**COMPONENTS AND ROUTING**

**Question 1**

**Component and Routing – Create View Employee Component**   
  
Earlier employee details were created in the default app component itself.

* This now has to be moved to a new component named "view-emp". Follow the steps below:
* Create a new component in angular-learning application created earlier with component name "view-emp".
* Copy and paste relevant code from html and component source from app component to this new component.
* Create another component "edit-emp" to have a placeholder for another component that will be developed later. This is primarily to check how router navigation works.
* Include routing with router link “View Employee” and "Edit Employee" pointing to this new component. Clicking either one of the link should display the respective component. Refer sample layout below.

**View Employee Component Layout** 

|  |
| --- |
| https://cognizant.e-box.co.in/uploads/Image/01fseangular/comp1.JPG |

**Edit Employee Component Layout** 

|  |
| --- |
| https://cognizant.e-box.co.in/uploads/Image/01fseangular/comp2.JPG |

If required, use the CSS below to underline the component title that is currently active. The class active-link is the one that needs to be applied in routerLinkActive.

|  |
| --- |
| body {  font-family: Segoe UI, Arial  }  a:visited {  color: blue  }  a {  text-decoration: none  }  .active-link {  border-bottom: Solid 1px blue  } |

**Answer 1**

**Employee.ts , Department.ts and Skill.ts same as previous queations**

**App.component.ts**

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

import {Employee} from './Employee';

import { Department } from './Department';

import { Skills } from './skills';

@**Component**({

selector: 'app-root',

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

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

})

export class **AppComponent** {

}

**App-routing.module.ts**

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

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

import { ViewEmpComponent } from './view-emp/view-emp.component';

import { EditEmpComponent } from './edit-emp/edit-emp.component';

const routes: **Routes** = [

{

path:'view',

component:ViewEmpComponent

},

{

path:'edit',

component:EditEmpComponent

}

];

@**NgModule**({

imports: [RouterModule.**forRoot**(routes)],

exports: [RouterModule]

})

export class **AppRoutingModule** { }

**App.component.html**

<b>Angular Learning</b>&nbsp;&nbsp;&nbsp;<a *routerLink*="view">View Employee</a>&nbsp;&nbsp;<a *routerLink*="edit">Edit Employee</a>

<hr>

<router-outlet></router-outlet>

**View-emp.component.ts**

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

import { Employee } from '../Employee';

import { Department } from '../Department';

import { Skills } from '../skills';

@**Component**({

selector: 'app-view-emp',

templateUrl: './view-emp.component.html',

styleUrls: ['./view-emp.component.css']

})

export class **ViewEmpComponent** implements **OnInit** {

constructor() { }

**ngOnInit**() {

}

favoriteMovie='Lord of the Rings';

public employee:**Employee**

={

Id:1,

Name:'John',

Salary:10000,

Permanent:true,

dateOfBirth:new **Date**('12/31/2000')

};

public dept:**Department**={

DeptId:1,

DeptName:'payroll'

};

public skill:**Skills**[]

=[

{

skillId:1,skillName:'HTML'

},

{

skillId:2,skillName:'Css'

},

{

skillId:3,skillName:'Javascript'

}

]

}

**View-emp.component.html**

<h3>Employee Details</h3><br/>

<div><strong>ID: </strong>{{employee.Id}}</div><br/>

<div><strong>Name: </strong>{{employee.Name}}</div><br/>

<div><strong>Salary: </strong>{{employee.Salary|currency}}</div><br/>

<div *\*ngIf*="employee.Permanent"> <strong>Permanent</strong>

{{employee.Permanent}}

</div><br/>

<div><strong>Department Id: </strong>{{dept.DeptId}}</div><br/>

<div><strong>Department Name: </strong>{{dept.DeptName}}</div><br/>

<strong>Skills</strong><br/><br/>

<strong>Skill Id Skill Name</strong>

<div *class*="row">

<div *\*ngFor*="let s of skill">

<div *class*="col-xs-1" >

{{s.skillId}}

</div>

<div *class*="col-xs-11" >

{{s.skillName}}

</div>

</div>

</div>

<br/>

<br/>

<strong>Date Of Birth : </strong>{{employee.dateOfBirth|date}}

**View-emp.component.css**

body {

font-family: Segoe UI, Arial

}

a:visited {

color: blue

}

a {

text-decoration: none

}

.active-link {

border-bottom: Solid 1px blue

}

**Edit-emp.component.ts**

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

@**Component**({

selector: 'app-edit-emp',

templateUrl: './edit-emp.component.html',

styleUrls: ['./edit-emp.component.css']

})

export class **EditEmpComponent** implements **OnInit** {

constructor() { }

**ngOnInit**() {

}

}

**Edit-emp.component.html**

<p>edit-emp works!</p>

**Question 2**

**Event Handling - Display value based on button click**   
  
In online food ordering apps, to add a specific food item with multiple quantity, there will be a plus and minus button that helps in increasing or decreasing the number of items. We are gradually going to build a component that will achieve this functionality.  
  
As part of this particular exercise we will create a button named "Click me". Clicking on this button should display the message "Click me button clicked!"  
  
Create a new component named "quantity-increment".  
  
In this component include a separate router link as per the layout below.  
  
Layout before button click

|  |
| --- |
| https://cognizant.e-box.co.in/uploads/Image/01fseangular/event1.JPG |

Layout after button click

|  |
| --- |
| https://cognizant.e-box.co.in/uploads/Image/01fseangular/event2.JPG |

Karma Testing

* Test if button click displays the message "Click me button clicked!"
* Include below imports:

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

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

* Define debugElement after fixture:

let debugElement: DebugElement;

* Include form module import in beforeEach(async) after declaration:

imports:[FormsModule]

* Include debug element in beforeEach() after component:

debugElement = fixture.debugElement;

* Get value from DOM using debugElement. Below code gets the button element from DOM:

let clickMeButton = debugElement.nativeElement.querySelector('#clickMe');

* Use click() method on clickMeButton to trigger the click event:
* Use fixture.detectChanges() to get the changes made because of the click event.
* Use textContent property of DOM element to get the text content within it:

debugElement.nativeElement.querySelector('#simpleClick').textContent

*Employee.ts, Department.ts Skill.ts, App.component.ts, App.component.html, View-emp.component.ts, View-emp.component.html, Edit-emp.component.ts and, Edit-emp.component.html same as previous questions*

**App-routing.module.ts**

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

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

import { ViewEmpComponent } from './view-emp/view-emp.component';

import { EditEmpComponent } from './edit-emp/edit-emp.component';

import { QuantityIncrementComponent } from './quantity-increment/quantity-increment.component';

const routes: **Routes** = [

{

path:'view',

component:ViewEmpComponent

},

{

path:'edit',

component:EditEmpComponent

},

{

path:'quant-incr',

component:QuantityIncrementComponent

}

];

@**NgModule**({

imports: [RouterModule.**forRoot**(routes)],

exports: [RouterModule]

})

export class **AppRoutingModule** { }

**quantity-increment.component.ts**

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

@**Component**({

selector: 'app-quantity-increment',

templateUrl: './quantity-increment.component.html',

styleUrls: ['./quantity-increment.component.css']

})

export class **QuantityIncrementComponent** implements **OnInit** {

constructor() { }

**ngOnInit**() {

}

}

**quantity-increment.component.html**

<p>Simple Button Click Event !!</p>

<button *(click)*="msg='Click me button clicked !'">Click me</button>

<br/>

{{msg}}

**Question 3**

**Event Handling and Two Way Binding - increment value in textbox based on button click**   
  
Component: quantity-selector  
  
Modify the “Click me” button as “Add”. Include a new textbox with default value zero. Clicking on the “Add” button should increment value in the textbox by 1.   
  
Sample Layout

|  |
| --- |
| https://cognizant.e-box.co.in/uploads/Image/01fseangular/event3.JPG |

**Answer 3**

*Employee.ts, Department.ts Skill.ts, App.component.ts, App.component.html, View-emp.component.ts, View-emp.component.html, Edit-emp.component.ts, Edit-emp.component.html, quantity-increment.component.html and quantity-increment.component.ts* ***same as previous questions***

**App-routing.module.ts**

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

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

import { ViewEmpComponent } from './view-emp/view-emp.component';

import { EditEmpComponent } from './edit-emp/edit-emp.component';

import { QuantityIncrementComponent } from './quantity-increment/quantity-increment.component';

import { QuantitySelectorComponent } from './quantity-selector/quantity-selector.component';

const routes: **Routes** = [

{

path:'view',

component:ViewEmpComponent

},

{

path:'edit',

component:EditEmpComponent

},

{

path:'quant-incr',

component:QuantityIncrementComponent

},

{

path:'quant-sel',

component:QuantitySelectorComponent

}

];

@**NgModule**({

imports: [RouterModule.**forRoot**(routes)],

exports: [RouterModule]

})

export class **AppRoutingModule** { }

**Quantitiy-selector.component.ts**

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

@**Component**({

selector: 'app-quantity-selector',

templateUrl: './quantity-selector.component.html',

styleUrls: ['./quantity-selector.component.css']

})

export class **QuantitySelectorComponent** implements **OnInit** {

num:number=0;

constructor() { }

**ngOnInit**() {

}

}

**Quantity-selector.component.html**

<p>Model Binding</p>

<br/>

<input *type*="text" *id*="num" *[(ngModel)]*="num" *name*="num"/>

&nbsp;&nbsp;&nbsp;<button *(click)*="num=num+1" >Add</button>

**Question 4**

**Event Handling and Two Way Binding – Implement quantity selector for Food App**   
  
Find below a sample layout for selecting quantity in a\ food app.  
  
This provides option to choose the quantity using a textbox having a plus icon in the right hand side and minus icon in the left hand side.  
  
The default value in the textbox should be zero.  
  
Clicking plus should increment the value. Clicking minus should decrement the value.  
  
If the value is zero, then minus icon should be disabled.  
  
Sample Layout 

|  |
| --- |
| https://cognizant.e-box.co.in/uploads/Image/01fseangular/event4.JPG |

Note: Minus button is disabled 

|  |
| --- |
| https://cognizant.e-box.co.in/uploads/Image/01fseangular/event5.JPG |

Note: Minus button is enabled   
  
**Karma Testing**

* Include test to check if increment works correctly.
* Include value attribute with interpolation to quantity textbox. This is required for Karma testing to pick the data.
* Include fixture.detectChanges() call in the start of the test method to ensure the test runs after initial value 0 is set in the DOM element.

**Answer 4**

*Employee.ts, Department.ts Skill.ts, App.component.ts, App.component.html, View-emp.component.ts, View-emp.component.html, Edit-emp.component.ts, Edit-emp.component.html, quantity-increment.component.html , quantity-increment.component.ts and app-routing.module.ts* ***same as previous questions***

**Quantitiy-selector.component.ts**

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

@**Component**({

selector: 'app-quantity-selector',

templateUrl: './quantity-selector.component.html',

styleUrls: ['./quantity-selector.component.css']

})

export class **QuantitySelectorComponent** implements **OnInit** {

num:number=0;

constructor() { }

**ngOnInit**() {

}

}

**Quantity-selector.component.html**

<p>Model Binding</p>

<br/>

<button *(click)*="num=num-1" *[disabled]*="num==0"><i *class*="fa fa-minus"></i></button>

<input *type*="text" *id*="add" *[(ngModel)]*="num" *name*="add"/>

&nbsp;&nbsp;&nbsp;<button *(click)*="num=num+1" ><i *class*="fa fa-plus"></i></button>