**Problem Objective**

Create a dynamic and responsive Java e-healthcare web application for ordering medicines of different categories.

**Source Code**

Backend

Admin Controller

package com.org.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.org.entity.Admin;

import com.org.service.AdminService;

@CrossOrigin(origins="http://localhost:4200")

@RestController

@RequestMapping(path="/medicare")

public class AdminController {

@Autowired

private AdminService adminService;

@PostMapping("/admin/insert")

public Admin addNewAdmin(@RequestBody Admin admin)

{

return adminService.addAdmin(admin);

}

@GetMapping("/admin/list")

public List<Admin> listAdmin()

{

return adminService.getAllAdmin();

}

@PutMapping("/admin/updatepwd")

public Admin updatePwd(@RequestBody Admin admin)

{

return adminService.updateAdmin(admin);

}

}

User Controller

package com.org.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.org.entity.Admin;

import com.org.entity.User;

import com.org.service.UserService;

@CrossOrigin(origins="http://localhost:4200")

@RestController

@RequestMapping(path="/medicare")

public class UserController {

@Autowired

private UserService userService;

@PostMapping("/user/insert")

public User addNewUser(@RequestBody User user)

{

return userService.addUser(user);

}

@GetMapping("/user/list")

public List<User> listUser()

{

return userService.getAllUser();

}

@PutMapping("/user/updatepwd")

public User updatePwd(@RequestBody User user)

{

return userService.updateUser(user);

}

}

Medicines Controller

package com.org.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.org.entity.Medicines;

import com.org.service.MedicinesService;

@CrossOrigin(origins="http://localhost:4200")

@RestController

@RequestMapping(path="/medicare")

public class MedicinesController {

@Autowired

MedicinesService medicinesService;

@PostMapping("/medicines/insert")

public Medicines addNewMedicine(@RequestBody Medicines medicines)

{

return medicinesService.addMedicine(medicines);

}

@GetMapping("/medicines/list")

public List<Medicines> listMedicines()

{

return medicinesService.getAllMedicines();

}

@GetMapping("/medicines/list/enable")

public List<Medicines> listMedicinesEnabled()

{

return medicinesService.getAllMedicinesEnabled();

}

@GetMapping("/medicines/list/category")

public List<Medicines> listMedicinesbyCat()

{

return medicinesService.getAllMedicinesbyCat();

}

@GetMapping("/medicines/list/price")

public List<Medicines> listMedicinesbyPrice()

{

return medicinesService.getAllMedicinesbyPrice();

}

@GetMapping("/medicines/list/date")

public List<Medicines> listMedicinesbyDate()

{

return medicinesService.getAllMedicinesbyDate();

}

@PutMapping("/medicines/edit")

public Medicines editMedicines(@RequestBody Medicines medicines)

{

return medicinesService.editMed(medicines);

}

@PutMapping("/medicines/search")

public List<Medicines> searchMedicines(@RequestBody Medicines medicines)

{

return medicinesService.searchforMeds(medicines);

}

@PutMapping("/medicines/enable")

public Medicines enableMedicine(@RequestBody Medicines medicines)

{

return medicinesService.enableMed(medicines);

}

@PutMapping("/medicines/disable")

public Medicines disableMedicine(@RequestBody Medicines medicines)

{

return medicinesService.disableMed(medicines);

}

@PutMapping("/medicines/delete")

public void deleteMedicine(@RequestBody Medicines medicines)

{

medicinesService.deleteMed(medicines);

System.out.println("Medicine Deleted");

}

}

Admin

package com.org.entity;

import java.io.Serializable;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

@JsonIgnoreProperties({"hibernateLazyInitializer"})

@Entity

public class Admin implements Serializable {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(nullable = false)

private int aid;

private String aname;

private String aemail;

private String apassword;

public Admin() {

}

public Admin(int aid, String aname, String aemail, String apassword) {

super();

this.aid = aid;

this.aname = aname;

this.aemail = aemail;

this.apassword = apassword;

}

public int getAid() {

return aid;

}

public void setAid(int aid) {

this.aid = aid;

}

public String getAname() {

return aname;

}

public void setAname(String aname) {

this.aname = aname;

}

public String getAemail() {

return aemail;

}

public void setAemail(String aemail) {

this.aemail = aemail;

}

public String getApassword() {

return apassword;

}

public void setApassword(String apassword) {

this.apassword = apassword;

}

}

Admin Repo

package com.org.entity;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.stereotype.Repository;

@Repository

public interface AdminRepo extends JpaRepository<Admin, Integer> {

@Query(value=("select \* from admin where admin.aname= ? and admin.apassword= ?"), nativeQuery = true)

public Admin findAdmin(String aname, String apassword);

@Query(value=("select \* from admin where admin.aname= ? "), nativeQuery = true)

public Admin searchAdmin(String aname);

}

User

package com.org.entity;

import java.io.Serializable;

import java.util.ArrayList;

import java.util.List;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToMany;

import com.fasterxml.jackson.annotation.JsonIgnore;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

@JsonIgnoreProperties({"hibernateLazyInitializer"})

@Entity

public class User implements Serializable {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(nullable = false)

private int uid;

private String uname;

private int uage;

private String uemail;

private String upassword;

@ManyToMany(mappedBy = "users")

@JsonIgnore

private List<Medicines> meds=new ArrayList<>();

public User() {

}

public User(int uid, String uname, int uage, String uemail, String upassword) {

super();

this.uid = uid;

this.uname = uname;

this.uage = uage;

this.uemail = uemail;

this.upassword = upassword;

this.meds = meds;

}

public int getUid() {

return uid;

}

public void setUid(int uid) {

this.uid = uid;

}

public String getUname() {

return uname;

}

public void setUname(String uname) {

this.uname = uname;

}

public int getUage() {

return uage;

}

public void setUage(int uage) {

this.uage = uage;

}

public String getUemail() {

return uemail;

}

public void setUemail(String uemail) {

this.uemail = uemail;

}

public String getUpassword() {

return upassword;

}

public void setUpassword(String upassword) {

this.upassword = upassword;

}

public List<Medicines> getMeds() {

return meds;

}

public void addMeds(Medicines med) {

this.meds.add(med);

}

}

User Repo

package com.org.entity;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.stereotype.Repository;

@Repository

public interface UserRepo extends JpaRepository<User, Integer> {

@Query(value=("select \* from user where user.uname= ? and user.upassword= ? "), nativeQuery = true)

public User findUser(String uname, String upassword);

@Query(value=("select \* from user where user.uname= ? "), nativeQuery = true)

public User searchUser(String uname);

}

Medicines

package com.org.entity;

import java.io.Serializable;

import java.util.ArrayList;

import java.util.List;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.JoinTable;

import javax.persistence.ManyToMany;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

@JsonIgnoreProperties({"hibernateLazyInitializer"})

@Entity

public class Medicines implements Serializable {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(nullable = false)

private int mid;

private String mname;

private int mprice;

private String mdesc;

private String mcategory;

private String expiredate;

private int mcontrol;

@ManyToMany

@JoinTable(name="orders",

joinColumns = @JoinColumn(name = "mid"),

inverseJoinColumns = @JoinColumn(name = "uid"))

private List<User> users=new ArrayList<>();

public Medicines() {

}

public Medicines(int mid, String mname, int mprice, String mdesc, String mcategory, String expiredate,

int mcontrol) {

super();

this.mid = mid;

this.mname = mname;

this.mprice = mprice;

this.mdesc = mdesc;

this.mcategory = mcategory;

this.expiredate = expiredate;

this.mcontrol = mcontrol;

this.users = users;

}

public int getMid() {

return mid;

}

public void setMid(int mid) {

this.mid = mid;

}

public String getMname() {

return mname;

}

public void setMname(String mname) {

this.mname = mname;

}

public int getMprice() {

return mprice;

}

public void setMprice(int mprice) {

this.mprice = mprice;

}

public String getMdesc() {

return mdesc;

}

public void setMdesc(String mdesc) {

this.mdesc = mdesc;

}

public String getMcategory() {

return mcategory;

}

public void setMcategory(String mcategory) {

this.mcategory = mcategory;

}

public String getExpiredate() {

return expiredate;

}

public void setExpiredate(String expiredate) {

this.expiredate = expiredate;

}

public int getMcontrol() {

return mcontrol;

}

public void setMcontrol(int mcontrol) {

this.mcontrol = mcontrol;

}

public List<User> getUsers() {

return users;

}

public void addUsers(User user) {

this.users.add(user);

}

}

Medicines Repo

package com.org.entity;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.stereotype.Repository;

@Repository

public interface MedicinesRepo extends JpaRepository<Medicines, Integer> {

@Query(value=("select \* from medicines order by medicines.mcategory"), nativeQuery = true)

public List<Medicines> MedsbyCat();

@Query(value=("select \* from medicines order by medicines.mprice"), nativeQuery = true)

public List<Medicines> MedsbyPrice();

@Query(value=("select \* from medicines order by medicines.expiredate"), nativeQuery = true)

public List<Medicines> MedsbyDate();

@Query(value=("select \* from medicines where medicines.mname= ? and medicines.mcontrol= 1"), nativeQuery = true)

public List<Medicines> searchMeds(String mname);

@Query(value=("select \* from medicines where medicines.mname= ? order by medicines.mcategory"), nativeQuery = true)

public List<Medicines> searchMedsByCat(String mname);

@Query(value=("select \* from medicines where medicines.mname= ? order by medicines.mprice"), nativeQuery = true)

public List<Medicines> searchMedsByPrice(String mname);

@Query(value=("select \* from medicines where medicines.mname= ? order by medicines.expiredate"), nativeQuery = true)

public List<Medicines> searchMedsByDate(String mname);

@Query(value=("select \* from medicines where medicines.mcontrol= 1 "), nativeQuery = true)

public List<Medicines> searchMedsEnabled();

}

Orders

package com.org.entity;

import java.io.Serializable;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

@JsonIgnoreProperties({"hibernateLazyInitializer"})

@Entity

public class Orders implements Serializable {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(nullable = false)

private int oid;

private int uid;

private int mid;

public Orders() {

}

public Orders(int oid, int uid, int mid) {

super();

this.oid = oid;

this.uid = uid;

this.mid = mid;

}

public int getOid() {

return oid;

}

public void setOid(int oid) {

this.oid = oid;

}

public int getUid() {

return uid;

}

public void setUid(int uid) {

this.uid = uid;

}

public int getMid() {

return mid;

}

public void setMid(int mid) {

this.mid = mid;

}

}

Orders Repo

package com.org.entity;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface OrdersRepo extends JpaRepository<Orders, Integer> {

}

Admin Service

package com.org.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.stereotype.Service;

import com.org.entity.Admin;

import com.org.entity.AdminRepo;

@Service

public class AdminService {

@Autowired

AdminRepo arepo;

public Admin addAdmin(Admin admin)

{

return arepo.save(admin);

}

public List<Admin> getAllAdmin() {

return arepo.findAll();

}

public Admin updateAdmin(Admin admin)

{

Admin a1=arepo.searchAdmin(admin.getAname());

a1.setApassword(admin.getApassword());

arepo.save(a1);

return a1;

}

}

User Service

package com.org.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.org.entity.Admin;

import com.org.entity.User;

import com.org.entity.UserRepo;

@Service

public class UserService {

@Autowired

UserRepo urepo;

public User addUser(User user)

{

return urepo.save(user);

}

public User updateUser(User user)

{

User u=urepo.searchUser(user.getUname());

u.setUpassword(user.getUpassword());

urepo.save(u);

return u;

}

public List<User> getAllUser() {

return urepo.findAll();

}

}

Medicines Service

package com.org.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.org.entity.Medicines;

import com.org.entity.MedicinesRepo;

@Service

public class MedicinesService {

@Autowired

MedicinesRepo mrepo;

public Medicines addMedicine(Medicines medicines)

{

return mrepo.save(medicines);

}

public List<Medicines> getAllMedicines() {

return mrepo.findAll();

}

public List<Medicines> getAllMedicinesEnabled() {

return mrepo.searchMedsEnabled();

}

public List<Medicines> getAllMedicinesbyCat()

{

return mrepo.MedsbyCat();

}

public List<Medicines> getAllMedicinesbyPrice()

{

return mrepo.MedsbyPrice();

}

public List<Medicines> getAllMedicinesbyDate()

{

return mrepo.MedsbyDate();

}

public Medicines editMed(Medicines medicines)

{

Medicines m=mrepo.getById(medicines.getMid());

m.setMname(medicines.getMname());

m.setMprice(medicines.getMprice());

m.setMdesc(medicines.getMdesc());

m.setMcategory(medicines.getMcategory());

m.setExpiredate(medicines.getExpiredate());

m.setMcontrol(medicines.getMcontrol());

mrepo.save(m);

return m;

}

public List<Medicines> searchforMeds(Medicines medicines)

{

return mrepo.searchMeds(medicines.getMname());

}

public Medicines enableMed(Medicines medicines)

{

Medicines m=mrepo.getById(medicines.getMid());

m.setMcontrol(1);

mrepo.save(m);

return m;

}

public Medicines disableMed(Medicines medicines)

{

Medicines m=mrepo.getById(medicines.getMid());

m.setMcontrol(0);

mrepo.save(m);

return m;

}

public void deleteMed(Medicines medicines)

{

mrepo.deleteById(medicines.getMid());

}

}

Frontend

Admin

export class Admin

{

    aid:number;

    aname:string;

    aemail:string;

    apassword:string;

}

User

export class User

{

    uid:number;

    uname:String;

    uage:number;

    uemail:String;

    upassword:String;

}

Medicines

export class Medicines

{

    mid:number;

    mname:String;

    mprice:number;

    mdesc:String;

    mcategory:String;

    expiredate:String;

    mcontrol:number;

}

app-routing module

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

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

import { AddmedicineComponent } from './addmedicine/addmedicine.component';

import { AdmindashboardComponent } from './admindashboard/admindashboard.component';

import { AdminlistComponent } from './adminlist/adminlist.component';

import { AdminloginComponent } from './adminlogin/adminlogin.component';

import { AdminlogoutComponent } from './adminlogout/adminlogout.component';

import { AdminregisterComponent } from './adminregister/adminregister.component';

import { BookingsuccessComponent } from './bookingsuccess/bookingsuccess.component';

import { ChangepwdadminComponent } from './changepwdadmin/changepwdadmin.component';

import { ChangepwduserComponent } from './changepwduser/changepwduser.component';

import { EditmedicineComponent } from './editmedicine/editmedicine.component';

import { EnabledisablemedComponent } from './enabledisablemed/enabledisablemed.component';

import { MedicineslistComponent } from './medicineslist/medicineslist.component';

import { PurchasemedicineComponent } from './purchasemedicine/purchasemedicine.component';

import { RemovemedicineComponent } from './removemedicine/removemedicine.component';

import { SearchmedicinesComponent } from './searchmedicines/searchmedicines.component';

import { UserdashboardComponent } from './userdashboard/userdashboard.component';

import { UserlistComponent } from './userlist/userlist.component';

import { UserloginComponent } from './userlogin/userlogin.component';

import { UserlogoutComponent } from './userlogout/userlogout.component';

import { UserregisterComponent } from './userregister/userregister.component';

const routes: Routes = [

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

  {"path":"admin/adminreg",component:AdminregisterComponent},

  {"path":"admin/admindash",component:AdmindashboardComponent},

  {"path":"admin/admindash/adminlist",component:AdminlistComponent},

  {"path":"admin/admindash/userlist",component:UserlistComponent},

  {"path":"admin/admindash/medslist",component:MedicineslistComponent},

  {"path":"admin/admindash/medslist/addmed",component:AddmedicineComponent},

  {"path":"admin/admindash/medslist/editmed",component:EditmedicineComponent},

  {"path":"admin/admindash/medslist/removemed",component:RemovemedicineComponent},

  {"path":"admin/admindash/endismed",component:EnabledisablemedComponent},

  {"path":"admin/admindash/changepwdadmin",component:ChangepwdadminComponent},

  {"path":"admin/admindash/adminlogout",component:AdminlogoutComponent},

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

  {"path":"user/userreg",component:UserregisterComponent},

  {"path":"user/userdash",component:UserdashboardComponent},

  {"path":"user/userdash/searchmed",component:SearchmedicinesComponent},

  {"path":"user/userdash/changepwduser",component:ChangepwduserComponent},

  {"path":"user/userdash/userlogout",component:UserlogoutComponent},

  {"path":"user/userdash/searchmed/purcmed",component:PurchasemedicineComponent},

  {"path":"user/userdash/searchmed/purcmed/book",component:BookingsuccessComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

admin service

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

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

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

import { Observable, of } from 'rxjs';

import { Admin } from './Admin';

@Injectable({

  providedIn: 'root'

})

export class AdminService {

  url:string="http://localhost:8081/medicare/admin/insert";

  url1:string="http://localhost:8081/medicare/admin/list";

  url2:string="http://localhost:8081/medicare/admin/updatepwd";

  flag:number=0;

  constructor(private http:HttpClient, private router:Router) { }

  registerAdmin(admin:any):Observable<Admin[]>

  {

    return this.http.post<Admin[]>(this.url,admin);

  }

  pwdchangeSuccess(name:String,admin:any):boolean

  {

    for(let a of admin)

    {

      if(a.aname===name)

      {

        this.flag=1;

      }

    }

    if(this.flag===1)

      return true;

    else

      return false;

  }

  login(name:String,password:String,admins:any):boolean

  {

    for(let a of admins)

    {

      if(a.aname===name && a.apassword===password)

      {

        this.router.navigateByUrl("admin/admindash");

        this.flag=1;

      }

    }

    if(this.flag===1)

      return true;

    else

      return false;

  }

  getAdmins():Observable<Admin[]>

  {

    return this.http.get<Admin[]>(this.url1);

  }

  changepwd(admin:any):Observable<Admin[]>

  {

    return this.http.put<Admin[]>(this.url2,admin);

  }

}

user service

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

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

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

import { BehaviorSubject, Observable } from 'rxjs';

import { User } from './User';

@Injectable({

  providedIn: 'root'

})

export class UserService {

  url:string="http://localhost:8081/medicare/user/list";

  url2:string="http://localhost:8081/medicare/user/login";

  url3:string="http://localhost:8081/medicare/user/insert";

  url4:string="http://localhost:8081/medicare/user/updatepwd";

  flag:number=0;

  constructor(private http:HttpClient, private router:Router) { }

  registerUser(user:any):Observable<User[]>

  {

    return this.http.post<User[]>(this.url3,user);

  }

  pwdchangeSuccess(name:String,user:any):boolean

  {

    for(let u of user)

    {

      if(u.uname===name)

      {

        this.flag=1;

      }

    }

    if(this.flag===1)

      return true;

    else

      return false;

  }

  login(name:String,password:String,users:any):boolean

  {

    for(let u of users)

    {

      if(u.uname===name && u.upassword===password)

      {

        this.router.navigateByUrl("user/userdash");

        this.flag=1;

      }

    }

    if(this.flag===1)

      return true;

    else

      return false;

  }

  getUsers():Observable<User[]>

  {

    return this.http.get<User[]>(this.url);

  }

  changepwd(user:any):Observable<User[]>

  {

    return this.http.put<User[]>(this.url4,user);

  }

}

medicines service

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

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

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

import { BehaviorSubject, map, Observable, of } from 'rxjs';

import { Medicines } from './Medicines';

@Injectable({

  providedIn: 'root'

})

export class MedicinesService {

  url:string="http://localhost:8081/medicare/medicines/list";

  url2:string="http://localhost:8081/medicare/medicines/insert";

  url3:string="http://localhost:8081/medicare/medicines/edit";

  url4:string="http://localhost:8081/medicare/medicines/delete";

  url5:string="http://localhost:8081/medicare/medicines/search";

  url6:string="http://localhost:8081/medicare/medicines/list/category";

  url7:string="http://localhost:8081/medicare/medicines/list/price";

  url8:string="http://localhost:8081/medicare/medicines/list/date";

  url9:string="http://localhost:8081/medicare/medicines/enable";

  url10:string="http://localhost:8081/medicare/medicines/list/enable";

  url11:string="http://localhost:8081/medicare/medicines/disable";

  r:Medicines=new Medicines();

  constructor(private http:HttpClient, private router:Router) { }

  getMedicines():Observable<Medicines[]>

  {

    return this.http.get<Medicines[]>(this.url);

  }

  getMedicinesEnabled():Observable<Medicines[]>

  {

    return this.http.get<Medicines[]>(this.url10);

  }

  getMedicinesbyCat():Observable<Medicines[]>

  {

    return this.http.get<Medicines[]>(this.url6);

  }

  getMedicinesbyPrice():Observable<Medicines[]>

  {

    return this.http.get<Medicines[]>(this.url7);

  }

  getMedicinesbyDate():Observable<Medicines[]>

  {

    return this.http.get<Medicines[]>(this.url8);

  }

  addMed(medicines:any):Observable<Medicines[]>

  {

    return this.http.post<Medicines[]>(this.url2,medicines);

  }

  editMed(medicines:any):Observable<Medicines[]>

  {

    return this.http.put<Medicines[]>(this.url3,medicines);

  }

  searchMeds(medicines:any):Observable<Medicines[]>

  {

    return this.http.put<Medicines[]>(this.url5,medicines);

  }

  removeMed(medicines:any):Observable<Medicines[]>

  {

    return this.http.put<Medicines[]>(this.url4,medicines);

  }

  enableMed(medicines:any):Observable<Medicines[]>

  {

    this.router.navigateByUrl("admin/admindash/medslist/endismed");

    return this.http.put<Medicines[]>(this.url9,medicines);

  }

  disableMed(medicines:any):Observable<Medicines[]>

  {

    this.router.navigateByUrl("admin/admindash/medslist");

    return this.http.put<Medicines[]>(this.url11,medicines);

  }

  addOrder(med:any)

  {

    this.r=med;

  }

  getOrderDetails()

  {

    console.log(this.r);

    return this.r;

  }

}