**Sporty Shoes**

**Screenshot**

**application.properties**

server.port=8083

#jpa-hibernate

spring.jpa.hibernate.ddl-auto=update

spring.jpa.hibernate.dialect=org.hibernate.dialect.MySQLDialect

spring.jpa.show-sql=true

#datasource

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/sportyshoes

spring.datasource.username=root

spring.datasource.password=123456

**Bank.java**

package com.example.demo;

import javax.persistence.Entity;

import javax.persistence.Id;

import lombok.Data;

@Data

@Entity

public class Bank {

@Id

private int bid;

private int balance;

}

**BankRepo.java**

package com.example.demo;

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

public interface BankRepo extends JpaRepository<Bank, Integer> {

}

**SecurityConfig.java**

package com.example.demo;

import org.springframework.context.annotation.Bean;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

import org.springframework.security.crypto.password.NoOpPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

@EnableWebSecurity

public class Securityconfig extends WebSecurityConfigurerAdapter{

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.inMemoryAuthentication()

.withUser("admin")

.password("admin")

.roles("ADMIN")

.and()

.withUser("user")

.password("user")

.roles("USER");

}

@Bean

public PasswordEncoder getpassword() {

return NoOpPasswordEncoder.getInstance();

}

@Override

protected void configure(HttpSecurity http) throws Exception {

http.authorizeRequests()

.antMatchers("/admin").hasRole("ADMIN")

.antMatchers("/user").hasRole("USER").antMatchers("/").hasRole("ADMIN").and().formLogin();

}

}

**Shoes.java**

package com.example.demo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import lombok.Data;

@Data

@Entity

public class Shoes {

@Id

private int sid;

private String sname;

private int scost;

private int squantity;

}

**ShoesDAO.java**

package com.example.demo;

import java.util.List;

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

import org.springframework.stereotype.Service;

@Service

public class ShoesDAO {

@Autowired

ShoesRepo srepo;

public Shoes insert(Shoes s){return srepo.save(s);}

public List<Shoes> getall(){return srepo.findAll();}

public Shoes getbyid(int id) {return srepo.getById(id);}

public void delete(int id) { srepo.deleteById(id);}

public Shoes updateByName(Shoes s) {

//1.we fetch the record from the table

Shoes ss=srepo.findById(s.getSid()).orElse(null);

ss.setSname(s.getSname());

ss.setScost(s.getScost());

ss.setSquantity(s.getSquantity());

return srepo.save(ss);

}

}

**ShoesRepo.java**

package com.example.demo;

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

public interface ShoesRepo extends JpaRepository<Shoes, Integer> {

}

**Users.java**

package com.example.demo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import lombok.Data;

@Data

@Entity

public class Users {

@Id

private int uid;

private String uname;

private String pwd;

private String email;

private String address;

}

**UsersDAO.java**

package com.example.demo;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

@Service

public class UsersDAO {

@Autowired

UsersRepo urepo;

public Users insert(Users u){return urepo.save(u);}

public List<Users> getall(){return urepo.findAll();}

public Users getbyid(int id) {return urepo.getById(id);}

public void delete(int id) {urepo.deleteById(id);}

public Users updateByName(Users u) {

Users uu=urepo.findById(u.getUid()).orElse(null);

uu.setUname(u.getUname());

uu.setPwd(u.getPwd());

uu.setEmail(u.getEmail());

uu.setAddress(u.getAddress());

return urepo.save(uu);

}

public Users checkById(int userid) {

Users u=urepo.findById(userid).orElse(null);

return u;

}

}

**UsersRepo.java**

package com.example.demo;

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

public interface UsersRepo extends JpaRepository<Users, Integer> {

}

**SUController.java**

package com.example.demo;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

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

import org.springframework.stereotype.Controller;

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

import org.springframework.web.servlet.ModelAndView;

@Controller

public class SUController {

@Autowired

ShoesDAO sdao;

@Autowired

UsersDAO udao;

@Autowired

BankRepo brepo;

@RequestMapping("/user")

public ModelAndView user() {

ModelAndView mv =new ModelAndView();

mv.setViewName("loginuser.jsp");

return mv;

}

@RequestMapping("/admin")

public ModelAndView admin() {

ModelAndView mv =new ModelAndView();

mv.setViewName("display.jsp");

return mv;

}

@RequestMapping("/")

public ModelAndView demo() {

ModelAndView mv =new ModelAndView();

mv.setViewName("display.jsp");

return mv;

}

@RequestMapping("/getallshoes")

public ModelAndView retrieveadmin(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

List<Shoes> list=sdao.getall();

mv.setViewName("displayshoes.jsp");

mv.addObject("list",list);

return mv;

}

@RequestMapping("/getallusers")

public ModelAndView retrieveadmin1(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

List<Users> list=udao.getall();

mv.setViewName("displayusers.jsp");

mv.addObject("list",list);

return mv;

}

@RequestMapping("/insertshoe")

public ModelAndView insert(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

Shoes s=new Shoes();

s.setSid(Integer.parseInt(request.getParameter("sid")));

s.setSname(request.getParameter("sname"));

s.setScost(Integer.parseInt(request.getParameter("scost")));

s.setSquantity(Integer.parseInt(request.getParameter("squantity")));

Shoes sh=sdao.insert(s);

if(sh!=null) {mv.setViewName("endinsert.jsp");}

return mv;

}

@RequestMapping("/insertuser")

public ModelAndView insert1(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

Users u=new Users();

u.setUid(Integer.parseInt(request.getParameter("uid")));

u.setUname(request.getParameter("uname"));

u.setPwd(request.getParameter("pwd"));

u.setEmail(request.getParameter("email"));

u.setAddress(request.getParameter("address"));

Users uh=udao.insert(u);

if(uh!=null) {mv.setViewName("endinsert.jsp");}

return mv;

}

@RequestMapping("/editshoe")

public ModelAndView editName(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

Shoes s=new Shoes();

s.setSname(request.getParameter("sname"));

s.setScost(Integer.parseInt(request.getParameter("scost")));

s.setSid(Integer.parseInt(request.getParameter("sid")));

s.setSquantity(Integer.parseInt(request.getParameter("squantity")));

Shoes list= sdao.updateByName(s);

mv.setViewName("endedit.jsp");

mv.addObject("list",list);

return mv;

}

@RequestMapping("/edituser")

public ModelAndView editName1(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

Users u=new Users();

u.setUid(Integer.parseInt(request.getParameter("uid")));

u.setUname(request.getParameter("uname"));

u.setPwd(request.getParameter("pwd"));

u.setEmail(request.getParameter("email"));

u.setAddress(request.getParameter("address"));

Users list= udao.updateByName(u);

mv.setViewName("endedit.jsp");

mv.addObject("list",list);

return mv;

}

@RequestMapping("/deleteshoe")

public ModelAndView deletebyid(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

int sno = Integer.parseInt(request.getParameter("sid"));

sdao.delete(sno);

mv.setViewName("enddelete.jsp");

mv.addObject("no",sno);

return mv;

}

@RequestMapping("/deleteuser")

public ModelAndView deletebyid1(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

int uno = Integer.parseInt(request.getParameter("uid"));

udao.delete(uno);

mv.setViewName("enddelete.jsp");

mv.addObject("no",uno);

return mv;

}

@RequestMapping("/verifyuser")

public ModelAndView verifyuser(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

String username=request.getParameter("uname");

String password=request.getParameter("pwd");

int userid=Integer.parseInt(request.getParameter("uid"));

Users list= udao.checkById(userid);

if(username.equals(list.getUname()) && (password.equals(list.getPwd()))){

mv.setViewName("/getallfn");}

else {mv.setViewName("invalid.jsp");}

return mv;

}

@RequestMapping("/getallfn")

public ModelAndView retrieveusershoe(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

List<Shoes> list=sdao.getall();

mv.setViewName("viewshoes.jsp");

mv.addObject("list",list);

return mv;

}

@RequestMapping("/checkbalance")

public ModelAndView pay(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

int price=Integer.parseInt(request.getParameter("amt"));

List<Shoes> s = sdao.getall();

for(Shoes ss:s)

{

if(price==ss.getScost())

{

int numshoe=ss.getSquantity();

if(numshoe>1)

{ int remainquan=numshoe-1;

ss.setSquantity(remainquan);

sdao.insert(ss);

}

else

{ mv.setViewName("amountnil.jsp");

return mv;

}

break;

}

}

Bank bb=brepo.findById(1).orElse(null);

if(price<bb.getBalance())

{ int remaining= bb.getBalance() - price;

bb.setBalance(remaining);

brepo.save(bb);

mv.setViewName("final.jsp");

mv.addObject("price",price);

return mv;

}

else

{ mv.setViewName("amountnil.jsp");

return mv;

}

}

}