<div class="container">

  <table class="table">

    <thead>

      <tr>

        <th>ID</th>

        <th>FirstName</th>

        <th>LastName</th>

        <th>Email</th>

        <th>Actions</th>

      </tr>

    </thead>

    <tbody>

      <tr \*ngFor="let user of todoList">

        <td>{{user.id}}</td>

        <td>{{user.firstname}}</td>

        <td>{{user.lastname}}</td>

        <td>{{user.email}}</td>

        <td>

        <!--<button type="button" class="btn btn-link" (click)="editTodo(todo)">Edit</button>

          -->

          <button class="btn btn-primary" (click)="show()">Edit</button>

          <button type="button" class="btn btn-danger mx-3" (click)="deleteTodo(user)">Delete</button>

        </td>

      </tr>

    </tbody>

  </table>

  <!--<button type="button" class="btn btn-success" (click)="createTodo()">Create</button>-->

</div>

<div \*ngIf="flag" style="background-color: yellowgreen;">

  <div class="jumbotron">

    <div class="container">

        <div style="width: 500px; margin: auto;">

            <div class="submit-form">

                <div class="form-group">

                  <label for="name">FirstName: {{user.firstname}}</label>

                  <input class="form-control" id="firstname"   [(ngModel)]="user.firstname"  name="firstname"/>

                </div>

                <div class="form-group">

                    <label for="country">LastName</label>

                    <input class="form-control" id="lastname" [(ngModel)]="user.lastname"  name="lastname" />

                  </div>

                <div class="form-group">

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

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

                </div>

                <button (click)="update(user.id)"  class="btn btn-success">Update</button>

            </div>

          </div>

    </div>

</div>

</div>

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

import { CrudHttpService } from '../crud-http.service';

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

@Component({

  selector: 'app-admin',

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

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

})

export class AdminComponent implements OnInit {

  title = 'EventManagementApp';

  todoList:any = [];

  constructor(private crudHttpService: CrudHttpService,private router:Router){}

  ngOnInit(): void {

    this.listTodos();

  }

  listTodos(){

    this.crudHttpService.list().subscribe((response)=>{

      this.todoList = response;

    },(error=>{

    }));

  }

  createTodo(){

    let todo = {

      id: new Date().getTime(),

      firstname:`Nikunj ` ,

      lastname:'Soni',

      email:`nikunj@gmail.com`,

    }

    this.crudHttpService.create(todo).subscribe((response)=>{

      this.listTodos();

    },(error=>{

    }));

  }

  user = {

    id: '',

    firstname:'',

    lastname:'',

    email: '',

  };

  isBookAdded = true;

  flag=false;

  show(){

    this.flag=!this.flag;

  }

  update(newuser:any){

    let data = {

      id:this.user.id,

      firstname:this.user.firstname ,

      lastname:this.user.lastname,

      email:this.user.email,

    }

    this.crudHttpService.update(newuser.id,data).subscribe(

      result=>this.todoList=

    this.router.navigateByUrl('/admin')

    );

  }

  editTodo(todo: any){

    let data = {

      id: new Date().getTime(),

      title:`Some Todo`

    }

    this.crudHttpService.update(todo.id,data).subscribe((response)=>{

      this.listTodos();

    },(error=>{

    }));

  }

  deleteTodo(id: any){

    this.crudHttpService.delete(id).subscribe((response)=>{

      this.listTodos();

    },(error=>{

    }));

  }

}

<div class="jumbotron">

  <div class="container">

      <div style="width: 500px; margin: auto;">

          <div class="submit-form">

            <div \*ngIf="!isBookAdded">

              <div class="form-group">

                <label for="name">FirstName</label>

                <input class="form-control" id="firstname" required [(ngModel)]="user.firstname" name="firstname"/>

              </div>

              <div class="form-group">

                  <label for="country">LastName</label>

                  <input class="form-control" id="lastname" required [(ngModel)]="user.lastname" name="lastname" />

                </div>

              <div class="form-group">

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

                <input class="form-control" id="email" required [(ngModel)]="user.email" name="email" />

              </div>

               <button (click)="createTodo()" class="btn btn-success">Create</button>

            </div>

            <div \*ngIf="isBookAdded">

              <h3>Data added SUccessFully</h3>

            </div>

          </div>

        </div>

  </div>

</div>

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

import { CrudHttpService } from '../crud-http.service';

import { User } from '../user';

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

@Component({

  selector: 'app-user',

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

  styleUrls: ['./user.component.css']

})

export class UserComponent implements OnInit {

  title = 'EventManagementApp';

  todo={id:"",firstname:"",lastname:"",email:""}

  todoList:any = [];

  constructor(private crudHttpService: CrudHttpService,private router:Router) { }

  ngOnInit(): void {

  }

  //user:User= new User();

  isBookAdded = false;

  user = {

    id: '',

    firstname:'',

    lastname:'',

    email: '',

  };

  createTodo(){

    let data = {

      //id: new Date().getTime(),

      firstname:this.user.firstname ,

      lastname:this.user.lastname,

      email:this.user.email,

    }

     this.crudHttpService.create(data).subscribe(result=>this.todoList=

    this.router.navigateByUrl('/admin'));

  }

}

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

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

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

import { UserComponent } from './user/user.component';

const routes: Routes = [

  {path:"admin",component:AdminComponent},

  {path:"user",component:UserComponent},

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

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

  <a href="#" class="navbar-brand">Personal Website</a>

  <ul class="navbar-nav">

    <li class="nav-item">

      <a href="#" class="nav-link" routerLink="user">User</a>

    </li>

    <li class="nav-item">

      <a href="#" class="nav-link" routerLink="admin">Admin</a>

    </li>

  </ul>

</nav>

<router-outlet></router-outlet>

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'EventManagementApp';

}

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

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

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

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

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

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

import { UserComponent } from './user/user.component';

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

@NgModule({

  declarations: [

    AppComponent,

    AdminComponent,

    UserComponent,

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    HttpClientModule,

    FormsModule,

    ReactiveFormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

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

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

import { catchError, Observable, throwError } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class CrudHttpService {

  apiUrl: string = 'http://localhost:3000/todos';

  headers = new HttpHeaders().set('Content-Type', 'application/json');

  constructor(private http: HttpClient) { }

  // Create

  create(data: any): Observable<any> {

    let API\_URL = `${this.apiUrl}`;

    return this.http.post(API\_URL, data)

      .pipe(

        catchError(this.handleError)

      )

  }

  // Read

  list() {

    return this.http.get(`${this.apiUrl}`);

  }

  // Update

  update(id: any, data: any): Observable<any> {

    let API\_URL = `${this.apiUrl}/${id}`;

    return this.http.put(API\_URL, data, { headers: this.headers }).pipe(

      catchError(this.handleError)

    )

  }

  // Delete

  delete(id: any): Observable<any> {

    var API\_URL = `${this.apiUrl}/${id}`;

    return this.http.delete(API\_URL).pipe(

      catchError(this.handleError)

    )

  }

  // Handle API errors

  handleError(error: HttpErrorResponse) {

    if (error.error instanceof ErrorEvent) {

      console.error('An error occurred:', error.error.message);

    } else {

      console.error(

        `Backend returned code ${error.status}, ` +

        `body was: ${error.error}`);

    }

    return throwError(

      'Something bad happened; please try again later.');

  };

}

{

  "todos": [

    {

      "firstname": "kavi",

      "lastname": "kavin",

      "email": "Kavi@gmail.com",

      "id": 1

    },

    {

      "firstname": "Mithu",

      "lastname": "G",

      "email": "mg@gmail.com",

      "id": 2

    },

    {

      "firstname": "Anu",

      "lastname": "Radha",

      "email": "anu@gmail.com",

      "id": 3

    },

    {

      "firstname": "vino",

      "lastname": "kavi",

      "email": "vino@gmail.com",

      "id": 4

    },

    {

      "firstname": "jack",

      "lastname": "rose",

      "email": "jack@gmail.com",

      "id": 5

    },

    {

      "firstname": "sri",

      "lastname": "bhavaan",

      "email": "sri@gmail.com",

      "id": 6

    },

    {

      "firstname": "hrithik",

      "lastname": "sai",

      "email": "sai@gmail.com",

      "id": 7

    },

    {

      "firstname": "heyuser",

      "lastname": "use",

      "email": "use@gmail.com",

      "id": 8

    },

    {

      "firstname": "demo",

      "lastname": "demo",

      "email": "demo@123",

      "id": 9

    },

    {

      "firstname": "demo",

      "lastname": "demo",

      "email": "demo@123",

      "id": 10

    },

    {

      "firstname": "Hey",

      "lastname": "user",

      "email": "hey@gmail.com",

      "id": 11

    }

  ]

}

export class User{

    id:any;

    firstname:string;

    lastname:string;

    email:string;

}

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>EventManagementApp</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@4.6.2/dist/css/bootstrap.min.css" integrity="sha384-xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N" crossorigin="anonymous">

</head>

<body>

  <app-root></app-root>

  <script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D6OP9dO5Vg3Q9ct" crossorigin="anonymous"></script>

</body>

</html>

{

  "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": "^14.2.0",

    "@angular/common": "^14.2.0",

    "@angular/compiler": "^14.2.0",

    "@angular/core": "^14.2.0",

    "@angular/forms": "^14.2.0",

    "@angular/platform-browser": "^14.2.0",

    "@angular/platform-browser-dynamic": "^14.2.0",

    "@angular/router": "^14.2.0",

    "rxjs": "~7.5.0",

    "tslib": "^2.3.0",

    "zone.js": "~0.11.4"

  },

  "devDependencies": {

    "@angular-devkit/build-angular": "^14.2.2",

    "@angular/cli": "~14.2.2",

    "@angular/compiler-cli": "^14.2.0",

    "@types/jasmine": "~4.0.0",

    "jasmine-core": "~4.3.0",

    "karma": "~6.4.0",

    "karma-chrome-launcher": "~3.1.0",

    "karma-coverage": "~2.2.0",

    "karma-jasmine": "~5.1.0",

    "karma-jasmine-html-reporter": "~2.0.0",

    "typescript": "~4.7.2"

  }

}