

Angular Assignment

1.

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 -->
</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="#">Angular</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"></sc
ript>
</body>
</html>

```

App.component.css

```

.navbar {
  background-color: #1a1a1a;
  color: #ccc;
  padding: 10px 0;
}

.navbar a.nav-link {
  color: #ccc;
  margin-right: 10px;
  font-weight: bold;
}

.navbar .dropdown-menu {
  background-color: #1a1a1a;
  border: none;
}

.navbar .dropdown-item {

```

```
    color: #ccc;
}

.navbar a.nav-link.active {
  background-color: #ccc;
  color: #1a1a1a;
  font-weight: bold;
}
```

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.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'> HOME</h1>")  
}
```

```
}
```

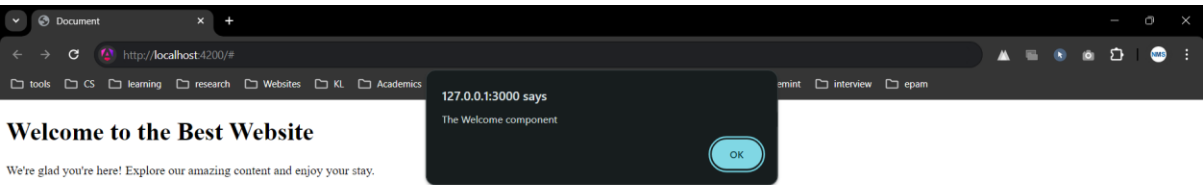
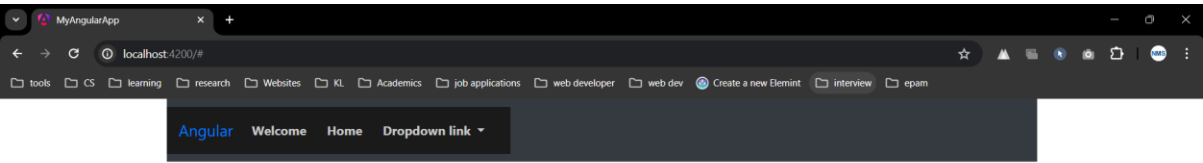
Home component.html

```
<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.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'>The Welcome  
Component.</h1>")  
  }  
}
```

Output:



2.

App.component.html

```
<div class="container">
  <app-register></app-register>
</div>
```

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)="registerSubmitted()"
class="border p-4 rounded shadow">
    <div class="text-center mb-4">
      <h2 class="fw-bold">Reactive Contact Form</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.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)])
  });
  registerSubmitted(){
    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;
    }

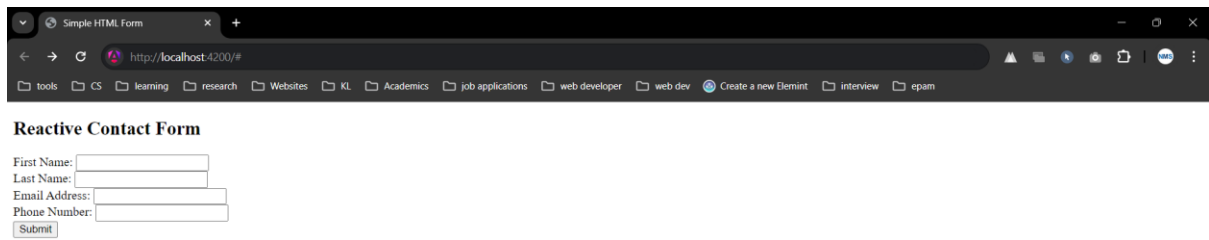
    get LastName():FormControl{
        return this.registerForm.get("lastname") as FormControl;
    }

    get Email():FormControl{
        return this.registerForm.get("email") as FormControl;
    }

    get Mobile():FormControl{
        return this.registerForm.get("mobile") as FormControl;
    }

}
```

OutPut:



A screenshot of a web browser window titled 'Simple HTML Form' at the URL 'http://localhost:4200/#'. The browser's address bar and tabs are visible. Below the browser window, a file explorer shows a directory structure with folders like 'tools', 'CS', 'learning', 'research', 'Websites', 'KL', 'Academics', 'job applications', 'web developer', 'web dev', 'Create a new Element', 'interview', and 'epam'. The main content area displays the title 'Reactive Contact Form' followed by four input fields labeled 'First Name:', 'Last Name:', 'Email Address:', and 'Phone Number:'. A 'Submit' button is located below the 'Phone Number' field. All input fields are currently empty.

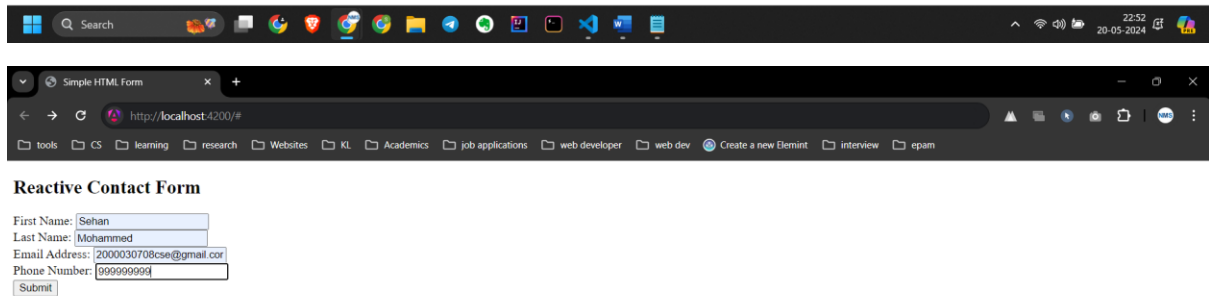
Reactive Contact Form

First Name:

Last Name:

Email Address:

Phone Number:



A screenshot of a Windows taskbar and a web browser window. The taskbar shows various application icons and the system clock indicating 22:52 on 20-05-2024. The browser window, titled 'Simple HTML Form', shows the same 'Reactive Contact Form' as the previous screenshot, but now the input fields are filled with the following text: 'First Name: Sehan', 'Last Name: Mohammed', 'Email Address: 2000030708cse@gmail.com', and 'Phone Number: 999999999'. The 'Submit' button remains below the fields.

Reactive Contact Form

First Name:

Last Name:

Email Address:

Phone Number:



A screenshot of a Windows taskbar showing application icons and the system clock indicating 22:59 on 20-05-2024.

3.

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

```
<div>
  <h2>Add Superhero</h2>
  <input #superheroName placeholder="Enter superhero name" />
  <button (click)="addSuperhero(superheroName.value)">Add</button>
</div>
```

Add-superhro.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-superheroes.component.html

```

<div>
  <h2>Superheroes List</h2>
  <ul>
    <li *ngFor="let superhero of superheroes">{{ superhero }}</li>
  </ul>
</div>

```

List-supetheroes.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

```

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

```

<nav>
  <ul>
    <li><a routerLink="/add-superhero">Add Superhero</a></li>
    <li><a routerLink="/list-superheroes">List Superheroes</a></li>
  </ul>

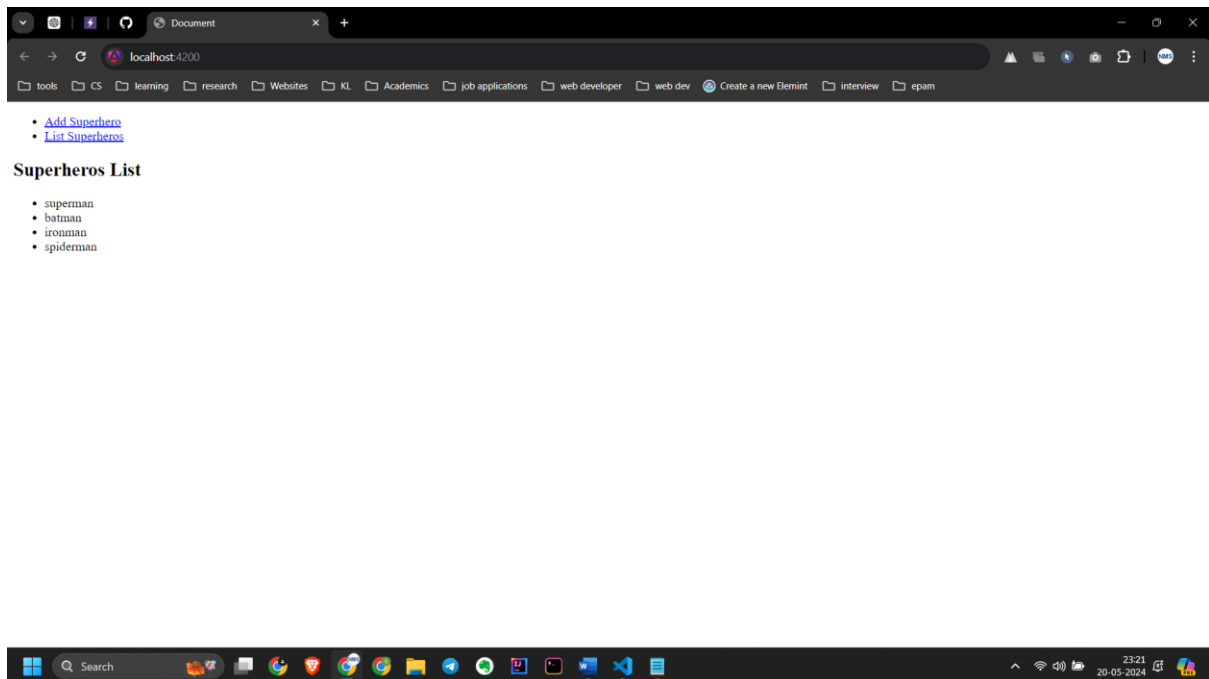
```



```
</nav>

<router-outlet></router-outlet>
```

Output:



4.

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';
}

```

App-routing.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';

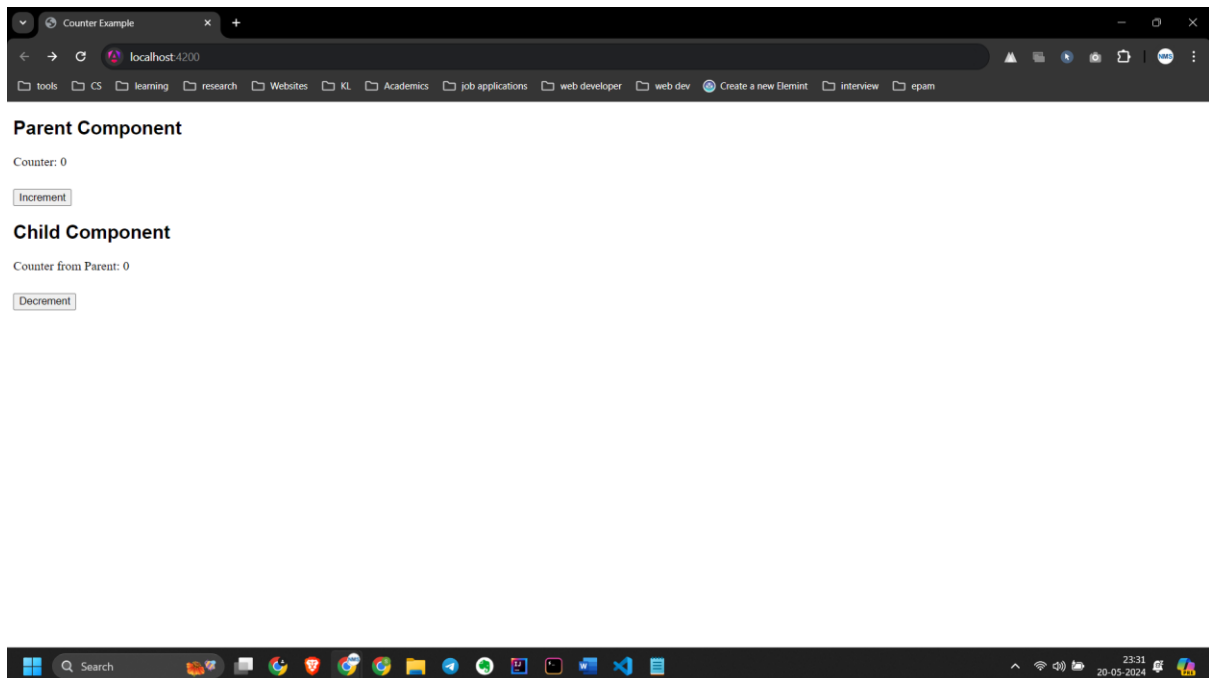
@NgModule({
  declarations: [
    AppComponent,
    ParentComponent,
    ChildComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

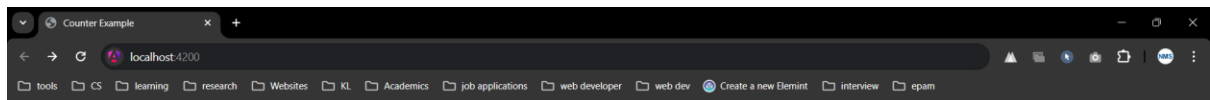
```
providers: [],  
bootstrap: [AppComponent],  
}))  
export class AppModule { }
```

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';  
}
```

Output:





Parent Component

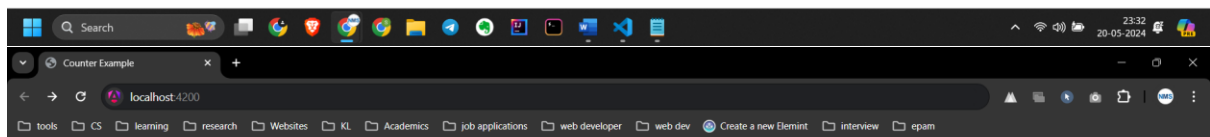
Counter: 5

Increment

Child Component

Counter from Parent: 5

Decrement



Parent Component

Counter: 2

Increment

Child Component

Counter from Parent: 2

Decrement



5.

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" />
    </div>
  `
})
```

```

        <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:

