**Project objective**

As a Full Stack Developer, complete the features of the application by planning the development and pushing the source code to the GitHub repository.

**Problem statement**

Sporty Shoes is a company that manufactures and sells sports shoes. They have a walk-in store, and now, they wish to launch their e-commerce portal sportyshoes.com.

**Source Code**

SportyShoesApplication

package com.org;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication

@ComponentScan(basePackages = "com.org")

public class SportyShoesApplication {

public static void main(String[] args) {

SpringApplication.run(SportyShoesApplication.class, args);

}

}

AdminController

package com.org.controller;

import java.util.List;

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

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

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.RequestParam;

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

import com.org.entity.Admin;

import com.org.entity.Category;

import com.org.entity.Orders;

import com.org.entity.Product;

import com.org.entity.User;

import com.org.service.AdminService;

import com.org.service.CategoryService;

import com.org.service.OrderService;

import com.org.service.ProductService;

import com.org.service.UserService;

@RestController

@RequestMapping("/api")

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/{aid}")

public Admin updatePwd(@PathVariable("aid")int aid,@RequestBody Admin admin)

{

return adminService.updateAdmin(aid,admin);

}

@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();

}

@GetMapping("/user/{uid}")

public User findUserById(@PathVariable("uid")int uid)

{

return userService.getUserById(uid);

}

@Autowired

private CategoryService categoryService;

@PostMapping("/category/insert")

public Category addNewCategory(@RequestBody Category category)

{

return categoryService.addCategory(category);

}

@GetMapping("/category/list")

public List<String> listCategory()

{

return categoryService.getAllCategory();

}

@Autowired

private ProductService productService;

@PostMapping("/product/insert")

public Product addNewProduct(@RequestBody Product product)

{

return productService.addProduct(product);

}

@GetMapping("/product/list")

public List<String> listProduct()

{

return productService.getAllProducts();

}

@GetMapping("/product/{pid}")

public List<String> findProductById(@PathVariable("pid") int pid)

{

return productService.ProductById(pid);

}

@PutMapping("/product/update/{pid}")

public Product updateProduct(@PathVariable("pid")int pid,@RequestBody Product product)

{

return productService.updateProduct(pid,product);

}

@DeleteMapping("/product/{pid}")

public void deleteProduct(@PathVariable("pid")int pid)

{

productService.deleteProduct(pid);

System.out.println("Deleted");

}

@Autowired

private OrderService orderService;

@GetMapping("/orders/list")

public List<Orders> listOrders()

{

return orderService.getAllOrders();

}

@GetMapping("/orders/filter")

public List<Orders> listOrdersByDate(@RequestParam(value="dateoforder") String dateoforder)

{

return orderService.findByDate(dateoforder);

}

@GetMapping("/orders/{cid}")

public List<String> listOrdersByCategory(@PathVariable("cid")int cid)

{

return categoryService.CategoryByCid(cid);

}

@PutMapping("/orders/update/{oid}")

public Orders updateOrder(@PathVariable("oid")int oid,@RequestBody Orders order)

{

return orderService.update(oid,order);

}

}

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;

}

}

AdminRepo

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> {

}

Category

package com.org.entity;

import java.io.Serializable;

import java.util.ArrayList;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import javax.persistence.Table;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

@JsonIgnoreProperties({"hibernateLazyInitializer"})

@Entity

public class Category implements Serializable {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(nullable = false)

private int cid;

private String cname;

@OneToMany(mappedBy = "category")

private List<Product> products = new ArrayList<>();

public Category() {

}

public Category(int cid, String cname, List<Product> products) {

super();

this.cid = cid;

this.cname = cname;

this.products = products;

}

public int getCid() {

return cid;

}

public void setCid(int cid) {

this.cid = cid;

}

public String getCname() {

return cname;

}

public void setCname(String cname) {

this.cname = cname;

}

public List<Product> getProducts() {

return products;

}

public void addProduct(Product product) {

this.products.add(product);

}

}

CategoryRepo

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 CategoryRepo extends JpaRepository<Category, Integer> {

@Query(value=("select \* from category"), nativeQuery = true)

public List<String> findCategory();

@Query(value=("select category.cid,category.cname,product.pid,product.pname,product.color,product.price from category inner join product on category.cid=product.cid where category.cid= ?"), nativeQuery=true)

public List<String> findCid(int cid);

}

Orders

package com.org.entity;

import java.io.Serializable;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

import org.springframework.format.annotation.DateTimeFormat;

import com.fasterxml.jackson.annotation.JsonFormat;

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;

//@DateTimeFormat(pattern="dd-MM-yyyy")

@JsonFormat(pattern="dd-MM-yyyy")

private String dateoforder;

private int pid;

private int uid;

public Orders() {

}

public Orders(int oid, String dateoforder) {

super();

this.oid = oid;

this.dateoforder = dateoforder;

}

public int getOid() {

return oid;

}

public void setOid(int oid) {

this.oid = oid;

}

public String getDateoforder() {

return dateoforder;

}

public void setDateoforder(String dateoforder) {

this.dateoforder = dateoforder;

}

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public int getUid() {

return uid;

}

public void setUid(int uid) {

this.uid = uid;

}

}

OrderRepo

package com.org.entity;

import java.sql.Date;

import java.util.List;

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

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

public interface OrderRepo extends JpaRepository<Orders, Integer> {

public List<Orders> findByDateoforder(String dateoforder);

}

Product

package com.org.entity;

import java.io.Serializable;

import java.math.BigDecimal;

import java.util.ArrayList;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

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 javax.persistence.ManyToOne;

import javax.persistence.Table;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

import com.fasterxml.jackson.annotation.JsonProperty;

@JsonIgnoreProperties({"hibernateLazyInitializer"})

@Entity

public class Product implements Serializable {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(nullable = false)

private int pid;

private String pname;

private String color;

private BigDecimal price;

@ManyToOne(fetch = FetchType.LAZY)

@JoinColumn(name = "cid")

private Category category;

@ManyToMany//(cascade = { CascadeType.MERGE, CascadeType.PERSIST })

@JoinTable(name="orders",

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

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

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

public Product() {

}

public Product(int pid, String pname, String color, BigDecimal price, Category category, List<User> users) {

super();

this.pid = pid;

this.pname = pname;

this.color = color;

this.price = price;

this.category = category;

this.users = users;

}

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getPname() {

return pname;

}

public void setPname(String pname) {

this.pname = pname;

}

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public BigDecimal getPrice() {

return price;

}

public void setPrice(BigDecimal price) {

this.price = price;

}

public Category getCategory() {

return category;

}

public void setCategory(Category category) {

this.category = category;

}

public List<User> getUsers() {

return users;

}

/\*public void setUsers(List<User> users) {

this.users = users;

}\*/

public void addUsers(User user) {

this.users.add(user);

}

}

ProductRepo

package com.org.entity;

import java.math.BigDecimal;

import java.util.List;

import org.hibernate.query.NativeQuery;

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

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

import org.springframework.stereotype.Repository;

@Repository

public interface ProductRepo extends JpaRepository<Product, Integer> {

@Query(value="select product.pid,product.pname,product.color,product.price,category.cid,category.cname from product inner join category on product.cid=category.cid", nativeQuery=true)

public List<String> findProduct();

@Query(value=("select pid,pname,color,price from Product where pid= ?"), nativeQuery = true)

public List<String> findPid(int pid);

}

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 javax.persistence.Table;

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<Product> products1=new ArrayList<>();

public User() {

}

public User(int uid, String uname, int uage, String uemail, String upassword, List<Product> products) {

super();

this.uid = uid;

this.uname = uname;

this.uage = uage;

this.uemail = uemail;

this.upassword = upassword;

this.products1 = products;

}

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<Product> getProducts1() {

return products1;

}

public void addProducts1(Product product) {

this.products1.add(product);

}

}

UserRepo

package com.org.entity;

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

import org.springframework.stereotype.Repository;

@Repository

public interface UserRepo extends JpaRepository<User, Integer> {

}

AdminService

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.AdminRepo;

import com.org.entity.Product;

@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(int aid, Admin admin)

{

Admin a=arepo.getById(aid);

a.setApassword(admin.getApassword());

arepo.save(a);

return a;

}

}

CategoryService

package com.org.service;

import java.util.ArrayList;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.org.entity.Category;

import com.org.entity.CategoryRepo;

import com.org.entity.Product;

@Service

public class CategoryService {

@Autowired

CategoryRepo crepo;

public Category addCategory(Category category)

{

return crepo.save(category);

}

public List<String> getAllCategory()

{

return crepo.findCategory();

}

public List<String> CategoryByCid(int cid)

{

return crepo.findCid(cid);

}

}

OrderService

package com.org.service;

import java.sql.Date;

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.OrderRepo;

import com.org.entity.Orders;

@Service

public class OrderService {

@Autowired

OrderRepo orepo;

public List<Orders> getAllOrders()

{

return orepo.findAll();

}

public List<Orders> findByDate(String dateoforder)

{

return orepo.findByDateoforder(dateoforder);

}

public Orders update(int oid, Orders order)

{

Orders o=orepo.getById(oid);

o.setDateoforder(order.getDateoforder());

o.setPid(order.getPid());

o.setUid(order.getUid());

orepo.save(o);

return o;

}

}

ProductService

package com.org.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.org.entity.Category;

import com.org.entity.Product;

import com.org.entity.ProductRepo;

@Service

public class ProductService {

@Autowired

ProductRepo prepo;

public Product addProduct(Product product)

{

Product p=prepo.save(product);

return p;

}

public List<String> getAllProducts()

{

return prepo.findProduct();

}

public List<String> ProductById(int pid)

{

return prepo.findPid(pid);

}

public Product updateProduct(int pid, Product product)

{

Product p=prepo.getById(pid);

p.setColor(product.getColor());

p.setPname(product.getPname());

p.setPrice(product.getPrice());

p.setCategory(product.getCategory());

prepo.save(p);

return p;

}

public void deleteProduct(int pid)

{

prepo.deleteById(pid);

}

}

UserService

**package** com.org.service;

**import** java.util.List;

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

**import** org.springframework.stereotype.Service;

**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** List<User> getAllUser()

{

**return** urepo.findAll();

}

**public** User getUserById(**int** uid)

{

**return** urepo.getById(uid);

}

}