**APP**

app.module.ts

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

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

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

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

import { ListProductsComponent } from './list-products/list-products.component';

import { ProductFormComponent } from './product-form/product-form.component';

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

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

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

import { ProductService } from './product.service';

import { FindProductComponent } from './find-product/find-product.component';

import { InMemoryWebApiModule } from 'angular-in-memory-web-api';

import { DataService } from './DataService';

import { ToastrModule } from 'ngx-toastr';

import { ShopCartComponent } from './shop-cart/shop-cart.component';

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

import { BillComponent } from './bill/bill.component';

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

import { RouteGaurdService } from './route-gaurd.service';

const menus:Routes=[

  {path: '', component: LoginComponent},

  {path: 'form', component: ProductFormComponent, canActivate : [RouteGaurdService]},

  {path: 'find', component: FindProductComponent, canActivate: [RouteGaurdService]},

  {path: 'list', component: ListProductsComponent, canActivate: [RouteGaurdService]},

  {path: 'cart', component: ShopCartComponent, canActivate: [RouteGaurdService]},

  {path: 'bill', component: BillComponent, canActivate: [RouteGaurdService]},

  {path: '\*\*', component: LoginComponent}

]

@NgModule({

  declarations: [

    AppComponent,

    ListProductsComponent,

    ProductFormComponent,

    FindProductComponent,

    ShopCartComponent,

    BillComponent,

    LoginComponent

  ],

  imports: [

    BrowserModule,

    BrowserAnimationsModule,

    AppRoutingModule,

    ReactiveFormsModule,

    HttpClientModule,

    InMemoryWebApiModule.forRoot(DataService),

    RouterModule.forRoot(menus),

    ToastrModule.forRoot()

  ],

  providers: [ProductService],

  bootstrap: [AppComponent]

})

export class AppModule { }

app.component.ts

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

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

import { ToastrService } from 'ngx-toastr';

import { DatashareService } from './datashare.service';

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'router-demo';

  constructor(private router:Router,public dataShare:DatashareService,private toastr:ToastrService){

  }

  goToCart(){

    this.router.navigate(["cart"]);

  }

  logout(){

    this.dataShare.setIsUserLoggedIn(false);

    this.dataShare.setUserType('');

    this.toastr.success('Loggedout Successfully');

    this.router.navigate(['']);

  }

}

app.component.css

.but{

    margin-top:10px;

}

app.component.html

<div class="container">

  <div class="alert alert-info text-center mt-3 ">

    <h1 >Online Food Order App

      <button \*ngIf="dataShare.getIsUserLoggedIn()" type="button" class="btn btn-success float-right but" (click)="logout()">Logout</button>

      <button \*ngIf="dataShare.getIsUserLoggedIn() && dataShare.getUserType()=='U'" type="button" class="btn btn-success float-right but mr-3" (click)="goToCart()">Go To Cart</button>

    </h1>

  </div>

  <div class="row">

    <div class="col-2 " \*ngIf="dataShare.getIsUserLoggedIn()">

      <ul class="nav flex-column">

        <li class="nav-item">

          <a class="nav-link" routerLink='/find'>Find</a>

        </li>

        <li class="nav-item">

          <a class="nav-link" \*ngIf="dataShare.getUserType()=='A'" routerLink='/form'>Add / Update Item</a>

        </li>

        <li class="nav-item">

          <a class="nav-link" routerLink='/list'>Menu</a>

        </li>

      </ul>

    </div>

    <div class="col-10">

     <router-outlet>

     </router-outlet>

    </div>

  </div>

</div>

**BILL**

bill.component.ts

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

import { DatashareService } from '../datashare.service';

@Component({

  selector: 'app-bill',

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

  styleUrls: ['./bill.component.css']

})

export class BillComponent implements OnInit {

  constructor(private dataShare: DatashareService) { }

  public bill:number = 0;

  ngOnInit(): void {

    this.bill = this.dataShare.getBillAmonut();

  }

}

bill.component.html

<div class="card">

<div class="card card-header">

<h5 class="text-center text-success">Thanks for ordering from our website. Your bill amount is <b>{{bill}}</b> paid successfully order is on the way</h5>

<img src="assets/food emoji.jpg">

</div>

</div>

find-product.component.ts

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

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

import { ProductService } from '../product.service';

@Component({

  selector: 'app-find-product',

  templateUrl: './find-product.component.html',

  styleUrls: ['./find-product.component.css']

})

export class FindProductComponent implements OnInit {

  searchForm:FormGroup;

  product:any;

  handleSubmit(){

    this.ps.findById(this.searchForm.get('id').value)

    .subscribe(productResponse=>this.product=productResponse)

    //this.ps.findById(this.)

  }

  constructor(private fb:FormBuilder,private ps:ProductService) { }

  ngOnInit(): void {

    this.searchForm=this.fb.group({

      id:['']

    })

  }

}

find-product.component.html

<div class="col-6">

<div class="card">

    <div class="card-header">Product Search Page</div>

    <div class="card-body">

   <form [formGroup]="searchForm" autocomplete="off" (ngSubmit)="handleSubmit()">

            <div class="form-group">

                <input placeholder="Product Id" class="form-control" formControlName="id" name="id"/>

            </div>

            <div class="text-center">

                <button class="btn btn-primary">Find</button>

            </div>

        </form>

    </div>

</div>

<hr>

<div \*ngIf="product">

    <table class="table table-bordered">

        <thead>

            <tr>

                <th>Product ID</th>

                <th>Name</th>

                <th>Price</th>

                <th>Description</th>

            </tr>

        </thead>

        <tbody>

            <tr>

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

                <td>{{product.name}}</td>

                <td>{{product.price}}</td>

                <td>{{product.description}}</td>

            </tr>

        </tbody>

    </table>

</div>

</div>

list-products.component.html

    <div class="card">

        <div class="card-header">Food Menu</div>

        <div class="card-body">

            <table class="table table-bordered">

                <thead>

                    <tr>

                        <th>Product ID</th>

                        <th>Name</th>

                        <th>Price</th>

                        <th>Description</th>

                        <th>Number Of Plates</th>

                        <th \*ngIf="dataShare.getUserType() == 'A'">Delete Item</th>

                    </tr>

                </thead>

                <tbody>

                    <tr \*ngFor="let product of products">

                        <td>

                           {{product.id}}

                        </td>

                        <td>{{product.name}}</td>

                        <td>{{product.price}}</td>

                        <td>{{product.description}}</td>

                        <td><select (change)="onChange($event,product.id)" name="items" id="foods">

                            <option value=0>Select</option>

                            <option value=1>1</option>

                            <option value=2>2</option>

                            <option value=3>3</option>

                            <option value=4>4</option>

                            <option value=5>5</option>

                            <option value=6>6</option>

                          </select></td>

                        <td \*ngIf="dataShare.getUserType() == 'A'"><button type="button" class="btn btn-primary" (click)="deleteItem(product)">Delete</button></td>

                    </tr>

                </tbody>

            </table>

            <button type="button" \*ngIf="dataShare.getUserType()=='U'" class="btn btn-primary float-right col-2" (click)="onClick()">Add to cart</button>

        </div>

    </div>

</div>

list-products.component.ts

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

import { DatashareService } from '../datashare.service';

import { food } from '../food';

import { ProductService } from '../product.service';

import { ToastrService } from 'ngx-toastr';

@Component({

  selector: 'app-product',

  templateUrl: './list-products.component.html',

  styleUrls: ['./list-products.component.css']

})

export class ListProductsComponent  implements OnInit {

    products:any

    cartOrder:food[] = [];

    constructor(

        private ps:ProductService,

        public dataShare:DatashareService,

        private toastr:ToastrService) { }

    renderProducts(){

      this.ps.findAll().subscribe(productsResult =>

        {

         this.products = productsResult

        });

    }

    ngOnInit(): void {

      this.renderProducts();

    }

    onChange(event,id){

        this.products[id-1].numberOfPlates=event.target.value;

    }

    onClick(){

        for(let i of this.products){

            if(i.numberOfPlates>0){

                this.cartOrder.push(i);

            }

        }

        this.dataShare.setFood(this.cartOrder);

        this.toastr.success('Item added to cart successfully');

    }

    deleteItem(product:food):void{

      this.products = this.products.filter(p => p !== product);

      this.ps.deleteProduct(product).subscribe();

    }

}

login.component.ts

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

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

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

import { ToastrService } from 'ngx-toastr';

import { DatashareService } from '../datashare.service';

import { users } from '../user';

@Component({

  selector: 'app-login',

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

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

})

export class LoginComponent implements OnInit {

  constructor(

    private fb:FormBuilder,

    private router:Router,

    private toastr:ToastrService,

    private dataShare:DatashareService

    ) { }

  uname:string='';

  pword:string='';

  flag:boolean=true;

  userDetails = users;

  loginForm:FormGroup;

  ngOnInit(): void {

    this.loginForm=this.fb.group(

      {

        uname:[''],

        pword:['']

      }

    )

    this.dataShare.setIsUserLoggedIn(false);

    this.dataShare.setUserType('');

  }

  onLogin(){

    for(let i of this.userDetails){

      if(i.name == this.loginForm.get('uname').value && i.password == this.loginForm.get('pword').value){

        this.toastr.success("Login successful");

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

        this.flag=false;

        this.dataShare.setIsUserLoggedIn(true);

        this.dataShare.setUserType(i.userType)

      }

    }

    if(this.flag){

      this.toastr.error("Login failed please enter valid credentials");

    }

  }

}

login.component.html

<div class="card login">

    <div class=" card card-header"> Login Page</div>

    <div class=" card card-body">

        <form [formGroup]="loginForm" autocomplete="off" (ngSubmit)="onLogin()">

            <div class="form-group">

                Username: <input type="text" name="username" class="form-control inp" id="uname" placeholder="Username" formControlName="uname" >

            </div>

            <div class="form-group">

                Password: <input type="password" name="password" class="form-control" id="pword" placeholder="Password" formControlName="pword">

            </div>

            <div class="form-group text-center" >

                <button type="submit" class="btn btn-primary btn-block" [disabled]="loginForm.get('uname').value == '' || loginForm.get('pword').value == ''" >Login</button>

            </div>

        </form>

    </div>

</div>

login.component.css

.login{

    margin-left:100px;

}

.inp{

    text-transform:uppercase

}

product-form.component.html

<div class="col-6">

    <div class="card">

        <div class="card-header">Product Form</div>

        <div class="card-body">

        <form [formGroup]="productForm" autocomplete="off" (ngSubmit)="handleSubmit()">

          <div class="form-group">

            <input class="form-control" name="id" placeholder="id" (keyup)="onChange($event)"  formControlName="id" [value]="newId"/>

          </div>

          <div class="form-group">

            <input class="form-control" name="name" placeholder="name" formControlName="name"/>

          </div>

          <div class="form-group">

            <input class="form-control" name="price" placeholder="price" formControlName="price"/>

          </div>

          <div class="form-group">

            <textarea class="form-control" name="description" placeholder="description" formControlName="description">

            </textarea>

          </div>

          <div class="form-group text-center">

            <button class="btn btn-primary btn-space" [disabled]="">{{btn}}</button>

            <button class="btn btn-warning btn-space">Reset</button>

            <button class="btn btn-danger btn-space">Cancel</button>

          </div>

        </form>

      </div>

</div>

product-form.component.ts

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

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

import { ProductService } from '../product.service';

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

import { ToastrService } from 'ngx-toastr';

@Component({

  selector: 'app-product-form',

  templateUrl: './product-form.component.html',

  styleUrls: ['./product-form.component.css']

})

export class ProductFormComponent implements OnInit {

  products:any

  btn:string = 'Save'

  // steps:

  // 1 - Define FormGroup element name

  productForm: FormGroup;

  public food: any;

  newId: number;

  saveButton: boolean=false;

  // step2: define form builder object

  constructor(

    private fb:FormBuilder,

    private ps:ProductService,

    private router:Router,

    private toastr:ToastrService) {}

  // step3: define / add form elements to the formGroup obj

  ngOnInit(){

    this.ps.findAll().subscribe(res=>{

      this.food=res;

      this.newId=this.food.length+1;

      if(this.food.length == 100){

        this.saveButton=true;

        this.toastr.info('You have crossed the limit of adding the items.')

      }

    });

    this.productForm=this.fb.group(

      {

        id:[],

        name:[''],

        price:[0.0],

        description:['']

      }

    );

  }

  onChange(event){

    let change = event.target.value;

    this.btn = change < this.newId ?  'Update' : 'Save';

    if(change > this.food.length+1){

    setTimeout(()=>{

      this.toastr.info('Available latest id is' + (this.food.length+1));

    },1000);

   }

  }

  // step 4: submit form function

  handleSubmit(){

    this.productForm.controls['id'].setValue(parseInt(this.productForm.controls['id'].value));

    if(this.productForm.controls['id'].value < this.newId){

      this.ps.updateProduct(this.productForm.value).subscribe(res=>{

      });

    }else{

     this.ps.saveProduct(this.productForm.value)

    .subscribe(product =>{

   })}

    this.router.navigate(['list']);

  }

}

shop-cart.component.html

<div>

    <div class="card">

        <div class="card-header">Cart </div>

        <div class="card-body">

            <table class="table table-bordered">

                <thead>

                    <tr>

                        <th>Number Of Plates</th>

                        <th>Product ID</th>

                        <th>Name</th>

                        <th>Price</th>

                        <th>Description</th>

                        <th>Amount</th>

                    </tr>

                </thead>

                <tbody>

                    <tr \*ngFor="let product of products">

                        <td>

                            {{product.numberOfPlates}}

                        </td>

                        <td>

                           {{product.id}}

                        </td>

                        <td>{{product.name}}</td>

                        <td>{{product.price}}</td>

                        <td>{{product.description}}</td>

                        <td>{{(product.price \* product.numberOfPlates)}}</td>

                    </tr>

                    <tr>

                        <td colspan="5" class="text-center">Total Amount</td>

                        <td>{{billAmount}}</td>

                    </tr>

                </tbody>

            </table>

            <button type="button" \*ngIf="dataShare.getUserType()=='U'" (click)="onPurchase()" class="btn btn-primary float-right" [disabled]="billAmount == 0">Pay Bill</button>

        </div>

    </div>

</div>

shop-cart.component.ts

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

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

import { DatashareService } from '../datashare.service';

@Component({

  selector: 'app-shop-cart',

  templateUrl: './shop-cart.component.html',

  styleUrls: ['./shop-cart.component.css']

})

export class ShopCartComponent implements OnInit {

  constructor(public dataShare:DatashareService,private router:Router) { }

  public products:any[];

  public billAmount:number = 0;

  ngOnInit(): void {

    this.products= this.dataShare.getFood();

    if(this.products !=null){

    for(let i of this.products){

      this.billAmount =  this.billAmount + (i.price \* i.numberOfPlates);

    }

    this.dataShare.setBillAmount(this.billAmount);

  }

 }

  onPurchase(){

    this.router.navigate(['bill'])

  }

}

food.ts

export class food{

    id:number;

    name:string;

    price:number;

    description:string;

    numberOfPlates:number;

}

DataService.ts

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

import { InMemoryDbService } from 'angular-in-memory-web-api';

@Injectable({

    providedIn: 'root'

  })

  export class DataService implements InMemoryDbService{

    constructor() { }

    createDb(){

     let  foods =  [

      {id:1, name:"Idly", price:20, description:"Four piece Idly with Chutney and Sambar", numberOfPlates:0},

      {id:2, name:"Pongal", price:30, description:"One cup Pongal with sambar", numberOfPlates:0},

      {id:3, name:"Puri", price:50, description:"Two Piece puri with kurma", numberOfPlates:0},

      {id:4, name:"Vada", price:30, description:"Three piece vada with Chutney and Sambar", numberOfPlates:0},

      {id:5, name:"Parotta", price:40, description:"Two piece Parotta with Kurma", numberOfPlates:0},

      {id:6, name:"Dosa", price:30, description:"Plain Dosa with Red and White chutney", numberOfPlates:0},

      {id:7, name:"Masala Dosa", price:60, description:"Masala dosa with red chutney", numberOfPlates:0},

      {id:8, name:"Onion Dosa", price:60, description:"Onion dosa with chutney", numberOfPlates:0},

      {id:9, name:"Ghee Karam Roast", price:70, description:"Nellore special Ghee Karam Roast", numberOfPlates:0},

      {id:10, name:"Upma Pesarattu", price:50, description:"Upma Pesarattu with allam chutney", numberOfPlates:0}

     ];

     return {foods};

    }

  }

user.ts

export const users=[

    //General Users

    {id:1, name:'testuser', password:'Test123', userType:'U'},

    {id:2, name:'tempuser', password:'Test123', userType:'U'},

    {id:3, name:'dummyuser', password:'Test123', userType:'U'},

    //Admin Users

    {id:4, name:'admin1', password:'Strict123', userType:'A'},

    {id:5, name:'admin2', password:'Strict123', userType:'A'},

]

datashare.service.ts

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

import { food } from './food';

@Injectable({

  providedIn: 'root'

})

export class DatashareService {

  constructor() { }

  private cart:food[];

  private billAmount:number=0;

  private isUserLoggedIn:boolean = false;

  private userType:string='';  // U || A

  getFood(){

    return this.cart;

  }

  setFood(cart){

    this.cart=cart;

  }

  getBillAmonut(){

    return this.billAmount;

  }

  setBillAmount(billAmount){

    this.billAmount=billAmount;

  }

  getIsUserLoggedIn(){

    return this.isUserLoggedIn;

  }

  setIsUserLoggedIn(isUserLoggedIn){

    this.isUserLoggedIn=isUserLoggedIn;

  }

  getUserType(){

    return this.userType;

  }

 setUserType(userType){

    this.userType=userType;

  }

}

datashare.service.spec.ts

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

import { DatashareService } from './datashare.service';

describe('DatashareService', () => {

  let service: DatashareService;

  beforeEach(() => {

    TestBed.configureTestingModule({});

    service = TestBed.inject(DatashareService);

  });

  it('should be created', () => {

    expect(service).toBeTruthy();

  });

});

product.service.ts

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

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

import { food } from './food';

import {  throwError } from 'rxjs';

import {  catchError } from 'rxjs/operators';

@Injectable({

  providedIn: 'root'

})

export class ProductService {

  baseUrl="api/foods"

  products:any;

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

  httpOptions = {

    headers: this.headers

  };

  constructor(private http: HttpClient) { }

  private handleError(error: any) {

    console.log(error);

    return throwError(error);

  }

  findAll() {

    return this.http.get<food[]>(this.baseUrl)

  }

  saveProduct(product){

    return this.http.post(this.baseUrl,product);

  }

  updateProduct(product){

    return this.http.put(this.baseUrl,product);

  }

  findById(id:number){

    let test=this.http.get(this.baseUrl+`/${id}`)

    return test;

  }

  deleteProduct(food:food){

     const id= food.id;

    // const url = `${this.baseUrl}/${id}`;

    // this.http.delete(url);

    const url = `${this.baseUrl}/${id}`;

    return this.http.delete<food>(url, this.httpOptions).pipe(

      catchError(this.handleError)

    );

  }

product.service.spec.ts

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

import { ProductService } from './product.service';

describe('ProductService', () => {

  let service: ProductService;

  beforeEach(() => {

    TestBed.configureTestingModule({});

    service = TestBed.inject(ProductService);

  });

  it('should be created', () => {

    expect(service).toBeTruthy();

  });

});

route-guard.service.ts

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

import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, Router } from '@angular/router';

import { DatashareService } from './datashare.service';

@Injectable({

  providedIn: 'root'

})

export class RouteGaurdService implements CanActivate{

  constructor(private service:DatashareService, private router:Router) { }

  canActivate(route: ActivatedRouteSnapshot, state:RouterStateSnapshot) {

    if(this.service.getIsUserLoggedIn())

    return true;

    else{

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

    return false;

    }

  }

}

route-guard.service.spec.ts

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

import { RouteGaurdService } from './route-gaurd.service';

describe('RouteGaurdService', () => {

  let service: RouteGaurdService;

  beforeEach(() => {

    TestBed.configureTestingModule({});

    service = TestBed.inject(RouteGaurdService);

  });

  it('should be created', () => {

    expect(service).toBeTruthy();

  });

});

styles.css

/\* You can add global styles to this file, and also import other style files \*/

@import '../node\_modules/bootstrap/dist/css/bootstrap.css';

.btn-space{

    margin:6px;

}

main.ts

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

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

import { AppModule } from './app/app.module';

import { environment } from './environments/environment';

if (environment.production) {

  enableProdMode();

}

platformBrowserDynamic().bootstrapModule(AppModule)

  .catch(err => console.error(err));

index.html

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Online Food</title>

  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

<body>

  <app-root></app-root>

</body>

</html>