

EVENT MANAGEMENT APP

```
<router-outlet></router-outlet>
```

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

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

```
import { HomepageComponent } from './homepage/homepage.component';
```

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

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

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

```
import { EmployeeListComponent } from './homepage/dashboard/employee-list/employee  
list.component';
```

```
import { ViewDetailsComponent } from './homepage/dashboard/view-details/view  
details.component';
```

```
import { AddEmployeeComponent } from './homepage/dashboard/add-employee/add  
employee.component';
```

```
import { UpdateComponent } from './homepage/dashboard/update/update.component';
```

```
const routes: Routes = [ {path:'',redirectTo:"homepage",pathMatch:"full"},
```

```
{path:'homepage',
```

```
component:HomepageComponent},
```

```
{path:'signin',
```

```
component:SigninComponent},
```

```
{path:'signup',
```

```
component:SignupComponent},
```

```
{path:'home',
```

```
component:DashboardComponent},
```

```
{path:'employeeelist',
```

```
component:EmployeeListComponent},
```

```
{path:'viewdetails/:id',
```

```
component:ViewDetailsComponent},
```

```
{path:'addEmployee',
```

```

    component: AddEmployeeComponent},
    { path: 'update/:id',
      component: UpdateComponent }
  ];
  @NgModule({
    imports: [RouterModule.forRoot(routes)],
    exports: [RouterModule]
  })
  export class AppRoutingModule { }

  import { Component } from '@angular/core';
  @Component({
    selector: 'app-root',
    templateUrl: './app.component.html',
    styleUrls: ['./app.component.css']
  })
  export class AppComponent {
    title = 'eventManagementApp';
  }

  import { TestBed } from '@angular/core/testing';
  import { RouterTestingModule } from '@angular/router/testing';
  import { AppComponent } from './app.component';
  describe('AppComponent', () => {
    beforeEach(() => TestBed.configureTestingModule({
      imports: [RouterTestingModule],
      declarations: [AppComponent]
    }));
    it('should create the app', () => {
      const fixture = TestBed.createComponent(AppComponent);
      const app = fixture.componentInstance;
      expect(app).toBeTruthy();
    });
  });

```

```

});

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

const fixture = TestBed.createComponent(AppComponent);

const app = fixture.componentInstance;

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

});

it('should render title', () => {

const fixture = TestBed.createComponent(AppComponent);

fixture.detectChanges();

const compiled = fixture.nativeElement as HTMLElement;

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

});

});

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

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

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

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

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

import { HomepageComponent } from './homepage/homepage.component';

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

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

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

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

import { EmployeeListComponent } from './homepage/dashboard/employee-list/employee
list.component';

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

import { ViewDetailsComponent } from './homepage/dashboard/view-details/view
details.component';

import { AddEmployeeComponent } from './homepage/dashboard/add-employee/add
employee.component';

```

```

import { UpdateComponent } from './homepage/dashboard/update/update.component';

@NgModule({
  declarations: [
    AppComponent,
    HomepageComponent,
    SignupComponent,
    SigninComponent,
    DashboardComponent,
    EmployeeListComponent,
    ViewDetailsComponent,
    AddEmployeeComponent,
    UpdateComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    HttpClientModule,
    ReactiveFormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})

export class AppModule { }

import { TestBed } from '@angular/core/testing';
import { EmployeeService } from './employee.service';

describe('EmployeeService', () => {
  let service: EmployeeService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(EmployeeService);
  });
});

```

```

});

it('should be created', () => {
  expect(service).toBeTruthy();
});

});

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

@Injectable({
  providedIn: 'root'
})

export class EmployeeService {
  constructor(private http: HttpClient) { }

  public getEmployees() {
    return this.http.get("http://localhost:3000/employees");
  }

  public getEmployeeById(id:any) {
    return this.http.get("http://localhost:3000/employees/"+id);
  }

  public deletebyid(id:any ) {
    return this.http.delete("http://localhost:3000/employees/"+id);
  }

  public putEmployee(id:any,EmployeeData: any) {
    return this.http.put("http://localhost:3000/employees/"+id, EmployeeData);
  }

  public postEmployee(EmployeeData: any) {
    return this.http.post("http://localhost:3000/employees", EmployeeData);
  }
}

{
  "employees": [

```

```
{
  "id": 1,
  "first_name": "Sebastian",
  "last_name": "Eschweiler",
  "email": "sebastian@codingthesmartway.com"
},
{
  "id": 2,
  "first_name": "Steve",
  "last_name": "Palmer",
  "email": "steve@codingthesmartway.com"
},
{
  "id": 3,
  "first_name": "Ann",
  "last_name": "Smith",
  "email": "ann@codingthesmartway.com"
},
{
  "first_name": "Abishek",
  "last_name": "Prabhu",
  "email": "abi@gmail.com",
  "id": 5
},
{
  "first_name": "pratheesh",
  "last_name": "kumar",
  "email": "a@c.c",
  "id": 6
},
{
```

```
    "first_name": "Pravanth",
    "last_name": "Kumar",
    "email": "pravan@gmail.com",
    "id": 7
  }
]
}
```

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>EventManagerApp</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@5.0.2/dist/css/bootstrap.min.css">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@5.15.4/css/all.min.css">
</head>
<body>
  <app-root></app-root>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

declare module "*.json" {
  const value: any;
  export default value;
```

```
}
```

(Create component ng generate component homepage (or) ng g c homepage.)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Header Page</title>
```

```
  <link rel="stylesheet"
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">
```

```
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
```

```
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font  
awesome/5.15.4/css/all.min.css">
```

```
  <style>
```

```
    @import
```

```
url('https://fonts.googleapis.com/css2?family=Inspiration&family=Poppins:wght@300;400;500;700&  
family=Roboto+Condensed:wght@300&display=swap');
```

```
    * {
```

```
      margin: 0;
```

```
      padding: 0;
```

```
    }
```

```
    #main{
```

```
      font-family: 'Inspiration', cursive;
```

```
      font-size: 30px;
```

```
    }
```

```
  </style>
```

```
</head>
```



```

<body>

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

    <div class="container-fluid">

      <a class="navbar-brand" id="main" href="#">Event Management App</a>

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

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse p-2" style="margin-left: 52px;"
id="navbarSupportedContent">

        <ul class="nav navbar-nav ms-auto">

          <li class="nav-item"><a class="btn btn-outline-success my-2 my-sm-0" routerLink="/signin"
style="margin: 5px; padding-right: 10px;">Sign In</a></li>

          <li class="nav-item"><a class="btn btn-outline-warning my-2 my-sm-0" routerLink="/signup"
style="margin: 5px;">Sign Up</a></li>

        </ul>

      </div>

    </div>

  </nav>

  <div class="container-fluid bg-dark">

    <div class="homepage bg-dark">

      <h1 class="text-light">Welcome to the Event Management Application</h1>

      <div class="buttons">

        <button class="signin-btn btn btn-outline-info my-2 my-sm-0" routerLink="/signin">Sign
In</button>

```

```
        <button class="signup-btn btn btn-outline-warning my-2 my-sm-0" routerLink="/signup">Sign
Up</button>

    </div>

</div>

</div>

</body>

</html>
```

```
.homepage {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  height: 100vh;
  background-color: #f2f2f2;
}
```

```
.homepage h1 {
  font-size: 2.5rem;
  margin-bottom: 2rem;
}
```

```
.buttons {
  display: flex;
  gap: 1rem;
}
```

```
.signin-btn,
.signup-btn {
  padding: 1rem 2rem;
  border: none;
```

```
border-radius: 4px;
font-size: 1.2rem;
font-weight: bold;
text-transform: uppercase;
cursor: pointer;
}
```

```
.signin-btn {
  background-color: #007bff;
  color: #fff;
}
```

```
.signup-btn {
  background-color: #28a745;
  color: #fff;
}
```

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

```
import { HomeComponent } from './homepage.component';
```

```
describe('HomepageComponent', () => {
  let component: HomeComponent;
  let fixture: ComponentFixture<HomepageComponent>;
  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [HomeComponent]
    });
    fixture = TestBed.createComponent(HomepageComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
});
```

```

it('should create', () => {
  expect(component).toBeTruthy();
});

});

import { Component } from '@angular/core';
@Component({
  selector: 'app-homepage',
  templateUrl: './homepage.component.html',
  styleUrls: ['./homepage.component.css']
})
export class HomepageComponent {

}

```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Header Page</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font
awesome/5.15.4/css/all.min.css">
  <style>
    @import
url('https://fonts.googleapis.com/css2?family=Inspiration&family=Poppins:wght@300;400;500;700&
family=Roboto+Condensed:wght@300&display=swap');
    * {
      margin: 0;

```

```

padding: 0;

}

#main{
    font-family: 'Inspiration', cursive;
    font-size: 30px;
}

</style>
</head>

<body>

<!-- navbar function -->

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

    <div class="container-fluid">

        <!--  -->

        <a class="navbar-brand" id="main" href="#">Event Management App</a>

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

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse p-2" style="margin-left: 52px;"
id="navbarSupportedContent">

            <ul class="nav navbar-nav ms-auto">

                <li class="nav-item"><a class="btn btn-outline-success my-2 my-sm-0" routerLink="/signin"
style="margin: 5px; padding-right: 10px;">Sign In</a></li>

                <li class="nav-item"><a class="btn btn-outline-warning my-2 my-sm-0" routerLink="/signup"
style="margin: 5px;">Sign Up</a></li>

            </ul>

```

```

    </div>

</div>

</nav>

<!-- signup content -->
<div class="container-fluid">

    <div class="row justify-content-center bg-dark text-muted">

        <div class="col-lg-6 col-md-6 col-sm-10">

            <form [formGroup]="regForm" (ngSubmit)="printDetails()">

                <!-- <form class="d-flex flex-column h-100"> -->

                    <div class="register-container">

                        <div class="text-center text-warning">

                            <h1>Sign Up</h1>

                        </div>

                        <div class="mb-3">

                            <label for="personname" class="form-label">Person Name</label><br />

                            <input type="text" class="form-control" formControlName="personname"
placeholder="Enter Name...."/>

                            <br />

                            <span class="errmsginfo text-danger" *ngIf="regForm.controls.personname.touched
&& regForm.controls.personname.invalid">

                                Person name should not be blank

                            </span>

                        </div>

                        <div class="mb-3">

                            <label for="phonenumber" class="form-label">Phone Number</label><br />

                            <input type="number" class="form-control" formControlName="phonenumber"
placeholder="Enter PhoneNumber...."/>

                            <br />

                            <span class="errmsginfo text-danger"

```

```

*ngIf="regForm.controls.phonenumber.touched && regForm.controls.phonenumber.invalid">
    Phone Number should not be blank
</span>
</div>
<div class="mb-3">
    <label for="email" class="form-label">Email</label><br />
    <input type="email" class="form-control" formControlName="email"
placeholder="Enter Email"/>
    <br />
    <span class="errmsginfo text-danger" *ngIf="regForm.controls.email.touched &&
regForm.controls.email.invalid">
        Email should not be blank
    </span>
</div>
<div class="mb-3">
    <label for="password" class="form-label">Password</label><br />
    <input type="password" class="form-control" formControlName="password"
placeholder="Enter Password" [(ngModel)]="regForm.value.password" />
    <br />
    <span class="errmsginfo text-danger" *ngIf="regForm.controls.password.touched &&
regForm.controls.password.invalid">
        Password should not be blank
    </span>
</div>
<div class="mt-auto text-center">
    <input type="submit" [disabled]="regForm.invalid" class="btn btn-lg btn-success"
value="Register Here" />
</div>
<br /><br />
    <div [innerHTML]="msg" class="text-center text-danger"></div>
</div>

```

```

        </form>
    </div>
</div>

</div>
<div class="col-md-3"></div>

</body>
</html>

.register-container{
max-width: 640px;
margin: 0 auto;
margin-top: 40px;
padding: 70px;
box-shadow: 0px 0px 40px 10px rgba(0,0,0, 0.6);
/* transition: transform 0.9s ease-out; */
border-radius: 16px;
}

.register-container h1 {
margin:20px 0px;
margin-bottom:50px;
}

.register-container .form-label{
margin-bottom:10px;
}

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { SignupComponent } from './signup.component';
describe('SignupComponent', () => {
let component: SignupComponent;
let fixture: ComponentFixture<SignupComponent>;
beforeEach(() => {

```



```

 TestBed.configureTestingModule({
  declarations: [SignupComponent]
});

 fixture = TestBed.createComponent(SignupComponent);
 component = fixture.componentInstance;
 fixture.detectChanges();
});

 it('should create', () => {
  expect(component).toBeTruthy();
});

});

import { Component, OnInit } from '@angular/core';
import { FormBuilder, Validators } 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 {
  constructor(private builder: FormBuilder,private router: Router) { }

  ngOnInit(): void {
  }

  result: any;
  msg: string = "";

  regForm = this.builder.group({
    personname: this.builder.control("", Validators.required),
    phonenumber: this.builder.control("", Validators.required),
    email: this.builder.control("", [Validators.required, Validators.email]),
    password: this.builder.control("", Validators.required)
  });

```

```

printDetails() {
  if (this.regForm.valid) {
    const registrationData = {
      personname: this.regForm.value.personname,
      phonenumber: this.regForm.value.phonenumber,
      email: this.regForm.value.email,
      password: this.regForm.value.password
    };
    // check the email is already exists...!
    const storedData = localStorage.getItem('registrationData');

    if (storedData) {
      const existingData = JSON.parse(storedData);
      if (existingData.email === registrationData.email) {
        this.msg = "Email already exists. Please choose a different email.";
        return;
      }
    }
    // if email is not exist store the data
    localStorage.setItem('registrationData', JSON.stringify(registrationData));

    this.msg = "Registration successful....!" ;
    setTimeout(() => {
      this.router.navigate(['/signin']);
    }, 700); // Wait for 0.7 seconds before redirecting

  } else {
    this.msg = "Invalid Form";
  }
}
}

```

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Header Page</title>

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">

  <!-- <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font
awesome/4.7.0/css/font-awesome.min.css"> -->

  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font
awesome/5.15.4/css/all.min.css">

  <style>

    @import
url('https://fonts.googleapis.com/css2?family=Inspiration&family=Poppins:wght@300;400;500;700&
family=Roboto+Condensed:wght@300&display=swap');

    * {

      margin: 0;

      padding: 0;

    }

    #main{

      font-family: 'Inspiration', cursive;

      font-size: 30px;

      /* font-weight: 400; */

    }

  </style>

</head>

<body>
```

```

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <!--  -->
    <a class="navbar-brand" id="main" href="#">Event Management App</a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs
target="#navbarSupportedContent" aria-controls="navbarNav" aria-expanded="false" aria
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse p-2" style="margin-left: 52px;"
id="navbarSupportedContent">
      <ul class="nav navbar-nav ms-auto">
        <li class="nav-item"><a class="btn btn-outline-success my-2 my-sm-0" routerLink="/signin"
style="margin: 5px; padding-right: 10px;">Sign In</a></li>
        <li class="nav-item"><a class="btn btn-outline-warning my-2 my-sm-0" routerLink="/signup"
style="margin: 5px;">Sign Up</a></li>
      </ul>
    </div>
  </div>
</nav>

```

```

<section id="team" class="py-4" style="background-image: url('https://img.freepik.com/free
vector/wavy-shape-business-background-paper-cut-style_1017
33399.jpg?w=1380&t=st=1688207002~exp=1688207602~hmac=2790ed29fd019f49088094c349dd5f
1d71db14130022fb570bdfa62d151eaac9');
background-repeat: no-repeat; background-attachment: fixed; background-size: cover;
width:100%;">

```

```

<div class="container">

```

```

<div class="login-container bg-dark text-light">

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

  <form #loginform="ngForm" (ngSubmit)="LoginCheck(loginform)">

    <div class="form-floating text-muted">

      <input type="text" [(ngModel)]="username" class="form-control" name="uname"
        required="required" placeholder="Username" #uctrl="ngModel" />

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

      <span class="errmsginfo text-danger" *ngIf="uctrl.touched && uctrl.invalid">
        Username can't be blank
      </span>

    </div>

    <!-- <br /> -->

    <div class="form-floating text-muted position-relative">

      <input type="password" class="form-control" [(ngModel)]="password" name="pswd"
        required="required" placeholder="Password" #pctrl="ngModel" />

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

      <span class="errmsginfo text-danger" *ngIf="pctrl.touched && pctrl.invalid">
        Password can't be blank
      </span>

    </div>

    <br /><br />

  </div>

  <div class="mb-3 form-check">

    <input type="checkbox" class="form-check-input" id="remember">

    <label class="form-check-label" for="remember">Remember me</label>

  </div>

  <div style="margin: 20px;">

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

  </div>

  <br /><div [innerHTML]="msg" class="text-center text-danger"></div>

  <div class="register-link">

    <p>Don't have an account? <a routerLink="/signup">Register</a></p>

```

```
        </div>
    </form>
</div>
</div>
<div class="col-md-3"></div>
```

```
</section>
```

```
.login-container {
    max-width: 450px;
    margin: 0 auto;
    margin-top: 100px;
    padding: 20px;
    background-color: #ffffff;
    box-shadow: 0px 0px 40px 10px rgba(0,0,0, 0.6);
    /* transition: transform 0.9s ease-out; */
    border-radius: 16px;
}
```

```
.login-container h2 {
    margin:30px;
    background: linear-gradient(to right, #56d8e4, #9f01ea);
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
}
```

```
.login-container .form-floating , .form-check{
    margin: 20px;
}
```

```
.login-container .btn-primary {
```

```
width: 100%;
```

```
}
```

```
.login-container .forgot-password , .register-link{
```

```
text-align: center;
```

```
margin-top: 10px;
```

```
}
```

```
.show-password-icon {
```

```
position: absolute;
```

```
top: 50%;
```

```
right: 10px;
```

```
transform: translateY(-50%);
```

```
cursor: pointer;
```

```
}
```

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

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

```
describe('SigninComponent', () => {
```

```
let component: SigninComponent;
```

```
let fixture: ComponentFixture<SigninComponent>;
```

```
beforeEach(() => {
```

```
TestBed.configureTestingModule({
```

```
declarations: [SigninComponent]
```

```
});
```

```
fixture = TestBed.createComponent(SigninComponent);
```

```
component = fixture.componentInstance;
```

```
fixture.detectChanges();
```

```
});
```

```
it('should create', () => {
```

```
expect(component).toBeTruthy();
```

```

});

});

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

@Component({
  selector: 'app-signin',
  templateUrl: './signin.component.html',
  styleUrls: ['./signin.component.css']
})
export class SigninComponent implements OnInit {

  username: string = "";
  password: string = "";
  msg: string = "";

  constructor(private router: Router) { }

  ngOnInit(): void {
  }

  //get the data from the localStorage to check the validation
  LoginCheck(frm: any) {
    if (frm.valid) {
      const storedData = localStorage.getItem('registrationData');

      if (storedData) {
        const registrationData = JSON.parse(storedData);

        if (this.username === registrationData.email && this.password === registrationData.password) {
          // Login successful
          localStorage.setItem('loggedInUser', JSON.stringify(registrationData));

```



```

    this.msg = "Login Successfully";

    setTimeout(() => {
        this.router.navigate(['/home']);
    }, 500); //wait for 0.5sec to redirect to home
} else {
    this.msg = "Incorrect..!<br />Please check your email and password";
}
} else {
    this.msg = "No registration data found";
}
} else {
    this.msg = "Invalid Form";
}
}

}

<nav class="navbar navbar-expand-lg navbar-light bg-light navbar-border">
    <div class="container-fluid">
        <a class="navbar-brand" routerLink="/home" style="margin-left: 1.2rem;"><b>Portal</b></a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs
target="#navbarSupportedContent" aria-controls="navbarNav" aria-expanded="false" aria
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse p-2" style="margin-left: 52px;"
id="navbarSupportedContent">
            <ul class="navbar-nav" routerLinkActive="active">
                <li class="nav-item active">
                    <a class="nav-link active" routerLink="/home">Home <span class="sr

```

```

only">(current)</span></a>

    </li>

    <li class="nav-item">

        <a class="nav-link" routerLink="/employeeelist" >employees</a>

    </li>

</ul>

<ul class="navbar-nav ms-auto" routerLinkActive="active">

    <div class="input-group">

        <input type="text" id="search" name="search" placeholder="Search by Id" class="form
control" [(ngModel)]="id" (ngModelChange)="findById(id)">

    </div>

</ul>

</div>

</div>

</nav>

<!-- dashboard content -->

<div class="container">

<h1>Welcome Admin,</h1>

<h3>Hello,<span class="text-primary">{{ loggedInUser.personname }}</span> </h3>

<!-- reset admin password -->

<div *ngIf="showResetForm">

<h4>Reset Password</h4>

<form (ngSubmit)="resetPassword()">

<label>New Password</label><br>

<input type="password" [(ngModel)]="newPassword" name="newPassword" required>

<br>

<input type="submit" value="Reset Password" class="btn btn-primary">

</form>

</div>

<button *ngIf="!showResetForm" (click)="showResetForm = true" class="btn btn-secondary">Reset

```

```

Password</button>

<div *ngIf="message">

<p class="text-danger">{{ message }}</p>

</div>

</div>

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { DashboardComponent } from './dashboard.component';
describe('DashboardComponent', () => {
  let component: DashboardComponent;
  let fixture: ComponentFixture<DashboardComponent>;
  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [DashboardComponent]
    });
    fixture = TestBed.createComponent(DashboardComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

import { Component, OnInit } from '@angular/core';
import { EmployeeService } from 'src/app/employee.service';
@Component({
  selector: 'app-dashboard',
  templateUrl: './dashboard.component.html',
  styleUrls: ['./dashboard.component.css']
})
export class DashboardComponent implements OnInit {
  loggedInUser: any;

```

```

showResetForm: boolean = false;

newPassword: string = "";

employees: any;

message = "";

id: any;

constructor(private service: EmployeeService) { }

ngOnInit(): void {
  const storedUser = localStorage.getItem('loggedInUser');
  if (storedUser) {
    this.loggedInUser = JSON.parse(storedUser);
  }
}

findById(id: number) {
  if (id) {
    let response = this.service.getEmployeeById(id);
    response.subscribe((data: any) => {
      this.employees = [data];
    });
  } else {
    let response = this.service.getEmployees();
    response.subscribe((data: any) => {
      this.employees = data;
    });
  }
}

resetPassword() {
  // Update the password for the logged-in user
  this.loggedInUser.password = this.newPassword;
}

```

```

// Store the updated user data in localStorage
localStorage.setItem('loggedInUser', JSON.stringify(this.loggedInUser));

// Reset the form and hide it
this.newPassword = '';
this.showResetForm = false;

// Set the message to be displayed
this.message = 'Password has been updated successfully.';
// You can change the message as needed
// For other updates, you can set a different message here
}
}

<!-- nav bar -->
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <!--  -->
    <a class="navbar-brand" routerLink="/home" style="margin-left:
1.2rem;"><b>Portal</b></a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs
target="#navbarSupportedContent" aria-controls="navbarNav" aria-expanded="false" aria
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse p-2" style="margin-left: 52px;"
id="navbarSupportedContent">
      <ul class="navbar-nav" routerLinkActive="active">
        <li class="nav-item active">

```

```

        <a class="nav-link" routerLink="/home">Home <span class="sr
only">(current)</span></a>
    </li>
    <li class="nav-item">
        <a class="nav-link active" routerLink="/employeeelist" >employees</a>
    </li>
</ul>
</div>
</nav>

```

```

<!-- add employees -->
<div style="text-align: center; margin-top: 5rem;">
    <div class="row">
        <div class="col-md-3"></div>
        <div class="col-md-6">

            <div class="card col-8 offset-2">
                <div class="card-header">
                    <h4>Add Employees</h4>
                </div>
                <div class="card-body">
                    <form [formGroup]="regForm" (ngSubmit)="printDetails()">

                        <label>First Name</label><br />
                        <input type="text" class="form-control" formControlName="first_name"
placeholder="Enter the First Name....."/>
                        <br />
                        <span class="errormsginfo text-danger"
*ngIf="regForm.controls.first_name.touched && regForm.controls.first_name.invalid">
                            First Name should not be blank
                        </span>

```

```

        <br />

        <label>Last Name</label><br />

        <input type="text" class="form-control" formControlName="last_name"
placeholder="Enter the Last Name....." />

        <br />

        <span class="errormsginfo text-danger"
*ngIf="regForm.controls.last_name.touched && regForm.controls.last_name.invalid">

            Last Name should not be blank

        </span>

        <br />

        <label>email</label><br />

        <input type="email" class="form-control" formControlName="email"
placeholder="Enter the Email....." />

        <br />

        <span class="errormsginfo text-danger" *ngIf="regForm.controls.email.touched &&
regForm.controls.email.invalid">

            Email should not be blank

        </span>

        <br />

        <br>

        <input type="submit" [disabled]="regForm.invalid" class="btn btn-success"
value="Register Here" />

        <br /><br />

        <div [innerHTML]="msg" style="color:red"></div>

    </form>

</div>

</div>

</div>

<div class="col-md-3"></div>

```

```
</div>
```

```
</div>
```

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { AddEmployeeComponent } from './add-employee.component';
describe('AddEmployeeComponent', () => {
  let component: AddEmployeeComponent;
  let fixture: ComponentFixture<AddEmployeeComponent>;
  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [AddEmployeeComponent]
    });
    fixture = TestBed.createComponent(AddEmployeeComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

import { Component, OnInit } from '@angular/core';
import { FormBuilder, Validators } from '@angular/forms';
import { Router } from '@angular/router';
import { EmployeeService } from 'src/app/employee.service';
@Component({
  selector: 'app-add-employee',
  templateUrl: './add-employee.component.html',
  styleUrls: ['./add-employee.component.css']
})
export class AddEmployeeComponent implements OnInit {
  employees: any;
```



```
constructor(  
  private service: EmployeeService,  
  private builder: FormBuilder,  
  private router: Router  
) {}  
  
ngOnInit(): void {}  
  
result: any;  
  
msg: string = "";  
  
regForm = this.builder.group({  
  first_name: this.builder.control("", Validators.required),  
  last_name: this.builder.control("", Validators.required),  
  email: this.builder.control("", Validators.required)  
});  
  
printDetails() {  
  if (this.regForm.valid) {  
    const EmployeeData = {  
      first_name: this.regForm.value.first_name,  
      last_name: this.regForm.value.last_name,  
      email: this.regForm.value.email  
    };  
  
    this.service.postEmployee(EmployeeData).subscribe(  
      (response) => {  
        console.log(response);  
      },  
      (error) => {  
        console.error(error);  
      }  
    );  
  }  
}
```

```

    this.msg = 'Employee Details Added successfully...! ' + JSON.stringify(EmployeeData);
    setTimeout(() => {
        this.router.navigate(['/employeeelist']);
    }, 700); // Wait for 0.7 seconds before redirecting
} else {
    this.msg = 'Invalid Form';
}
}
}

<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
        <!--  -
    >

        <a class="navbar-brand" routerLink="/home" style="margin-left: 1.2rem;"><b>Portal</b></a>

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

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse p-2" style="margin-left: 52px;"
id="navbarSupportedContent">
            <ul class="navbar-nav" routerLinkActive="active">
                <li class="nav-item active">
                    <a class="nav-link" routerLink="/home">Home <span class="sr
only">(current)</span></a>
                </li>
                <li class="nav-item">
                    <a class="nav-link active" routerLink="/employeeelist">employees</a>
                </li>

```

```

</ul>

<ul class="navbar-nav ms-auto" routerLinkActive="active">

    <div class="input-group">

        <input type="text" id="search" name="search" placeholder="Search by Id" class="form
control" [(ngModel)]="id" (ngModelChange)="findById(id)">

    </div>

</ul>

</div>

</nav>

```

```

<div class="container" style="margin-top: 5rem; margin-right: 15rem;">

    <div class="row">

        <div class="col-8 offset-2">

            <div class="text-center mb-4">

                <td><a [routerLink]="['/addEmployee']" class="btn btn-secondary">Add
employee</a></td>

            </div>

            <table class="table table-bordered" style="width: 120%;">

                <thead>

                    <tr class="text-center">

                        <th>#</th>

                        <th>Firstname</th>

                        <th>LastName</th>

                        <th>Email</th>

                        <th></th>

                    </tr>

                </thead>

```

```

<tbody>
  <tr *ngFor="let employee of employees" class="text-center">
    <td>{{ employee.id }}</td>
    <td>{{ employee.first_name }}</td>
    <td>{{ employee.last_name }}</td>
    <td>{{ employee.email }}</td>
    <td><a [routerLink]="['/viewdetails', employee.id]" class="btn btn
secondary">viewDetails</a></td>
  </tr>
</tbody>
</table>
</div>
</div>
</div>

```

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

```
import { EmployeeListComponent } from './employee-list.component';
```

```
describe('EmployeeListComponent', () => {
```

```
  let component: EmployeeListComponent;
```

```
  let fixture: ComponentFixture<EmployeeListComponent>;
```

```
  beforeEach(() => {
```

```
    TestBed.configureTestingModule({
```

```
      declarations: [EmployeeListComponent]
```

```
    });
```

```
    fixture = TestBed.createComponent(EmployeeListComponent);
```

```
    component = fixture.componentInstance;
```

```
    fixture.detectChanges();
```

```
  });
```

```
it('should create', () => {  
  expect(component).toBeTruthy();  
});  
});
```

```
import { Component,OnInit } from '@angular/core';  
import { EmployeeService } from 'src/app/employee.service';  
@Component({  
  selector: 'app-employee-list',  
  templateUrl: './employee-list.component.html',  
  styleUrls: ['./employee-list.component.css']  
})  
export class EmployeeListComponent implements OnInit{  
  employees:any;  
  message="";  
  id:any;
```

```
  constructor(private service:EmployeeService) { }  
  ngOnInit(): void {  
    let response=this.service.getEmployees();  
    response.subscribe((data:any)=>this.employees=data);  
  }
```

```
  findById(id: number) {  
    if (id) {  
      let response = this.service.getEmployeeById(id);  
      response.subscribe((data: any) => {  
        this.employees = [data];  
      });  
    }
```

```

    } else {

        let response = this.service.getEmployees();

        response.subscribe((data: any) => {

            this.employees = data;

        });

    }

}

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

    <div class="container-fluid">

        <a class="navbar-brand" routerLink="/home" style="margin-left: 1.2rem;"><b>Portal</b></a>

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

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse p-2" style="margin-left: 52px;"
id="navbarSupportedContent">

            <ul class="navbar-nav" routerLinkActive="active">

                <li class="nav-item active">

                    <a class="nav-link" routerLink="/home">Home <span class="sr
only">(current)</span></a>

                </li>

                <li class="nav-item">

                    <a class="nav-link active" routerLink="/employeelist" >employees</a>

                </li>

            </ul>

        </div>

```

```
</nav>
```

```
<!-- update form -->
```

```
<div class="container" style="margin-top: 5rem;">
```

```
  <div class="row justify-content-center">
```

```
    <div class="col-4 col-sm-8">
```

```
      <h1 class="">Update Form</h1>
```

```
      <h5 class="text-danger"> {{ message }}</h5>
```

```
    <form>
```

```
      <div class="mb-3">
```

```
        <label for="id" class="form-label">id:</label>
```

```
        <input type="number" class="form-control" id="id" name="id"
```

```
        [(ngModel))="employees.id">
```

```
      </div>
```

```
      <div class="mb-3">
```

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

```
        <input type="text" class="form-control" id="first_name" name="first_name"
```

```
        [(ngModel))="employees.first_name">
```

```
      </div>
```

```
      <div class="mb-3">
```

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

```
        <input type="text" class="form-control" id="last_name" name="last_name"
```

```
        [(ngModel))="employees.last_name">
```

```
      </div>
```

```
    <div class="mb-3">
```

```

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

        <input type="email" class="form-control" id="email" name="email"
[(ngModel)]="employees.email">

    </div>

    <button type="button" class="btn btn-primary"
(click)="updateNow(employees.id)">Update</button>

    <td><a [routerLink]="['/employeeelist']" class="btn btn
secondary">EmployeeList</a></td>

</form>
</div>
</div>
</div>

```

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { UpdateComponent } from './update.component';

describe('UpdateComponent', () => {
  let component: UpdateComponent;
  let fixture: ComponentFixture<UpdateComponent>;

  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [UpdateComponent]
    });
    fixture = TestBed.createComponent(UpdateComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```



```

});

import { ActivatedRoute } from '@angular/router';
import { Component,OnInit } from '@angular/core';
import { EmployeeService } from 'src/app/employee.service';

```

```

@Component({
  selector: 'app-update',
  templateUrl: './update.component.html',
  styleUrls: ['./update.component.css']
})
export class UpdateComponent implements OnInit {
  employees:any;
  message = "";
  constructor(private route: ActivatedRoute,private service:EmployeeService) { }
  ngOnInit(): void {
    const id = this.route.snapshot.paramMap.get('id');
    let response=this.service.getEmployeeById(id);
    response.subscribe((data:any)=>this.employees=data);
  }
  updateNow(id: number) {
    let response = this.service.putEmployee(id, this.employees);
    response.subscribe((data: any) => {
      this.message = "Updated...! id: "+id;
    });
  }
}

```

→create component inside the src/app/homepage/dashboard/ ng g view-details

<!-- nav bar -->

```

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

  <div class="container-fluid">

    <!--  -
  >

    <a class="navbar-brand" routerLink="/home" style="margin-left: 1.2rem;"><b>Portal</b></a>

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

      <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse p-2" style="margin-left: 52px;"
id="navbarSupportedContent">

      <ul class="navbar-nav" routerLinkActive="active">

        <li class="nav-item active">

          <a class="nav-link" routerLink="/home">Home <span class="sr
only">(current)</span></a>

        </li>

        <li class="nav-item">

          <a class="nav-link active" routerLink="/employeelist" >employees</a>

        </li>

      </ul>

    </div>

  </div>

</nav>

<!-- view Details -->

<div class="container" style="margin-top: 5rem;">

  <div class="row justify-content-center">

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

```

```
<h5 class="text-danger">{{message}}</h5>

<table class="table table-bordered">

  <thead class="table">

    <tr>

      <td colspan="2"><span>Details of</span> <b><i>{{ employees.first_name }} {{
employees.last_name }}</i></b></td>

    </tr>

  </thead>

  <tbody>

    <tr>

      <td class="label-col">ID:</td>

      <td>{{ employees.id }}</td>

    </tr>

    <tr>

      <td class="label-col">First Name:</td>

      <td>{{ employees.first_name }}</td>

    </tr>

    <tr>

      <td class="label-col">Last Name:</td>

      <td>{{ employees.last_name }}</td>

    </tr>

    <tr>

      <td class="label-col">Email:</td>

      <td>{{ employees.email }}</td>

    </tr>

  </tbody>

  <tfoot>

    <tr>

      <td colspan="2">

        <div class="button-group">

          <a [routerLink]="['/employeeelist']" class="btn btn-info">Employee List</a>

        </div>

      </td>

    </tr>

  </tfoot>

</table>
```

```
        <button class="btn btn-danger"
(click)="deleteEmployee(employees.id)">Delete</button>

        <a [routerLink]="['/update', employees.id]" class="btn btn-secondary">Update</a>
    </div>
</td>
</tr>
</tfoot>
</table>
</div>
</div>
</div>
```

```
.table thead td {
    border: 1px solid #000;
}

.table tbody td {
    border: 1px solid #ccc;
}

.table tfoot td {
    border: 1px solid #000;
}

.label-col {
    width: 240px;
}

.button-group {
    display: flex;
    justify-content: flex-start;
    gap: 10px;
}
```

```

.button-group {
  display: flex;
  justify-content: flex-start;
  gap: 10px;
}

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { ViewDetailsComponent } from './view-details.component';
describe('ViewDetailsComponent', () => {
  let component: ViewDetailsComponent;
  let fixture: ComponentFixture<ViewDetailsComponent>;
  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [ViewDetailsComponent]
    });
    fixture = TestBed.createComponent(ViewDetailsComponent);
    component = fixture.componentInstance;
    f
    ixture.detectChanges();
  });
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

import { ActivatedRoute } from '@angular/router';
import { Component, OnInit, ChangeDetectorRef } from '@angular/core';
import { EmployeeService } from 'src/app/employee.service';
import { Router } from '@angular/router';
@Component({
  selector: 'app-view-details',
  templateUrl: './view-details.component.html',
  styleUrls: ['./view-details.component.css']
})

```

```

})

export class ViewDetailsComponent implements OnInit {
  employees: any;
  message = '';
  constructor(
    private route: ActivatedRoute,
    private service: EmployeeService,
    private router: Router,
    private cdr: ChangeDetectorRef
  ) {}
  ngOnInit(): void {
    const id = this.route.snapshot.paramMap.get('id');
    let response = this.service.getEmployeeById(id);
    response.subscribe((data: any) => {
      this.employees = data;
      this.cdr.detectChanges(); // Detect changes after updating the employees data
    });
  }
  // update
  updateNow(id: number) {
    let response = this.service.putEmployee(id, this.employees);
    response.subscribe((data: any) => {
      this.message = 'Updated...! id: ' + id;
    });
  }
  // delete with changedetect function
  deleteEmployee(id: number) {
    let response = this.service.deletebyid(id);
    response.subscribe((data: any) => {
      this.employees = data;
      this.message = 'Deleted it...! id: ' + id;
    });
  }
}

```

```
this.cdr.detectChanges(); // Detect changes after updating the employees data  
setTimeout(() => {  
  this.router.navigate(['/employeeelist']);  
}, 1000); // Wait for 1 seconds before redirecting  
});  
}  
}
```