Admin.component.html

<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>

Admin.component.ts

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:`Nive ` ,

      lastname:'Kamal',

      email:`nivekamal@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=>{

    }));

  }

}

User.component.html

<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>

User.component.ts

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'));

  }

}

App-routing.module.ts

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 { }

app.component.html

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

    <a href="#" class="navbar-brand">EVENT MANAGEMENT </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>

App.module.ts

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 { }

crud-http.service.ts

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.');

  };

}