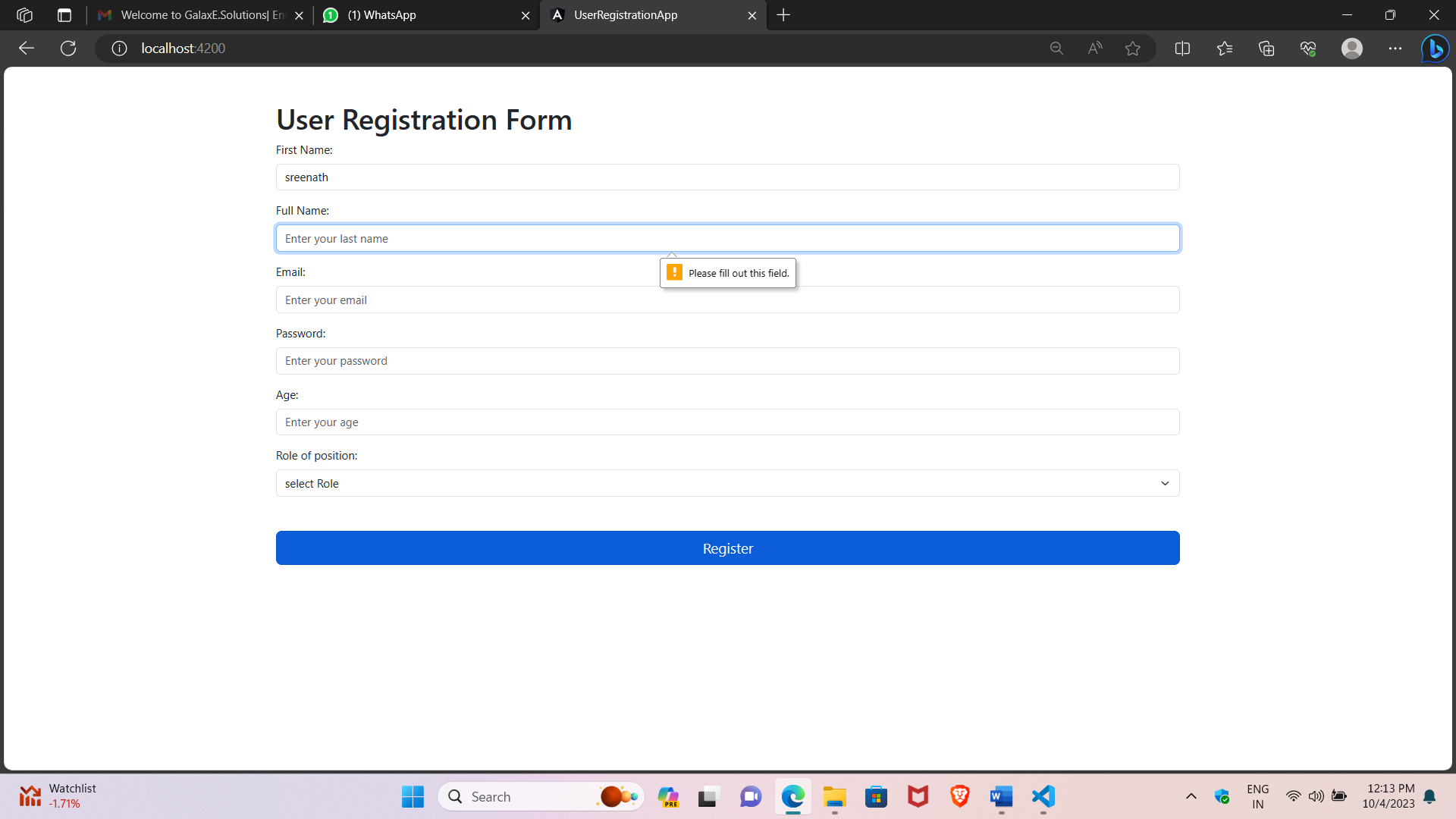
})

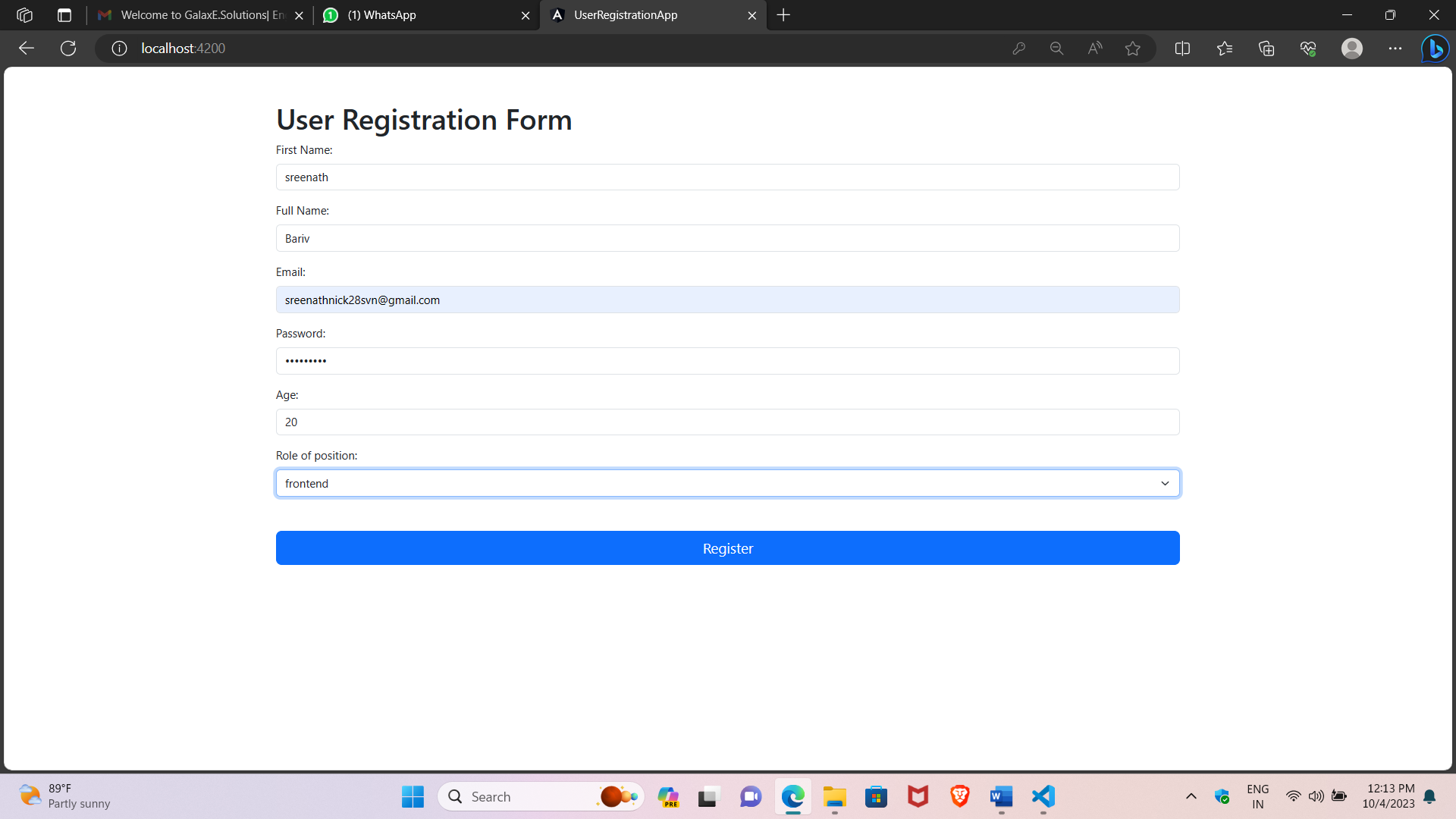
export class AppModule { }

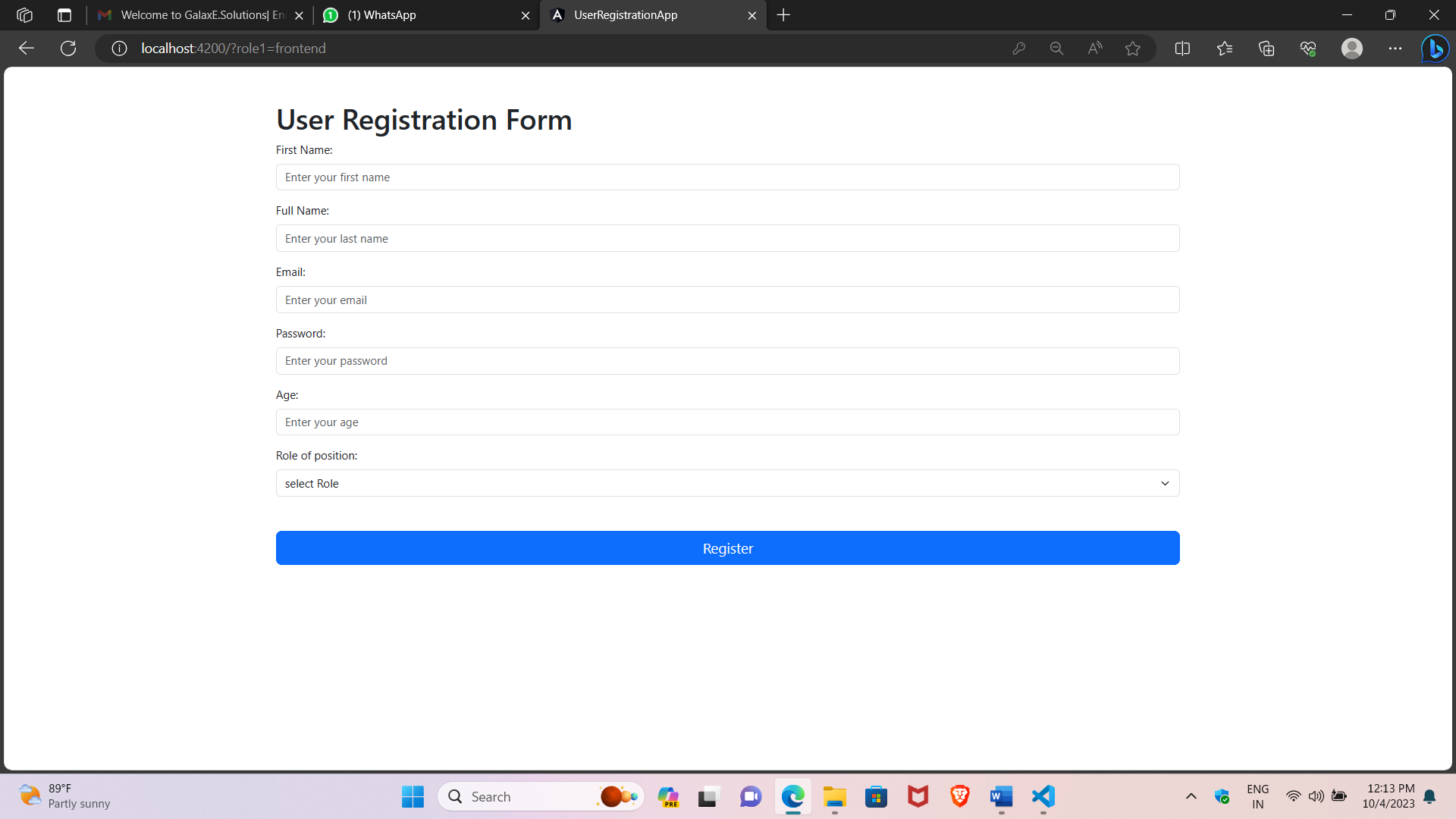


**Output:**



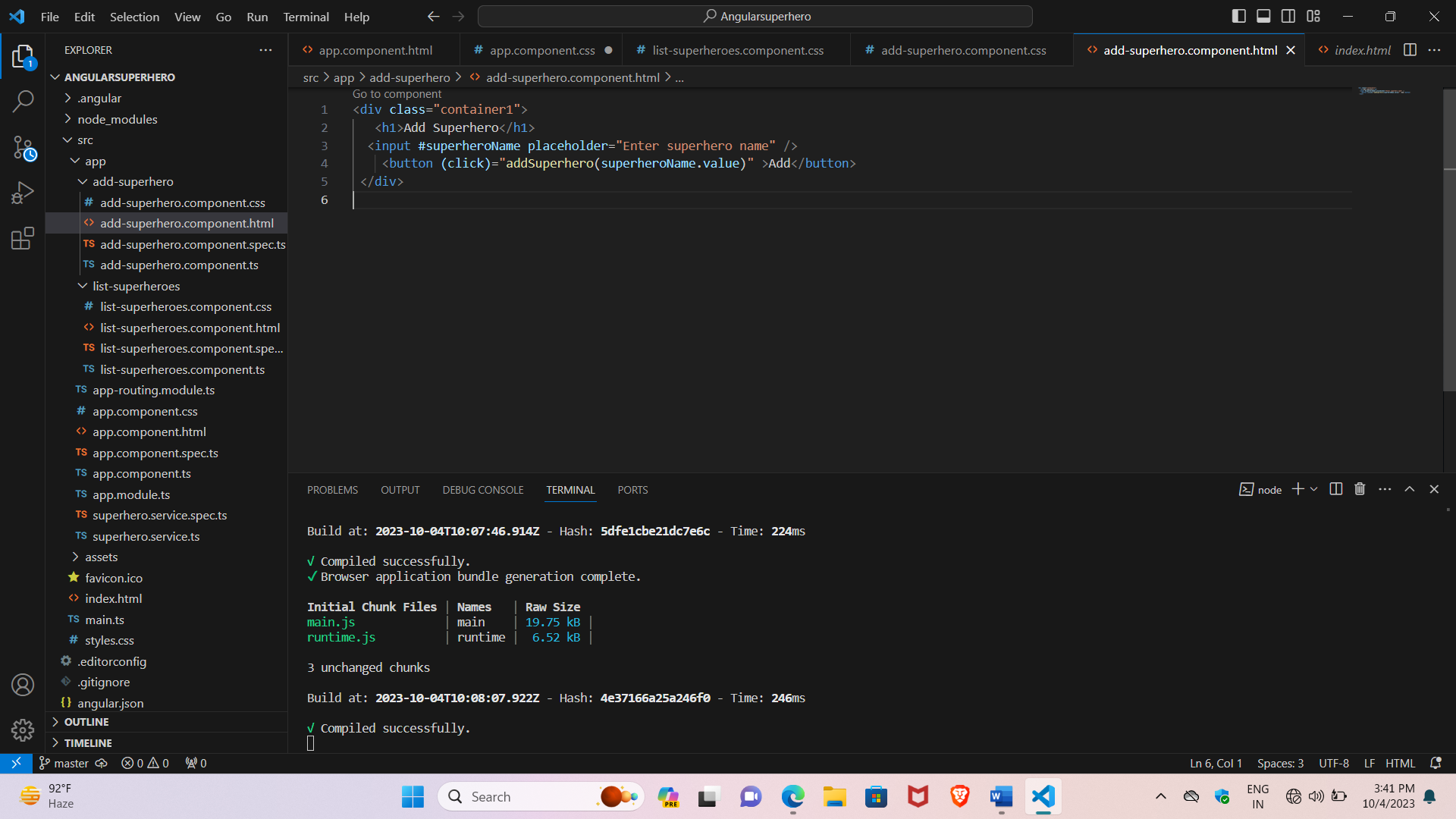






3) Write a program to add and retrieve super heroes using services?

Source:



Add-superhero component:

Html:

<div class="container1">

   <h1>Add Superhero</h1>

  <input #superheroName placeholder="Enter superhero name" />

    <button (click)="addSuperhero(superheroName.value)" >Add</button>

 </div>

CSS:

.container1{

    background-image: linear-gradient(to right, #3dc0c0, #3c8ca0);

    padding: 50px;

    border-radius: 10px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

    max-width: 100%;

    margin: 50px;

    text-align: left;

}

input {

        width: 100%;

        padding: 12px 20px;

        margin: 8px 0;

        box-sizing: border-box;

        border: none;

        background-color: white;

        color: black;

        font-size: larger;

    }

button{

    font-size: 20px;

    border-radius: 12px;

    background-color: gray;

}

Typescript:

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

import { SuperheroService } from '../superhero.service';

@Component({

  selector: 'app-add-superhero',

  templateUrl: './add-superhero.component.html',

  styleUrls: ['./add-superhero.component.css']

})

export class AddSuperheroComponent {

  constructor(private superheroService: SuperheroService) {}

  addSuperhero(name: string) {

    if (name.trim() !== '') {

      this.superheroService.addSuperhero(name);

    }

  }

}

List-superhero component:

Html:

<div class="container">

        <h1>Superheroes List</h1>

        <ul>

          <li \*ngFor="let superhero of superheroes">{{ superhero }}</li>

        </ul>

      </div>

Css:

.container{

    background-image: linear-gradient(to right, #e4e4e4, #ebebeb);

    padding: 50px;

    border-radius: 10px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

    max-width: 100%;

    margin: 50px;

    text-align: left;

}

li{

    font-size: 30px;

}

Typescript:

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

import { SuperheroService } from '../superhero.service';

@Component({

  selector: 'app-list-superheroes',

  templateUrl: './list-superheroes.component.html',

  styleUrls: ['./list-superheroes.component.css']

})

export class ListSuperheroesComponent {

  superheroes: string[] = [];

  constructor(private superheroService: SuperheroService) {}

  ngOnInit() {

    this.superheroes = this.superheroService.getSuperheroes();

  }

}

App-component:

Html:

<nav>

    <ul>

      <li id="li1"><a routerLink="/add-superhero">Add Superhero</a></li>

      <li id ="li2"><a routerLink="/list-superheroes">List Superheroes</a></li>

    </ul>

  </nav>

  <router-outlet></router-outlet>

Css:

ul {

    list-style-type: none;

    margin: 0;

    padding: 0;

    overflow: hidden;

  }

li{

float: left;

}

li a {

    display: block;

    padding: 20px;

    font-size: 30px ;

    background-color: #dddddd;

  }

Typescript:

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'superhero';

}

App module.ts

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

import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { AddSuperheroComponent } from './add-superhero/add-superhero.component';

import { ListSuperheroesComponent } from './list-superheroes/list-superheroes.component';

@NgModule({

  declarations: [

    AppComponent,

    AddSuperheroComponent,

    ListSuperheroesComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

superheroservices.ts

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

@Injectable({

providedIn: 'root'

})

export class SuperheroService {

private superheroes: string[] = [];

constructor() { }

addSuperhero(superhero: string) {

 this.superheroes.push(superhero);

}

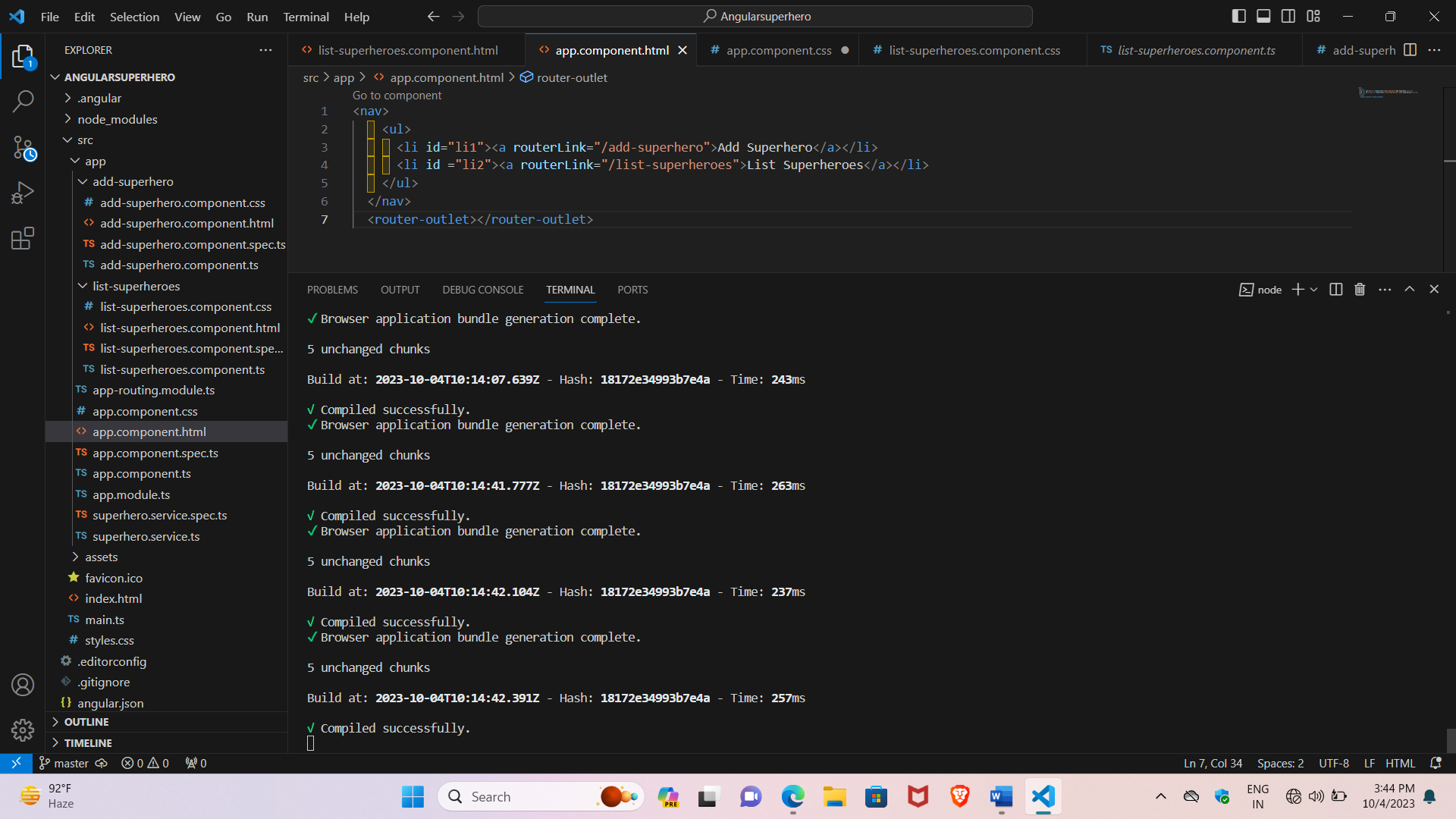
getSuperheroes() {

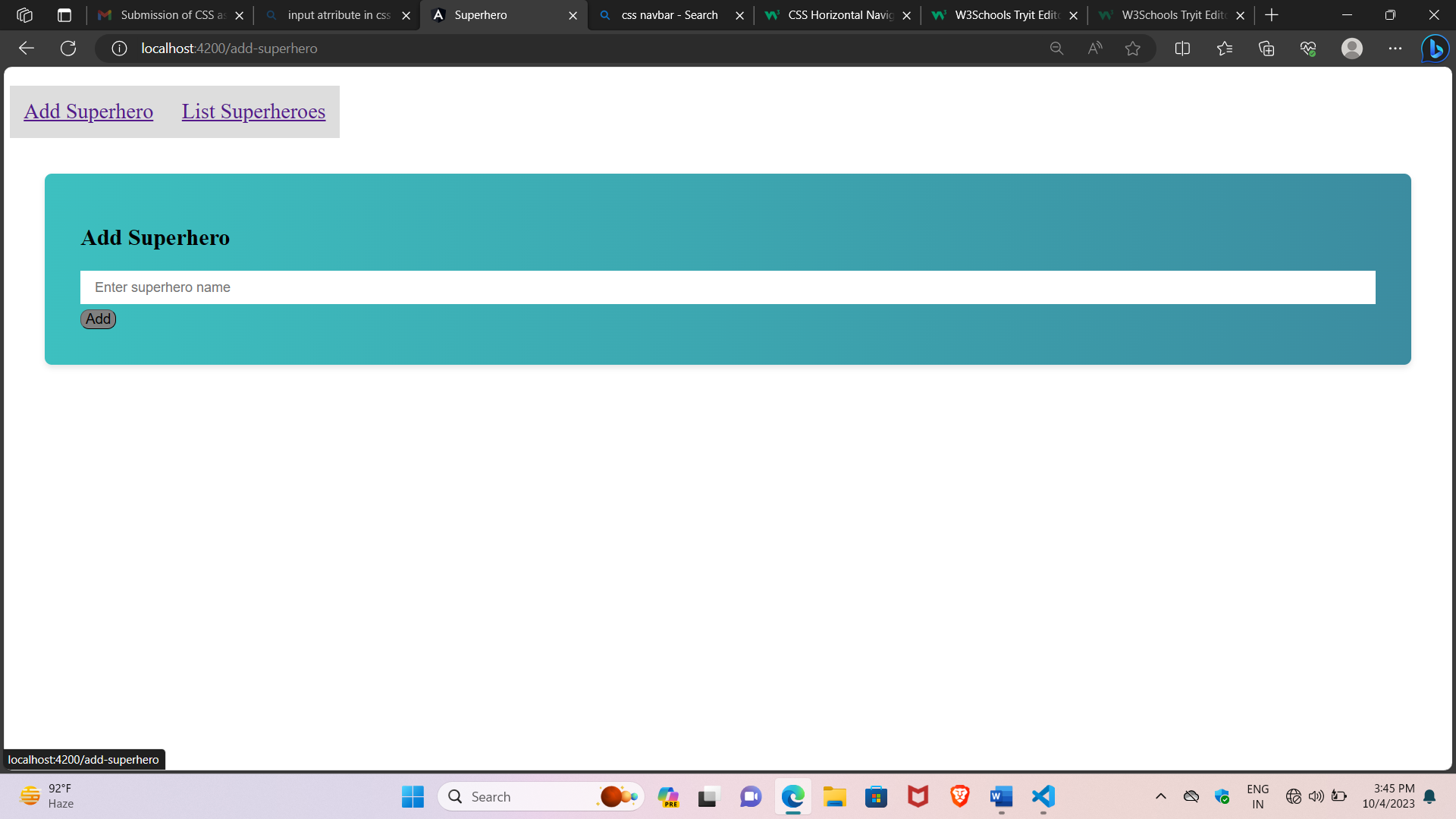
return this.superheroes;

}

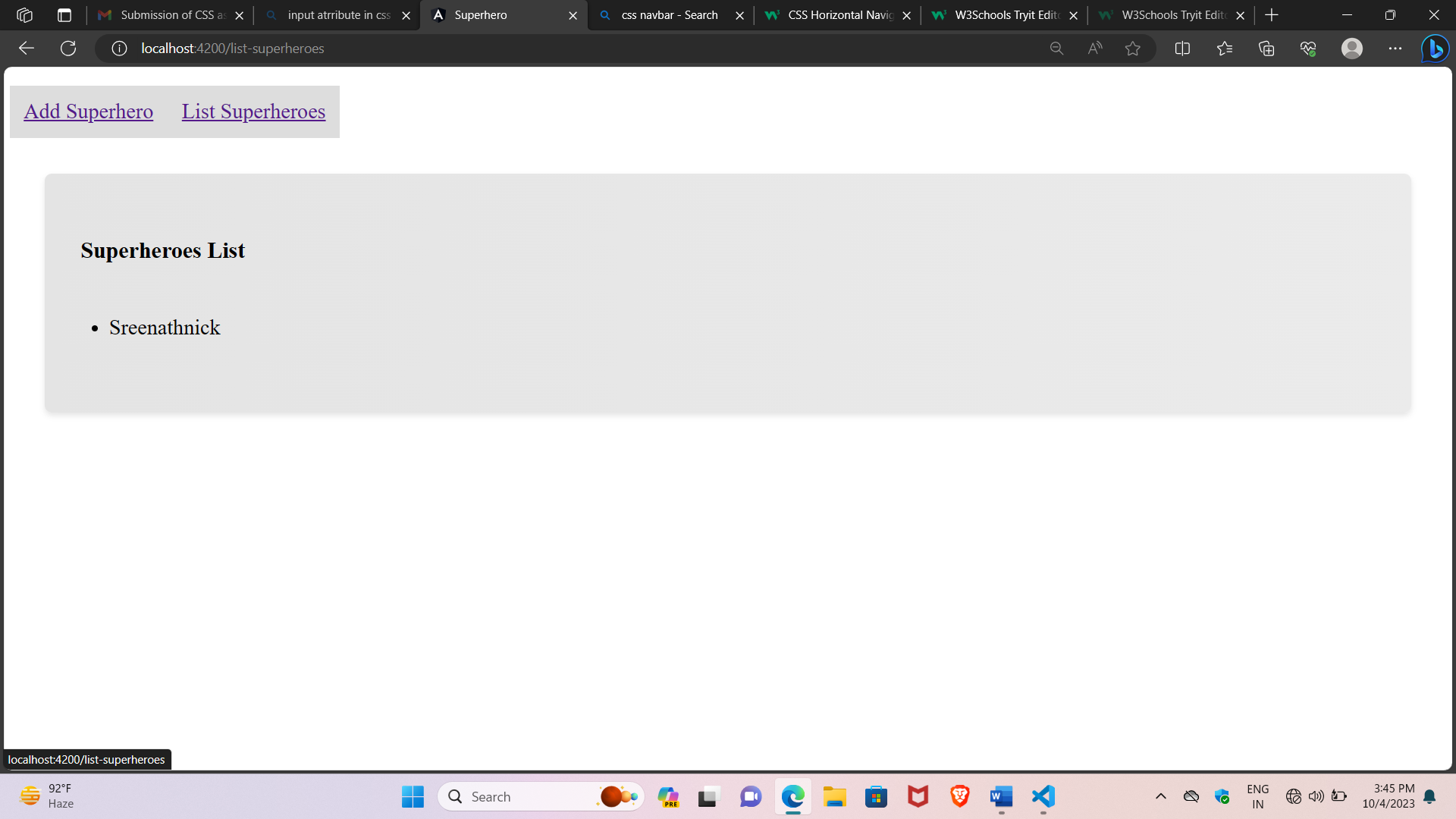
}

Output:



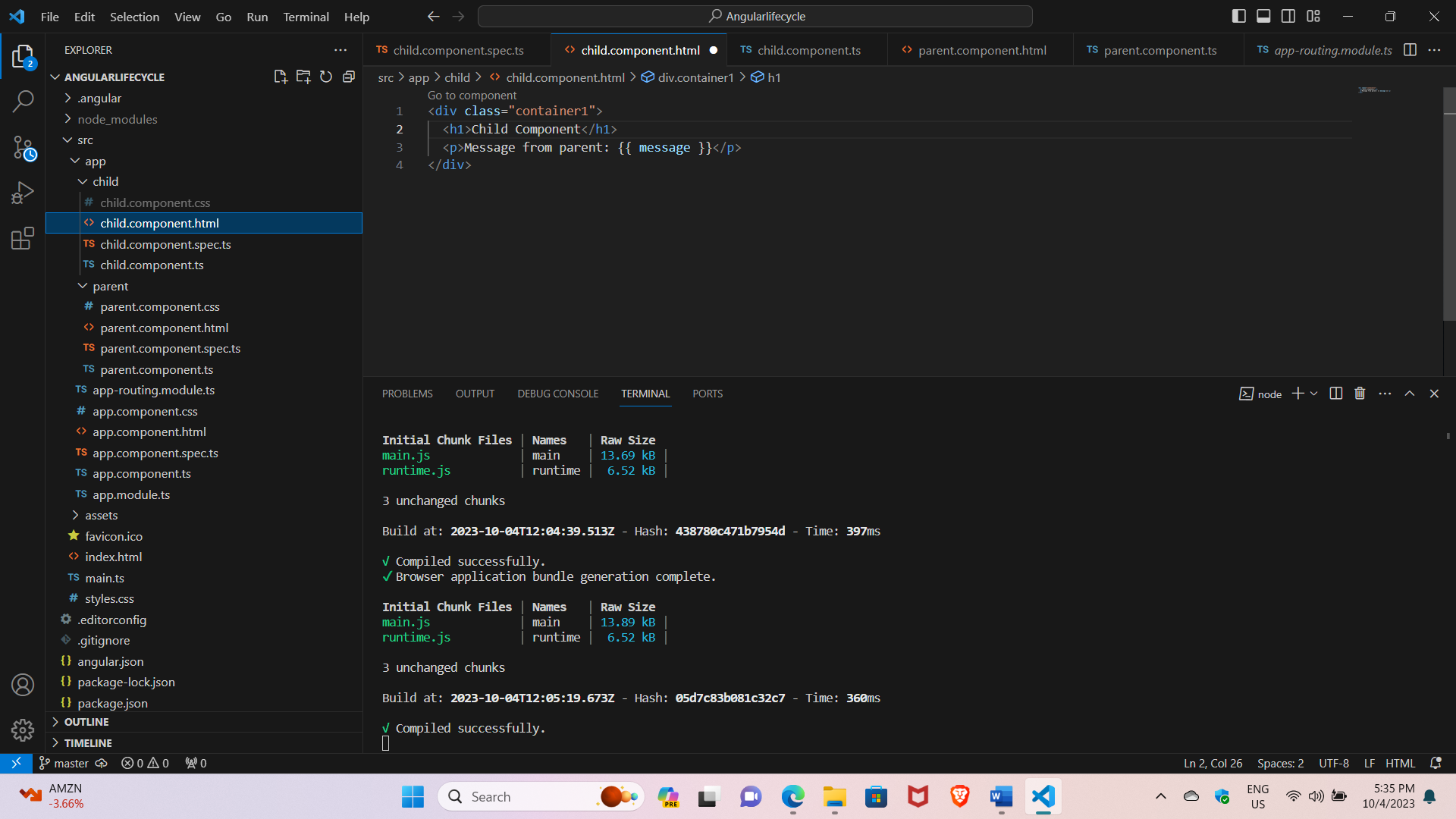






4) Write a program to demonstrate angular life cycle hooks with component communication?

Source:



Child component:

Html:

<div class="container1">

  <h1>Child Component</h1>

  <p>Message from parent: {{ message }}</p>

</div>

Css:

p{

    font-size: 20px;

}

Typescript:

// child.component.ts

import { Component, Input, OnChanges, SimpleChanges, OnInit, OnDestroy, AfterViewInit } from '@angular/core';

@Component({

  selector: 'app-child',

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

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

})

export class ChildComponent implements OnInit, OnChanges, OnDestroy, AfterViewInit {

  @Input() message: string | undefined;

  constructor() {

    console.log('Child Component - Constructor');

  }

  ngOnInit() {

    console.log('Child Component - ngOnInit');

  }

  ngOnChanges(changes: SimpleChanges) {

    console.log('Child Component - ngOnChanges', changes);

  }

  ngAfterViewInit() {

    console.log('Child Component - ngAfterViewInit');

  }

  ngOnDestroy() {

    console.log('Child Component - ngOnDestroy');

  }

}

Parent component:

Html:

  <div class="container">

    <h1>Parent Component</h1>

  <button class="bt1" (click)="changeMessage()"> New message from parent </button>

  <button class="bt1" (click)="changeMessage1()"> Alert2 </button>

  <button class="bt1" (click)="changeMessage2()"> Alert3 </button>Typescript:

Typescript:

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

@Component({

  selector: 'app-parent',

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

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

})

export class ParentComponent {

  message: string = 'Initial message from parent';

  changeMessage() {

    this.message = 'New message from parent';

  }

  changeMessage1() {

    this.message = 'Alert 2 from parent ';

  }

  changeMessage2() {

    this.message = 'Alert 3 from parent ';

  }

  constructor() {

    console.log('Parent Component - Constructor');

  }

  ngOnInit() {

    console.log('Parent Component - ngOnInit');

  }

  ngOnChanges() {

    console.log('Parent Component - ngOnChanges');

  }

  ngAfterViewInit() {

    console.log('Parent Component - ngAfterViewInit');

  }

  ngOnDestroy() {

    console.log('Parent Component - ngOnDestroy');

  }

}

App component:

Html:

<div class="container">

<app-parent></app-parent>

</div>

Css:

.container{

    background-image: linear-gradient(to right, #e4e4e4, #ebebeb);

    padding: 50px;

    border-radius: 10px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

    max-width: 100%;

    margin: 50px;

    text-align: left;

}

Typescript:

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'lifecycle';

}

App module.ts:

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

import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { ParentComponent } from './parent/parent.component';

import { ChildComponent } from './child/child.component';

@NgModule({

  declarations: [

    AppComponent,

    ParentComponent,

    ChildComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule

  ],

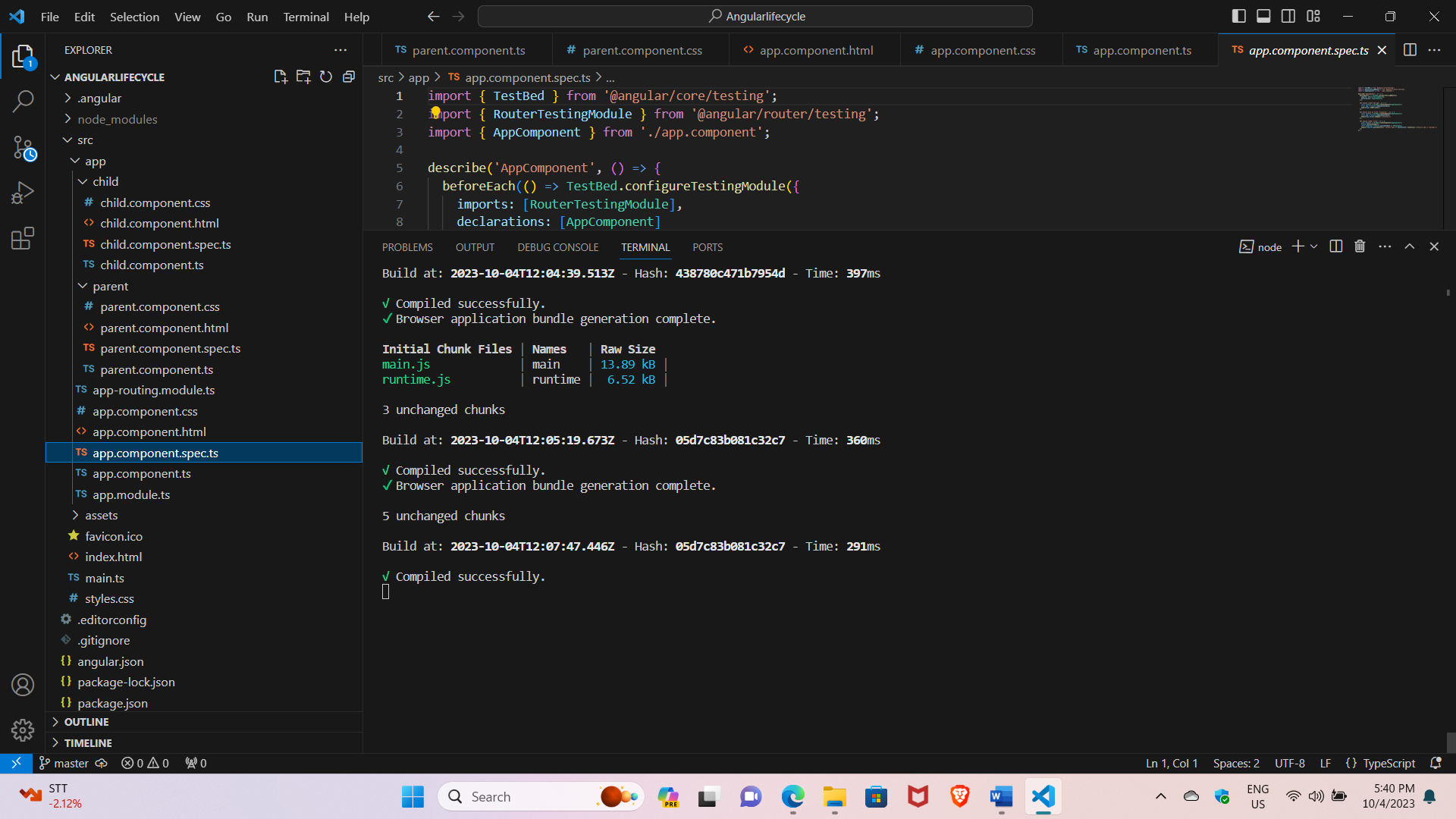
  providers: [],

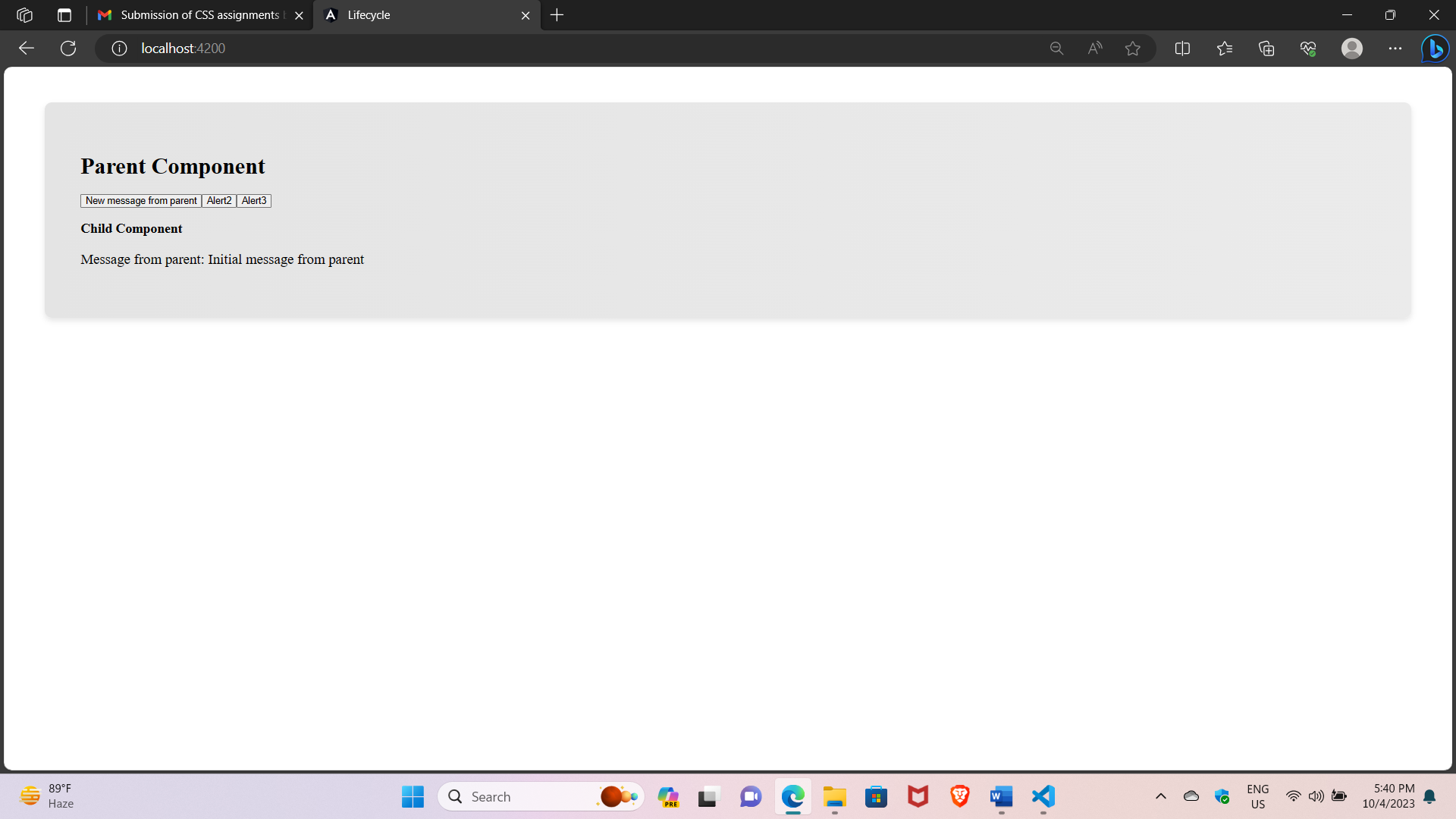
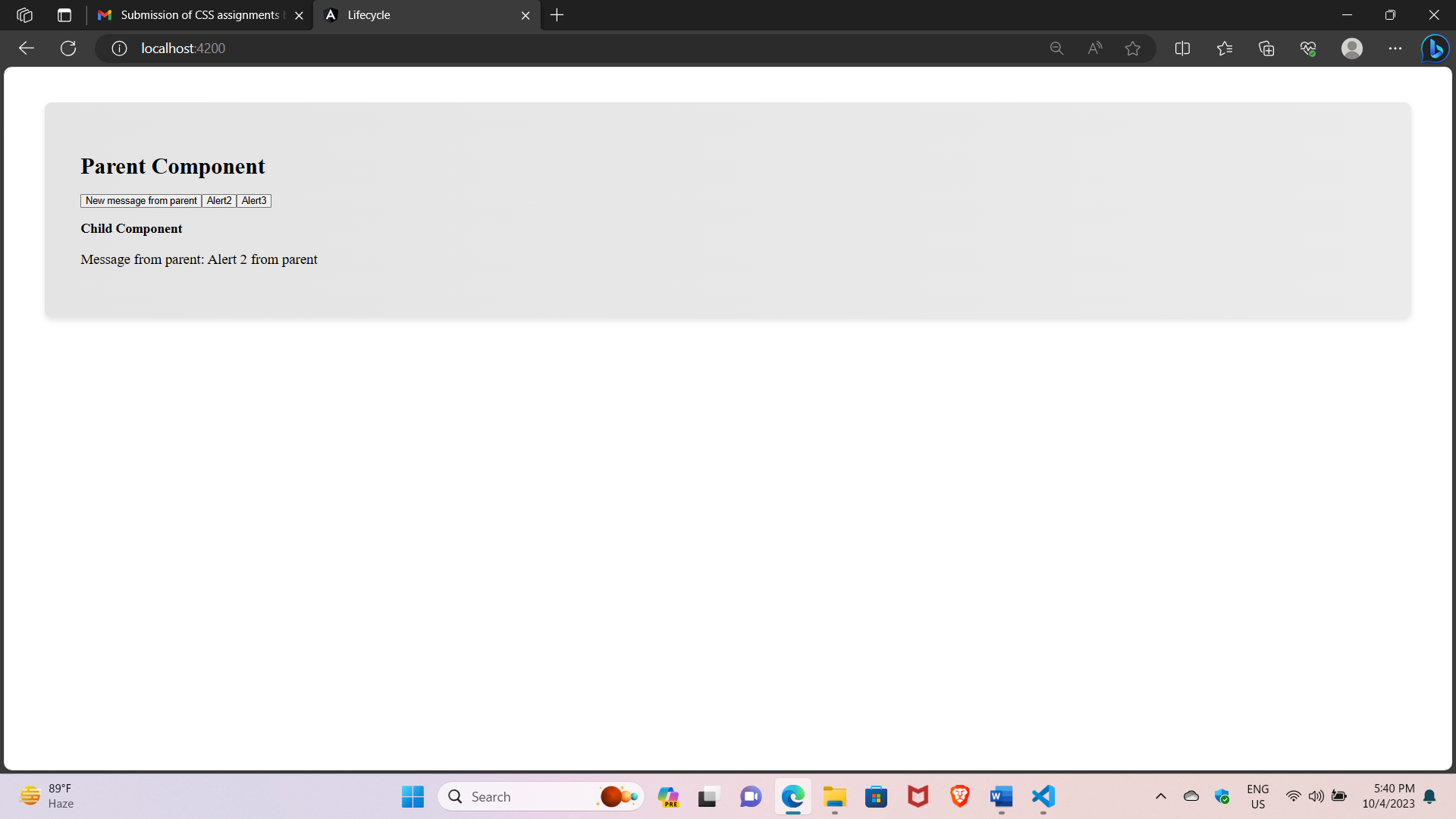
  bootstrap: [AppComponent]

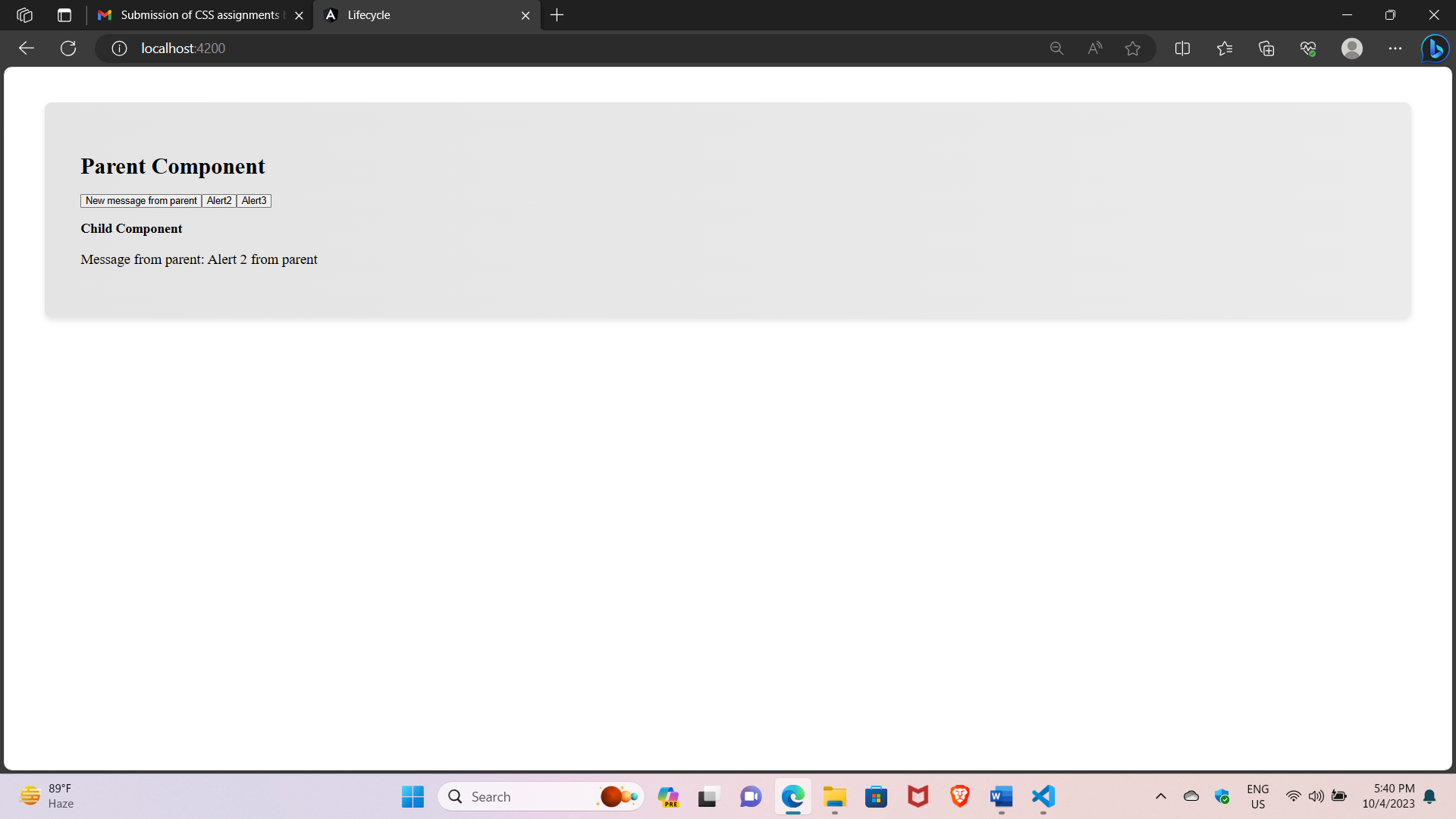
})

export class AppModule { }

Output:

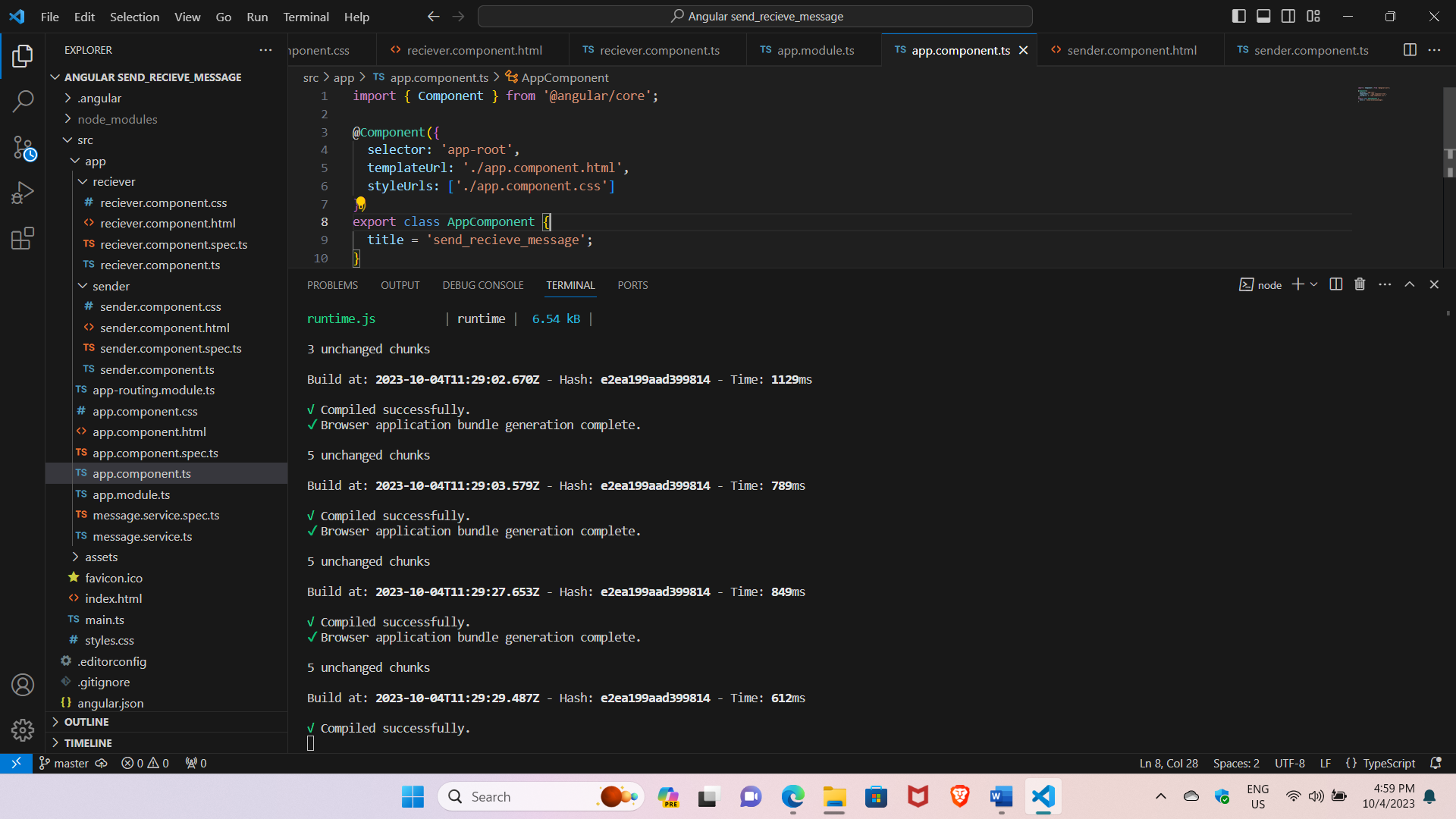






5) Write a program to send message using angular subject

Source:



Sender component:

Html:

<div class="container">

    <h1>Send Messages</h1>

    <p>Please enter the message you want to send</p>

          <input [(ngModel)]="message" placeholder="Type a message" />

          <button (click)="sendMessage()">Send</button>

        </div>

Css:

.container{

 background-image: linear-gradient(to right, #e4e4e4, #ebebeb);

 padding: 50px;

 border-radius: 10px;

 box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

 max-width: 100%;

 margin: 50px;

 text-align: left;

 }

 p{

    font-size: 20px;

 }

 input{

    width: 100%;

        padding: 12px 20px;

        margin: 8px 0;

        box-sizing: border-box;

        border: none;

        background-color: white;

        color: black;

        font-size: larger;

 }

 button{

    font-size: 20px;

    border-radius: 20px;

    background-color: whitesmoke;

}

Typescript:

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

import { MessageService } from '../message.service';

@Component({

  selector: 'app-sender',

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

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

})

export class SenderComponent {

  message = '';

  constructor(private messageService: MessageService) {}

  sendMessage() {

    if (this.message.trim()) {

      this.messageService.sendMessage(this.message);

      this.message = '';

    }

  }

}

**Receiver component:**

**Html:**

<div class="container1">

          <h1>Received Messages</h1>

          <ul>

            <li \*ngFor="let message of messages">{{ message }}</li>

          </ul>

          <button (click)="clearMessages()">Clear</button>

        </div>

**Css:**

.container1{

    background-image: linear-gradient(to right,  #e4e4e4, #ebebeb);

    padding: 50px;

    border-radius: 10px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

    max-width: 100%;

    margin: 50px;

    text-align: left;

}

li{

    font-size: 30px;

}

button{

    font-size: 20px;

    border-radius: 20px;

    background-color: whitesmoke;

}

**Typescript:**

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

import { MessageService } from '../message.service';

@Component({

  selector: 'app-receiver',

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

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

})

export class ReceiverComponent implements OnInit {

  messages: string[] = [];

  constructor(private messageService: MessageService) {}

  ngOnInit() {

    this.messageService.getMessage().subscribe((message: string) => {

      if (message) {

        this.messages.push(message);

      }

    });

  }

  clearMessages() {

    this.messages = [];

    this.messageService.clearMessages();

  }

}

**App component:**

**Html:**

<div>

    <app-sender></app-sender>

    <app-receiver></app-receiver>

  </div>

**Typescript:**

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'send\_recieve\_message';

}

**Appmodule.ts**

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

import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { ParentComponent } from './parent/parent.component';

import { ChildComponent } from './child/child.component';

@NgModule({

  declarations: [

    AppComponent,

    ParentComponent,

    ChildComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule

  ],

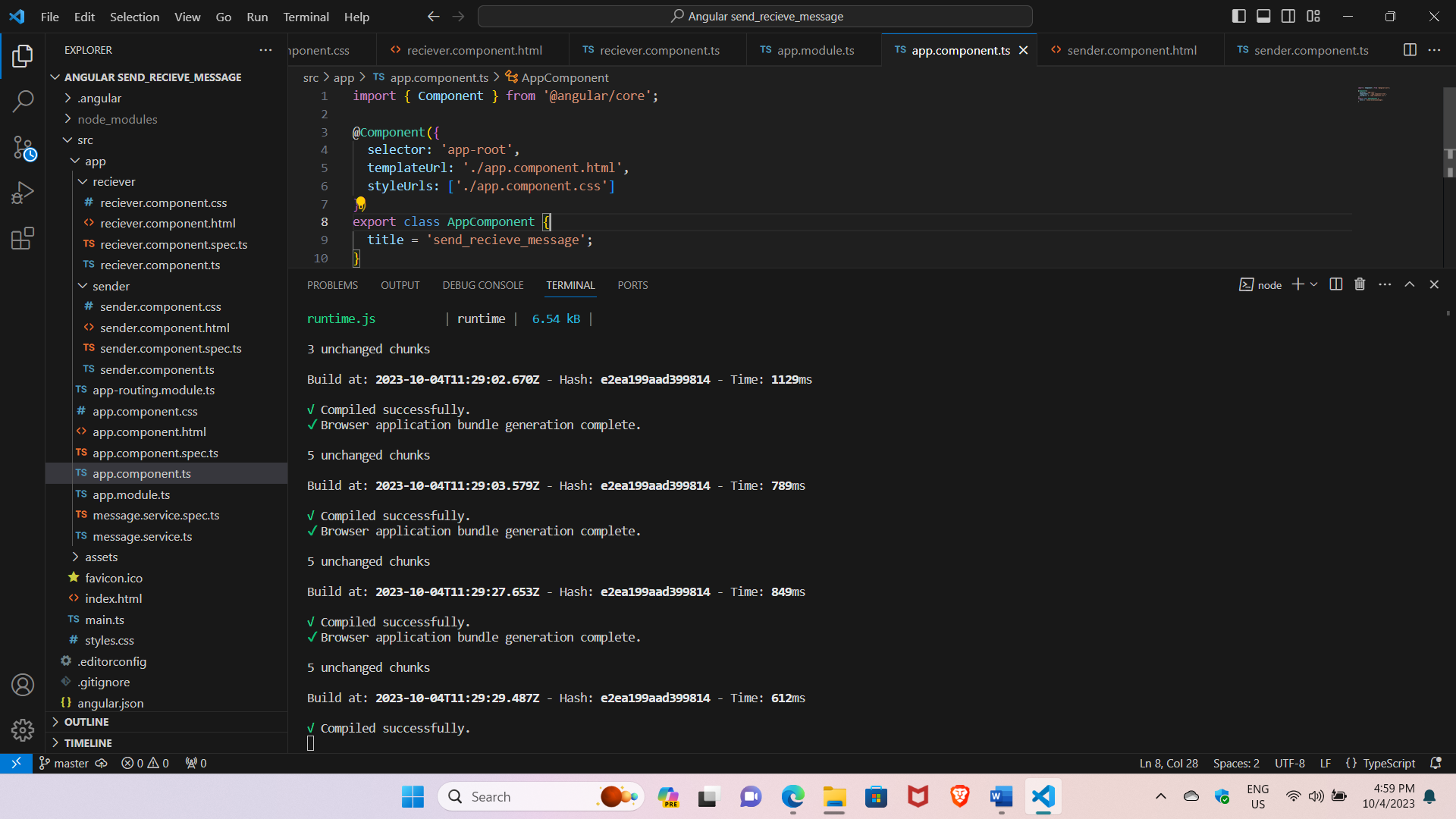
  providers: [],

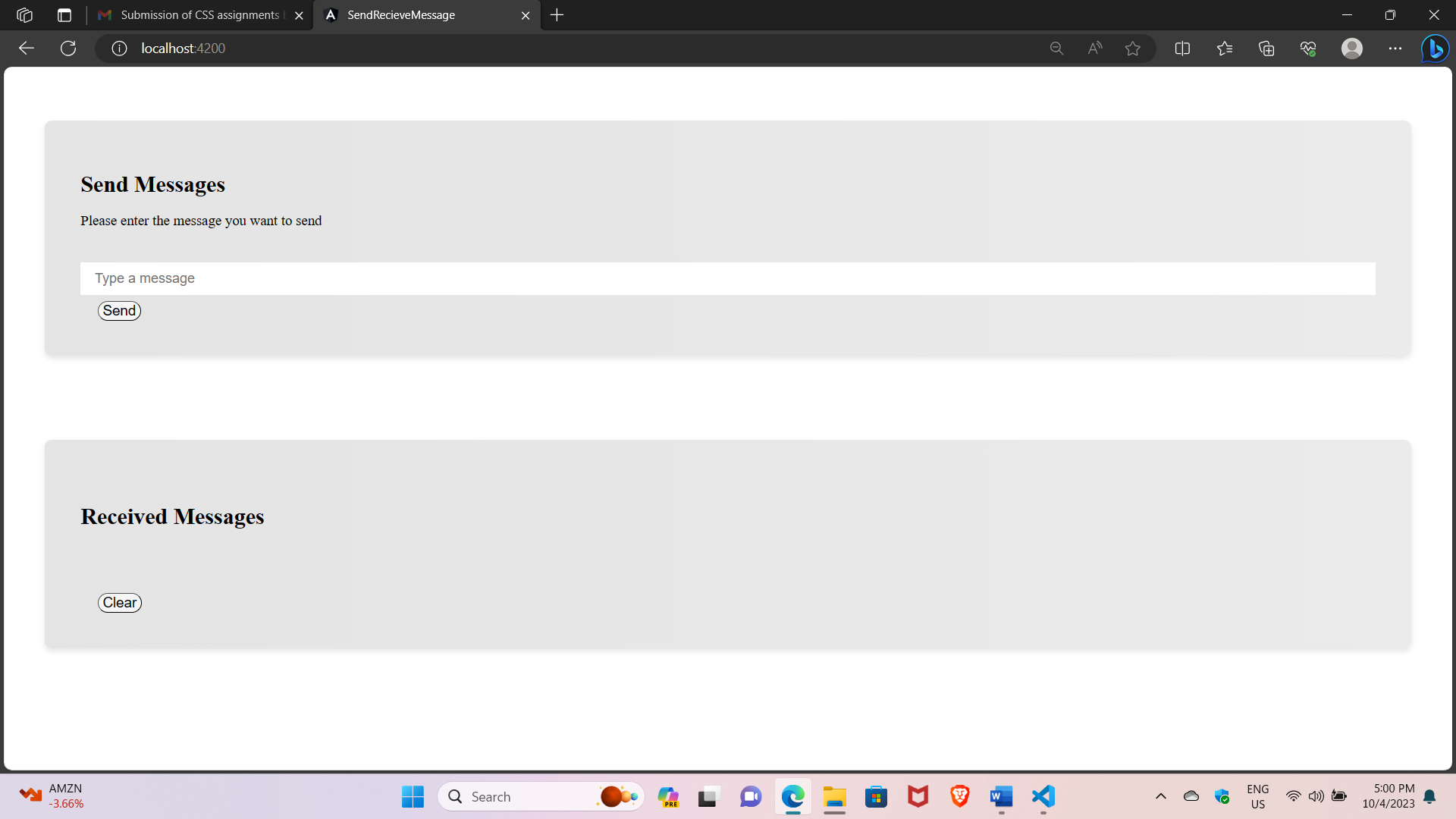
  bootstrap: [AppComponent]

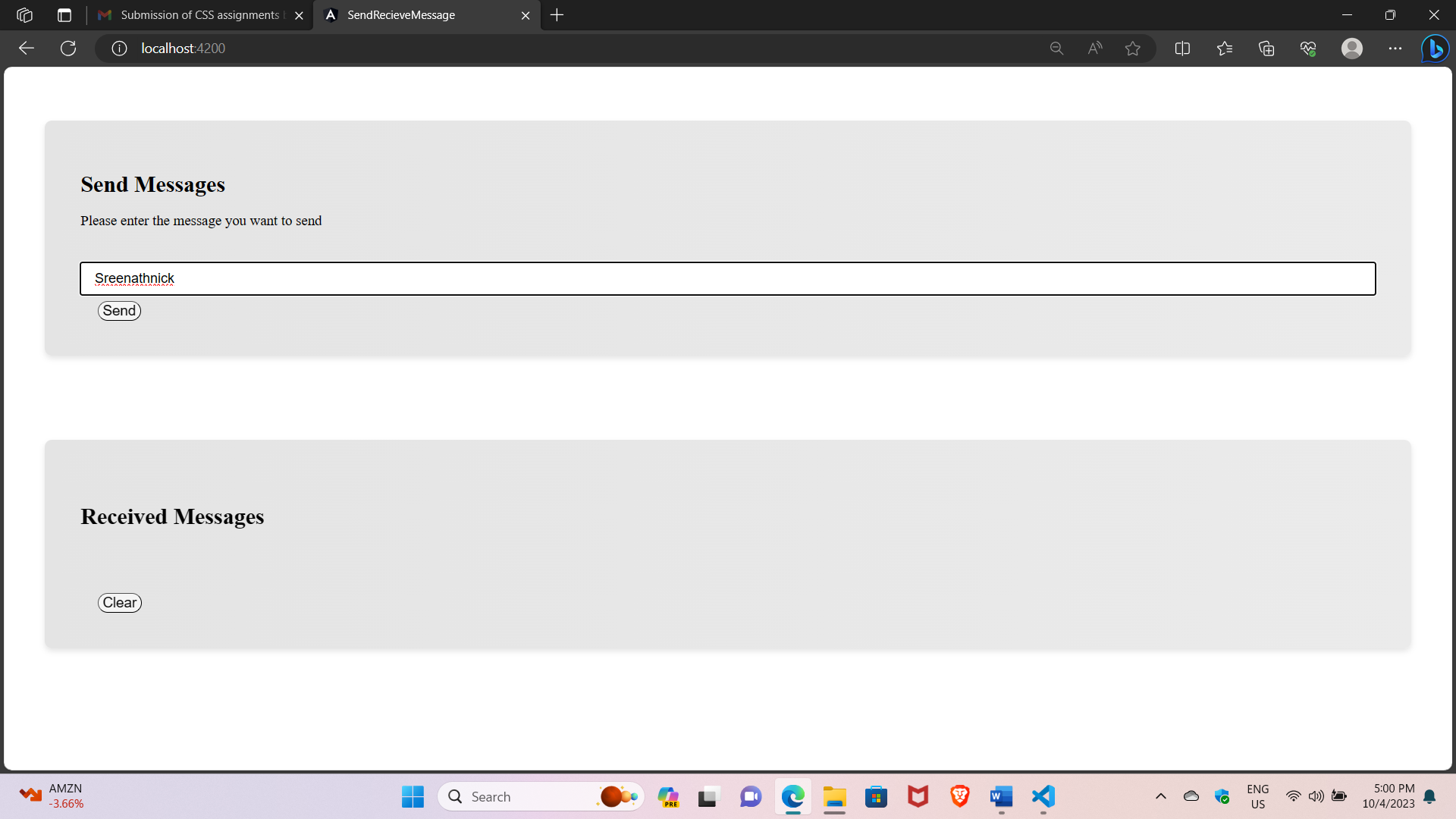
})

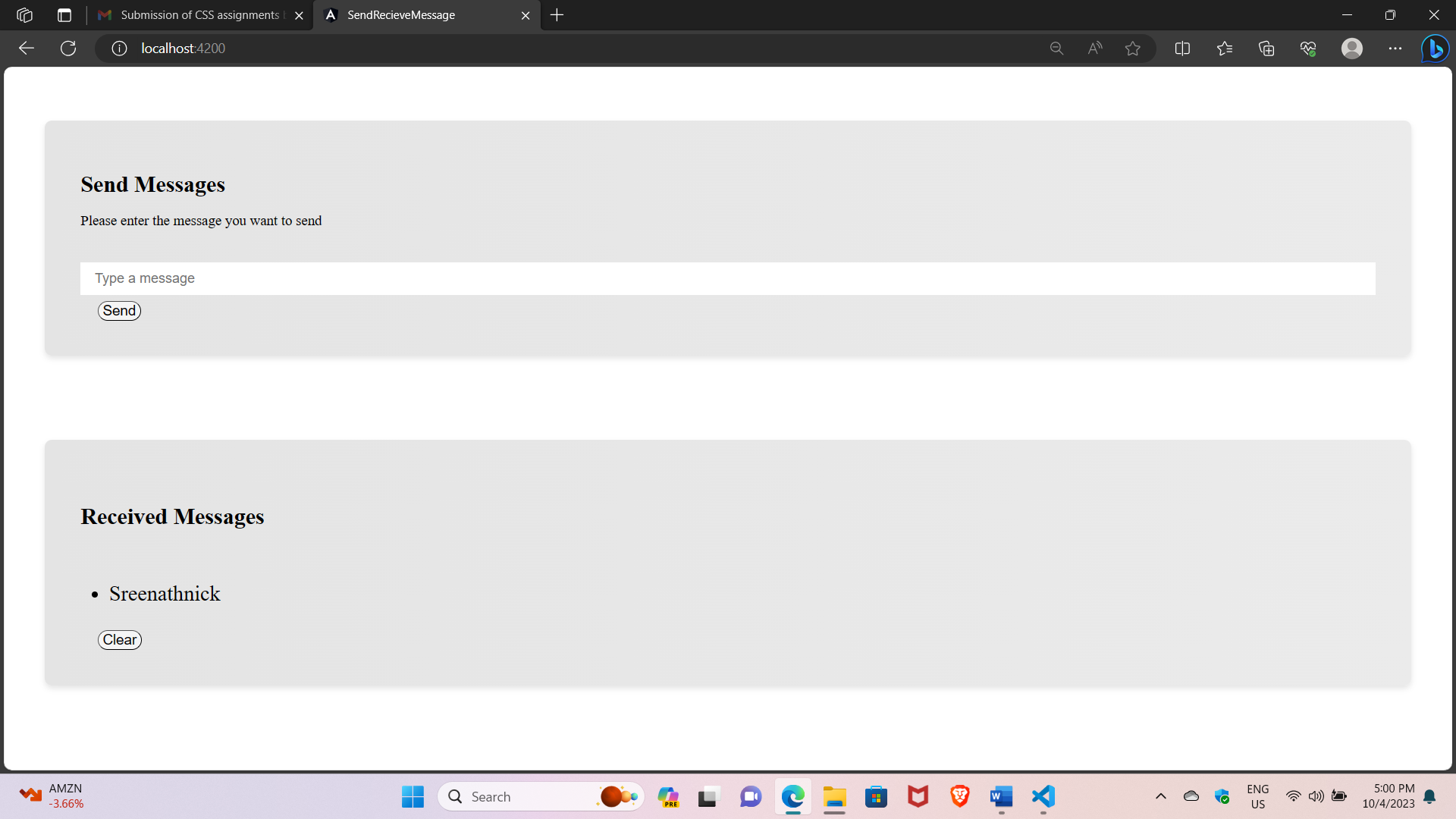
export class AppModule { }

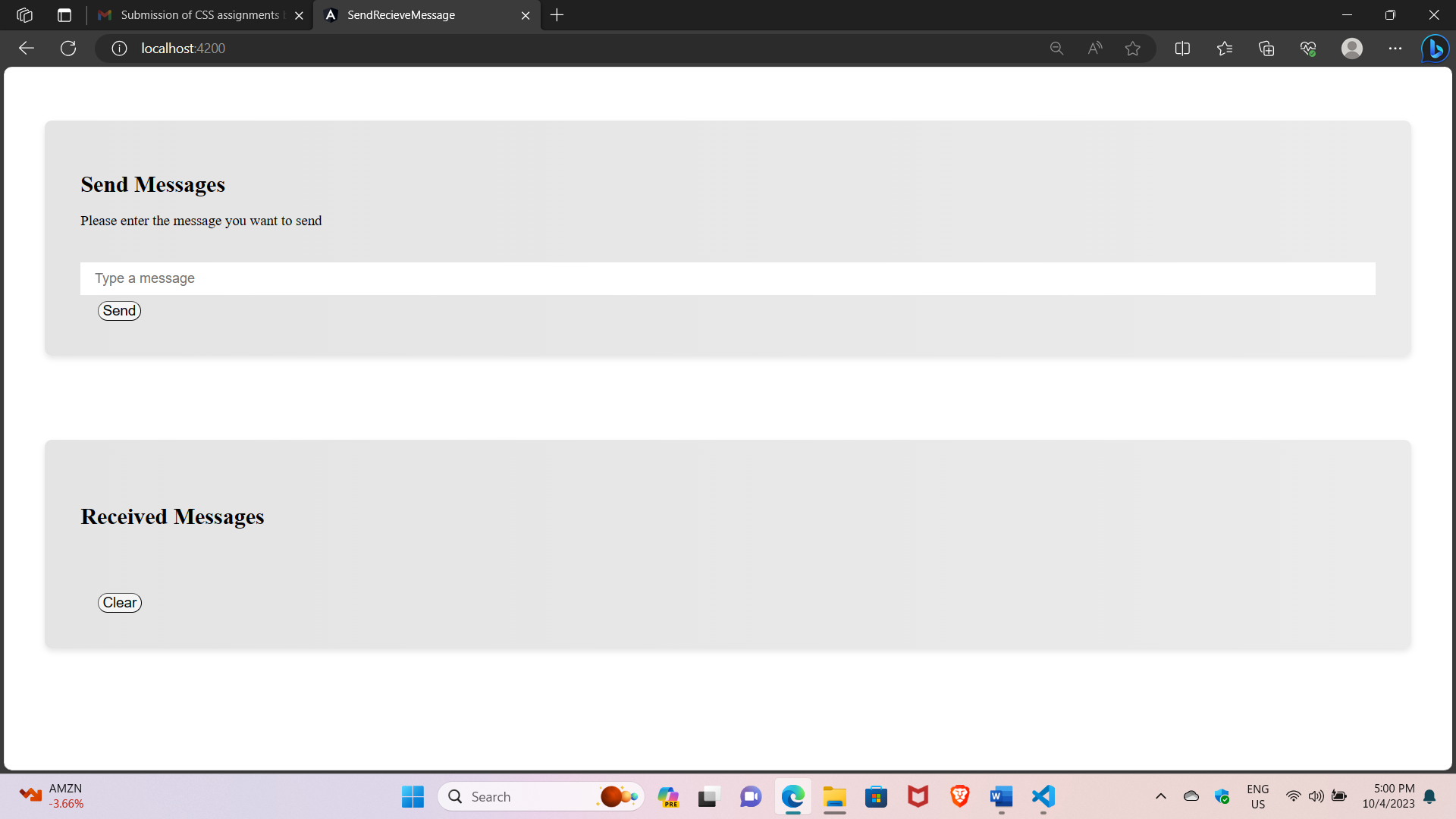
**Output:**











**Submitted by**

**B.Sreenath**