**Event Management App Source Code**

**api.service.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/employees", data)

    .pipe(map((res:any)=>{

      return res;

    }))

  }

  getEmployee(){

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

    .pipe(map((res:any)=>{

      return res;

    }))

  }

  updateEmployee(data : any, id: number){

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

    .pipe(map((res:any)=>{

      return res;

    }))

  }

  deleteEmployee(id : number){

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

    .pipe(map((res:any)=>{

      return res;

    }))

  }

}

**app-routing.module.ts**

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

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

import { EmpolyeeDashboardComponent } from './empolyee-dashboard/empolyee-dashboard.component';

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

import { SignupComponent } from './signup/signup.component';

const routes: Routes = [

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

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

  {path:'signup', component:SignupComponent},

  {path:'dashboard', component:EmpolyeeDashboardComponent }

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

**app.component.html**

<router-outlet></router-outlet>

**app.component.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 'EventManagement'`, () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

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

  });

  it('should render title', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.nativeElement as HTMLElement;

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

  });

});

**app.component.ts**

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

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'EventManagement';

}

**app.module.ts**

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

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

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

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

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

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

import { EmpolyeeDashboardComponent } from './empolyee-dashboard/empolyee-dashboard.component';

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

import { SignupComponent } from './signup/signup.component';

@NgModule({

  declarations: [

    AppComponent,

    EmpolyeeDashboardComponent,

         LoginComponent,

         SignupComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    ReactiveFormsModule,

    HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

**Employee Dashboard Component**

**employee-dashboard.model.ts**

export class EmployeeModel{

    id:number = 0;

    firstName:string = '';

    lastName: string = '';

    email : string = '';

    mobile : string = '';

    salary : string = '';

}

**empolyee-dashboard.component.css**

body{

    margin: 0;

    padding: 0;

    background: linear-gradient(120deg,#2980b9, #8e44ad);

    height: 100vh;

    overflow: hidden;

  }

  @import url('https://fonts.googleapis.com/css2?family=Noto+Sans:wght@700&family=Poppins:wght@400;500;600&display=swap');

\*{

  box-sizing: border-box;

  font-family: "Poppins", sans-serif;

}

h1{

    font-family:'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

    letter-spacing: 2px;

    text-shadow:  2px 2px 4px white;

}

.navbar{

    text-align: center;

}

**empolyee-dashboard.component.html**

<body>

  <div class="conatainer">

    <nav class="navbar">

      <h1 style="color: rgb(1, 9, 17)"><b>Event Management</b></h1>

      <div class="d-flex">

        <button

          (click)="clickAddEMployee()"

          type="button"

          class="btn btn-success"

          data-bs-toggle="modal"

          data-bs-target="#staticBackdrop"

        >

          Add Employee

        </button>

        <a routerLink="/login" type="button" class="btn btn-danger mx-2">

          Logout

        </a>

      </div>

    </nav>

  </div>

  <table style="font-weight: 700" class="table table-striped mt-3">

    <thead style="color: rgb(253, 253, 253)" class="thead-dark">

      <tr>

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

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

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

        <th scope="col">Email No</th>

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

        <th scope="col">Salary</th>

        <th scope="col">Action</th>

      </tr>

    </thead>

    <tbody>

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

        <td>{{ row.id }}</td>

        <td>{{ row.firstName }}</td>

        <td>{{ row.lastName }}</td>

        <td>{{ row.email }}</td>

        <td>{{ row.mobile }}</td>

        <td>{{ row.salary }}</td>

        <td>

          <button

            (click)="onEdit(row)"

            type="button"

            data-bs-toggle="modal"

            data-bs-target="#staticBackdrop"

            class="btn btn-info"

          >

            Edit

          </button>

          <button (click)="deleteEmployee(row)" class="btn btn-danger mx-4">

            Delete

          </button>

        </td>

      </tr>

    </tbody>

  </table>

</body>

<!-- Modal -->

<div

  class="modal fade"

  id="staticBackdrop"

  data-bs-backdrop="static"

  data-bs-keyboard="false"

  tabindex="-1"

  aria-labelledby="staticBackdropLabel"

  aria-hidden="true"

>

  <div class="modal-dialog">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title" id="staticBackdropLabel">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">

            <label for="exampleInputEmail1" class="form-label"

              >Frist Name</label

            >

            <input

              type="text"

              class="form-control"

              formControlName="firstName"

              id="exampleInputEmail1"

              aria-describedby="emailHelp"

            />

          </div>

          <div class="mb-3">

            <label for="exampleInputPassword1" class="form-label"

              >Last Name</label

            >

            <input

              type="text"

              class="form-control"

              formControlName="lastName"

              id="exampleInputPassword1"

            />

          </div>

          <div class="mb-3">

            <label for="exampleInputEmail1" class="form-label">Email Id</label>

            <input

              type="text"

              class="form-control"

              formControlName="email"

              id="exampleInputEmail1"

              aria-describedby="emailHelp"

            />

          </div>

          <div class="mb-3">

            <label for="exampleInputEmail1" class="form-label">Mobile No</label>

            <input

              type="text"

              class="form-control"

              formControlName="mobile"

              id="exampleInputEmail1"

              aria-describedby="emailHelp"

            />

          </div>

          <div class="mb-3">

            <label for="exampleInputEmail1" class="form-label">Salary</label>

            <input

              type="text"

              class="form-control"

              id="exampleInputEmail1"

              formControlName="salary"

              aria-describedby="emailHelp"

            />

          </div>

        </form>

      </div>

      <div class="modal-footer">

        <button

          type="button"

          id="cancel"

          class="btn btn-secondary"

          data-bs-dismiss="modal"

        >

          Close

        </button>

        <button

          \*ngIf="showAdd"

          type="button"

          (click)="postEmployeeDetails()"

          class="btn btn-primary"

        >

          Add

        </button>

        <button

          \*ngIf="showUpdate"

          type="button"

          (click)="updateEmployeeDetails()"

          class="btn btn-primary"

        >

          Update

        </button>

      </div>

    </div>

  </div>

</div>

**empolyee-dashboard.component.ts**

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

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

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

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

@Component({

  selector: 'app-empolyee-dashboard',

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

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

})

export class EmpolyeeDashboardComponent implements OnInit {

  formValue !: FormGroup;

  employeeModelobj : EmployeeModel = new EmployeeModel();

  employeeData !: any;

  showAdd !: boolean;

  showUpdate !: boolean;

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

  ngOnInit(): void {

    this.formValue = this.formbuilber.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 Successfully")

        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(row: any){

      this.api.deleteEmployee(row.id)

      .subscribe(res=>{

        alert("Employee Deleted")

        this.getAllEmployee();

      })

    }

    onEdit(row : any){

      this.showAdd = false;

      this.showUpdate = true;

      this.employeeModelobj.id = row.id;

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

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

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

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

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

    }

    updateEmployeeDetails(){

      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();

      })

    }

}

**Login Component**

**.html**

<div class="container">

  <div class="row">

    <div class="col-md-6">

      <div class="card">

        <div class="text-center">

          <h1>SignUp</h1>

          <h6>Register Yourself!!</h6>

        </div>

        <form>

          <form [formGroup]="signupForm" (ngSubmit)="signUp()">

            <div class="mb-3">

              <label for="exampleInputEmail1" class="form-label"

                >Full Name</label

              >

              <input

                type="text"

                class="form-control"

                formControlName="fullname"

                id="exampleInputEmail1"

                aria-describedby="emailHelp"

              />

            </div>

            <div class="mb-3">

              <label for="exampleInputPassword1" class="form-label"

                >Mobile Number</label

              >

              <input

                type="text"

                class="form-control"

                formControlName="mobile"

                id="exampleInputPassword1"

              />

            </div>

            <div class="mb-3">

              <label for="exampleInputPassword1" class="form-label"

                >Email</label

              >

              <input

                type="email"

                class="form-control"

                formControlName="email"

                id="exampleInputPassword1"

              />

            </div>

            <div class="mb-3">

              <label for="exampleInputPassword1" class="form-label"

                >Password</label

              >

              <input

                type="password"

                class="form-control"

                formControlName="password"

                id="exampleInputPassword1"

              />

            </div>

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

          </form>

          <a routerLink="/login">Already Register? Click to login!!</a>

        </form>

      </div>

    </div>

  </div>

</div>

**login.component.css**

@import url('https://fonts.googleapis.com/css2?family=Noto+Sans:wght@700&family=Poppins:wght@400;500;600&display=swap');

\*{

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: "Poppins", sans-serif;

}

body{

  margin: 0;

  padding: 0;

  background: linear-gradient(120deg,#FDF5DF,#5EBEC4, #F92C85);

height: 100vh;

  overflow: hidden;

}

.center{

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

  width: 400px;

  background: white;

  border-radius: 10px;

  box-shadow: 10px 10px 15px rgba(0,0,0,0.05);

}

.center h1{

  text-align: center;

  padding: 20px 0;

  border-bottom: 1px solid silver;

}

.center form{

  padding: 0 40px;

  box-sizing: border-box;

}

form .txt\_field{

  position: relative;

  border-bottom: 2px solid #adadad;

  margin: 30px 0;

}

.txt\_field input{

  width: 100%;

  padding: 0 5px;

  height: 40px;

  font-size: 16px;

  border: none;

  background: none;

  outline: none;

}

.txt\_field label{

  position: absolute;

  top: 50%;

  left: 5px;

  color: #adadad;

  transform: translateY(-50%);

  font-size: 16px;

  pointer-events: none;

  transition: .5s;

}

.txt\_field span::before{

  content: '';

  position: absolute;

  top: 40px;

  left: 0;

  width: 0%;

  height: 2px;

  background: #2691d9;

  transition: .5s;

}

.txt\_field input:focus ~ label,

.txt\_field input:valid ~ label{

  top: -5px;

  color: #2691d9;

}

.txt\_field input:focus ~ span::before,

.txt\_field input:valid ~ span::before{

  width: 100%;

}

.pass{

  margin: -5px 0 20px 5px;

  color: #a6a6a6;

  cursor: pointer;

}

.pass:hover{

  text-decoration: underline;

}

input[type="submit"]{

  width: 100%;

  height: 50px;

  border: 1px solid;

  background: #2691d9;

  border-radius: 25px;

  font-size: 18px;

  color: #e9f4fb;

  font-weight: 700;

  cursor: pointer;

  outline: none;

}

input[type="submit"]:hover{

  border-color: #2691d9;

  transition: .5s;

}

.signup\_link{

  margin: 30px 0;

  text-align: center;

  font-size: 16px;

  color: #666666;

}

.signup\_link a{

  color: #2691d9;

  text-decoration: none;

}

.signup\_link a:hover{

  text-decoration: underline;

}

**login.component.html**

<body>

  <div class="center">

    <h1>Login</h1>

    <form [formGroup]="loginForm" (ngSubmit)="login()">

      <div class="txt\_field">

        <input

          type="email"

          class="form-control"

          formControlName="email"

          id="exampleInputEmail1"

          aria-describedby="emailHelp"

          placeholder="Email"

        />

        <span></span>

        <label for="exampleInputEmail1" class="form-label"></label>

      </div>

      <div class="txt\_field">

        <input

          type="password"

          class="form-control"

          formControlName="password"

          id="exampleInputPassword1"

          placeholder="Password"

        />

        <span></span>

        <label for="exampleInputPassword1" class="form-label"></label>

      </div>

      <input type="submit" value="Login" />

      <div class="signup\_link">

        Not a member? <a routerLink="/signup">Signup</a>

      </div>

    </form>

  </div>

</body>

**login.component.ts**

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

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

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

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

@Component({

  selector: 'app-login',

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

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

})

export class LoginComponent implements OnInit {

  public loginForm !: FormGroup

  constructor(private formBulider : FormBuilder, private http : HttpClient, private router : Router) { }

  ngOnInit(): void {

    this .loginForm = this.formBulider.group({

      email :[''],

      password:['']

    })

  }

  login(){

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

    .subscribe(res=>{

      const user = res.find((a : any)=>{

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

      });

      if(user){

        alert("Login Success");

        this.loginForm.reset();

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

      }else{

        alert("User not found");

      }

    },err=>{

      alert("Something went wrong!!")

    })

  }

}

**Signup Component**

**signup.component.css**

@import url('https://fonts.googleapis.com/css2?family=Noto+Sans:wght@700&family=Poppins:wght@400;500;600&display=swap');

\*{

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: "Poppins", sans-serif;

}

body{

  margin: 0;

  padding: 0;

  background: linear-gradient(120deg,#425664,#C6AD8F);

height: 100vh;

  overflow: hidden;

}

.center{

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

  width: 400px;

  background: white;

  border-radius: 10px;

  box-shadow: 10px 10px 15px rgba(0,0,0,0.05);

}

.center h1{

  text-align: center;

  padding: 20px 0;

  border-bottom: 1px solid silver;

}

.center form{

  padding: 0 40px;

  box-sizing: border-box;

}

form .txt\_field{

  position: relative;

  border-bottom: 2px solid #adadad;

  margin: 30px 0;

}

.txt\_field input{

  width: 100%;

  padding: 0 5px;

  height: 40px;

  font-size: 16px;

  border: none;

  background: none;

  outline: none;

}

.txt\_field label{

  position: absolute;

  top: 50%;

  left: 5px;

  color: #adadad;

  transform: translateY(-50%);

  font-size: 16px;

  pointer-events: none;

  transition: .5s;

}

.txt\_field span::before{

  content: '';

  position: absolute;

  top: 40px;

  left: 0;

  width: 0%;

  height: 2px;

  background: #2691d9;

  transition: .5s;

}

.txt\_field input:focus ~ label,

.txt\_field input:valid ~ label{

  top: -5px;

  color: #2691d9;

}

.txt\_field input:focus ~ span::before,

.txt\_field input:valid ~ span::before{

  width: 100%;

}

.pass{

  margin: -5px 0 20px 5px;

  color: #a6a6a6;

  cursor: pointer;

}

.pass:hover{

  text-decoration: underline;

}

input[type="submit"]{

  width: 100%;

  height: 50px;

  border: 1px solid;

  background: #2691d9;

  border-radius: 25px;

  font-size: 18px;

  color: #e9f4fb;

  font-weight: 700;

  cursor: pointer;

  outline: none;

}

input[type="submit"]:hover{

  border-color: #2691d9;

  transition: .5s;

}

.signup\_link{

  margin: 30px 0;

  text-align: center;

  font-size: 16px;

  color: #666666;

}

.signup\_link a{

  color: #2691d9;

  text-decoration: none;

}

.signup\_link a:hover{

  text-decoration: underline;

}

**signup.component.html**

<body>

  <div class="center">

    <h1>Signup</h1>

    <form [formGroup]="signupForm" (ngSubmit)="signUp()">

      <div class="txt\_field">

        <input

          type="text"

          class="form-control"

          formControlName="fullname"

          id="exampleInputEmail1"

          aria-describedby="emailHelp"

          placeholder="Full name"

        />

        <span></span>

        <label for="exampleInputEmail1" class="form-label"></label>

      </div>

      <div class="txt\_field">

        <input

          type="text"

          class="form-control"

          formControlName="mobile"

          id="exampleInputPassword1"

          placeholder="Mobile Number"

        />

        <span></span>

        <label for="exampleInputPassword1" class="form-label"></label>

      </div>

      <div class="txt\_field">

        <input

          type="text"

          class="form-control"

          formControlName="email"

          id="exampleInputPassword1"

          placeholder="Email"

        />

        <span></span>

        <label for="exampleInputPassword1" class="form-label"></label>

      </div>

      <div class="txt\_field">

        <input

          type="password"

          class="form-control"

          formControlName="password"

          id="exampleInputPassword1"

          placeholder="Password"

        />

        <span></span>

        <label for="exampleInputPassword1" class="form-label"></label>

      </div>

      <input type="submit" value="Register" />

      <div class="signup\_link">

        Already member?<a routerLink="/login"> Click to login</a>

      </div>

    </form>

  </div>

</body>

**signup.component.ts**

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

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

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

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

@Component({

  selector: 'app-signup',

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

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

})

export class SignupComponent implements OnInit {

  public signupForm !: FormGroup;

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

  ngOnInit(): void {

   this.signupForm = this.formBuilder.group({

    fullname:[''],

    email:[''],

    password:[''],

    mobile:['']

   })

  }

  signUp(){

    this.http.post<any>("http://localhost:3000/signupUsers", this.signupForm.value)

    .subscribe(res=>{

      alert("Signup Succesfully");

      this.signupForm.reset();

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

    },err=>{

      alert("Something went wrong")

    })

  }

}