**Employee.component.html**

<div class="jumbotron mt-4 p-5 text-white base rounded">

  <div class="container my-5">

    <nav class="navbar navbar-expand-sm">

      <div class="container-fluid">

        <h3>Employee Information</h3>

        <div class="d-flex">

          <button

            type="button"

            class="btn btn-primary"

            data-bs-toggle="modal"

            data-bs-target="#exampleModal"

            (click)="clickAdd()"

          >

            Add Employee

          </button>

        </div>

      </div>

    </nav>

  </div>

  <div class="container my-5">

    <table class="table base1 m-2 table-hover">

      <thead style="background-color: aquamarine">

        <tr>

          <th>Id</th>

          <th>Firstname</th>

          <th>Lastname</th>

          <th>Email</th>

          <th>Action</th>

        </tr>

      </thead>

      <tbody>

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

          <td>{{ v.id }}</td>

          <td>{{ v.fname }}</td>

          <td>{{ v.lname }}</td>

          <td>{{ v.email }}</td>

          <td>

            <button

              type="button"

              class="btn btn-info"

              data-bs-toggle="modal"

              data-bs-target="#exampleModal"

              (click)="editButton(v)"

            >

              Edit

            </button>

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

              Delete

            </button>

          </td>

        </tr>

      </tbody>

    </table>

  </div>

  <!-- Modal -->

  <div

    class="modal fade"

    id="exampleModal"

    tabindex="-1"

    aria-labelledby="exampleModalLabel"

    aria-hidden="true"

  >

    <div class="modal-dialog" style="color: black">

      <div class="modal-content">

        <div class="modal-header">

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

          <button

            type="button"

            class="btn-close"

            data-bs-dismiss="modal"

            aria-label="Close"

          ></button>

        </div>

        <div class="modal-body">

          <form [formGroup]="formValue">

            <div class="mb-3 mt-3">

              <label for="email" class="form-label">First Name:</label>

              <input

                type="text"

                class="form-control"

                id="fname"

                placeholder="Enter Firstname"

                formControlName="fname"

              />

            </div>

            <div class="mb-3 mt-3">

              <label for="email" class="form-label">Last Name:</label>

              <input

                type="text"

                class="form-control"

                id="lname"

                placeholder="Enter Last name"

                formControlName="lname"

              />

            </div>

            <div class="mb-3 mt-3">

              <label for="email" class="form-label">Email:</label>

              <input

                type="email"

                class="form-control"

                id="email"

                placeholder="Enter email"

                formControlName="email"

              />

            </div>

          </form>

        </div>

        <div class="modal-footer">

          <button

            type="button"

            id="cancel"

            class="btn btn-secondary"

            data-bs-dismiss="modal"

          >

            Close

          </button>

          <button

            type="button"

            \*ngIf="showAdd"

            (click)="postEmployeeDetails()"

            class="btn btn-primary"

          >

            Save

          </button>

          <button

            type="button"

            \*ngIf="showUpdate"

            (click)="updateEmployeeDetails()"

            class="btn btn-primary"

          >

            Update

          </button>

        </div>

      </div>

    </div>

  </div>

</div>

**Employee.comonent.spec.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { EmployeeComponent } from './employee.component';

describe('EmployeeComponent', () => {

  let component: EmployeeComponent;

  let fixture: ComponentFixture<EmployeeComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ EmployeeComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(EmployeeComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Employee.component.ts**

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

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

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

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

@Component({

  selector: 'app-employee',

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

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

})

export class EmployeeComponent implements OnInit {

  formValue !: FormGroup;

  employeeModelobj: EmployeeModel=new EmployeeModel();

  employeeData !: any ;

  showAdd !:boolean;

  showUpdate !:boolean;

  constructor( private formbuilber:FormBuilder,

    private api: ApiService) { }

  ngOnInit(): void {

    this.getAllEmployee();

    this.formValue=this.formbuilber.group({

      fname:[''],

      lname:[''],

      email:['']

    })

  }

  clickAdd(){

    this.formValue.reset();

    this.showAdd=true;

    this.showUpdate=false;

  }

  postEmployeeDetails(){

    this.employeeModelobj.fname=this.formValue.value.fname;

    this.employeeModelobj.lname=this.formValue.value.lname;

    this.employeeModelobj.email=this.formValue.value.email;

    this.api.postEmployee(this.employeeModelobj)

    .subscribe(res=>{

  alert("Employee Added Succesfully")

  this.formValue.reset();

  this.getAllEmployee();

let refo=document.getElementById("cancel");

refo?.click();

 },

  err=>{

    alert("Something went wrong!!")

  })

  }

  getAllEmployee(){

    this.api.getEmployee()

    .subscribe(res=>{

   this.employeeData=res;

  })

  }

  deleteEmployee(v:any){

    this.api.deleteEmployee(v.id)

    .subscribe(res=>{

   alert("Employee Deleted !!")

   this.getAllEmployee();

  })

  }

  editButton(v:any){

    this.employeeModelobj.id=v.id;

    this.formValue.controls['fname'].setValue(v.fname);

    this.formValue.controls['lname'].setValue(v.lname);

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

    this.showAdd=false;

    this.showUpdate=true;

}

updateEmployeeDetails(){

  this.employeeModelobj.fname=this.formValue.value.fname;

    this.employeeModelobj.lname=this.formValue.value.lname;

    this.employeeModelobj.email=this.formValue.value.email;

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

    .subscribe(res=>{

  alert("Updated Succesfully ");

  this.formValue.reset();

  this.getAllEmployee();

let refo=document.getElementById("cancel");

refo?.click();

    })

}}

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

<div

  class="container my-4"

  style="position: relative; text-align: center; color: white"

>

  <img

    src="..\assets\home2.jpg "

    class="img-fluid mt-4"

    alt=""

    style="height: 550px; width: 1200px"

  />

  <div class="centered">

    <pre>

    <h1><b>WELCOME TO HOMEPAGE</b></h1></pre>

  </div>

</div>

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

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

@Component({

  selector: 'app-home',

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

  styles: [

  ]

})

export class HomeComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

**App.component.html**

<div \*ngIf="!showAdmin">

  <div class="jumbotron" style="margin: 45px">

    <h2 style="text-align: center; padding-top: 5%">Admin</h2>

    <p style="text-align: center">

      Enter the Admin E-mail address and the password!

    </p>

    <form action="">

      <div class="container">

        <p style="text-align: center; padding-bottom: 20px">

          E-mail

          <input

            type="text"

            name="email"

            placeholder="Confirm email address"

            email

            required

            style="padding-right: 60px; margin-left: 35px"

          />

        </p>

      </div>

      <div class="container">

        <p style="text-align: center">

          Password

          <input

            type="text"

            name="email"

            placeholder="Password"

            required

            style="padding-right: 60px; margin-left: 15px"

          />

          <br />

          <button

            class="btn btn-primary"

            style="margin-top: 20px"

            (click)="clickLogin()"

          >

            Login

          </button>

        </p>

      </div>

    </form>

  </div>

</div>

<div class="container" \*ngIf="showHome">

  <div class="row">

    <div class="col">

      <ul class="nav nav-tabs">

        <li

          role="presentation"

          class="nav-item"

          routerLinkActive="active"

          [routerLinkActiveOptions]="{ exact: true }"

        >

          <h5><a class="nav-link" routerLink="/">Home</a></h5>

        </li>

        <li>

          <h5><a class="nav-link" routerLink="employee">Employee</a></h5>

        </li>

      </ul>

    </div>

  </div>

  <div class="row">

    <div class="col">

      <router-outlet></router-outlet>

    </div>

  </div>

</div>

**App.componet.spec.ts**

import { TestBed } from '@angular/core/testing';

import { RouterTestingModule } from '@angular/router/testing';

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

describe('AppComponent', () => {

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      imports: [

        RouterTestingModule

      ],

      declarations: [

        AppComponent

      ],

    }).compileComponents();

  });

  it('should create the app', () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app).toBeTruthy();

  });

  it(`should have as title 'EventApp'`, () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app.title).toEqual('EventApp');

  });

  it('should render title', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.nativeElement as HTMLElement;

    expect(compiled.querySelector('.content span')?.textContent).toContain('EventApp app is running!');

  });

});

**Index.html**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <title>EventApp</title>

    <base href="/" />

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

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

    <link

      rel="stylesheet"

      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"

      integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"

      crossorigin="anonymous"

    />

  </head>

  <body>

    <app-root></app-root>

    <script

      src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.5/dist/umd/popper.min.js"

      integrity="sha384-Xe+8cL9oJa6tN/veChSP7q+mnSPaj5Bcu9mPX5F5xIGE0DVittaqT5lorf0EI7Vk"

      crossorigin="anonymous"

    ></script>

    <script

      src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.min.js"

      integrity="sha384-ODmDIVzN+pFdexxHEHFBQH3/9/vQ9uori45z4JjnFsRydbmQbmL5t1tQ0culUzyK"

      crossorigin="anonymous"

    ></script>

  </body>

</html>

**Style.css**

/\* You can add global styles to this file, and also import other style files \*/

body{

    color: black;

    /\* background-color: bisque; \*/

    background-image: url(assets\bg5.jpg);

}

.base1{

    background-color: white;

    text-emphasis-color: rgb(96, 177, 38);

}

.centered {

    color: black;

    position: absolute;

    top: 50%;

    left: 50%;

    transform: translate(-50%, -50%);

  }