3**. Event Management App**

**Home.component.html**

<head>

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"

    integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"

    integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

    crossorigin="anonymous"></script>

  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"

    integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"

    crossorigin="anonymous"></script>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"

    integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"

    crossorigin="anonymous"></script>

</head>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

  <h2>Employee Managment</h2>

        <button class="nav-link" (click)="mainlist()">Home </button>

        <div class="btn"><button class="nav-link" (click)="logout()">Logout </button></div>

</nav>

<!-- Add Form -->

<div class="maincontainer" \*ngIf="showAddform">

  <form>

    <div class="text-center" \*ngIf="showMsg">

      <div class="col-xs-12">

        <p class="alert alert-success">

          User Added Succesfully.

        </p>

      </div>

    </div>

    <h2 class="text-center">Add Employee</h2>

    <div class="form-outline mb-4">

      Employee Id:<input type="text" name="name" class="form-control " [(ngModel)]="employe.empid">

    </div>

    <div class="form-outline mb-4">

      <label> Employee FirstName</label>

      <input type="text" name="fname" class="form-control " [(ngModel)]="employe.firstname">

    </div>

    <div class="form-outline mb-4">

      Employee Last Name:

      <input type="text" name="lname" class="form-control " [(ngModel)]="employe.lastname"><br>

    </div>

    <div class="form-outline mb-4">

      Employee email:<input type="text" name="email" class="form-control" [(ngModel)]="employe.email"><br>

    </div>

    <div class="text-center">

      <input type="submit" class="btn btn-primary" value="Register" (click)="registerNow()">&nbsp;&nbsp;

      <button (click)="emplist()" class="btn btn-primary">Close</button>

    </div>

  </form>

</div>

<!-- Main list -->

<div \*ngIf="showList" class="list">

  <div \*ngIf="showdelMsg">

    <div class="col-xs-6">

      <p class="alert alert-success">

        Delete Succesfully.

      </p>

    </div>

  </div>

  <div class="addnewbtn">

    <button (click)="addform()" class="btn btn-dark">Add New Employee</button>

  </div>

  <table>

    <tr>

      <th>#</th>

      <th>FirstName</th>

      <th>LastName</th>

      <th>Email</th>

      <th>Action</th>

    </tr>

    <tr \*ngFor="let emp of employee">

      <td>{{emp.id}}</td>

      <td>{{emp.firstname}}</td>

      <td>{{emp.lastname}}</td>

      <td>{{emp.email}}</td>

      <td><button class="btn btn-info" (click)="viewdetail(emp.id)">View</button></td>

    </tr>

    <!-- Add more rows if needed -->

  </table>

</div>

<!-- cardview -->

<div class="pcard" \*ngIf="showView">

  <div class="card">

    <p>Details of <b>{{employee.firstname}} {{employee.lastname}}</b> </p>

    <div class="card-body">

      <p><strong>Employee id</strong>&nbsp;&nbsp;&nbsp;&nbsp;{{employee.empid}}</p>

      <p><strong>Employee First Name</strong>&nbsp;&nbsp;&nbsp;&nbsp;{{employee.firstname}}</p>

      <p><strong>Employee Last Name</strong>&nbsp;&nbsp;&nbsp;&nbsp;{{employee.lastname}}</p>

      <p><strong>Employee Email</strong>&nbsp;&nbsp;&nbsp;&nbsp;{{employee.email}}</p>

      <button class="btn btn-primary" (click)="elist()">Emplist </button>&nbsp;

      <button class="btn btn-primary" (click)="updateemp(employee.empid)">Update </button> &nbsp;

      <button class="btn btn-primary" (click)="deleteemp(employee.empid)">Delete </button> &nbsp;

    </div>

  </div>

</div>

<!-- updateForm -->

<div \*ngIf="showUpdateForm">

  <div class="maincontainer">

    <form>

      <div class="text-center" \*ngIf="showUpdateMsg">

        <div class="col-xs-12">

          <p class="alert alert-success">

            Employee Update Succesfully.

          </p>

        </div>

      </div>

      <h2 class="text-center">Update Employee</h2>

      <div class="form-outline mb-4">

        <input type="hidden" name="name" class="form-control " [(ngModel)]="employe.empid">

      </div>

      <div class="form-outline mb-4">

        <label> Employee FirstName</label>

        <input type="text" name="fname" class="form-control " [(ngModel)]="employe.firstname">

      </div>

      <div class="form-outline mb-4">

        Employee Last Name:

        <input type="text" name="lname" class="form-control " [(ngModel)]="employe.lastname"><br>

      </div>

      <div class="form-outline mb-4">

        Employee email:<input type="text" name="email" class="form-control" [(ngModel)]="employe.email"><br>

      </div>

      <div class="text-center">

        <input type="submit" class="btn btn-primary" value="update" (click)="update()">&nbsp;

        <button (click)="updempcard(employe.empid)" class="btn btn-primary">Close</button>

      </div>

    </form>

  </div>

</div>

[**Home.component.ts**](http://Home.component.ts)

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

import { emp } from '../Emp';

import { EmpserService } from '../empser.service';

import { Router } from '@angular/router';

import { Location } from '@angular/common';

@Component({

  selector: 'app-home',

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

  styleUrls: ['./home.component.css']

})

export class HomeComponent implements OnInit  {

  employee:any;

  employe:emp=new emp();

  showAddform: boolean=false;

  showList: boolean = true;

  showView: boolean=false;

  showUpdateForm:boolean=false;

  ids:any;

  constructor(private service:EmpserService,private route:Router,private router: Router){}

  //Global Func

  mainlist(){

   this.showUpdateForm=false;

  this.showAddform=false;

  this.showView=false;

  this.ngOnInit();

  this.showList=true;

  }

  logout(){

    this.router.navigate(['login']);

  }

  // Addform function

  addform(){

     this.showAddform=true;

      this.showList=false;

      this.showMsg=false;

  }

  message:"";

  showMsg: boolean = false;

  isDisabled: boolean = false;

  public  registerNow(){

    //json resposne

  let response=this.service.registration(this.employe);

  response.subscribe((data:any)=>this.message=data);

  this.showMsg=true;

  this.employe=new emp();

  }

  emplist(){

    this.showAddform=false;

    this.ngOnInit();

    this.showList=true;

  }

 // View function

  viewdetail(id:number){

    this.ids=id;

    let response=this.service.getbyid(this.ids);

    response.subscribe((data:any)=>this.employee=data);

    this.showView=true;

    this.showList=false;

    // this.route.navigate(['view',id]);

  }

  elist(){

    this.showView=false;

    this.ngOnInit();

    this.showList=true;

  }

  //  update detail

  showUpdateMsg:boolean=false;

  updateemp(id:number){

    this.ids=id;

    let response=this.service.getbyid(this.ids);

    response.subscribe((data:any)=>this.employe=data);

  this.showView=false;

  this.showUpdateForm=true;

 }

 update(){

  let response=this.service.update(this.employe);

  response.subscribe((data:any)=>this.employe=data);

  this.showUpdateMsg=true;

  this.employe=new emp();

 }

 updempcard(id:any){

  this.ids=id;

  let response=this.service.getbyid(this.ids);

  response.subscribe((data:any)=>this.employee=data);

  this.showView=true;

  this.showUpdateForm=false;

 }

  //  delete detail

emplo:any[]=[];

showdelMsg:boolean=false;

  public deleteemp(id:any){

    this.service.deleteemp(id).subscribe(()=>{this.emplo=this.emplo.filter(emplo=>emplo.id!==id)});

    this.showView=false;

    this.showdelMsg=true;

    this.ngOnInit();

    this.showList=true;

    this.employe=new emp();

  }

  ngOnInit(): void {

    let response= this.service.getall();

    response.subscribe((data:any)=>this.employee=data);

  }

}

**Home.component.css**

table {

    border-collapse: collapse;

    width: 100%;

  }

  th, td {

    border: 1px solid #ddd;

    padding: 8px;

    text-align: left;

  }

  th {

    background-color: #f2f2f2;

  }

  tr:nth-child(even) {

    background-color: #f9f9f9;

  }

  tr:hover {

    background-color: #e9e9e9;

  }

  .maincontainer{

    margin: auto;

    width: 50%;

}

form{

  margin-top: 10px;

  padding: 20px;

  border: 1px solid#000000;

}

p{

  border-bottom: 1px solid black;

}

.pcard{

  padding-top: 20px;

}

.card{

  border: 1px solid black;

  display: flex;

  flex-direction: column;

  margin: auto;

  width: 80%;

  justify-content: center;

}

.list .addnewbtn{

  float: right;

  padding: 10px 5px;

}

**Login.component.css**

.maincontainer{

    margin: auto;

    width: 50%;

    padding-top: 250px;

}

form{

    border: 1px solid black;

    margin: auto;

    height: 60%;

    padding:10px;

    width: 75%;

    display: flex;

    flex-direction: column;

    align-items: center;

}

form div{

    padding: 0 15px;

}

**Login.component.html**

<head>

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"

      integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"

      integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

      crossorigin="anonymous"></script>

  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"

      integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"

      crossorigin="anonymous"></script>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"

      integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"

      crossorigin="anonymous"></script>

</head>

<div class="maincontainer">

<form>

  <h2 class="text-center"> Login Form</h2>

  <div class="text-center"  \*ngIf="message">

    <div class="col-xs-12">

      <p  class="alert alert-danger" >

          Wrong id and Password;

      </p>

    </div>

  </div>

    <!-- Email input -->

    <div class="form-outline mb-4">

      <label class="form-label" for="form2Example1">Email address</label>

      <input type="email" id="form2Example1" class="form-control" name="email" [(ngModel)]="user" />

    </div>

    <!-- Password input -->

    <div class="form-outline mb-4">

      <label class="form-label" for="form2Example2">Password</label>

      <input type="password" id="form2Example2" class="form-control"  name="password" [(ngModel)]="password" />

    </div>

    <!-- Submit button -->

    <div class="text-center">

    <button  class="btn btn-success" (click) = "login()" > Login </button>

    </div>

  </form>

  </div>

Login.component.ts

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

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

import { EmpserService } from '../empser.service';

import { Router } from '@angular/router';

@Component({

  selector: 'app-login',

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

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

})

export class LoginComponent {

  admin:admin=new admin();

  user="";

  password="";

  message: boolean=false;

  constructor(private service : EmpserService,private router:Router){}

  ngOnInit():void{

    let response=this.service.login();

    response.subscribe((data:any)=>this.admin=data);

  }

  login(){

   if(this.admin.email===this.user&& this.admin.password===this.password){

    this.router.navigate(['home']);

   }

   else{

  this.message=true;

  this.router.navigate(['login']);

  }

  }

}

Admin.ts

export class admin{

   email:string;

   password:string;

}

App.module.ts

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

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

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

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

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

import { HttpClientModule } from '@angular/common/http'

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

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

import { HomeComponent } from './home/home.component';

import { LoginComponent } from './login/login.component';

const routes:Routes=[

  {path:"",redirectTo:"login",pathMatch:"full"},

  {path:"login",component:LoginComponent},

  {path:"home",component:HomeComponent},

  ]

@NgModule({

  declarations: [

    AppComponent,

    HomeComponent,

    LoginComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    FormsModule,

    HttpClientModule,

    BrowserAnimationsModule,

    RouterModule.forRoot(routes)

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

app.component.html

<router-outlet></router-outlet>

Emp.ts

export class emp{

    empid:string;

    firstname:string;

    lastname:string;

    email:string;

}

Empser.service.ts

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

import { HttpClient } from '@angular/common/http';

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class EmpserService {

  constructor(private http:HttpClient) { }

   public login(){

    let res =this.http.get("http://localhost:3000/Admin");

   return res;

   }

  public registration(value:any){

    return this.http.post("http://localhost:3000/employee",value,{responseType:'text' as 'json'});

  }

  public getall(){

    return this.http.get("http://localhost:3000/employee");

  }

  public getbyid(id:any){

  let res =this.http.get("http://localhost:3000/employee/"+id);

   return res;

  }

  public update(value:any):Observable<any>{

    return this.http.put("http://localhost:3000/employee/"+value.id,value);

  }

  public deleteemp(id:number):Observable<any>{

    return this.http.delete("http://localhost:3000/employee/"+id);

     }

}

Json-typing.ts

declare module "\*.json" {

    const value: any;

    export default value;

  }

Index.html

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Employee Managment</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>

  <app-root></app-root>

</body>

</html>

Db.json

{

  "posts": [

    {

      "id": 1,

      "title": "json-server",

      "author": "typicode"

    }

  ],

  "comments": [

    {

      "id": 1,

      "body": "some comment",

      "postId": 1

    }

  ],

  "profile": {

    "name": "typicode"

  }

}

Emp.json

{

  "Admin": {

    "email": "admin@admin.com",

    "password": "admin"

  },

  "employee": [

    {

      "empid": "1",

      "firstname": "Sebastian",

      "lastname": "Eschweiler",

      "email": "sebastian@codingthesmartway.com",

      "id": 1

    },

    {

      "empid": "2",

      "firstname": "Steve",

      "lastname": "Palmer",

      "email": "steve@codingthesmartway.com",

      "id": 2

    },

    {

      "empid": "3",

      "firstname": "Ann",

      "lastname": "Smith",

      "email": "ann@codingthesmartway.com",

      "id": 3

    }

  ]

}

Package.json

"json-server": "json-server --watch db.json emp.json"