**Event management app**

**Code:**

**Add-emp.component.html**

<div class="col-md-6">

    <h3 class="text-center" style=" text-decoration: underline;color:blue;text-shadow: 0 0 3px #00e1ff, 0 0 5px #e6e6f1;">{{empformlabel}}</h3>

    <form [formGroup]="addForm" novalidate class="form" style="margin: auto; padding-top: 5px; padding-bottom: 10px;width: 400px;">

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="empId">Employee Id:</label>

        <input type="number" formControlName="id" placeholder="Id" name="empId" class="form-control" id="empId">

      </div>

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="empName">Employee Name:</label>

        <input formControlName="employee\_name" placeholder="Employee Name" name="empName" class="form-control" id="empName">

        <div class="alert  alert-danger" \*ngIf="addForm.get('employee\_name')!.hasError('required') && addForm.get('employee\_name')!.touched">

          Employee Name is required

        </div>

      </div>

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="empEmail">Email ID:</label>

        <input formControlName="employee\_email" placeholder="Employee Email ID" name="empEmail" class="form-control" id="empEmail">

        <div class="alert  alert-danger" \*ngIf="addForm.get('employee\_email')!.hasError('required') && addForm.get('employee\_email')!.touched">

          Employee Email ID is required

        </div>

      </div>

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="empSalary">Employee Salary:</label>

        <input formControlName="employee\_salary" placeholder="Employee Salary" name="employee\_salary" class="form-control" id="employee\_salary">

        <div class="alert  alert-danger" \*ngIf="addForm.get('employee\_salary')!.hasError('maxlength') && addForm.get('employee\_salary')!.touched">

          Employee Salary is required and should less than 9 characters.

        </div>

      </div>

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="empAge">Employee Age:</label>

        <input formControlName="employee\_age" placeholder="Employee Age" name="empAge" class="form-control" id="empAge">

        <div class="alert  alert-danger" \*ngIf=" addForm.get('employee\_age')!.hasError('maxlength') && addForm.get('employee\_age')!.touched">

          Age is required and should less than 3 characters.

        </div>

      </div >

      <button class="btn btn-success" [disabled]='addForm.invalid' \*ngIf="btnvisibility" (click)="onSubmit()">Save</button>

      <button class="btn btn-success" [disabled]='addForm.invalid' \*ngIf="!btnvisibility" (click)="onUpdate()">Update</button>

      <p style=" padding: 12px 20px;">Form value: {{ addForm.value | json }}</p>

    </form>

  </div>

**Add-emp.component.ts**

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

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

import { EmployeeService } from '../employee.service';

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

@Component({

  selector: 'app-add-emp',

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

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

})

export class AddEmpComponent implements OnInit {

  empformlabel = 'Add Employee Form';

  empformbtn = 'Save';

  constructor(private formBuilder: FormBuilder, private router: Router, private empService: EmployeeService) {}

  editForm:FormGroup;

  addForm: FormGroup;

  btnvisibility = true;

  ngOnInit() {

    this.addForm = this.formBuilder.group({

      id: [],

      employee\_name: ['', Validators.required],

      employee\_email: ['', Validators.required],

      employee\_salary: ['', [Validators.required, Validators.maxLength(9)]],

      employee\_age: ['', [Validators.required, Validators.maxLength(3)]]

    });

  }

  onSubmit() {

    console.log('Create fire');

    this.empService.createUser(this.addForm.value)

      .subscribe(data => {

        this.router.navigate(['list-emp']);

      },

      error => {

        alert(error);

      });

  }

  onUpdate() {

    console.log('Update fire');

    this.empService.updateEmployee(this.editForm.value).subscribe(data => {

      this.router.navigate(['list-emp']);

    },

      error => {

        alert(error);

      });

  }

  }

**Admin.component.html**

<!DOCTYPE html>

<html>

<head>

  <meta charset="utf-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <title>Page Title</title>

</head>

<body style="padding:40px ">

<div >

<div class="jumbotron">

  <div class="container">

      <div class="row">

          <div class="col-md-6 offset-md-3">

              <h2  style=" text-decoration: underline;color:blue;text-shadow: 0 0 3px #00e1ff, 0 0 5px #e6e6f1;">Admin Login Form</h2>

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

                  <div class="form-group">

                  <div class="form-group">

                      <label>Email</label>

                      <input type="text" formControlName="email" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.email.errors }" />

                      <div \*ngIf="submitted && f.email.errors" class="invalid-feedback">

                          <div \*ngIf="f.email.errors.required">Email is required</div>

                          <div \*ngIf="f.email.errors.email">Email must be a valid email address</div>

                      </div>

                  </div>

                  <div class="form-group">

                      <label>Password</label>

                      <input type="password" formControlName="password" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.password.errors }" />

                      <div \*ngIf="submitted && f.password.errors" class="invalid-feedback">

                          <div \*ngIf="f.password.errors.required">Password is required</div>

                          <div \*ngIf="f.password.errors.minlength">Password must be at least 6 characters</div>

                      </div>

                  </div>

                  <div class="form-group">

                      <button  class="btn btn-primary"  >Login</button>

                  </div>

                  </div>

              </form>

          </div>

      </div>

  </div>

</div>

</div>

</body>

</html>

**Admin.component.ts**

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

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

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

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

@Component({

  selector: 'app-admin',

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

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

})

export class AdminComponent implements OnInit {

  routerLinkActive="active";

  public loginform:FormData;

  constructor(private formBuilder: FormBuilder,private http:HttpClient,private route:Router) { }

  title = 'admin';

  email = '';

  password = '';

  submitted = false;

 registerForm: FormGroup=new FormGroup({});

  ngOnInit() {

    this.registerForm = this.formBuilder.group({

        email: ['', [Validators.required, Validators.email]],

        password: ['', [Validators.required, Validators.minLength(6)]]

    });

  }

  get f() { return this.registerForm.controls; }

  baseUrl = 'http://localhost:3000/admin/';

  onSubmit() {

    this.http.get<any>(this.baseUrl)

    .subscribe(res=>{

      const u=res.find((a:any)=>{

        return a.email===this.registerForm.value.email && a.password===this.registerForm.value.password

      })

      if(u){

        alert("Login success");

        this.registerForm.reset();

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

      }else{

        alert("Invalid Credentials")

      }

    })

  }

}

**Admin-edit.component.ts**

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

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

import { EmployeeService } from '../employee.service';

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

@Component({

  selector: 'app-admin-edit',

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

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

})

export class AdminEditComponent implements OnInit {

  empformlabel = 'Edit Admin';

  empformbtn = 'Update';

  constructor(private formBuilder: FormBuilder, private router: Router, private empService: EmployeeService) {

  }

  addForm: FormGroup;

  editForm: FormGroup;

  ngOnInit() {

    this.editForm = this.formBuilder.group({

      email: ['', Validators.required],

      password: ['', [Validators.required, Validators.maxLength(14)]]

    });

    const empid = Number(localStorage.getItem('editEmpId'));

    if (+empid > 0) {

      this.empService.getadminById(+empid).subscribe(data => {

        this.editForm.patchValue(data);

      });

    }

  }

  onUpdate() {

    console.log('Update fire');

    this.empService.updateadmin(this.editForm.value).subscribe(data => {

      this.router.navigate(['']);

    },

      error => {

        alert(error);

      });

  }

}

**Admin-edit.component.html**

<div class="col-md-6">

    <h2 class="text-center" style=" text-decoration: underline;color:blue;text-shadow: 0 0 3px #00e1ff, 0 0 5px #e6e6f1;">{{empformlabel}}</h2>

    <form [formGroup]="editForm" novalidate class="form" style="margin: auto; padding-top: 5px; padding-bottom: 10px;width: 400px;">

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="email">Admin Email:</label>

        <input formControlName="email" placeholder="Email" name="email" class="form-control" id="email">

        <div class="alert  alert-danger" \*ngIf="editForm.get('email')!.hasError('required') ">

          Email is required

        </div>

      </div>

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="password">Password:</label>

        <input formControlName="password" placeholder="Password" name="password" class="form-control" id="password">

        <div class="alert  alert-danger" \*ngIf="editForm.get('password')!.hasError('required') ">

          Password is required

        </div>

      </div>

       <button class="btn btn-success" [disabled]='editForm.invalid'  (click)="onUpdate()">Update</button>

      <p>Form value: {{ editForm.value | json }}</p>

    </form>

  </div>

**Dashboard.component.ts**

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

@Component({

  selector: 'app-dashboard',

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

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

})

export class DashboardComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

**Dashboard.component.hml**

<app-home></app-home>

<h3>Welcome to Event Management App !!</h3>

<img src="https://media.istockphoto.com/vectors/vector-illustration-icon-with-the-employee-management-system-concept-vector-id1084494738?k=20&m=1084494738&s=170667a&w=0&h=RXqjXE5bvKZXkL7uHxfafaV8oxbCfNGcw-xP\_5GupiI=" style="width:500px ;">

**Edit-emp.component.html**

<div class="col-md-6">

    <h2 class="text-center" style=" text-decoration: underline;color:blue;text-shadow: 0 0 3px #00e1ff, 0 0 5px #e6e6f1;">{{empformlabel}}</h2>

    <form [formGroup]="editForm" novalidate class="form" style="margin: auto; padding-top: 5px; padding-bottom: 10px;width: 400px;">

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="empId">Employee Id:</label>

        <input type="number" formControlName="id" placeholder="Id" name="empId" class="form-control" id="empId">

      </div>

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="empName">Employee Name:</label>

        <input formControlName="employee\_name" placeholder="Employee Name" name="empName" class="form-control" id="empName">

        <div class="alert  alert-danger" \*ngIf="editForm.get('employee\_name')!.hasError('required') && addForm.get('employee\_name')!.touched">

          Employee Name is required

        </div>

      </div>

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="empEmail">Email ID:</label>

        <input formControlName="employee\_email" placeholder="Employee Email ID" name="empEmail" class="form-control" id="empEmail">

        <div class="alert  alert-danger" \*ngIf="editForm.get('employee\_email')!.hasError('required') && addForm.get('employee\_email')!.touched">

          Employee Email ID is required

        </div>

      </div>

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="empSalary">Employee Salary:</label>

        <input formControlName="employee\_salary" placeholder="Employee Salary" name="employee\_salary" class="form-control" id="employee\_salary">

        <div class="alert  alert-danger" \*ngIf="editForm.get('employee\_salary')!.hasError('maxlength') && addForm.get('employee\_salary')!.touched">

          Employee Salary is required and should less than 9 characters.

        </div>

      </div>

      <div class="form-group" style=" padding: 12px 20px;">

        <label for="empAge">Employee Age:</label>

        <input formControlName="employee\_age" placeholder="Employee Age" name="empAge" class="form-control" id="empAge">

        <div class="alert  alert-danger" \*ngIf=" editForm!.get('employee\_age')!.hasError('maxlength') && addForm.get('employee\_age')!.touched">

          Age is required and should less than 3 characters.

        </div>

      </div>

       <button class="btn btn-success" [disabled]='editForm.invalid'  (click)="onUpdate()">Update</button>

      <p>Form value: {{ editForm.value | json }}</p>

    </form>

  </div>

**Edit-emp.component.ts**

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

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

import { EmployeeService } from '../employee.service';

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

@Component({

  selector: 'app-edit-emp',

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

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

})

export class EditEmpComponent implements OnInit {

  empformlabel = 'Edit Employee';

  empformbtn = 'Update';

  constructor(private formBuilder: FormBuilder, private router: Router, private empService: EmployeeService) {

  }

  addForm: FormGroup;

  editForm: FormGroup;

  ngOnInit() {

    this.editForm = this.formBuilder.group({

      id: [],

      employee\_name: ['', Validators.required],

      employee\_email: ['', Validators.required],

      employee\_salary: ['', [Validators.required, Validators.maxLength(9)]],

      employee\_age: ['', [Validators.required, Validators.maxLength(3)]]

    });

    const empid = Number(localStorage.getItem('editEmpId'));

    if (+empid > 0) {

      this.empService.getEmployeeById(+empid).subscribe(data => {

        this.editForm.patchValue(data);

      });

    }

  }

  onUpdate() {

    console.log('Update fire');

    this.empService.updateEmployee(this.editForm.value).subscribe(data => {

      this.router.navigate(['list-emp']);

    },

      error => {

        alert(error);

      });

  }

}

**Emp-detail.component.ts**

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

import { EmployeeService } from '../employee.service';

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

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

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

@Component({

  selector: 'app-emp-detail',

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

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

})

export class EmpDetailComponent implements OnInit {

  employeeID: number;

  employees: Employee[];

  employeess:Employee;

  constructor(private empService: EmployeeService,private formBuilder: FormBuilder, private router: Router, ) {

   }

  editForm: FormGroup;

  ngOnInit() {

      this.getEmployeeByID();

  }

   empid = Number(localStorage.getItem('editEmpId'));

  getEmployeeByID() {

    this.empService.getEmployeeById(this.empid)

    .subscribe((data: Employee) => {

      this.employeess = data;

    });

  }

  deleteEmp(employee: Employee): void {

    this.empService.deleteEmployees(employee.id!)

      .subscribe(data => {

        this.employees = this.employees.filter(u => u !== employee);

      });

  }

  editEmp(employee: Employee): void {

    localStorage.removeItem('editEmpId');

    localStorage.setItem('editEmpId', employee.id!.toString());

    this.router.navigate(['update-emp']);

  }

  addEmp(): void {

    localStorage.removeItem('editEmpId');

    this.router.navigate(['add-emp']);

  }

}

**Emp-detail.component.html**

<app-home></app-home>

<br>

<br>

<div style="border: 2px solid black;background-color: rgb(197, 163, 230);width:600px; padding-top: 5px; padding-bottom: 10px; margin-left: auto;margin-right: auto;padding: 10px;">

  <p>Detail of </p><h3>{{employeess.employee\_name}}</h3>

</div>

    <table class="table table-bordered " style="border: 2px solid black;width:600px; padding-top: 10px; padding-bottom: 20px; margin-left: auto;margin-right: auto;padding: 20px;" \*ngIf="employeess">

      <tbody>

        <tr>

          <td>Id</td>

          <td>{{employeess.id}}</td>

        </tr>

        <tr>

          <td>name</td>

          <td>{{employeess.employee\_name}}</td>

        </tr>

        <tr>

          <td>Email ID</td>

          <td>{{employeess.employee\_email}}</td>

        </tr>

        <tr>

          <td>Age</td>

          <td>{{employeess.employee\_age}}</td>

        </tr>

        <tr>

          <td>Salary</td>

          <td>{{employeess.employee\_salary}}</td>

        </tr>

        <tr  style=" padding: 20px;padding-top: 10px;

        padding-bottom: 20px;">

          <button (click)="deleteEmp(employeess)" routerLink="/list-emp" class="btn btn-danger"style="border: 2px solid black;border-spacing: 30px;background-color:red"> Delete</button>

        <button (click)="editEmp(employeess)" style="margin-left: 20px;border: 2px solid black;background-color:rgb(161, 153, 153)" class="btn btn-secondary"> Edit</button>

        <button routerLink="/list-emp" style="margin-left: 20px;border: 2px solid black;background-color:rgb(0, 247, 255)" class="btn btn-primary"> Employees List</button>

      </tr>

      </tbody>

    </table>

**Home.component.ts**

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

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

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

import { EmployeeService } from '../employee.service';

@Component({

  selector: 'app-home',

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

  styleUrls: ['./home.component.css']

})

export class HomeComponent implements OnInit {

  constructor(private empService: EmployeeService, private router: Router) { }

  ngOnInit(): void {

  }

  editadminn(admin: admin): void {

    localStorage.removeItem('editEmpId');

    localStorage.setItem('editEmpId', admin.id!.toString());

    this.router.navigate(['admin']);

  }

}

**Home.component.html**

<ul class="nav nav-pills  navbar-dark bg-dark " routerLinkActive="active">

    <li class="nav-item">

      <a class="nav-link active" aria-current="page" routerLink="/dash">Home</a>

    </li>

    <li class="nav-item">

      <a class="nav-link" routerLink="/list-emp">Employees</a>

    </li>

    <li class="nav-item">

      <a class="nav-link" routerLink="/admin">Settings</a>

    </li>

    <li class="nav-item">

      <a class="nav-link" href="#">Logout</a>

    </li>

  </ul>

**List-emp.component.ts**

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

import { EmployeeService } from '../employee.service';

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

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

@Component({

  selector: 'app-list-emp',

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

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

})

export class ListEmpComponent implements OnInit {

  employees: Employee[];

  constructor(private empService: EmployeeService, private router: Router, ) { }

  ngOnInit() {

    this.empService.getEmployees()

      .subscribe((data: Employee[]) => {

        this.employees = data;

      });

  }

  detailEmp(employee: Employee): void {

    localStorage.removeItem('editEmpId');

    localStorage.setItem('editEmpId', employee.id!.toString());

    this.router.navigate(['/detail']);

  }

  addEmp(): void {

    localStorage.removeItem('editEmpId');

    this.router.navigate(['add-emp']);

  }

}

**List-emp.component.html**

<app-home></app-home>

<br><br>

<div class="col-md-12">

    <button (click)="addEmp()" class="btn btn-secondary"> Add Employee</button>

    <br><br>

  <h3 style="color:rgb(52, 164, 172)"> Employees List</h3>

  <div class="table-responsive table-container">

    <table class="table" style="width: 1200px; margin-left: auto;margin-right: auto;">

      <thead>

        <tr>

          <th>Id</th>

          <th>Employee Name</th>

          <th>Email ID</th>

        </tr>

      </thead>

      <tbody>

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

          <td class="hidden">{{emp.id}}</td>

          <td>{{emp.employee\_name}}</td>

          <td>{{emp.employee\_email}}</td>

          <td>

            <button (click)="detailEmp(emp)" class="btn btn-info"> View Detail</button>

          </td>

        </tr>

      </tbody>

    </table>

  </div>

</div>

**Employee.services.ts**

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

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

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

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

@Injectable({

  providedIn: 'root'

})

export class EmployeeService {

  constructor(private http: HttpClient) { }

  baseUrl = 'http://localhost:3000/employees/';

adminurl='http://localhost:3000/admin/';

  getEmployees() {

    return this.http.get<Employee[]>(this.baseUrl);

  }

  deleteEmployees(id: number) {

    return this.http.delete<Employee[]>(this.baseUrl + id);

  }

  createUser(employee: Employee) {

    return this.http.post(this.baseUrl, employee);

  }

  getEmployeeById(id: number) {

    return this.http.get<Employee>(this.baseUrl  + id);

  }

  updateEmployee(employee: Employee) {

    return this.http.put(this.baseUrl  + employee.id, employee);

  }

  updateadmin(admin: admin) {

    return this.http.put(this.adminurl  + 1, admin);

  }

  getadminById(id:number) {

    return this.http.get<admin>(this.adminurl  + 1);

  }

}

**Employee.model.ts**

export class Employee {

        id?: number;

        employee\_email?:string;

        employee\_name?: string;

        employee\_salary?: number;

        employee\_age?: number;

    }

    export class admin {

        id=1;

        email?: string;

        password?: string;

    }

**App.module.ts**

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

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

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

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

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

import { AddEmpComponent } from './add-emp/add-emp.component';

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

import { ListEmpComponent } from './list-emp/list-emp.component';

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

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

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

import { EmpDetailComponent } from './emp-detail/emp-detail.component';

import { AdminEditComponent } from './admin-edit/admin-edit.component';

import { DashboardComponent } from './dashboard/dashboard.component';

@NgModule({

  declarations: [

    AppComponent,

    AddEmpComponent,

    EditEmpComponent,

    ListEmpComponent,

    HomeComponent,

    AdminComponent,

    EmpDetailComponent,

    AdminEditComponent,

    DashboardComponent,

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    ReactiveFormsModule,

    HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

**app.component.ts**

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'EventManagementApp';

}

**App.component.html**

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center;margin: auto;text-shadow: 3px 4px 7px rgba(81,67,21,0.8);" >

  <h2>

   {{ title }}

    <br>

  </h2>

  </div>

<router-outlet></router-outlet>

**App-routing.module.ts**

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

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

import { ListEmpComponent } from './list-emp/list-emp.component';

import { AddEmpComponent } from './add-emp/add-emp.component';

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

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

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

import { EmpDetailComponent } from './emp-detail/emp-detail.component';

import { AdminEditComponent } from './admin-edit/admin-edit.component';

import { DashboardComponent } from './dashboard/dashboard.component';

const routes: Routes = [

  { path: '', component:AdminComponent  },

  { path: 'detail', component:EmpDetailComponent  },

  { path: 'dash', component:DashboardComponent },

  { path: 'admin', component:AdminEditComponent  },

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

  { path: 'list-emp', component: ListEmpComponent },

  { path: 'add-emp', component: AddEmpComponent },

  { path: 'update-emp', component: EditEmpComponent }

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule],

  declarations: []

})

export class AppRoutingModule { }

**db.json**

{

  "employees": [

    {

      "id": 11,

      "employee\_name": "mahalakshmi",

      "employee\_email": "rani@gmail.com",

      "employee\_salary": 15000,

      "employee\_age": "30"

    },

    {

      "id": 2,

      "employee\_name": "arun",

      "employee\_email": "arun@gmail.com",

      "employee\_salary": "60000",

      "employee\_age": "27"

    },

    {

      "id": 3,

      "employee\_name": "sowmya",

      "employee\_email": "sowmya@gmail.com",

      "employee\_salary": "200000",

      "employee\_age": "35"

    },

    {

      "id": 4,

      "employee\_name": "monisha",

      "employee\_email": "monisha@gmail.com",

      "employee\_salary": "40000",

      "employee\_age": "22"

    },

    {

      "id": 20,

      "employee\_name": "shyam",

      "employee\_email": "shyam@gmail.com",

      "employee\_salary": "34000",

      "employee\_age": "28"

    }

  ],

  "admin": [

    {

      "email": "saranya@gmail.com",

      "password": "saranya611",

      "id": 1

    }

  ]

}