**Developer Name – Vinayak Shinde**

Src/main/java

By default spring create package com.sportyshoes

package com.sportyshoes;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication

@ComponentScan(basePackages = "com.sportyshoes")

public class SportyShoesApplication {

public static void main(String[] args) {

SpringApplication.run(SportyShoesApplication.class, args);

}

}

Create package com.sportyshoes.controllers

Create adminconttoller.java

package com.sportyshoes.controllers;

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

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

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

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

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

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

import com.sportyshoes.services.AdminService;

@RestController

@RequestMapping("/api")

public class AdminController {

@Autowired

private AdminService adminService;

@PostMapping("/")

public String adminLogin(@RequestParam String email, @RequestParam String password)

{

if(adminService.verifyAdmin(email, password))

{

return "Login Successfull";

}

else

{

return "Login Failed";

}

}

@PostMapping("/changePassword/{id}&{newPassword}")

public String changePassword(@PathVariable int id, @PathVariable String newPassword)

{

adminService.changePassword(id, newPassword);

return "Password Changed Successfully";

}

}

Create CategoryController.java

package com.sportyshoes.controllers;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

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

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

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

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

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

import com.sportyshoes.entities.Category;

import com.sportyshoes.services.CategoryService;

@RestController

@RequestMapping("/api/category")

public class CategoryController {

@Autowired

private CategoryService categoryService;

@PostMapping("/")

public ResponseEntity<Category> addCategory(@RequestBody Category category)

{

Category obj= categoryService.addCategory(category);

return new ResponseEntity<Category>(obj,HttpStatus.CREATED);

}

@DeleteMapping("/{categoryId}")

public String deleteCategory(@PathVariable int categoryId)

{

categoryService.deleteCategory(categoryId);

return "Category Deleted";

}

@PostMapping("/update/")

public String updateCategory(@RequestBody Category category)

{

int categoryId = category.getId();

if(categoryService.updateCategory(categoryId, category))

{

return categoryService.searchCategory(categoryId).toString();

}

else

{

return "No such Category exists";

}

}

}

Create ProductController.java

package com.sportyshoes.controllers;

import java.util.List;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.RequestBody;

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

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

import com.sportyshoes.entities.Product;

import com.sportyshoes.services.ProductService;

@RestController

@RequestMapping("/api/product")

public class ProductController {

@Autowired

private ProductService productService;

@PostMapping("/")

public ResponseEntity<Product> addProduct(@RequestBody Product product)

{

Product obj= productService.addProduct(product);

return new ResponseEntity<Product>(obj,HttpStatus.CREATED);

}

@DeleteMapping("/{productId}")

public String deleteProduct(@PathVariable int productId)

{

productService.deleteProduct(productId);

return "Product Deleted";

}

@PostMapping("/update/")

public String updateProduct(@RequestBody Product product)

{

int productId = product.getId();

if(productService.updateProduct(productId, product))

{

return productService.searchProduct(productId).toString();

}

else

{

return "No such Product exists";

}

}

@GetMapping("/")

public List<Product> getProducts()

{

List<Product> prods = productService.getAllProducts();

return prods;

}

}

Create PurchaseController.java

package com.sportyshoes.controllers;

import java.util.List;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.RequestBody;

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

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

import com.sportyshoes.entities.Purchase;

import com.sportyshoes.services.PurchaseService;

@RestController

@RequestMapping("/api/purchase")

public class PurchaseController {

@Autowired

private PurchaseService purchaseService;

@PostMapping("/")

public ResponseEntity<Purchase> addPurchase(@RequestBody Purchase purchase)

{

Purchase obj= purchaseService.addPurchase(purchase);

return new ResponseEntity<Purchase>(obj,HttpStatus.CREATED);

}

@GetMapping("/")

public List<Purchase> getAllProduct()

{

return purchaseService.getAllProducts();

}

@GetMapping("/byCategory/{categoryId}")

public List<Purchase> getAllProductByCategory(@PathVariable int categoryId)

{

return purchaseService.getProductsByCategory(categoryId);

}

}

Create UserController.java

package com.sportyshoes.controllers;

import java.util.List;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.RequestBody;

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

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

import com.sportyshoes.entities.User;

import com.sportyshoes.services.UserService;

@RestController

@RequestMapping("/api/user")

public class UserController {

@Autowired

private UserService userService;

@GetMapping("/")

public List<User> getUsers()

{

List<User> users = userService.getAllUsers();

return users;

}

@GetMapping("/{userId}")

public String getUser(@PathVariable int userId)

{

if(userService.searchUser(userId).getId()==0)

{

return "User not Available";

}

else

{

return userService.searchUser(userId).toString();

}

}

@DeleteMapping("/{userId}")

public String deleteUser(@PathVariable int userId)

{

userService.deleteUser(userId);

return "User Deleted";

}

@PostMapping("/")

public ResponseEntity<User> addPerson(@RequestBody User user)

{

User obj= userService.addUser(user);

return new ResponseEntity<User>(obj,HttpStatus.CREATED);

}

}

Create package com.sportyshoes.entities

Create admin.java

package com.sportyshoes.entities;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Admin {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String email;

private String password;

public Admin(String email, String password) {

super();

this.email = email;

this.password = password;

}

public Admin() {}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

@Override

public String toString() {

return "Admin [id=" + id + ", email=" + email + ", password=" + password + "]";

}

}

Create Category.java

package com.sportyshoes.entities;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Category {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String details;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDetails() {

return details;

}

public void setDetails(String details) {

this.details = details;

}

public Category(String name, String details) {

super();

this.name = name;

this.details = details;

}

public Category() {}

@Override

public String toString() {

return "Category [id=" + id + ", name=" + name + ", details=" + details + "]";

}

}

Create product.java

package com.sportyshoes.entities;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToOne;

@Entity

public class Product {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private int price;

@ManyToOne

private Category category;

public Product(String name, int price, Category category) {

super();

this.name = name;

this.price = price;

this.category = category;

}

public Product() {}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getPrice() {

return price;

}

public void setPrice(int price) {

this.price = price;

}

public Category getCategory() {

return category;

}

public void setCategory(Category category) {

this.category = category;

}

@Override

public String toString() {

return "Product [id=" + id + ", name=" + name + ", price=" + price + ", category=" + category + "]";

}

}

Create Purchase.java

package com.sportyshoes.entities;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToOne;

@Entity

public class Purchase {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String date;

@ManyToOne

private User user;

@ManyToOne

private Product product;

public Purchase(String date, int price, User user, Product product) {

super();

this.date = date;

this.user = user;

this.product = product;

}

public Purchase() {}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getDate() {

return date;

}

public void setDate(String date) {

this.date = date;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public Product getProduct() {

return product;

}

public void setProduct(Product product) {

this.product = product;

}

@Override

public String toString() {

return "Purchase [id=" + id + ", date=" + date + ", user=" + user + ", product=" + product

+ "]";

}

}

Create User.java

package com.sportyshoes.entities;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String email;

private String password;

private String contact;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getContact() {

return contact;

}

public void setContact(String contact) {

this.contact = contact;

}

public User(String name, String email, String password, String contact) {

super();

this.name = name;

this.email = email;

this.password = password;

this.contact = contact;

}

public User() {}

@Override

public String toString() {

return "User [id=" + id + ", name=" + name + ", email=" + email + ", password=" + password + ", contact="

+ contact + "]";

}

}

Create package com.sportyshoes.repo

Create AdminRepository.java

package com.sportyshoes.repo;

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

import org.springframework.stereotype.Repository;

import com.sportyshoes.entities.Admin;

@Repository

public interface AdminRepository extends JpaRepository<Admin, Integer>{

}

Create CategoryRepository.java

package com.sportyshoes.repo;

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

import org.springframework.stereotype.Repository;

import com.sportyshoes.entities.Category;

@Repository

public interface CategoryRepository extends JpaRepository<Category, Integer>{

}

Create ProductRepository.java

package com.sportyshoes.repo;

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

import org.springframework.stereotype.Repository;

import com.sportyshoes.entities.Product;

@Repository

public interface ProductRepository extends JpaRepository<Product, Integer>{

}

Create PurchaseRepository.java

package com.sportyshoes.repo;

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

import org.springframework.stereotype.Repository;

import com.sportyshoes.entities.Purchase;

@Repository

public interface PurchaseRepository extends JpaRepository<Purchase, Integer>{

}

Create UserRepository.java

package com.sportyshoes.repo;

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

import org.springframework.stereotype.Repository;

import com.sportyshoes.entities.User;

@Repository

public interface UserRepository extends JpaRepository<User, Integer>{

}

Create package com.sportyshoes.response

Create CategoryResponse.java

package com.sportyshoes.response;

public class CategoryResponse {

private int id;

private String name;

private String details;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDetails() {

return details;

}

public void setDetails(String details) {

this.details = details;

}

public CategoryResponse() {}

@Override

public String toString() {

return "CategoryResponse [id=" + id + ", name=" + name + ", details=" + details + "]";

}

}

Create ProductResponse.java

package com.sportyshoes.response;

import com.sportyshoes.entities.Category;

public class ProductResponse {

private int id;

private String name;

private int price;

private Category category;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getPrice() {

return price;

}

public void setPrice(int price) {

this.price = price;

}

public Category getCategory() {

return category;

}

public void setCategory(Category category) {

this.category = category;

}

public ProductResponse() {}

@Override

public String toString() {

return "ProductResponse [id=" + id + ", name=" + name + ", price=" + price + ", category=" + category + "]";

}

}

Create UserResponse.java

package com.sportyshoes.response;

public class UserResponse {

private int id;

private String name;

private String email;

private String password;

private String contact;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getContact() {

return contact;

}

public void setContact(String contact) {

this.contact = contact;

}

public UserResponse(int id, String name, String email, String password, String contact) {

super();

this.id = id;

this.name = name;

this.email = email;

this.password = password;

this.contact = contact;

}

public UserResponse() {

}

@Override

public String toString() {

return "UserResponse [id=" + id + ", name=" + name + ", email=" + email + ", password=" + password

+ ", contact=" + contact + "]";

}

}

Create package com.sportyshoes.services

Create AdminService.java

package com.sportyshoes.services;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.entities.Admin;

import com.sportyshoes.repo.AdminRepository;

@Service

public class AdminService {

@Autowired

private AdminRepository adminRepository;

public Boolean verifyAdmin(String email, String password)

{

List<Admin> admins = adminRepository.findAll();

for(Admin admin : admins)

{

if(admin.getEmail().equals(email) && admin.getPassword().equals(password))

{

return true;

}

}

return false;

}

public void changePassword(int id, String newPassword)

{

Admin admin = adminRepository.findById(id).get();

admin.setPassword(newPassword);

adminRepository.save(admin);

System.out.println("Password Changed Successfully");

}

}

Create CategoryService.java

package com.sportyshoes.services;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.entities.Category;

import com.sportyshoes.repo.CategoryRepository;

import com.sportyshoes.response.CategoryResponse;

@Service

public class CategoryService {

@Autowired

private CategoryRepository categoryRepository;

public Category addCategory(Category category) {

return categoryRepository.save(category);

}

public void deleteCategory(int id)

{

categoryRepository.deleteById(id);

}

public Boolean updateCategory(int id, Category category)

{

Category cat1 = categoryRepository.findById(id).get();

if(cat1.getId()==0)

{

return false;

}

else

{

cat1 = category;

categoryRepository.save(cat1);

return true;

}

}

public CategoryResponse searchCategory(int id)

{

CategoryResponse cr = new CategoryResponse();

Optional<Category> searchedCategory = categoryRepository.findById(id);

if(searchedCategory.isPresent())

{

Category cat = searchedCategory.get();

cr.setId(cat.getId());

cr.setName(cat.getName());

cr.setDetails(cat.getDetails());

}

return cr;

}

}

Create ProductService.java

package com.sportyshoes.services;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.entities.Product;

import com.sportyshoes.repo.ProductRepository;

import com.sportyshoes.response.ProductResponse;

@Service

public class ProductService {

@Autowired

private ProductRepository productRepository;

public Product addProduct(Product product) {

return productRepository.save(product);

}

public void deleteProduct(int id)

{

productRepository.deleteById(id);

}

public Boolean updateProduct(int id, Product product)

{

Product prod = productRepository.findById(id).get();

if(prod.getId()==0)

{

return false;

}

else

{

prod = product;

productRepository.save(prod);

return true;

}

}

public List<Product> getAllProducts()

{

List<Product> prods = productRepository.findAll();

return prods;

}

public ProductResponse searchProduct(int id)

{

ProductResponse pr = new ProductResponse();

Optional<Product> searchedProduct = productRepository.findById(id);

if(searchedProduct.isPresent())

{

Product prod = searchedProduct.get();

pr.setId(prod.getId());

pr.setName(prod.getName());

pr.setPrice(prod.getPrice());

pr.setCategory(prod.getCategory());

}

return pr;

}

}

Create PurchaseService.java

package com.sportyshoes.services;

import java.util.ArrayList;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.entities.Purchase;

import com.sportyshoes.repo.PurchaseRepository;

@Service

public class PurchaseService {

@Autowired

private PurchaseRepository purchaseRepository;

public Purchase addPurchase(Purchase purchase)

{

return purchaseRepository.save(purchase);

}

public List<Purchase> getAllProducts()

{

List<Purchase> purchases = purchaseRepository.findAll();

return purchases;

}

public List<Purchase> getProductsByCategory(int categoryId)

{

List<Purchase> purchase = purchaseRepository.findAll();

List<Purchase> purchasedByCategory = new ArrayList<>();

for(Purchase pur : purchase)

{

if(pur.getProduct().getCategory().getId()==categoryId)

{

purchasedByCategory.add(pur);

}

}

return purchasedByCategory;

}

}

Create UserService.java

package com.sportyshoes.services;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.entities.User;

import com.sportyshoes.repo.UserRepository;

import com.sportyshoes.response.UserResponse;

@Service

public class UserService {

@Autowired

private UserRepository userRepository;

public List<User> getAllUsers()

{

List<User> users = userRepository.findAll();

return users;

}

public UserResponse searchUser(int id)

{

UserResponse ur = new UserResponse();

Optional<User> searchedUser = userRepository.findById(id);

if(searchedUser.isPresent())

{

User us = searchedUser.get();

ur.setId(us.getId());

ur.setContact(us.getContact());

ur.setEmail(us.getEmail());

ur.setName(us.getName());

ur.setPassword(us.getPassword());

}

return ur;

}

public void deleteUser(int id)

{

userRepository.deleteById(id);

}

public User addUser(User person) {

return userRepository.save(person);

}

}

Create adminLogin.html

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Admin Login</title>

</head>

<body>

<form action="/api/" method="post">

Enter Email : <input type="email" name="email"><br><br>

Enter Password : <input type="password" name="password"><br><br>

<input type="submit" value="Login">

</form>

</body>

</html>

Create applicationproperties.java

server.port=8969

spring.datasource.username=root

spring.datasource.password=Vinayak@123

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

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.mvc.view.prefix: /static/

spring.mvc.view.suffix: .html

src/main/test

package com.sportyshoes.main;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class SportyShoesApplicationTests {

@Test

void contextLoads() {

}

}

Open pom.xml and add dependencies

<?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>2.6.4</version>

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

</parent>

<groupId>com.sportyshoes</groupId>

<artifactId>Sporty\_Shoes</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Sporty\_Shoes</name>

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

<properties>

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

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<scope>runtime</scope>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<dependency>

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

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

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>