**HANDS-ON STAGE-3 ANGULAR DAY-70**

**EDITEMPLOYEE.COMPONENT.HTML**

<strong>Edit Employees</strong>

<br>

<br>

<form><label><strong>Name:</strong><input type="text" name="name1" #name1></label>

    {{name1.value}}

    <br>

    <br></form>

<hr>

 <label>

     <strong>Lname:</strong>

    <input type="text" name="name" [(ngModel)]="employees" #name=ngModel PlaceHolder="Lname" pattern="[a-z]+" minlength="3" maxlength="10">

    {{employees}}

    <div>name.invalid - {{name.invalid}}</div>

    <div>name.touched - {{name.touched}}</div>

    <div>name.dirty - {{name.dirty}}</div>

    <div>name.errors.minlength or maxlength - {{name.errors?.minlength | json}}</div>

 </label>

<hr>

<br><br>

<label>

    <strong>Name:</strong>

    <input type="text" name="t1" [(ngModel)]="myname" minlength="2" maxlength="10" #t1=ngModel required="true"><br>

</label>

<span class="c1" \*ngIf="t1.invalid && t1.dirty">Name must be 2 characters long</span>

<br>

<br>

<hr>

<form #empForm="ngForm" (submit)="OnSubmit()">

<div class="block">

    <label>

        <strong>Name:</strong>

        <input type="text" [(ngModel)]="emp.Name" [ngModelOptions]="{standalone: true}">

    </label>

    <br>

    <br>

    <label>

        <strong>Salary:</strong>

        <input type="number" [(ngModel)]="emp.Salary" Pattern="\d+" [ngModelOptions]="{standalone: true}">

    </label>

    <br><br>

    <label>

        <strong>Is Permanent or Not?</strong>

        <input type="radio" name="permanent" [(ngModel)]="emp.Permanent" [value]="true">Yes

        <input type="radio" name="permanent" [(ngModel)]="emp.Permanent" [value]="false">No

    </label>

    <label>

    </label>

    <button type="submit" [disabled]="!empForm.valid">Save</button>

</div>

</form>

<div class="block">

    <strong>Employee Model</strong><br>

    <pre>{{emp | json }}</pre>

</div>

**EDITEMPLOYEE.COMPONENT.TS**

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

@Component({

  selector: 'app-editemp',

  templateUrl: './editemp.component.html'

})

export class EditempComponent{

  public employees:string;

  public myname:string;

  constructor() {

    this.employees = "";

    this.myname="";

  }

  public emp:employee={Id:1,Name:"John",Salary:10000,Permanent:false, Dept:{Id:1,Name:"Payroll"},skills:[{id:1,Name:"html"},{id:2,Name:"Css"},{id:3,Name:"Javascript"}]};

  public departmentslist:any[]=[{Id:1,Name:"Payroll"},{Id:2,Name:"Internal"},{Id:3,Name:"Hr"}];

  public OnSubmit():void{

    console.log(this.emp);

  }

}

class employee{

  public Id:number=0;

  public Name:string="";

  public Salary:number=0;

  public Permanent:boolean=true;

  public Dept:Department={Id:1,Name:"Payroll"};

  public skills:any[]=[{}];

}

class Department{

  public Id:number=1;

  public Name:string="";

}

**OUTPUT:**

**Graphical user interface, text, application, email

Description automatically generated**

<h2>Edit Employee</h2>

<label>

    Name: <input type="text" [formControl]="Name">

</label>

<br>

<br>

<div>

    <form [formGroup]="dept">

        <h3>Department</h3>

        <select formControlName="department">

        <option \*ngFor="let department of departments"

        [ngValue]="department"

        [selected]="department.Id == emp.Dept.Id">

        {{department.Name}}

        </option>

        </select>

    </form>

</div>

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

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

@Component({

  selector: 'app-edit-emp-reactive',

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

})

export class EditEmpReactiveComponent{

  public emp:employee={Id:1,Name:"John",Salary:10000,Permanent:false, Dept:{Id:1,Name:"Payroll"},skills:[{id:1,Name:"html"},{id:2,Name:"Css"},{id:3,Name:"Javascript"}]};

  public departments:any[]=[{Id:1,Name:"Payroll"},{Id:2,Name:"Internal"},{Id:3,Name:"Hr"}];

  Name=new FormControl('John');

  public empForm:FormGroup = new FormGroup({

    'name': new FormControl(this.emp.Name, [

    Validators.required,

    Validators.minLength(4),

    Validators.maxLength(20)

    ])

    });

    public dept:FormGroup = new FormGroup(

      'department': new FormControl("")

          });

}

class employee{

  public Id:number=0;

  public Name:string="";

  public Salary:number=0;

  public Permanent:boolean=true;

  public Dept:Department={Id:1,Name:"Payroll"};

  public skills:any[]=[{}];

}

class Department{

  public Id:number=1;

  public Name:string="";

}

**OUTPUT:**

**Graphical user interface, text, application

Description automatically generated**