

Angular Assignment – 8

1. Create application which contains two components as app and Child. Child component contains one textbook and one button. User enters text in that text box and when user press the button data from that textbook should be sent to parent component ie app. After getting data from Child component app component will display the data using string interpolation.

src ▸ app ▸ child ▸ <> child.component.html ▸ input

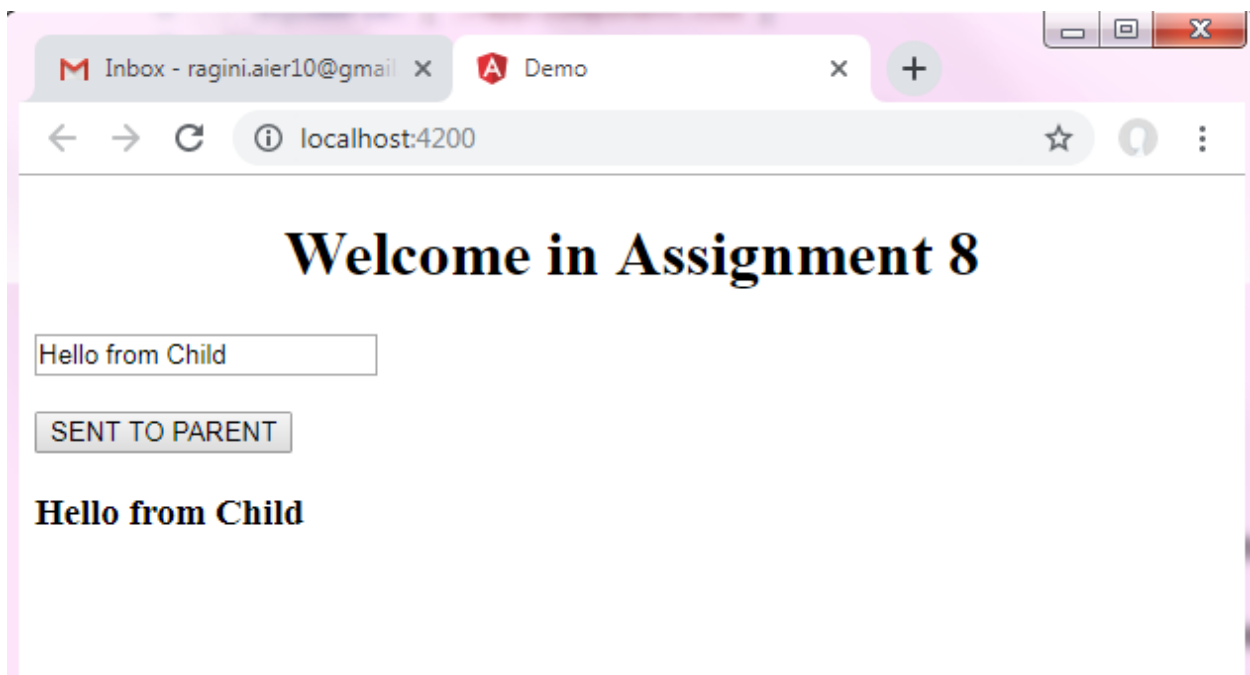
```
1 <input [(ngModel)]= "parent_data" type="text"><br><br>
2 <button (click)="sender()">SENT TO PARENT</button>
```

src ▸ app ▸ child ▸ TS child.component.ts ▸ ChildComponent

```
1 import { Component, OnInit, Output, EventEmitter } from '@angular/core';
2 //import { EventEmitter } from 'events';
3
4 @Component({
5   selector: 'app-child',
6   templateUrl: './child.component.html',
7   styleUrls: ['./child.component.css']
8 })
9 export class ChildComponent implements OnInit {
10
11   constructor() { }
12
13   ngOnInit() {
14   }
15   @Output() public eeobj=new EventEmitter();
16   parent_data="";
17   public sender()
18   {
19     this.eeobj.emit(this.parent_data);
20   }
21
22 }
```

```
src > app > <> app.component.html > div > h1
1 <!--The content below is only a placeholder and can be replaced.-->
2 <div style="text-align:center">
3   <h1>
4   | Welcome in {{ title }}
5   </h1>
6 </div>
7 <app-child (eeobj)="message= $event"></app-child>
8 <h3>{{message}}</h3>
9
10
```

```
src > app > TS app.component.ts > AppComponent > title
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'Assignment 8';
10  public message="";
11 }
12
```



2. Create application which contains two components as app and Child. App component send “Hello from Parent” to the child component and Child component send “Hello from Child” to the app component ie app. In this application parent and child components communicate with each other by sending the data.

src ▸ app ▸ TS app.component.ts ▸ ...

```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'Assignment 8';
10
11    public childmsg = "Hello from Parent";
12    public message = "";
13  }
```

src ▸ app ▸ <> app.component.html ▸ ...

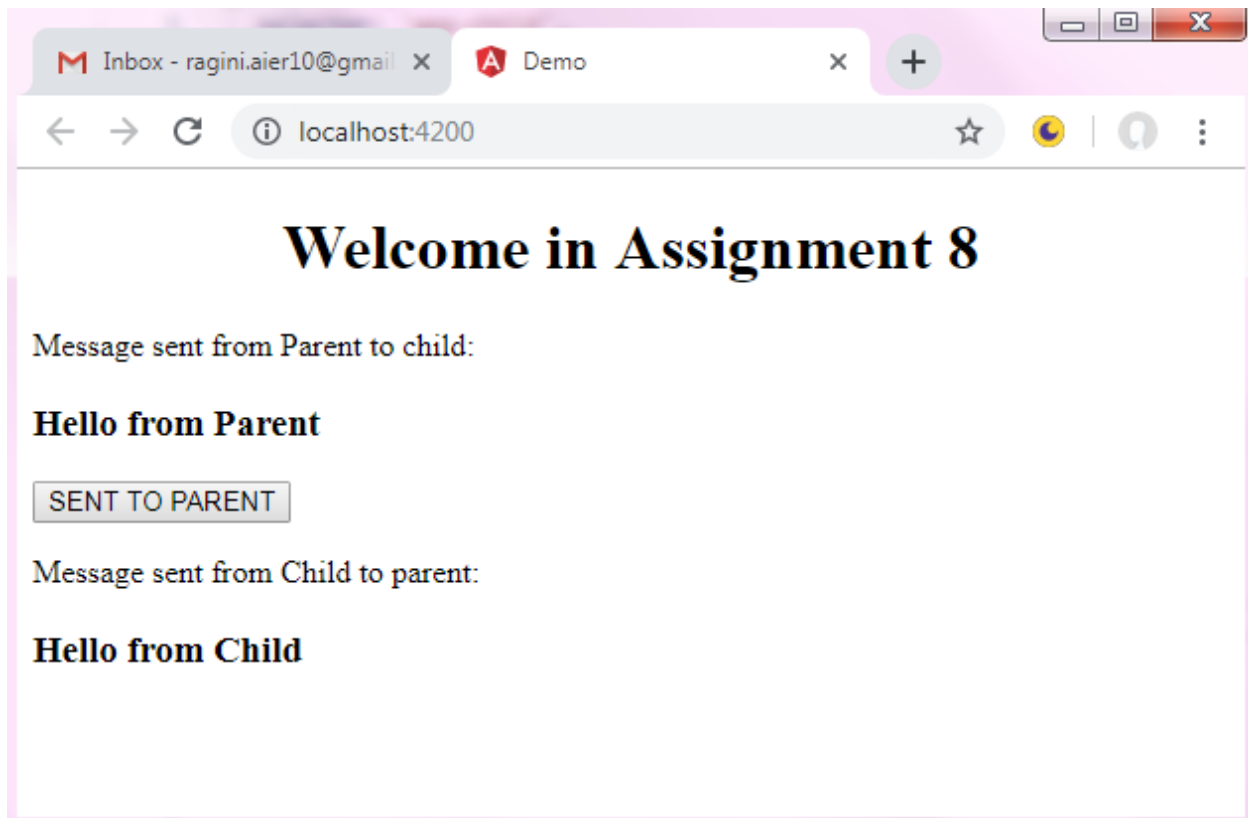
```
1  <!--The content below is only a placeholder and can be replaced.-->
2  <div style="text-align:center">
3    <h1>
4      Welcome in {{ title }}
5    </h1>
6  </div>
7  <app-child [childmsg]="childmsg" (eeobj)="message= $event"></app-child>
8  <p>Message sent from Child to parent: </p>
9  <h3>{{message}}</h3>
```

src ▸ app ▸ child ▸ TS child.component.ts ▸ ChildComponent

```
1 import { Component, OnInit, Output, EventEmitter, Input } from '@angular/core';
2 //import { EventEmitter } from 'events';
3
4 @Component({
5   selector: 'app-child',
6   templateUrl: './child.component.html',
7   styleUrls: ['./child.component.css']
8 })
9 export class ChildComponent implements OnInit {
10
11   constructor() { }
12   ngOnInit() {
13   }
14   @Input() public childmsg="";
15   @Output() public eeobj=new EventEmitter();
16   //parent_data="";
17   public sender()
18   {
19     //this.eeobj.emit(this.parent_data);
20     this.eeobj.emit("Hello from Child");
21   }
22 }
```

src ▸ app ▸ child ▸ child.component.html ▸ button

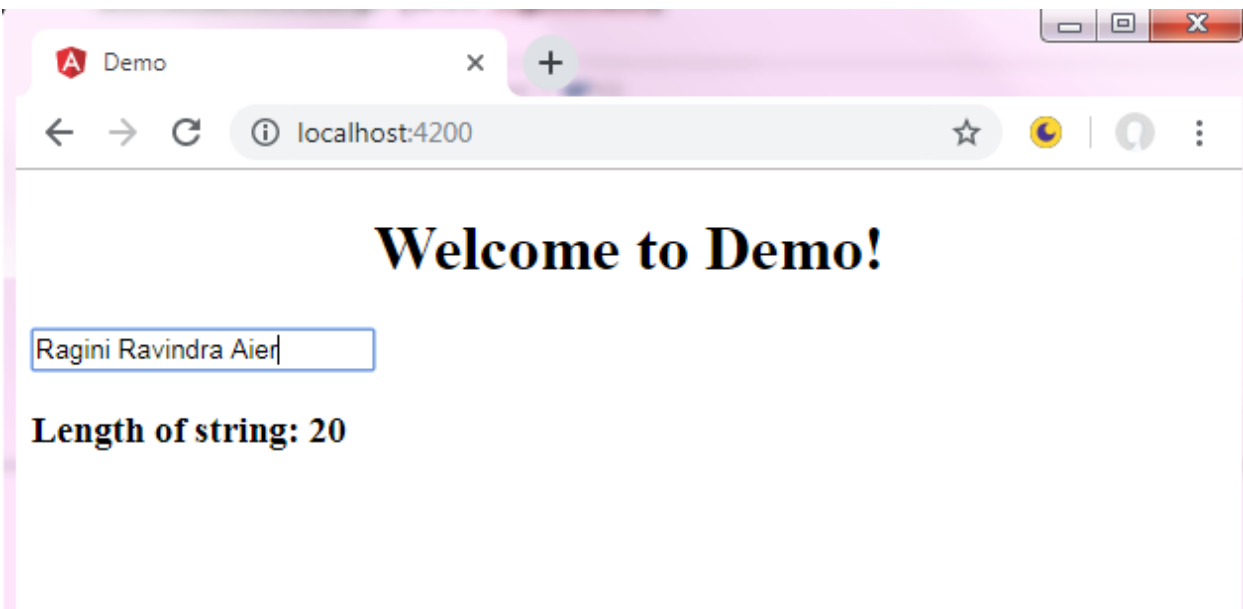
```
1 <p>Message sent from Parent to child:</p>
2 <h3>{{childmsg}}</h3>
3
4 <!--<input [(ngModel)]="parent_data" type="text"><br><br>-->
5 <button (click)="sender()">SENT TO PARENT</button>
```



3. Create application which contains only one component as app. app component contains one text box. When user enters data in that text box we have to display length of that string simultaneously. (Use ngModel)

```
src > app > <> app.component.html > h3
1  <!--The content below is only a placeholder and can be replaced.-->
2  <div style="text-align:center">
3    <h1>
4    | Welcome to {{ title }}!
5    </h1>
6  </div>
7  <input [(ngModel)]="str" type="text">
8  <h3>Length of string: {{str.length}}</h3>
9
```

```
src ▸ app ▸ TS app.component.ts ▸ AppComponent ▸ str
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'Demo';
10   public str;
11 }
```



4. Recreate the above application without using ngModel directive. (Use Event listeners on text box).

src ▸ app ▸ <> app.component.html ▸ input

```
1  <!--The content below is only a placeholder and can be replaced.-->
2  <div style="text-align:center">
3    <h1>
4      Welcome in {{ title }}
5    </h1>
6  </div>
7  <!--<input [(ngModel)]="str" type="text"-->
8  <input #data type="text" (keyup)="fun(data.value)">
9
10 <h3>Length of string: {{fun(data.value)}}</h3>
11
```

src ▸ app ▸ TS app.component.ts ▸ ...

```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'Assignment 8';
10
11    public fun(str)
12    {
13      return str.length;
14    }
15  }
```

