

Marvellous Infosystems : Angular Assignment 8

Name: Shrirang Jagdish Nikam

Enrollment No: 396AM_Shrirang

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

Answer:

app.component.html:

```
<app-child (childData)="onChildDataReceived($event)"></app-child>
<p>Text from child component: {{ childText }}</p>
```

app.component.ts:

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  childText: string = "";

  onChildDataReceived(data: string) {
    this.childText = data;
  }
}
```

Child.component.html:

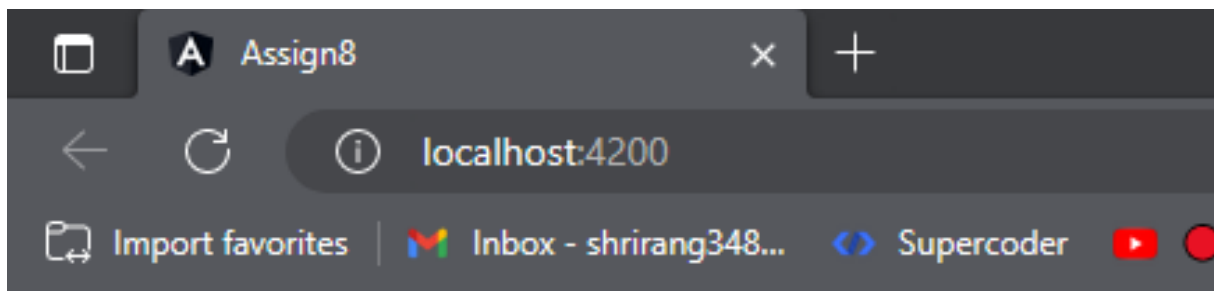
```
<div>
  <input type="text" #textInput>
  <button (click)="sendData(textInput.value)">Send Data</button>
</div>
```

child.component.ts:

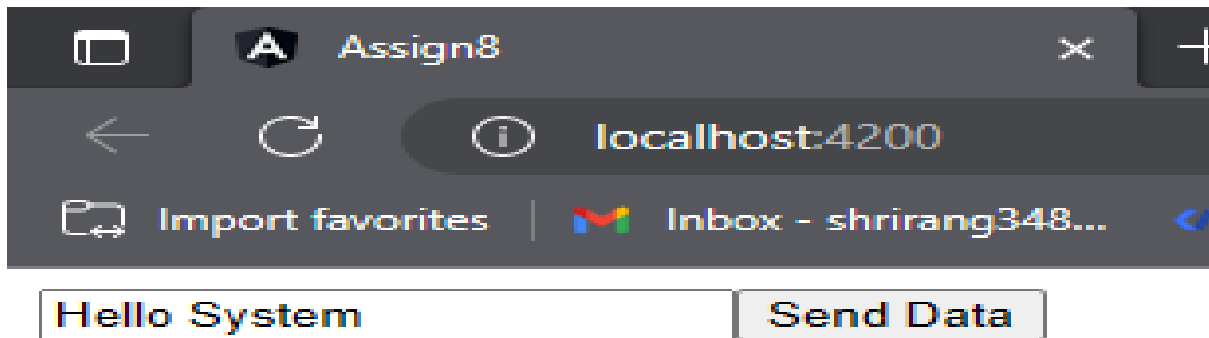
```
import { Component, EventEmitter, Output } from '@angular/core';

@Component({
  selector: 'app-child',
  templateUrl: './child.component.html',
  styleUrls: ['./child.component.css']
})
export class ChildComponent {
  @Output() childData = new EventEmitter<string>();

  sendData(data: string) {
    this.childData.emit(data);
  }
}
```



Text from child component:



Text from child component: Hello System

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 app. In this application parent and child components communicate with each other by sending the data.

Answer:

parent.component.ts:

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

@Component({
  selector: 'app-parent',
  template: `
    <h1>Parent Component</h1>
    <app-child (messageEvent)="receiveMessage($event)"></app-child>
    <p>{{ message }}</p>
  `,
})
export class ParentComponent {
  message!: string;

  receiveMessage($event: string) {
    this.message = $event;
  }
}
```

child.component.ts:

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

@Component({
```

```

selector: 'app-child',
template: `
  <h2>Child Component</h2>
  <button (click)="sendMessage()">Send Message to Parent</button>
`,
})
export class ChildComponent {
  @Output() messageEvent = new EventEmitter<string>();

  message = 'Hello from Child';

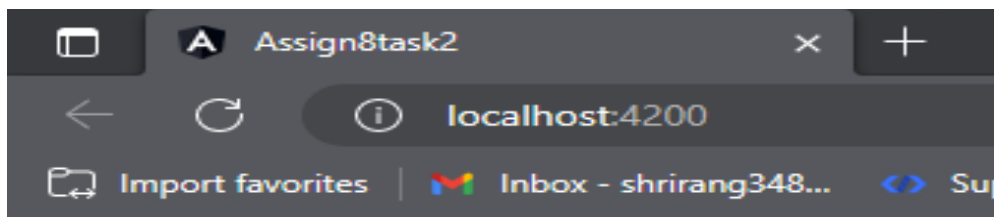
  sendMessage() {
    this.messageEvent.emit(this.message);
  }
}

```

app.component.html:

```
<app-parent></app-parent>
```

Output:



Parent Component

Child Component

Send Message to Parent

Hello from Child

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)

Answer:

app.component.html:

```
<input [(ngModel)]="inputText" (input)="updateLength()" placeholder="Enter  
some text">  
<p>The length of the text is: {{ textLength }}</p>
```

app.component.ts:

```
import { Component } from '@angular/core';  
  
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
  inputText = '';  
  textLength = 0;  
  
  updateLength() {  
    this.textLength = this.inputText.length;  
  }  
}
```

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

Answer:

app.component.html:

```
<input [(ngModel)]="text" (keyup)="onKeyUp($event)">
```

app.component.ts:

```
import { Component } from '@angular/core';  
  
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})
```

```
})  
export class AppComponent {  
  text: string;  
  
  onKeyUp(event: any): void {  
    console.log(event.target.value);  
  }  
}
```

app.component.css:

```
input {  
  font-size: 20px;  
  padding: 10px;  
}
```