Angular

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

app.component.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Your Angular App</title>

  <!-- Include Bootstrap CSS -->

  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

  <!-- Include your custom CSS file if needed -->

  <!-- <link rel="stylesheet" href="path/to/your/custom.css"> -->

</head>

<body>

  <div class="container">

    <nav class="navbar bg-dark border-bottom border-body" data-bs-theme="dark"  style="align-items: center;">

      <nav class="navbar navbar-expand-lg bg-body-tertiary" >

        <div class="container-fluid">

          <a class="navbar-brand" href="#">Rachana G Raikar</a>

          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

          </button>

          <div class="collapse navbar-collapse" id="navbarNavDropdown">

            <ul class="navbar-nav">

              <li class="nav-item">

                <a class="nav-link" routerLink="welcome" (click)="welcomecom()">Welcome</a>

              </li>

              <li class="nav-item">

                <a class="nav-link" routerLink="home" (click)="homecom()">Home</a>

              </li>

              <li class="nav-item dropdown">

                <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="true">

                  Dropdown link

                </a>

                <ul class="dropdown-menu">

                  <li> <a class="dropdown-item" routerLink="welcome" (click)="welcomecom()">Welcome</a></li>

                  <li><a class="dropdown-item" routerLink="home" (click)="homecom()">Home</a></li>

                </ul>

              </li>

            </ul>

          </div>

        </div>

      </nav>

    </nav>

  </div>

  <router-outlet></router-outlet>

  <!-- Include Bootstrap JS and Popper.js (if needed) -->

  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>

  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

App.component.css

/\* app.component.css \*/

/\* Apply styles to the navbar \*/

.navbar {

    background-color: #333; /\* Background color \*/

    color: #fff; /\* Text color \*/

    padding: 10px 0; /\* Add padding to top and bottom \*/

  }

  /\* Style the navigation links \*/

  .navbar a.nav-link {

    color: #fff; /\* Link text color \*/

    margin: 0 10px; /\* Add spacing between links \*/

  }

  /\* Style the dropdown menu \*/

  .navbar .dropdown-menu {

    background-color: #333; /\* Dropdown background color \*/

    border: none; /\* Remove borders \*/

  }

  /\* Style the dropdown items \*/

  .navbar .dropdown-item {

    color: #fff; /\* Dropdown item text color \*/

  }/\* Add styles for the navbar and its links \*/

/\* Navbar background color and text color \*/

.navbar {

    background-color: #333;

    color: #fff;

  }

  /\* Navbar links color and spacing \*/

  .navbar a.nav-link {

    color: #fff;

    margin-right: 10px;

    font-weight: bold; /\* Example: Set font-weight to bold \*/

  }

  /\* Dropdown background and text color \*/

  .navbar .dropdown-menu {

    background-color: #333;

  }

  /\* Dropdown item text color \*/

  .navbar .dropdown-item {

    color: #fff;

  }

  /\* Active link styles (if using Bootstrap) \*/

  .navbar a.nav-link.active {

    background-color: #fff;

    color: #333;

  }

  /\* Change color of active link \*/

  .navbar a.nav-link.active {

    font-weight: bold;

    color: #ff0;

  }

App.component.ts

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

welcomecom() {

alert("welcome component is working.....!!!");

}

homecom() {

  alert("home component is working......!!!");

}

  title = 'ROUTING';

}

app.routing.module.ts

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

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

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

import { WelcomeComponent } from './welcome/welcome.component';

const routes: Routes = [

    {

      path:'',redirectTo : 'welcome', pathMatch : "full"

    },

    {

      path:"home",component : HomeComponent

    },

    {

      path:"welcome",component : WelcomeComponent

    }

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

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

import { WelcomeComponent } from './welcome/welcome.component';

@NgModule({

  declarations: [

    AppComponent,

    HomeComponent,

    WelcomeComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

welcome.component.html

<body>

    <div class="welcome-container">

      <h1>Welcome to the Best Website</h1>

      <p>We're glad you're here! Explore our amazing content and enjoy your stay.</p>

      <button class="explore-button" (click)="home()">HOME</button>

    </div>

  </body>

welcome.component.css

body {

    font-family: Arial, sans-serif;

    background-color: #a37892;

    margin: 0;

    padding: 0;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

}

.welcome-container {

  text-align: center;

  padding: 30px;

  background-color: #8ca4e7;

  border-radius: 10px;

  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1 {

  color: #04114f;

  font-size: 2em;

  margin-bottom: 10px;

}

p {

    color: #990a16;

    font-size: 1.2em;

    margin-bottom: 20px;

  }

  .explore-button {

    background-color: #a14087;

    color: #fff;

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    font-size: 1.2em;

    cursor: pointer;

  }

  .explore-button:hover {

    background-color: #2980b9;

  }

welcome.component.ts

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

@Component({

  selector: 'app-welcome',

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

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

})

export class WelcomeComponent {

home() {

  document.write("<h1 style='text-align:center'>RAIKAR's HOME Component is working..!!!</h1>")

}

}

[home.component.html](http://home.component.html)

<body>

    <div class="home-container">

      <h1>Welcome</h1>

      <p>to the Best Website</p>

      <button class="explore-link" (click)="welcomecom()">Welcome</button>

    </div>

  </body>

[home.component.css](http://home.component.css)

body {

    font-family: Arial, sans-serif;

    background-color: #f0f0f0;

    margin: 0;

    padding: 0;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

  }

  .home-container {

    text-align: center;

    padding: 30px;

    background-color: #ffffff;

    border-radius: 10px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

  }

  h1 {

    color: #333;

    font-size: 2em;

    margin-bottom: 10px;

}

p {

  color: #666;

  font-size: 1.2em;

  margin-bottom: 20px;

}

.explore-link {

  background-color: #3498db;

  color: #fff;

  padding: 10px 20px;

  border: none;

  border-radius: 5px;

  font-size: 1.2em;

  text-decoration: none;

  cursor: pointer;

  transition: background-color 0.3s;

}

.explore-link:hover {

  background-color: #b929ad;

}

[home.component.ts](http://home.component.ts)

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

@Component({

  selector: 'app-home',

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

  styleUrls: ['./home.component.css']

})

export class HomeComponent {

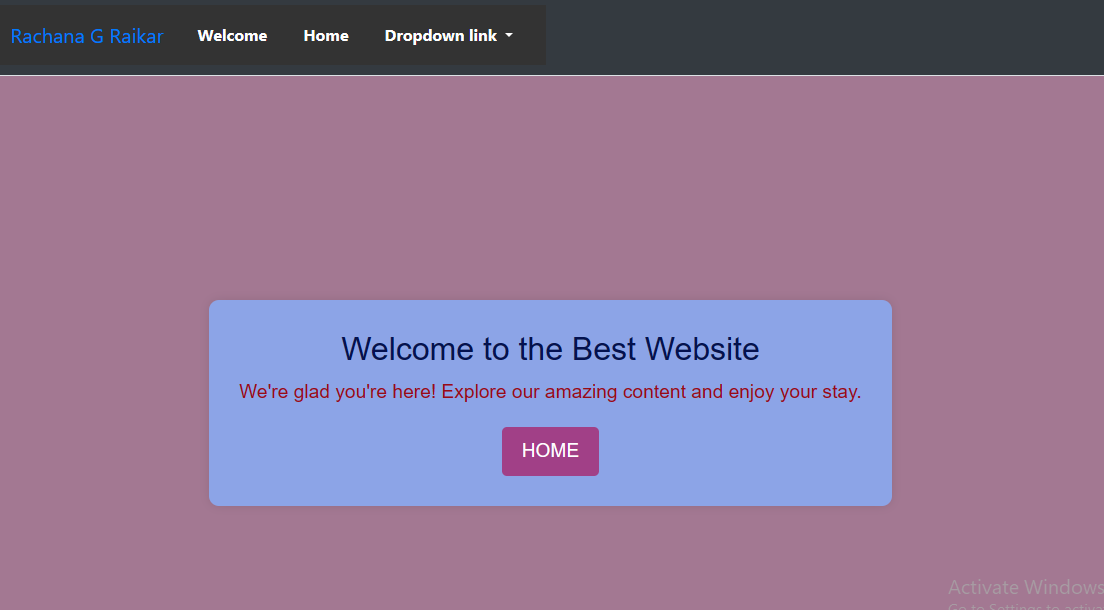
  welcomecom() {

    document.write("<h1 style='text-align:center'>RAIKAR's WELCOME Component is working..!!!</h1>")

    }

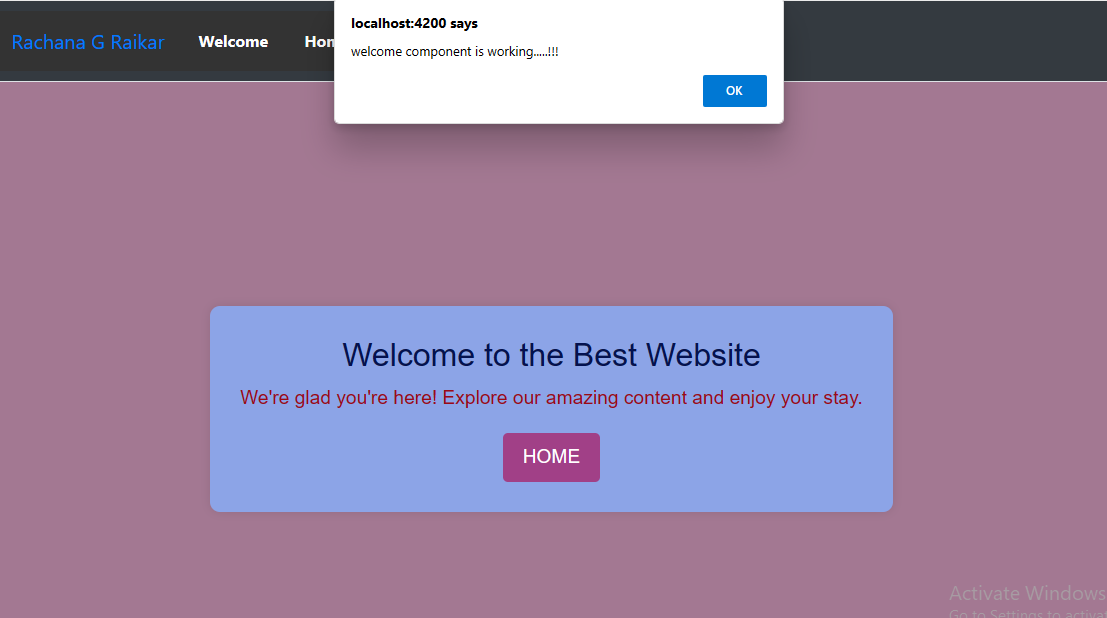
}

**OUTPUT :**

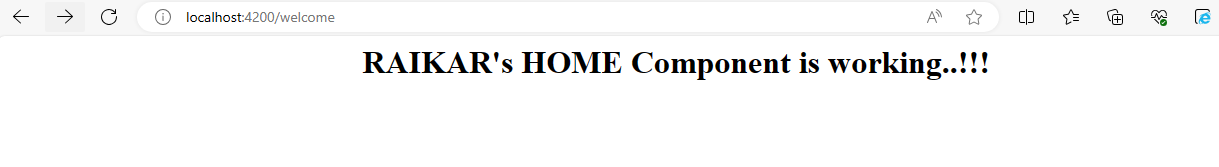


**Welcome page**

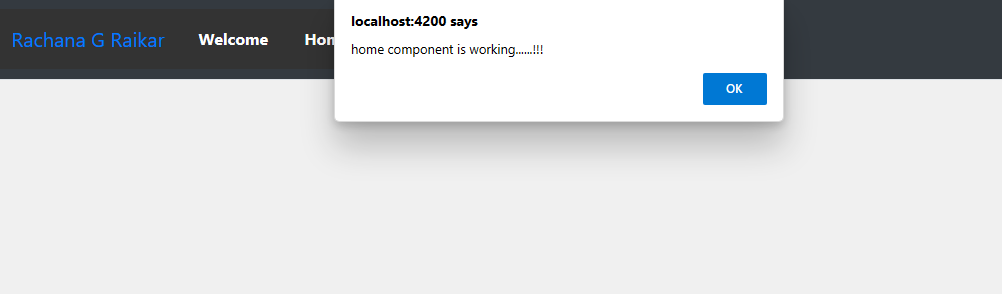
**(Default page when loads)**



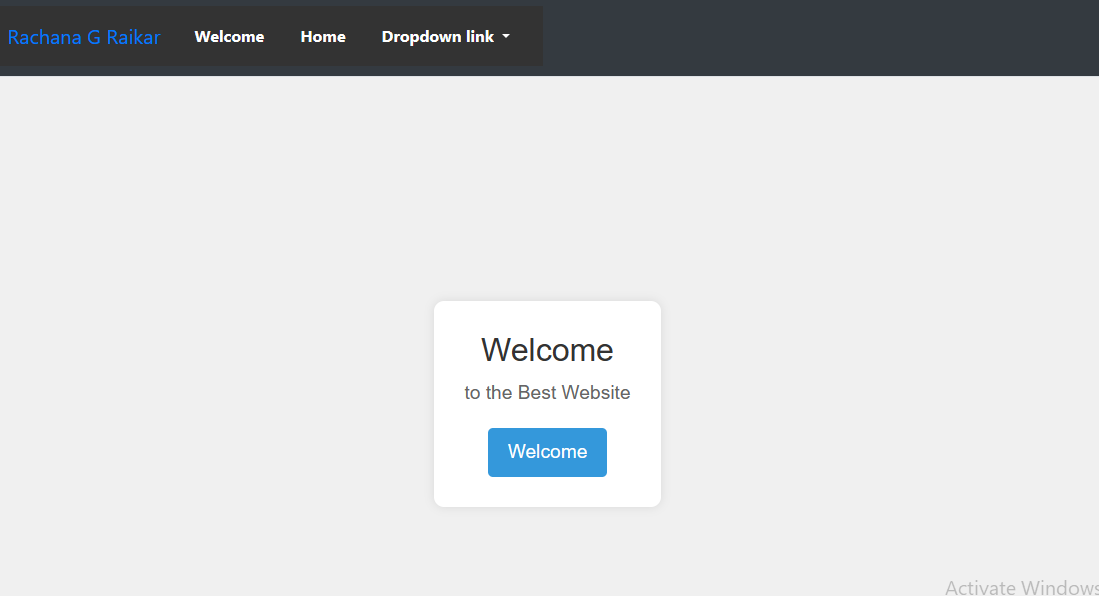
**Popup when click on welcome on navigation bar**



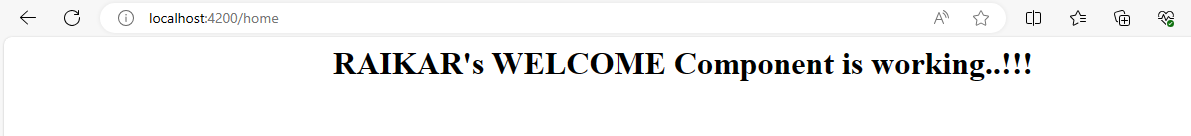
Output when click on home button



**Popup when click on home on navigation bar**



**Home page**



Output when click on welcome button

1. Write a program to create user with address fields with validation using reactive forms?

app.component.html

<div class="container">

  <app-register></app-register>

</div>

app.component.css

.navbar {

    background-color: #343a40; /\* Dark background color \*/

    color: #ffffff; /\* Text color \*/

  }

  .navbar-brand {

    font-size: 24px;

  }

  .navbar-nav .nav-link {

    color: #ffffff; /\* Navbar link color \*/

  }

  .navbar-nav .nav-link:hover {

    color: #17a2b8; /\* Hover color \*/

  }

  /\* Welcome component styles \*/

  .welcome-container {

    text-align: center;

    background-color: #f0f0f0; /\* Background color for the welcome component \*/

    padding: 30px;

    border-radius: 10px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

  }

  .welcome-container h1 {

    color: #333; /\* Heading color \*/

    font-size: 2em;

    margin-bottom: 10px;

  }

  .welcome-container p {

    color: #666; /\* Paragraph text color \*/

    font-size: 1.2em;

    margin-bottom: 20px;

  }

  .explore-button {

    background-color: #3498db; /\* Button background color \*/

    color: #fff; /\* Button text color \*/

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    font-size: 1.2em;

    cursor: pointer;

  }

  .explore-button:hover {

    background-color: #2980b9; /\* Hover background color \*/

  }

  /\* Home component styles \*/

  .home-container {

    text-align: center;

    background-color: #ffffff; /\* Background color for the home component \*/

    padding: 30px;

    border-radius: 10px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

  }

  .home-container h1 {

    color: #333; /\* Heading color \*/

    font-size: 2em;

    margin-bottom: 10px;

  }

  .home-container p {

    color: #666; /\* Paragraph text color \*/

    font-size: 1.2em;

    margin-bottom: 20px;

  }

  .explore-link {

    background-color: #3498db; /\* Button background color \*/

    color: #fff; /\* Button text color \*/

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    font-size: 1.2em;

    text-decoration: none;

    cursor: pointer;

    transition: background-color 0.3s;

  }

  .explore-link:hover {

    background-color: #2980b9; /\* Hover background color \*/

  }

app.module.ts

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

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

import { ReactiveFormsModule, FormsModule } from '@angular/forms';

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

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

import { RegisterComponent } from './register/register.component';

@NgModule({

  declarations: [

    AppComponent,

    RegisterComponent,

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    ReactiveFormsModule,

    FormsModule,

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

register.component.html

<div class="container">

    <form [formGroup]="registerForm" (ngSubmit)="registerSubmited()" class="border p-4 rounded shadow">

      <div class="text-center mb-4">

        <h2 class="fw-bold">Register</h2>

      </div>

      <div class="mb-3">

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

        <input type="text" class="form-control" id="firstname" formControlName="firstname"><br><br>

        <div class="form-text text-danger" \*ngIf="FirstName.errors?.['required'] && FirstName.touched">

          First name is required

        </div>

        <div class="form-text text-danger" \*ngIf="FirstName.errors?.['pattern'] && FirstName.touched">

          First name should start with an alphabet

        </div>

      </div>

      <div class="mb-3">

        <label for="lastname" class="form-label">Last Name:</label>

        <input type="text" class="form-control" id="lastname" formControlName="lastname"><br><br>

        <div class="form-text text-danger" \*ngIf="LastName.errors?.['required'] && LastName.touched">

          Last name is required

        </div>

        <div class="form-text text-danger" \*ngIf="LastName.errors?.['pattern'] && LastName.touched">

          Last name should start with an alphabet

        </div>

      </div>

      <div class="mb-3">

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

        <input type="email" class="form-control" id="email" formControlName="email"><br><br>

        <div class="form-text text-danger" \*ngIf="Email.errors?.['required'] && (Email.dirty || Email.touched)">

          Email is required

        </div>

        <div class="form-text text-danger" \*ngIf="Email.errors?.['email'] && (Email.dirty || Email.touched)">

          Email is invalid

        </div>

      </div>

      <div class="mb-3">

        <label for="mobile" class="form-label">Phone Number:</label>

        <input type="text" class="form-control" id="mobile" formControlName="mobile"><br><br>

        <div class="form-text text-danger" \*ngIf="Mobile.errors?.['required'] && (Mobile.dirty || Mobile.touched)">

          Number is required

        </div>

        <div class="form-text text-danger" \*ngIf="Mobile.errors?.['pattern'] && (Mobile.dirty || Mobile.touched)">

          Number is invalid

        </div>

        <div class="form-text text-danger" \*ngIf="Mobile.errors?.['minlength'] && (Mobile.dirty || Mobile.touched)">

          Number is too short

        </div>

        <div class="form-text text-danger" \*ngIf="Mobile.errors?.['maxlength'] && (Mobile.dirty || Mobile.touched)">

          Number is too long

        </div>

      </div>

      <div class="text-center">

        <button type="submit" [disabled]="registerForm.invalid" class="btn btn-primary">Submit</button>

      </div>

    </form>

  </div>

register.component.css

/\* Custom CSS for the Registration Form \*/

/\* Center the form vertically and horizontally \*/

.container {

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

  }

  /\* Style the form container \*/

  .border {

    background-color: #f8f9fa; /\* Light gray background \*/

    padding: 20px;

    border-radius: 10px;

  }

  /\* Style the form title \*/

  h2 {

    color: #343a40; /\* Dark text color \*/

    margin-bottom: 20px;

    text-align: center;

  }

  /\* Style form labels \*/

  .form-label {

    font-weight: bold;

  }

  /\* Style form inputs \*/

  .form-control {

    border-color: #ced4da; /\* Gray border color \*/

  }

  /\* Style the submit button \*/

  .btn-primary {

    background-color: #007bff; /\* Primary color \*/

    color: #fff; /\* White text color \*/

    border: none;

    border-radius: 5px;

    padding: 10px 20px;

  }

  /\* Style form error messages \*/

  .form-text.text-danger {

    color: #dc3545; /\* Red error message color \*/

    font-size: 14px;

    margin-top: 5px;

  }

register.component.ts

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

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

@Component({

  selector: 'app-register',

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

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

})

export class RegisterComponent implements OnInit{

  constructor(){}

  ngOnInit(): void {

  }

  registerForm =new FormGroup({

 firstname:new FormControl("",[Validators.required,Validators.pattern("[a-zA-Z].\*")]),

 lastname:new FormControl("",[Validators.required,Validators.pattern("[a-zA-Z].\*")]),

email:new FormControl("",[Validators.required,Validators.email]),

mobile:new FormControl("",[Validators.required,Validators.pattern("[0-9]\*"),Validators.minLength(10),Validators.maxLength(10)])

  });

  registerSubmited(){

    console.log(this.registerForm.get("firstname"));

    console.log(this.registerForm.get("lastname"));

    console.log(this.registerForm.get("email"));

    console.log(this.registerForm.get("mobile"));

  }

get FirstName():FormControl{

  return this.registerForm.get("firstname") as FormControl;

}

get LastName():FormControl{

  return this.registerForm.get("lastname") as FormControl;

}

get   Email():FormControl{

  return this.registerForm.get("email") as FormControl;

}

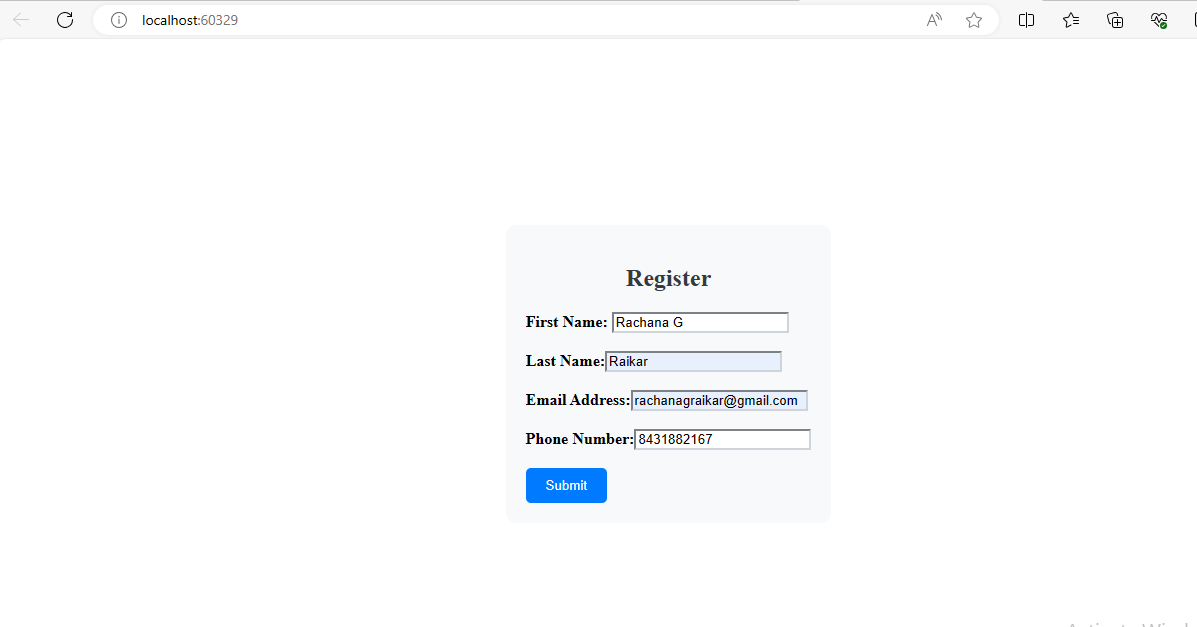
get Mobile():FormControl{

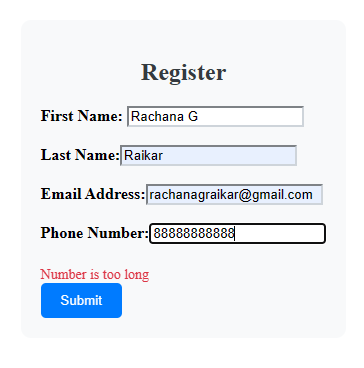
  return this.registerForm.get("mobile") as FormControl;

}

}

**OUTPUT :**





**Reactive form**

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

SuperHero.Service.ts

/ src/app/superhero.service.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;

  }

}

Add.Superhero.component.html

<!-- src/app/add-superhero/add-superhero.component.html -->

<div>

    <h2>Add Superhero</h2>

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

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

  </div>

Add-superhro.Component.ts

/ src/app/add-superhero/add-superhero.component.ts

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-superheros.component.html

<!-- src/app/list-superheroes/lisy-sup.component.html -->

<div>

    <h2>Superheroes List</h2>

    <ul>

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

    </ul>

  </div>

List-supetheroes.Component.ts

/ src/app/list-superheroes/list-superheroes.component.ts

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-routing.module.ts

// src/app/app-routing.module.ts

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

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

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

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

const routes: Routes = [

  { path: '', redirectTo: '/add-superhero', pathMatch: 'full' },

  { path: 'add-superhero', component: AddSuperheroComponent },

  { path: 'list-superheroes', component: ListSuperheroesComponent },

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

App-component.html

!-- src/app/app.component.html -->

<nav>

  <ul>

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

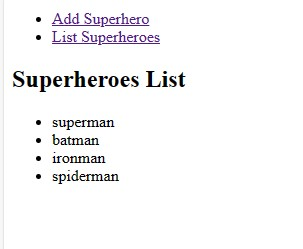
    <li><a routerLink="/list-superheroes">List Superheroes</a></li>

  </ul>

</nav>

<router-outlet></router-outlet>

Output



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

Child.component.ts

import { Component, Input, Output, EventEmitter } from '@angular/core';

@Component({

  selector: 'app-child',

  template: `

    <h2>Child Component</h2>

    <p>Counter from Parent: {{ counter }}</p>

    <button (click)="decrement()">Decrement</button>

  `,

})

export class ChildComponent {

  @Input() counter: number = 0; // Initialize the property

  @Output() counterChange = new EventEmitter<number>();

  constructor() {}

  decrement() {

    if (this.counter > 0) {

      this.counter--;

      this.counterChange.emit(this.counter);

    }

  }

}

Child.component.html

<p>child works!</p>

<app-child></app-child>

Parent.component.ts

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'AngularAssignment4';

}

Parent.component.html

<app-parent></app-parent>

App-rounting.module.ts

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

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

const routes: Routes = [];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

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 { }

app.component.ts

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

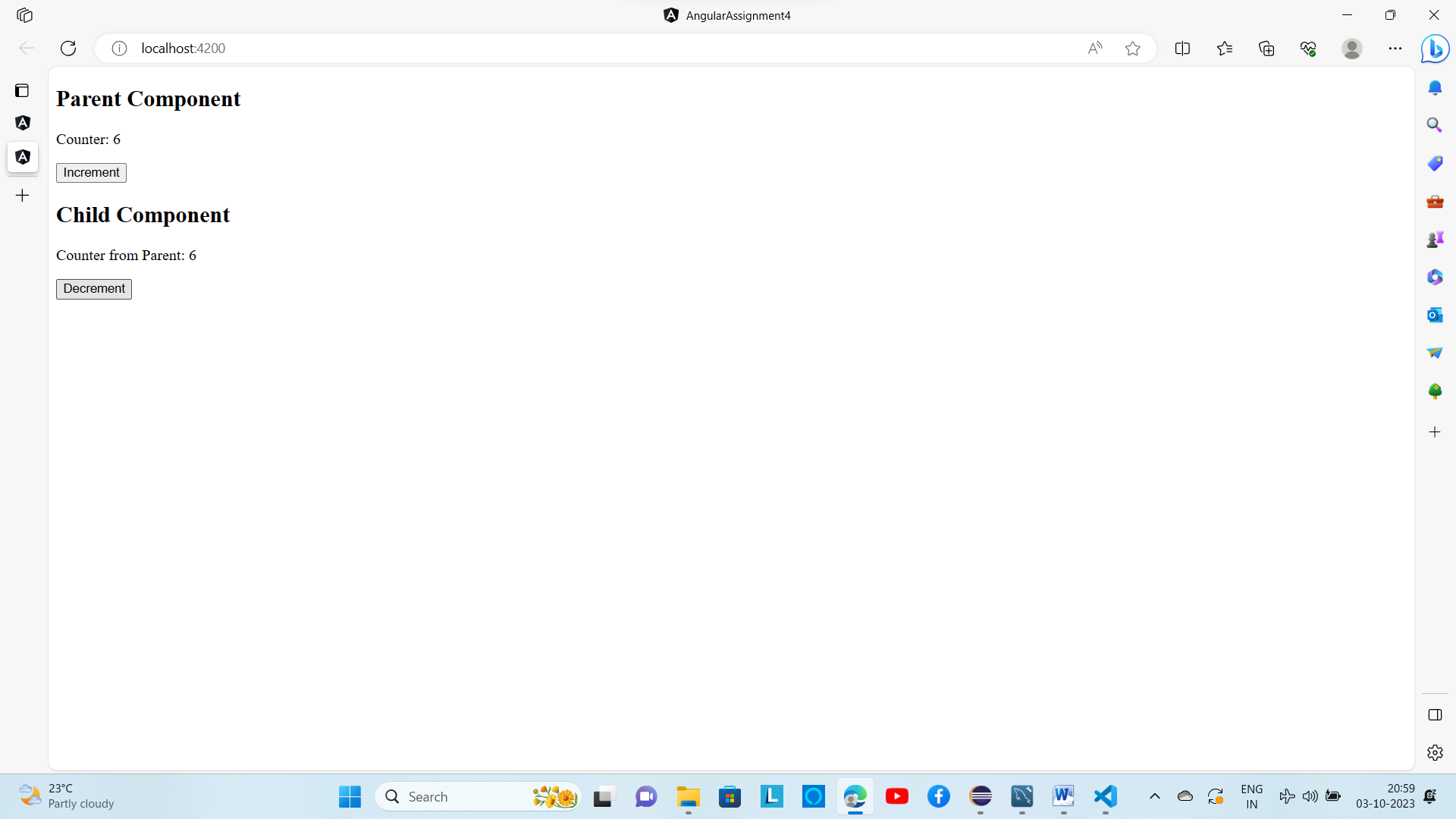
export class AppComponent {

  title = 'AngularAssignment4';

}

App.component.html

<app-parent></app-parent>

Output:  


5. Write a program to send message using angular subject

Message,service.ts

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

import { Subject } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class MessageService {

  private messageSubject = new Subject<string>();

  sendMessage(message: string) {

    this.messageSubject.next(message);

  }

  clearMessages() {

    this.messageSubject.next(''); // Provide a default value (e.g., undefined)

  }

  getMessage() {

    return this.messageSubject.asObservable();

  }

Sender.component.ts

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

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

@Component({

  selector: 'app-sender',

  template: `

    <div>

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

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

    </div>

  `

})

export class SenderComponent {

  message = '';

  constructor(private messageService: MessageService) {}

  sendMessage() {

    if (this.message.trim()) {

      this.messageService.sendMessage(this.message);

      this.message = '';

    }

  }

}

Reciever.component.ts

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

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

@Component({

  selector: 'app-receiver',

  template: `

    <div>

      <h2>Received Messages</h2>

      <ul>

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

      </ul>

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

    </div>

  `

})

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>

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 { SenderComponent } from './sender/sender.component';

import { ReceiverComponent } from './receiver/receiver.component';

import { FormsModule } from '@angular/forms'; // Import FormsModule

@NgModule({

  declarations: [

    AppComponent,

    SenderComponent,

    ReceiverComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    FormsModule,

  ],

  providers: [],

  bootstrap: [AppComponent]

})

Output

