## WRITEUP:

GITHUB LINK: <https://github.com/SweathaRJ/Event-Management-App>

Step 1: create one folder in your drive named “Event Management Project”.

Step 2: open folder in visual code.

Step 3: create new app by using below commands

* ng new EventManagement
* ng g c employee-dashboard
* ng g c signin
* create one folder named shared and inside create one file named api.service.ts

Step 4: create on db.json file on right clicking the app and apply the code

{

  "posts": [

    {

      "id": 9,

      "firstName": "Rahul",

      "lastName": "swea",

      "email": "rahul@gmail.com",

      "mobile": "123456780",

      "salary": "60000"

    },

    {

      "id": 11,

      "firstName": "Kaviya",

      "lastName": "S",

      "email": "kavi@gmail.com",

      "mobile": "9867453201",

      "salary": "75000"

    }

  ],

  "comments": [],

  "profile": {

    "name": "typicode"

  }

}

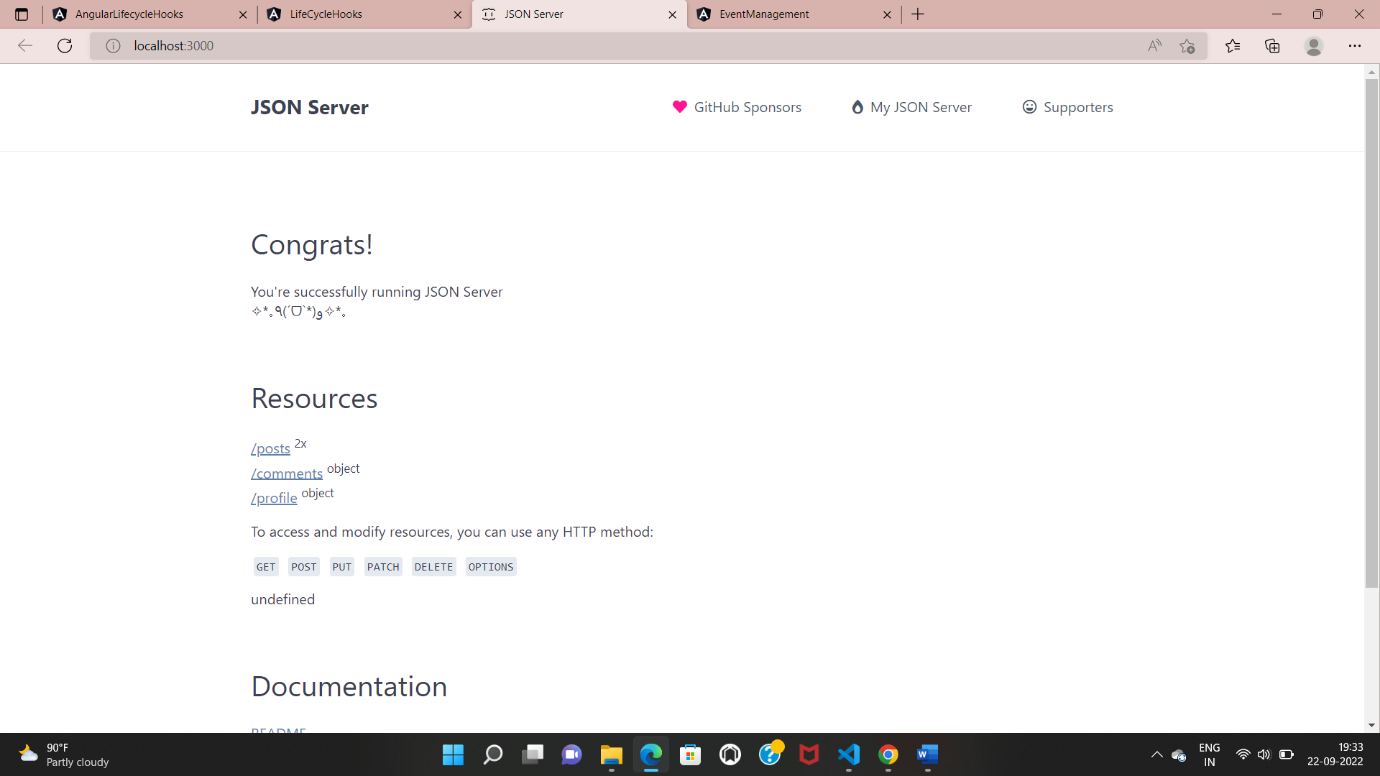
Give this command inPackage.json

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

Then in command prompt give

>npm install -g json-server

>npm run json:server



Step 6: goto these component files which we created and include the code.

app-routing.module.ts

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

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

import { EmployeeDashboardComponent } from './employee-dashboard/employee-dashboard.component';

import { SigninComponent } from './signin/signin.component';

const routes: Routes = [

  {path: "",component:SigninComponent},

{path: "dashboard",component:EmployeeDashboardComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

app.component.html

<router-outlet></router-outlet>

app.module.ts

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

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

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

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

import { EmployeeDashboardComponent } from './employee-dashboard/employee-dashboard.component';

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

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

import { SigninComponent } from './signin/signin.component';

@NgModule({

  declarations: [

    AppComponent,

    EmployeeDashboardComponent,

    SigninComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    ReactiveFormsModule,

    HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

employee-dashborad.model.ts

export class EmployeeModel{

    id : number = 0;

    firstName : string = '';

    lastName : string = '';

    email : string = '';

    mobile : string = '';

    salary : string = '';

}

Employee.dashboard.component.html

<nav class="navbar navbar-light bg-primary">

    <div class="container-fluid">

      <h1 style="color: white">Event Management Employee Admin Panel</h1>

      <div class="d-fles">

        <button

        (click)="clickAddEmployee()"

          type="button"

          data-toggle="modal"

          data-target="#exampleModal"

          class="btn btn-success"

        >

          Add Employee

        </button>

      </div>

    </div>

  </nav>

  <table class="table mt-3">

    <thead>

      <tr>

        <th scope="col">Employee Id</th>

        <th scope="col">First Name</th>

        <th scope="col">Last Name</th>

        <th scope="col">Email Id</th>

        <th scope="col">Mobile No.</th>

        <th scope="col">Salary</th>

        <th scope="col">Action</th>

      </tr>

    </thead>

    <tbody>

      <tr \*ngFor="let temp of employeeData">

        <td>{{ temp.id }}</td>

        <td>{{ temp.firstName }}</td>

        <td>{{ temp.lastName }}</td>

        <td>{{ temp.email }}</td>

        <td>{{ temp.mobile }}</td>

        <td>{{ temp.salary }}</td>

        <td>

          <button

            type="button"

            data-toggle="modal"

            data-target="#exampleModal"

            class="btn btn-info"

            (click)="onEdit(temp)"

          >

            Edit

          </button>

          <button (click)="deleteEmployee(temp)" class="btn btn-danger mx-3">

            Delete

          </button>

        </td>

      </tr>

    </tbody>

  </table>

  <!-- Button trigger modal -->

  <!-- Modal -->

  <div

    class="modal fade"

    id="exampleModal"

    tabindex="-1"

    role="dialog"

    aria-labelledby="exampleModalLabel"

    aria-hidden="true"

  >

    <div class="modal-dialog" role="document">

      <div class="modal-content">

        <div class="modal-header">

          <h5 class="modal-title" id="exampleModalLabel">Employee Details</h5>

          <button

            type="button"

            class="close"

            data-dismiss="modal"

            aria-label="Close"

          >

            <span aria-hidden="true">&times;</span>

          </button>

        </div>

        <div class="modal-body">

          <form [formGroup]="formValue">

            <div class="form-group">

              <label for="exampleInputEmail1">First Name</label>

              <input

                type="text"

                class="form-control"

                id="exampleInputEmail1"

                aria-describedby="emailHelp"

                formControlName="firstName"

              />

            </div>

            <div class="form-group">

              <label for="exampleInputPassword1">Last Name</label>

              <input

                type="text"

                class="form-control"

                id="exampleInputPassword1"

                formControlName="lastName"

              />

            </div>

            <div class="form-group">

              <label for="exampleInputPassword1">Email</label>

              <input

                type="email"

                class="form-control"

                id="exampleInputPassword1"

                formControlName="email"

              />

            </div>

            <div class="form-group">

              <label for="exampleInputPassword1">Mobile No.</label>

              <input

                type="text"

                class="form-control"

                id="exampleInputPassword1"

                formControlName="mobile"

              />

            </div>

            <div class="form-group">

              <label for="exampleInputPassword1">Salary</label>

              <input

                type="text"

                class="form-control"

                id="exampleInputPassword1"

                formControlName="salary"

              />

            </div>

          </form>

        </div>

        <div class="modal-footer">

          <button

            type="button"

            id="cancel"

            class="btn btn-secondary"

            data-dismiss="modal"

          >

            Close

          </button>

          <button

            type="button"

            (click)="postEmployeeDetails()"

            class="btn btn-primary"

            \*ngIf="showAdd"

          >

            Add

          </button>

          <button

            type="button"

            (click)="updateEmployee()"

            class="btn btn-primary"

            \*ngIf="showUpdate"

          >

            Update

          </button>

        </div>

      </div>

    </div>

  </div>

Employee-dashboard.componenet.ts

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

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

import { ApiService } from '../shared/api.service';

import { EmployeeModel } from './employee-dash board.model';

@Component({

  selector: 'app-employee-dashboard',

  templateUrl: './employee-dashboard.component.html',

  styleUrls: ['./employee-dashboard.component.css']

})

export class EmployeeDashboardComponent implements OnInit {

  formValue !: FormGroup;

  employeeModelObj : EmployeeModel = new EmployeeModel();

  employeeData : any;

  showAdd !: boolean;

  showUpdate !: boolean;

  constructor(private formbuiler: FormBuilder, private api : ApiService) { }

  ngOnInit(): void {

    this.formValue = this.formbuiler.group({

      firstName : [''],

      lastName : [''],

      email : [''],

      mobile : [''],

      salary : ['']

    })

    this.getAllEmployee();

  }

  clickAddEmployee()

  {

    this.formValue.reset();

    this.showAdd= true;

    this.showUpdate= false;

  }

  postEmployeeDetails()

  {

    this.employeeModelObj.firstName = this.formValue.value.firstName;

    this.employeeModelObj.lastName = this.formValue.value.lastName;

    this.employeeModelObj.email = this.formValue.value.email;

    this.employeeModelObj.mobile = this.formValue.value.mobile;

    this.employeeModelObj.salary = this.formValue.value.salary;

    this.api.postEmployee(this.employeeModelObj)

    .subscribe(res=>{

      console.log(res);

      alert("employee Added")

      let ref = document.getElementById('cancel')

      ref?.click();

      this.formValue.reset();

      this.getAllEmployee();

    },

    err=>{

      alert("Something went wrong")

    })

  }

  getAllEmployee(){

    this.api.getEmployee()

    .subscribe(res=>{

      this.employeeData = res;

    })

  }

  deleteEmployee(temp : any){

    this.api.deleteEmployee(temp.id)

    .subscribe(res=>{

      alert("employee Deleted");

      this.getAllEmployee();

    })

  }

  onEdit(temp : any)

  {

    this.showAdd= false;

    this.showUpdate= true;

    this.employeeModelObj.id = temp.id;

    this.formValue.controls['firstName'].setValue(temp.firstName);

    this.formValue.controls['lastName'].setValue(temp.lastName);

    this.formValue.controls['email'].setValue(temp.email);

    this.formValue.controls['mobile'].setValue(temp.mobile);

    this.formValue.controls['salary'].setValue(temp.salary);

  }

  updateEmployee()

  {

    this.employeeModelObj.firstName = this.formValue.value.firstName;

    this.employeeModelObj.lastName = this.formValue.value.lastName;

    this.employeeModelObj.email = this.formValue.value.email;

    this.employeeModelObj.mobile = this.formValue.value.mobile;

    this.employeeModelObj.salary = this.formValue.value.salary;

    this.api.updateEmployee(this.employeeModelObj, this.employeeModelObj.id)

    .subscribe(res=>{

      alert("Updated Successfully")

      let ref = document.getElementById('cancel')

      ref?.click();

      this.formValue.reset();

      this.getAllEmployee();

    })

  }

}

Api.servive.ts

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

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

import { map } from 'rxjs/operators'

@Injectable({

  providedIn: 'root'

})

export class ApiService {

  constructor(private http: HttpClient) { }

  postEmployee(data : any){

    return this.http.post<any>("http://localhost:3000/posts", data)

    .pipe(map((res:any)=>{

      return res;

    }))

  }

  getEmployee(){

    return this.http.get<any>("http://localhost:3000/posts")

    .pipe(map((res:any)=>{

      return res;

    }))

  }

  updateEmployee(data : any, id: number){

    return this.http.put<any>("http://localhost:3000/posts/"+id, data)

    .pipe(map((res:any)=>{

      return res;

    }))

  }

  deleteEmployee(id : number){

    return this.http.delete<any>("http://localhost:3000/posts/"+id)

    .pipe(map((res:any)=>{

      return res;

    }))

  }

}

Signin.componenet.html

<div class="pos-f-t">

    <nav class="navbar navbar-dark bg-dark">

      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarToggleExternalContent" aria-controls="navbarToggleExternalContent" aria-expanded="false" aria-label="Toggle navigation">

        <span style="font-size: 35px ;">Amazing Portfolio App</span>

      </button>

    </nav>

  </div>

  <div class="container mt-5 w-50">

    <form [formGroup]="registerForm" (ngSubmit)="OnSubmit()">

      <div class="form-group pt-5">

        <label for="exampleInputEmail1"><i class="fa-solid fa-envelope"></i> Email address</label>

        <input type="email" formControlName="uemail"  class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email">

        <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>

      </div>

      <div class="form-group mt-5">

        <label for="exampleInputPassword1"><i class="fa-sharp fa-solid fa-key"></i> Password </label>

        <input type="password" formControlName="upassword" class="form-control" id="exampleInputPassword1" placeholder="Password">

      </div >

      <div class="text-center">

        <button type="submit" class="btn btn-primary ">Submit</button>

      </div>

        </form>

  </div>

Signin.component.ts

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

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

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

@Component({

  selector: 'app-signin',

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

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

})

export class SigninComponent implements OnInit {

  registerForm!: FormGroup;

  uemail: string = '';

  upassword: string = '';

  submitted: boolean = false;

  constructor(private builder: FormBuilder, private router: Router) { }

  ngOnInit(): void {

    this.registerForm = this.builder.group({

      uemail: [''],

      upassword: [''],

    });

  }

  OnSubmit() {

    this.submitted = true;

    this.uemail = this.registerForm.get('uemail')?.value;

    this.upassword = this.registerForm.get('upassword')?.value;

    if (this.uemail == 'swea@gmail.com' && this.upassword == '123') {

     // alert("Form Submitted Successfully"+this.upassword);

      this.router.navigate(['dashboard']);

    } else {

      alert('wrong credential');

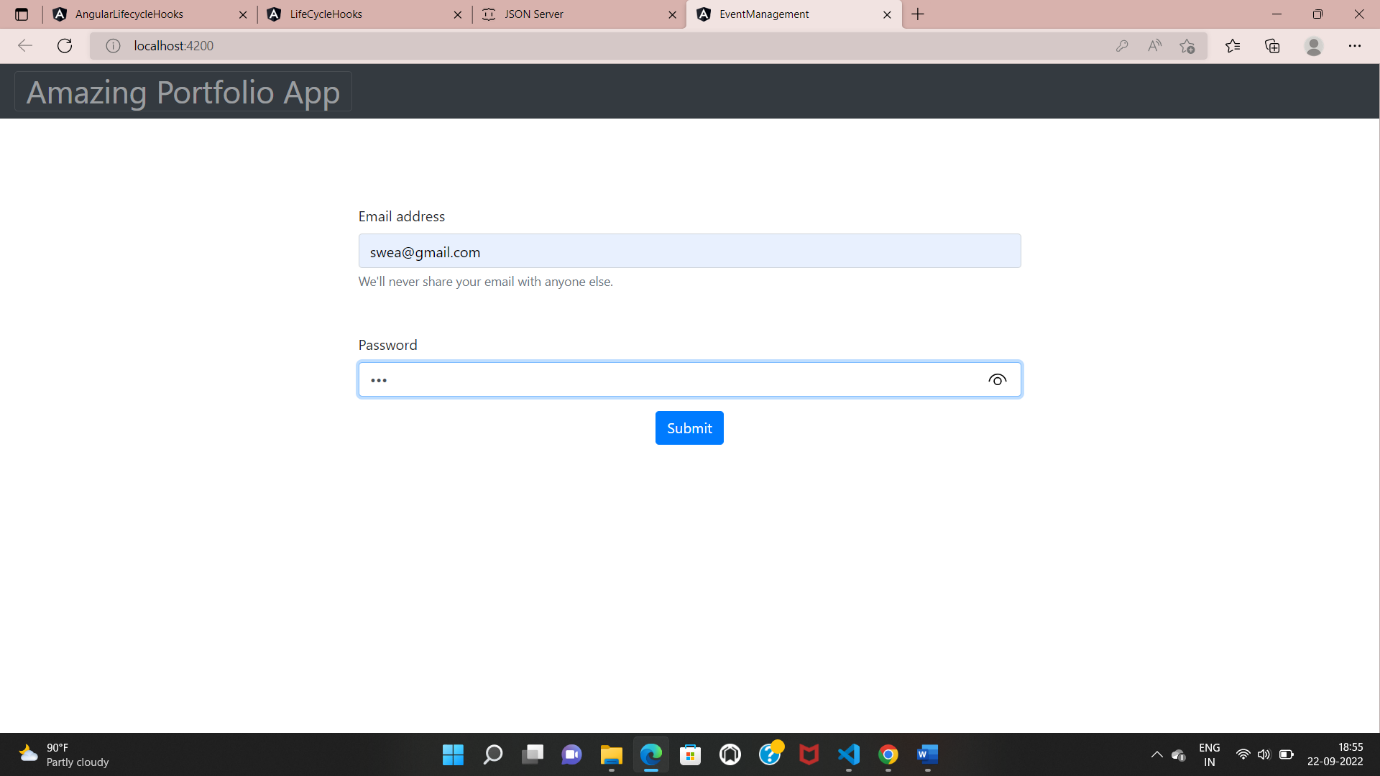
    }

  }

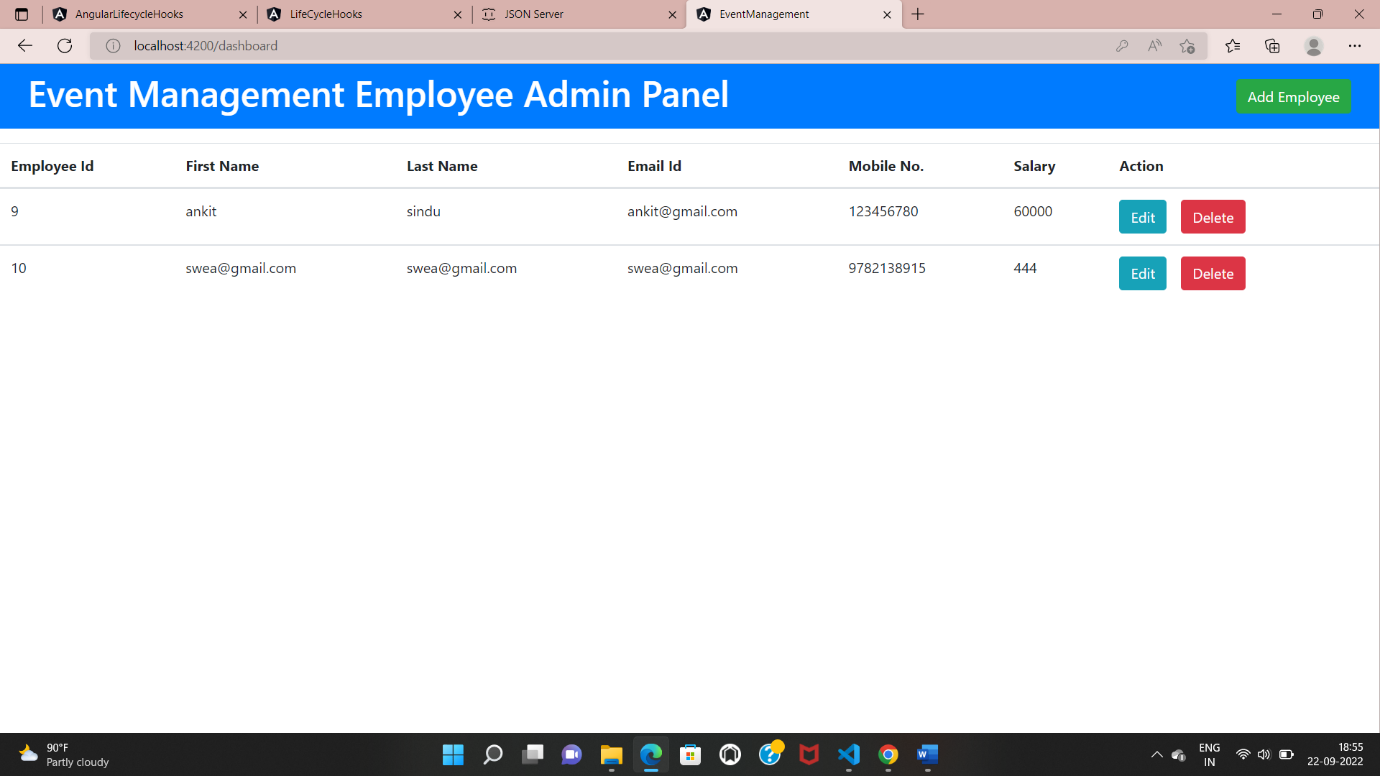
}

Step 7: save all and run as ng serve

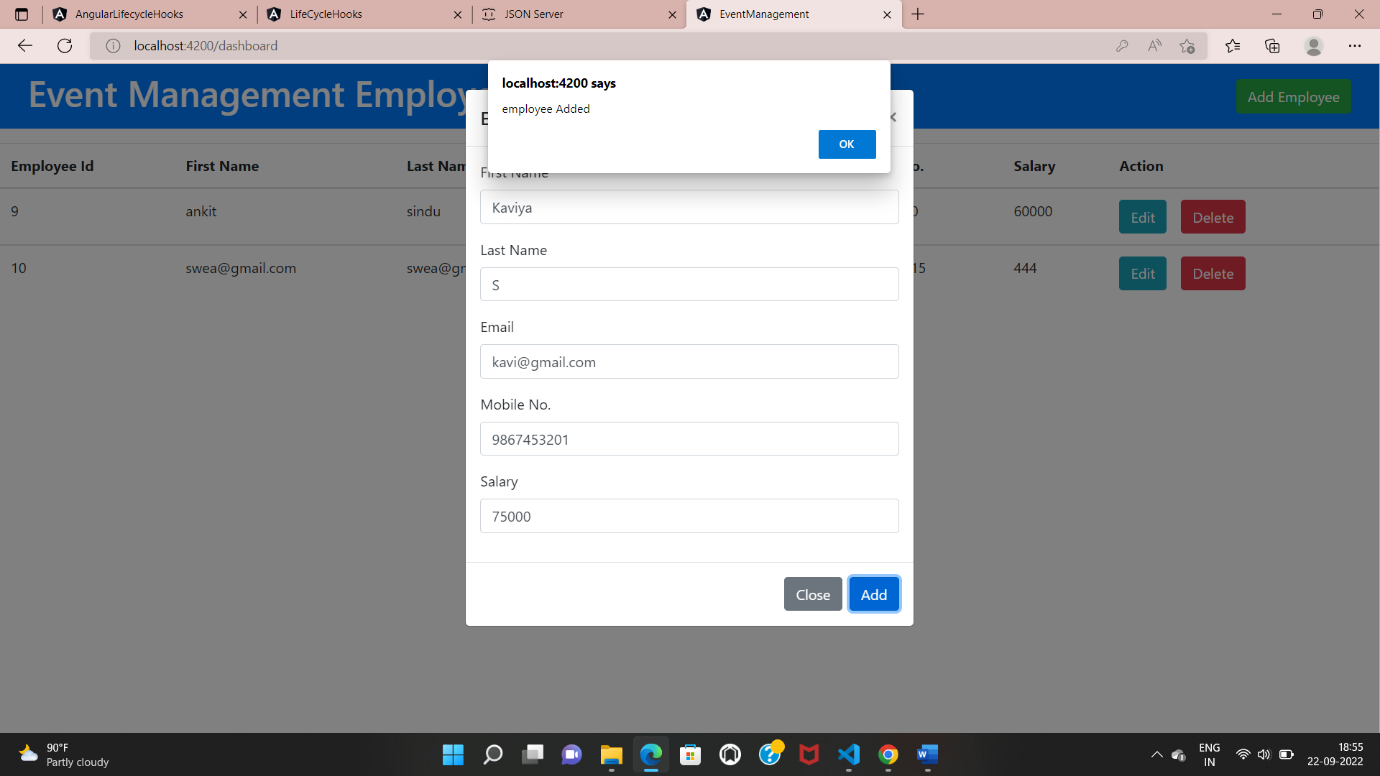
**Sign in page**



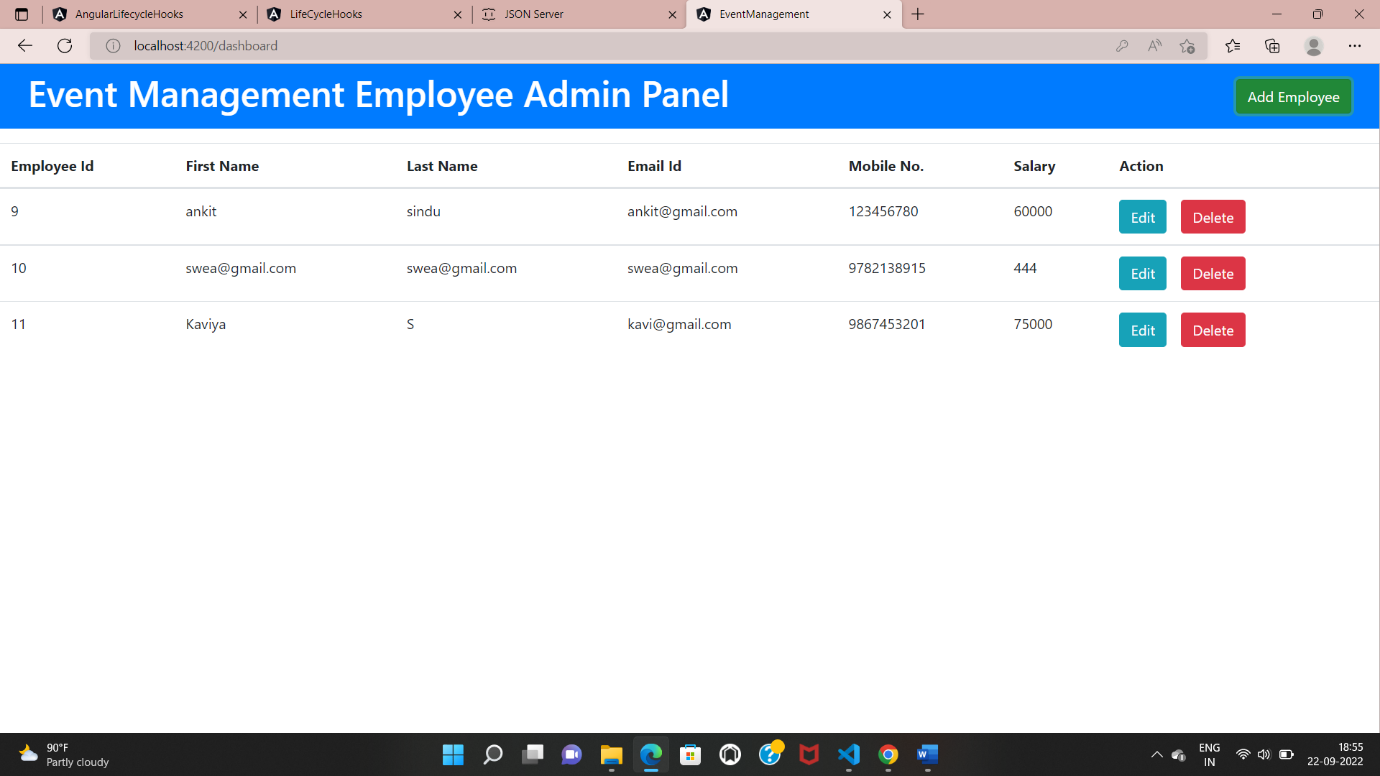
**Admin panel(retrieve)**



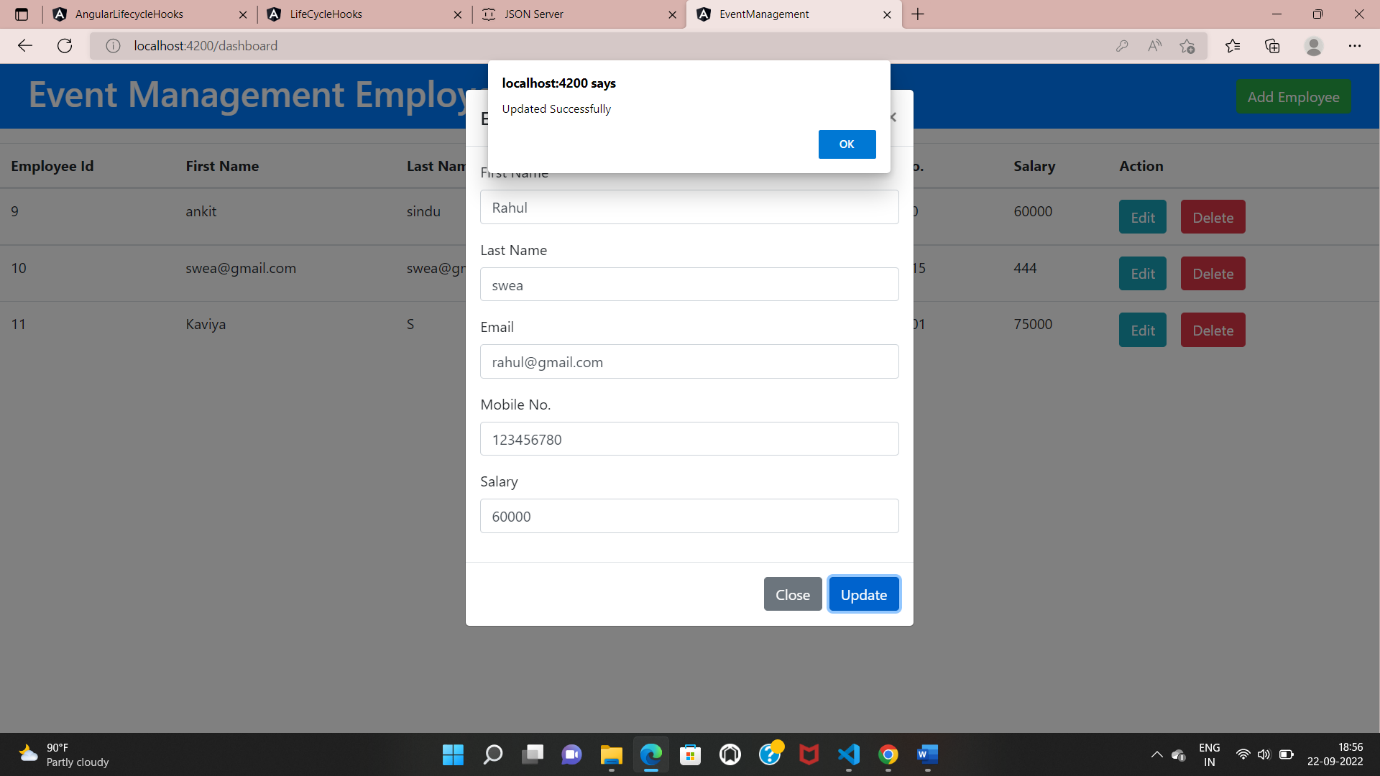
**Adding employee(create)**



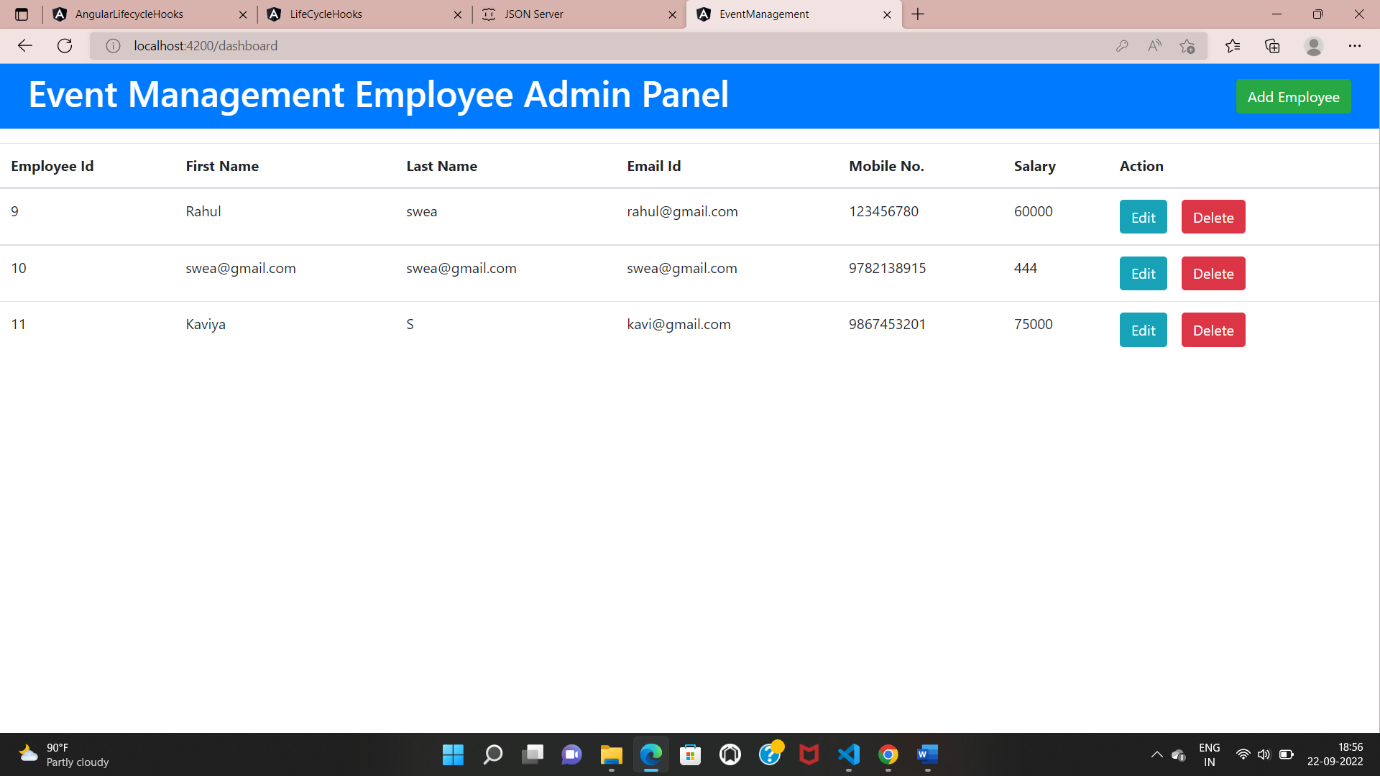
**After added**



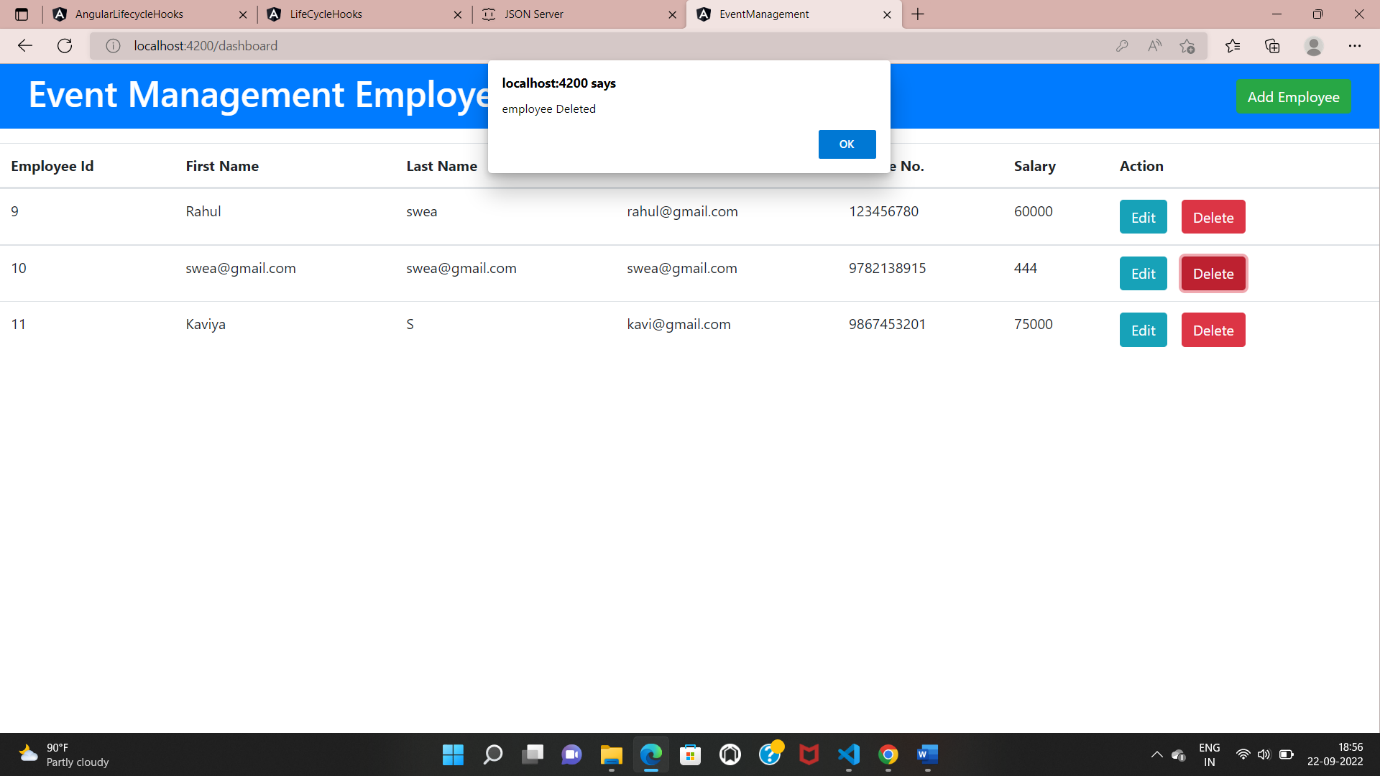
**Editing (update)**



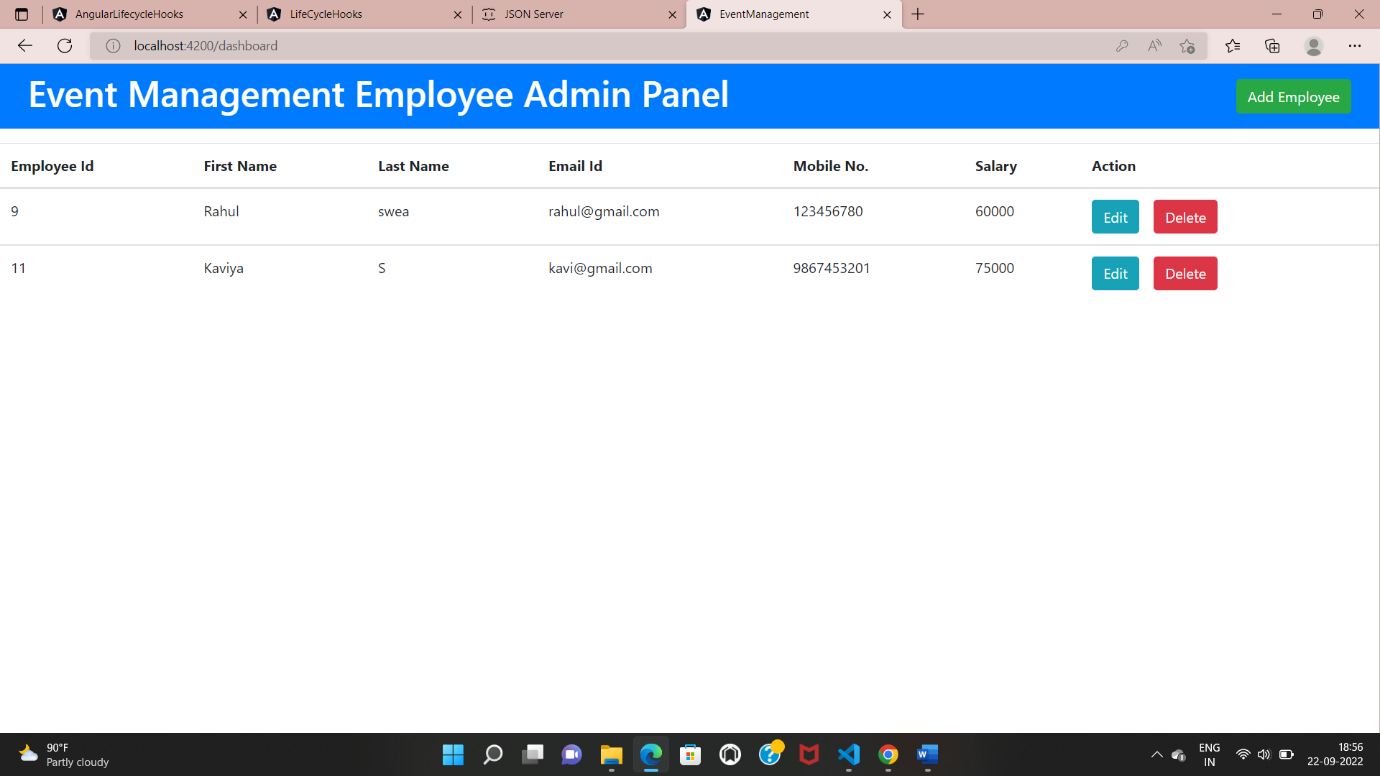
**After update**



**Deleting (delete)**



**After deletion**



Here we have performed the crud operation using angular, bootstrap

CRUD (Create, Retrieve, Update, Delete).