**Product Model Class**

**package** com.project.app.model;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.NamedQuery;

**import** javax.persistence.Table;

**import** org.springframework.transaction.annotation.EnableTransactionManagement;

@Entity

@Table(name = "product")

@EnableTransactionManagement

**public** **class** Product {

@Id

@GeneratedValue(strategy = GenerationType.***AUTO***)

@Column(name = "Product\_ID")

**private** **int** productID;

@Column(name = "Product\_Name")

**private** String productname;

@Column(name = "Season")

**private** String season;

@Column(name = "Brand")

**private** String brand;

@Column(name = "Category")

**private** String category;

@Column(name = "Price")

**private** **int** price;

@Column(name = "Color")

**private** String color;

@Column(name = "Discount")

**private** String discount;

@Column(name = "Created\_Date")

**private** String createddate;

**public** **int** getProductID() {

**return** productID;

}

**public** **void** setProductID(**int** productID) {

**this**.productID = productID;

}

**public** String getProductname() {

**return** productname;

}

**public** **void** setProductname(String productname) {

**this**.productname = productname;

}

**public** String getSeason() {

**return** season;

}

**public** **void** setSeason(String season) {

**this**.season = season;

}

**public** String getBrand() {

**return** brand;

}

**public** **void** setBrand(String brand) {

**this**.brand = brand;

}

**public** String getCategory() {

**return** category;

}

**public** **void** setCategory(String category) {

**this**.category = category;

}

**public** **int** getPrice() {

**return** price;

}

**public** **void** setPrice(**int** price) {

**this**.price = price;

}

**public** String getColor() {

**return** color;

}

**public** **void** setColor(String color) {

**this**.color = color;

}

**public** String getDiscount() {

**return** discount;

}

**public** **void** setDiscount(String discount) {

**this**.discount = discount;

}

**public** String getCreateddate() {

**return** createddate;

}

**public** **void** setCreateddate(String createddate) {

**this**.createddate = createddate;

}

}

**User Model Class  
package** com.project.app.model;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.NamedQuery;

**import** javax.persistence.Table;

**import** org.springframework.transaction.annotation.EnableTransactionManagement;

@Entity

@Table(name = "user")

@EnableTransactionManagement

**public** **class** User {

@Id

@GeneratedValue(strategy = GenerationType.***AUTO***)

@Column(name = "user\_ID")

**private** **int** userID;

@Column(name = "user\_name")

**private** String username;

@Column(name = "user\_password")

**private** String password;

@Column(name = "user\_email")

**private** String email;

@Column(name = "user\_phone\_number")

**private** String phone\_number;

**public** **int** getUserID() {

**return** userID;

}

**public** **void** setUserID(**int** userID) {

**this**.userID = userID;

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getPhone\_number() {

**return** phone\_number;

}

**public** **void** setPhone\_number(String phone\_number) {

**this**.phone\_number = phone\_number;

}

}

**Product Repository Class  
package** com.project.app.repository;

**import** org.springframework.data.domain.Pageable;

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

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

**import** org.springframework.data.repository.CrudRepository;

**import** org.springframework.data.repository.query.Param;

**import** org.springframework.stereotype.Repository;

**import** java.util.\*;

**import** com.project.app.model.Product;

@Repository

**public** **interface** ProductRepository **extends** JpaRepository<Product, Integer>{

//@Query("SELECT p from Product p")

//public List<Product> findByAll();

@Query("SELECT p from Product p where p.season=:c")

**public** List<Product> findBySeason(@Param("c") String season);

@Query("SELECT p from Product p where p.brand=:b")

**public** List<Product> findByBrand(@Param("b") String brand);

@Query("SELECT p from Product p where p.category=:cg")

**public** List<Product> findByCategory(@Param("cg") String category);

@Query("SELECT p from Product p where p.color=:co")

**public** List<Product> findByColor(@Param("co") String color);

@Query("SELECT p from Product p order by price asc")

**public** List<Product> findByLowPrice();

@Query("SELECT p from Product p order by price desc")

**public** List<Product> findByHighPrice();

@Query(nativeQuery = **true**,

value = "select \* from product p where p.Brand=?1 order by Created\_Date desc limit 0, 1")

**public** List<Product> findByLatest(String brand);

}

**User Repository Class  
package** com.project.app.repository;

**import** org.springframework.data.domain.Pageable;

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

**import** org.springframework.data.jpa.repository.Modifying;

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

**import** org.springframework.data.repository.CrudRepository;

**import** org.springframework.data.repository.query.Param;

**import** org.springframework.stereotype.Repository;

**import** org.springframework.transaction.annotation.Transactional;

**import** java.util.\*;

**import** com.project.app.model.User;

@Repository

**public** **interface** UserRepository **extends** JpaRepository<User, Integer>{

@Transactional

@Modifying

@Query(nativeQuery = **true**,

value = "update user u set u.user\_password=?1 where u.user\_email=?2")

**public** **void** findByChangePass(String newpass, String email);

//@Query("update user set u.user\_password=:pass where u.user\_email=:mail")

//public List<User> findByChangePass2(@Param("pass") String newpasss, @Param("mail") String email);

}

**Product Controller Class  
package** com.project.app.controller;

**import** java.util.List;

**import** java.util.Optional;

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

**import** org.springframework.data.domain.PageRequest;

**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.RestController;

**import** com.project.app.model.Product;

**import** com.project.app.repository.ProductRepository;

@RestController

@RequestMapping("product")

**public** **class** ProductController {

@Autowired

ProductRepository productrepository;

// to get all the products available

@GetMapping("getallproducts")

**public** List<Product> getAllproduct()

{

List<Product> product = (List<Product>) productrepository.findAll();

**return** product;

}

// to add a completely new product

@PostMapping("addproduct")

**public** Product addproduct(@RequestBody Product product)

{

**return** productrepository.save(product);

}

// to delete the product

@DeleteMapping("delete/product/{id}")

**public** **void** deleteStudent(@PathVariable **int** id)

{

productrepository.deleteById(id);

}

// update the any product details with the particular id of that product

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

**public** Product updateproduct(@RequestBody Product product)

{

**return** productrepository.save(product);

}

// to filter by season

@GetMapping("filter/season/{season}")

**public** List<Product> getFilteredbySeason(@PathVariable String season) {

**return** productrepository.findBySeason(season);

}

// to filter by brand

@GetMapping("filter/brand/{brand}")

**public** List<Product> getFilteredbyBrand(@PathVariable String brand) {

**return** productrepository.findByBrand(brand);

}

// to filter by category

@GetMapping("filter/category/{category}")

**public** List<Product> getFilteredbyCategory(@PathVariable String category) {

**return** productrepository.findByCategory(category);

}

// to filter by color

@GetMapping("filter/color/{color}")

**public** List<Product> getFilteredbyColor(@PathVariable String color) {

**return** productrepository.findByColor(color);

}

// to filter from low price to high price

@GetMapping("filter/price/lowtohigh")

**public** List<Product> findByLowPrice() {

**return** productrepository.findByLowPrice();

}

// to filter from high price to low price

@GetMapping("filter/price/hightolow")

**public** List<Product> findByHighPrice() {

**return** productrepository.findByHighPrice();

}

// to filter latest shoe of a particular brand

@GetMapping("filter/{brand}/latest")

**public** List<Product> getFilteredLatest(@PathVariable String brand) {

**return** productrepository.findByLatest(brand);

}

}

**User Controller Class**

package com.project.app.controller;

import java.util.List;

import java.util.Optional;

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.RestController;

import com.project.app.model.Product;

import com.project.app.model.User;

import com.project.app.repository.UserRepository;

@RestController

@RequestMapping("user")

public class UserController {

@Autowired

UserRepository userrepository;

// to get all the users registered

@GetMapping("getallusers")

public List<User> getAllUsers()

{

List<User> user=(List<User>) userrepository.findAll();

return user;

}

// to change admin password only

String email = "admin@yahoo.com";

@PutMapping("changepass/admin/{newpass}")

public void updatepass(@PathVariable String newpass)

{

userrepository.findByChangePass(newpass,email);

}

}

**pom.xml File**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.2.4.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.project</groupId>

<artifactId>SpringBootProject</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>SpringBootProject</name>

<description>Demo project for Spring Boot</description>

<!-- <properties>

<java.version>11</java.version>

</properties>-->

<properties>

<maven-jar-plugin.version>3.1.1</maven-jar-plugin.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.0.8</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.19.1</version>

</plugin>

</plugins>

</build>

</project>

**Application properties**# MySQL database connecting utility

spring.datasource.url=jdbc:mysql://localhost:3306/db\_world

spring.datasource.username=root

spring.datasource.password=root

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

spring.jpa.database=mysql

# JPA property utility

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.show\_sql=true

**SpringBootProjectApplication**

package com.project.app;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import org.springframework.transaction.annotation.EnableTransactionManagement;

@SpringBootApplication

@EnableTransactionManagement

@EnableJpaRepositories

public class SpringBootProjectApplication {

public static void main(String[] args) {

SpringApplication.run(SpringBootProjectApplication.class, args);

}

}