

Employee.component.html

```
<div class="container">  
  <h3>Add New Employee</h3>  
  <form [formGroup]="empRef" (ngSubmit)="storeEmp()">  
    <input  
      type="number"  
      placeholder="Enter Employee ID"  
      FormControlName="id"  
      ngModel  
      required  
    >&nbsp;  <br>  
    <input type="text" placeholder="Enter First Name"  
FormControlName="first_name">&nbsp;  <br>  
    <input type="text" placeholder="Enter Last Name"  
FormControlName="last_name">&nbsp;  <br>  
    <input type="text" placeholder="Enter Email"  
FormControlName="email"><br><br>  
    <input type="submit" value="ADD EMPLOYEE" class="btn btn-success">  
  </form>  
  <br>  
  <h3>Employee Details</h3><br>  
  <table class="table table-bordered">  
    <tr>  
      <th scope="col">Employee ID</th>  
      <th scope="col">First Name</th>  
      <th scope="col">Second Name</th>  
      <th scope="col">Email</th>  
      <th scope="col">View Details</th>  
    </tr>  
    <tr *ngFor="let employee of employees">  
      <th scope="row">{{employee.id}}</th>  
      <td>{{employee.first_name}}</td>  
      <td>{{employee.last_name}}</td>  
      <td>{{employee.email}}</td>  
      <td><input type="button" value="VIEW DETAILS"  
(click)="viewDetails(employee)"></td>  
    </tr>  
  </table>  
</div>
```

Employee.component.spec.ts

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

```

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

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

  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [EmployeeComponent]
    });
    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 { Router } from '@angular/router';
import { Employee } from '../employee';
import { EmployeeService } from '../employee.service';
import { FormGroup, FormControl } from "@angular/forms";
@Component({
  selector: 'app-employee',
  templateUrl: './employee.component.html',
  styleUrls: ['./employee.component.css']
})
export class EmployeeComponent implements OnInit {

  empRef = new FormGroup({
    id: new FormControl(),
    first_name: new FormControl(),
    last_name: new FormControl(),
    email: new FormControl()
  })

  employees:Array<Employee>=[]

  constructor(public es:EmployeeService,public router:Router) { }

  ngOnInit(): void {

```

```

        this.loadEmpDetails();
    }

    loadEmpDetails(){
        this.es.loadEmpData().subscribe({
            next:(data:any)=>this.employees=data,
            error:(error:any)=>console.log(error),
            complete:()=>console.log("Completed")
        })
    }

    storeEmp(){
        let employee = this.empRef.value;
        this.es.storeEmpData(employee).subscribe({
            next:(data:any)=>console.log(data),
            error:(error:any)=>console.log(error),
            complete:()=>this.loadEmpDetails()
        })
        this.empRef.reset();
    }

    viewDetails(employee:any){
        sessionStorage.setItem("empInfo",JSON.stringify(employee));
        this.router.navigate(["employee-operation"]);
    }
}

```

Employee-operation.component.html

```

<div>
    <h4 class="font-weight-bold">Details of {{employee.first_name}}</h4><br>
    <table class="table">
        <tr>
            <th>Employee ID</th>
            <th>First Name</th>
            <th>Second Name</th>
            <th>Email</th>
        </tr>
        <br>
        <tr>
            <th>{{employee.id}}</th>
            <td>{{employee.first_name}}</td>
            <td>{{employee.last_name}}</td>
            <td>{{employee.email}}</td>
        </tr>
    </table>

```

```

</table>
<hr><br>
<input type="button" class="btn btn-primary" value="View All Employee"
(click)="ViewAll()"/>&nbsp;
<input type="button" class="btn btn-secondary" value="Update Employee"
(click)="updateEmployee(employee)"/>&nbsp;
<input type="button" class="btn btn-danger" value="Delete Employee"
(click)="deleteEmployee(employee.id)"/>&nbsp;<br>
<input type="button" class="btn btn-link" value="Back To Home"
(click)="ViewAll()"/>&nbsp;
</div>

```

Employee-operation.component.spec.ts

```

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

import { EmployeeOperationComponent } from './employee-operation.component';

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

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

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

```

Employee-operation.component.ts

```

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

@Component({

```

```

    selector: 'app-employee-operation',
    templateUrl: './employee-operation.component.html',
    styleUrls: ['./employee-operation.component.css']
  })
  export class EmployeeOperationComponent implements OnInit {
    employee:any;
    constructor(public router:Router,public es:EmployeeService) { }

    ngOnInit(): void {
      let obj = sessionStorage.getItem("empInfo");
      if(obj != null){
        this.employee= JSON.parse(obj);
      }
    }

    ViewAll(){
      this.router.navigate(["employees"])
    }

    deleteEmployee(id:any){
      this.es.delete(id).subscribe({
        next:(result:any)=>console.log(result),
        error:(error:any)=>console.log(error),
        complete:()=>this.ViewAll()

      })
    }

    updateEmployee(employee:any){
      sessionStorage.setItem("empUpd",JSON.stringify(employee));
      this.router.navigate(["employee-update"]);
    }
  }

```

Login.component.html

```

<div class="container">
  <div class="row">
    <div class="col-md-6">
      <div class="card">
        <div class="text-center">
          <h1>Login</h1>
          <h6>Please enter email & password !!</h6>
        </div>
        <form [formGroup]="loginRef" (ngSubmit)="checkUser()">

```



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

Login.component.ts

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

  msg: string = "";
  loginRef = new FormGroup({
    email: new FormControl(),
    pass: new FormControl()
  });

  constructor(public router: Router) { }

  ngOnInit(): void {
  }

  checkUser() {
    let login = this.loginRef.value;
    if (login.email == "admin@gmail.com" && login.pass == "1234") {
      this.router.navigate(["employees"]);
    } else {
      this.msg = "Failure! Try Again!";
    }
  }
}
```

Update-employee.component.html

```
<div>
  <h4 class="font-weight-bold">Update Details of
  {{employee.first_name}}</h4><br>
```

```

    <form [formGroup]="empuRef" (ngSubmit)="updateEmpDetails()">
      Employee ID
      <input type="number" [(ngModel)]=employee.id formControlName="id"
readonly>
      First Name
      <input type="text" [(ngModel)]=employee.first_name
formControlName="first_name">
      Last Name
      <input type="text" [(ngModel)]=employee.last_name
formControlName="last_name">
      Email
      <input type="text" [(ngModel)]=employee.email
formControlName="email"><br><br>
      <input type="submit" class="btn btn-success" value="UPDATE EMPLOYEE" >
    </form>
    <input type="button" class="btn btn-link" value="Back To Home"
(click)="ViewAll()"/>&nbsp;  
</div>

```

Update-employee.component.spec.ts

```

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

import { UpdateEmployeeComponent } from './update-employee.component';

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

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

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

```


Update-employee.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl } from "@angular/forms";
import { Router } from '@angular/router';
import { EmployeeService } from '../employee.service';

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

  constructor(public es:EmployeeService,public router:Router) { }
  employee:any;

  empuRef = new FormGroup({
    id: new FormControl(),
    first_name: new FormControl(),
    last_name: new FormControl(),
    email: new FormControl()
  })

  ngOnInit(): void {
    let obj = sessionStorage.getItem("empUpd");
    if(obj != null){
      this.employee= JSON.parse(obj);
    }
  }

  ViewAll(){
    this.router.navigate(["employees"])
  }

  updateEmpDetails(){
    let employee = this.empuRef.value;
    let id = this.empuRef.value.id;
    this.es.updateEmp(id,employee).subscribe({
      next:(data:any)=>this.employee=data,
      error:(error:any)=>console.log(error),
      complete:()=>this.ViewAll()
    })
  }
}
```

App-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { EmployeeOperationComponent } from '../employee-operation/employee-operation.component';
import { EmployeeComponent } from '../employee/employee.component';
import { LoginComponent } from '../login/login.component';
import { UpdateEmployeeComponent } from '../update-employee/update-employee.component';

const routes: Routes = [
  { path: '', component: LoginComponent },
  { path: 'employees', component: EmployeeComponent },
  { path: 'employee-operation', component: EmployeeOperationComponent },
  { path: 'employee-update', component: UpdateEmployeeComponent },
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule],
})
export class AppRoutingModule {}
```

app.component.css

```
.container {
  position: absolute;
  background-image: linear-gradient(antiquewhite, white);
  background-repeat: no-repeat;
  background-repeat: no-repeat;
  background-size: cover;
  min-width: 100%;
  min-height: 100%;
  font-family: cursive;
}
```

App.component.html

```
<div class="container">
  <br>
  <h2 class="display-6,font-weight-bold">EVENT MANAGEMENT APP</h2><br>
  <router-outlet></router-outlet>
</div>
```

App.component.spec.ts

```
import { TestBed } from '@angular/core/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(() => TestBed.configureTestingModule({
    declarations: [AppComponent]
  }));

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'event-management-app'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('event-management-app');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.content
span')?.textContent).toContain('event-management-app app is running!');
  });
});
```

App.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Event-Management-App';
}
```

App.module.ts

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

```

import { HttpClientModule } from '@angular/common/http';
import { AppRoutingModuleModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { LoginComponent } from './login/login.component';
import { ReactiveFormsModule } from '@angular/forms';
import { FormsModule } from '@angular/forms';
import { EmployeeComponent } from './employee/employee.component';
import { UpdateEmployeeComponent } from './update-employee/update-employee.component';
import { EmployeeOperationComponent } from './employee-operation/employee-operation.component';

@NgModule({
  declarations: [
    AppComponent,
    LoginComponent,
    EmployeeComponent,
    UpdateEmployeeComponent,
    EmployeeOperationComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModuleModule,
    ReactiveFormsModule,
    HttpClientModule,
    FormsModule,
  ],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}

```

employee.service.spec.ts

```

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();
    });
  });
});
```

Employee.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { Employee } from './employee';
@Injectable({
  providedIn: 'root'
})
export class EmployeeService {

  constructor(public http:HttpClient) { }

  loadEmpData():Observable<Employee[]>{
    return this.http.get<Employee[]>("http://localhost:3000/employees")
  }

  storeEmpData(employee:any):Observable<Employee>{
    return
this.http.post<Employee>("http://localhost:3000/employees",employee)
  }

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

  updateEmp(id:any,employee:any):Observable<Employee>{
    return
this.http.put<Employee>("http://localhost:3000/employees/"+id,employee);
  }
}
```

Employee.spec.ts

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

describe('Employee', () => {
  it('should create an instance', () => {
    expect(new Employee(1, 'John', 'Doe', 'john@example.com')).toBeTruthy();
  });
});
```

```
});
```

Employee.ts

```
export class Employee {  
  constructor(public id:number,public first_name:string, public  
last_name:string,public email:string){  
  
  }  
}
```

Db.json

```
{  
  "employees": [  
    {  
      "id": 2,  
      "first_name": "Steve",  
      "last_name": "Palmer",  
      "email": "steve@codingthesmartway.com"  
    },  
    {  
      "id": 4,  
      "first_name": "Shri",  
      "last_name": "Ram",  
      "email": "Ramji@gmail.com"  
    },  
    {  
      "id": 5,  
      "first_name": "Jai Bajrnag",  
      "last_name": "bali",  
      "email": "Hanuman@gmail.com"  
    },  
    {  
      "id": 6,  
      "first_name": "shri",  
      "last_name": "Laxman",  
      "email": "Laxman@gmail.com"  
    }  
  ]  
}
```

Package.json

```
{  
  "name": "event-management-app",
```

```
"version": "0.0.0",
"scripts": {
  "ng": "ng",
  "start": "ng serve",
  "build": "ng build",
  "watch": "ng build --watch --configuration development",
  "test": "ng test",
  "json:server": "json-server --watch db.json"
},
"private": true,
"dependencies": {
  "@angular/animations": "^16.1.0",
  "@angular/common": "^16.1.0",
  "@angular/compiler": "^16.1.0",
  "@angular/core": "^16.1.0",
  "@angular/forms": "^16.1.0",
  "@angular/platform-browser": "^16.1.0",
  "@angular/platform-browser-dynamic": "^16.1.0",
  "@angular/router": "^16.1.0",
  "rxjs": "~7.8.0",
  "tslib": "^2.3.0",
  "zone.js": "~0.13.0"
},
"devDependencies": {
  "@angular-devkit/build-angular": "^16.1.4",
  "@angular/cli": "~16.1.4",
  "@angular/compiler-cli": "^16.1.0",
  "@types/jasmine": "~4.3.0",
  "jasmine-core": "~4.6.0",
  "karma": "~6.4.0",
  "karma-chrome-launcher": "~3.2.0",
  "karma-coverage": "~2.2.0",
  "karma-jasmine": "~5.1.0",
  "karma-jasmine-html-reporter": "~2.1.0",
  "typescript": "~5.1.3"
}
}
```