Index.html

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Formswork</title>

<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

<body>

<h4>Angular Routing</h4>

<app-root></app-root>

</body>

</html>

app.component.html

<div class="container">

<div class="row">

<div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">

<ul class="nav nav-tabs">

<li role="presentation" routerLinkActive="active" [routerLinkActiveOptions]="{exact:true}"><a routerLink="/editemp">Edit-employee-Template</a></li>

</ul>

</div>

</div>

</div>

<div class="row">

<div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">

<router-outlet></router-outlet>

</div>

</div>

app.component.ts

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

title = 'formswork';

constructor(private route :Router){

this.route.navigate(['/editemp'])

}

}

app.module.ts

import { BrowserModule } from '@angular/platform-browser';

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

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { EditEmpTemplateDrivenComponent } from './edit-emp-template-driven/edit-emp-template-driven.component';

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

@NgModule({

declarations: [

AppComponent,

EditEmpTemplateDrivenComponent

],

imports: [

BrowserModule,

AppRoutingModule,

FormsModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

app.routing.module.ts

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

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

import { EditEmpTemplateDrivenComponent } from './edit-emp-template-driven/edit-emp-template-driven.component';

const routes: Routes = [{

path:"editemp",

component:EditEmpTemplateDrivenComponent

}

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

employee.ts

import { Department } from './Department';

import { Skill } from './Skill';

export interface Employee extends Department{

id:number;

name:string;

salary:number;

skills:Skill[];

permenamt:string;

DateOfBirth:Date;

}

Department.ts

export interface Department {

deptId:number;

deptName:string;

}

Skill.ts

export interface Skill{

skill\_id:number;

skill\_name:string;

}

edit-emp-template-driven.component.ts

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

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

@Component({

selector: 'app-edit-emp-template-driven',

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

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

})

export class EditEmpTemplateDrivenComponent implements OnInit {

constructor() {

}

ngOnInit(): void {

}

employee:Employee={

id:10

,name:'giri',

salary:40000,

deptId:1,

deptName:"HR",

skills:[{skill\_id:122,skill\_name:"Java"},

{skill\_id:123,skill\_name:"c"},

{skill\_id:124,skill\_name:"c++"}],

permenamt:"yes",

DateOfBirth : new Date()

};

}

edit-emp-template-driven.component.html

<div class="container-fluid">

<h4>Edit Employee</h4>

<form #empForm="ngForm">

<div class="form-group">

<label>Name</label>

<input type="text" class="form-control" name="name" [(ngModel)]="employee.name" #name="ngModel"

[class.is-invalid]="name. Invalid && name.touched && name.dirty" minlength="2" maxlength="10">{{employee.name}}

<div \*ngIf="name.errors && (name.invalid || name.touched || name.dirty)">

<small class="text-danger" \*ngIf="name.errors.required">Name is required</small>

<small class="text-danger" \*ngIf="name.errors.minlength">Name must be atleast 2 charachters long</small>

<small class="text-danger" \*ngIf="name.errors.maxlength">Name must not contain more than 10 charachters</small>

</div>

</div>

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

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

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

<div>name.errors.required - {{name.errors?.required}}</div>

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

<div class="form-group">

<label>Salary</label>

<input type="text" class="form-control" name="salary" [(ngModel)]="employee.salary" #salary="ngModel">

</div>

<div class="mb-3">

<label>Is Permanant or not??</label>

<div class="form-check">

<input class="form-check-input" type="radio" name="permenamt" value="yes" [(ngModel)]="employee.permenamt" ngModel>

<label class="form-check-label">Yes</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" name="permenamt" value="no" [(ngModel)]="employee.permenamt" ngModel>

<label class="form-check-label">No</label>

</div>

</div>

<button class="btn btn-primary"[disabled]="empForm.form.invalid" type="submit">Save</button>

</form>

</div>

<div class="block">

<strong>Employee Model</strong>

<pre>{{employee| json }}</pre>

</div>

<div class="block">

<strong>Angular Generated Form Model</strong><br>

<pre>{{empForm.value | json}}</pre>

</div>

edit-emp-template-driven.component.css

.block {

float:left;

width:33%

}

