**SOURCE CODE OF SPORTYSHOES API**

**Util package:**

package com.sportyshoes.utils;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import javax.annotation.PostConstruct;

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

import org.springframework.stereotype.Component;

@Component

public class DatabaseConnection {

private Connection connection = null;

DatabaseConnection(

@Value("${db.url}")String url,

@Value("${db.username}")String username,

@Value("${db.password}")String password) throws SQLException {

super();

this.connection = DriverManager.getConnection(url, username, password);

}

@PostConstruct

public void init() throws SQLException {

printConnectionDetails();

}

@SuppressWarnings("unused")

private void printConnectionDetails() throws SQLException {

System.out.println("Connected to database server "

+ this.connection.getMetaData().getDatabaseProductName()

+ " version: "

+ this.connection.getMetaData().getDatabaseProductVersion()

+ "\n");

}

public Connection getConnection() {

return connection;

}

}

**Model package:**

package com.sportyshoes.models;

import io.swagger.annotations.ApiModel;

import io.swagger.annotations.ApiModelProperty;

@ApiModel(description="Details about the user")

public class User {

@ApiModelProperty(notes="The unique userId of the user")

Integer userId;

@ApiModelProperty(notes="The username of the user")

String username;

@ApiModelProperty(notes="The password of the user")

String password;

@ApiModelProperty(notes="The name of the user")

String name;

public User() {

super();

}

public User(Integer userId, String username, String password, String name) {

super();

this.userId = userId;

this.username = username;

this.password = password;

this.name=name;

}

public User(String username, String password) {

super();

this.username = username;

this.password = password;

}

public User(Integer userId) {

super();

this.userId = userId;

}

public User(String name) {

super();

this.name = name;

}

public Integer getUserId() {

return userId;

}

public void setUserId(Integer 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 getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "User [userId=" + userId + ", username=" + username + ", password=" + password + ",name= "+ name +"]";

}

}

package com.sportyshoes.models;

import io.swagger.annotations.ApiModel;

import io.swagger.annotations.ApiModelProperty;

@ApiModel(description="Details about the admin")

public class Admin {

@ApiModelProperty(notes="The unique adminId of the admin")

Integer adminId;

@ApiModelProperty(notes="The name of the admin")

String adminName;

@ApiModelProperty(notes="The password of the admin")

String password;

public Admin() {

super();

}

public Admin(Integer adminId) {

super();

this.adminId = adminId;

}

public Admin(String adminName, String password) {

super();

this.adminName = adminName;

this.password = password;

}

public Admin(Integer adminId, String adminName, String password) {

super();

this.adminId = adminId;

this.adminName = adminName;

this.password = password;

}

public Integer getAdminId() {

return adminId;

}

public void setAdminId(Integer adminId) {

this.adminId = adminId;

}

public String getAdminName() {

return adminName;

}

public void setAdminName(String adminName) {

this.adminName = adminName;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

@Override

public String toString() {

return "Admin [adminId=" + adminId + ", adminName=" + adminName + ", password=" + password + "]";

}

}

package com.sportyshoes.models;

import io.swagger.annotations.ApiModel;

import io.swagger.annotations.ApiModelProperty;

@ApiModel(description="Details about the product")

public class Product {

@ApiModelProperty(notes="The unique productId of the product")

Integer productId;

@ApiModelProperty(notes="The name of the product")

String productName;

@ApiModelProperty(notes="The MSRP of the product")

Double MSRP;

@ApiModelProperty(notes="The quantinty in stock of the product")

Integer quantityInStock;

@ApiModelProperty(notes="The product vendor ")

String productVendor;

public Product() {

super();

}

public Product(Integer productId) {

super();

this.productId = productId;

}

public Product(String productName) {

super();

this.productName = productName;

}

public Product(Double mSRP) {

super();

MSRP = mSRP;

}

public Product(Integer productId, String productName, Double mSRP, Integer quantityInStock, String productVendor) {

super();

this.productId = productId;

this.productName = productName;

MSRP = mSRP;

this.quantityInStock = quantityInStock;

this.productVendor = productVendor;

}

public Integer getProductId() {

return productId;

}

public void setProductId(Integer productId) {

this.productId = productId;

}

public String getProductName() {

return productName;

}

public void setProductName(String productName) {

this.productName = productName;

}

public Double getMSRP() {

return MSRP;

}

public void setMSRP(Double mSRP) {

MSRP = mSRP;

}

public Integer getQuantityInStock() {

return quantityInStock;

}

public void setQuantityInStock(Integer quantityInStock) {

this.quantityInStock = quantityInStock;

}

public String getProductVendor() {

return productVendor;

}

public void setProductVendor(String productVendor) {

this.productVendor = productVendor;

}

@Override

public String toString() {

return "Product [productId=" + productId + ", productName=" + productName + ", MSRP=" + MSRP

+ ", quantityInStock=" + quantityInStock + ", productVendor=" + productVendor + "]";

}

}

package com.sportyshoes.models;

import java.sql.Date;

import io.swagger.annotations.ApiModel;

import io.swagger.annotations.ApiModelProperty;

@ApiModel(description="Details about the order")

public class Order {

@ApiModelProperty(notes="The unique orderId of the order")

Integer orderId;

@ApiModelProperty(notes="The unique productId of the product")

Integer productId;

@ApiModelProperty(notes="Datetime of purchase of product")

Date date;

@ApiModelProperty(notes="The unique userId of the user")

Integer userId;

public Order() {

super();

}

public Order(Integer productId) {

super();

this.productId = productId;

}

public Order(Integer orderId, Integer productId, Date date, Integer userId) {

super();

this.orderId = orderId;

this.productId = productId;

this.date = date;

this.userId = userId;

}

public Integer getOrderId() {

return orderId;

}

public void setOrderId(Integer orderId) {

this.orderId = orderId;

}

public Integer getProductId() {

return productId;

}

public void setProductId(Integer productId) {

this.productId = productId;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

public Integer getUserId() {

return userId;

}

public void setUserId(Integer userId) {

this.userId = userId;

}

@Override

public String toString() {

return "Order [orderId=" + orderId + ", productId=" + productId + ", date=" + date + ", userId=" + userId + "]";

}

}

**Daos package:**

package com.sportyshoes.daos;

import java.sql.SQLException;

import java.util.ArrayList;

import com.sportyshoes.models.User;

public interface UserDao {

// Create Operation

Integer createUser(User user) throws SQLException ;

// Read/Retrieve Operations

User findUserByUserNameandPassword(String userName, String password) throws SQLException ;

User findUserByName(String name) throws SQLException ;

ArrayList<User> findAllUsers() throws SQLException;

// Update Operations

Integer updateUser(User user) throws SQLException ;

// Delete Operations

Integer deleteUserByUserId(Integer userId) throws SQLException;

User findUserByUserId(Integer userId) throws SQLException;

Integer updateUserName(User user) throws SQLException;

Integer updateUserPassword(User user) throws SQLException;

}

package com.sportyshoes.daos;

import java.sql.SQLException;

import java.util.ArrayList;

import com.sportyshoes.models.Admin;

public interface AdminDao {

// Create Operation

Integer createAdmin(Admin admin) throws SQLException ;

// Read/Retrieve Operations

Admin findAdminByAdminNameandPassword(String adminName, String password) throws SQLException ;

ArrayList<Admin> findAllAdmin() throws SQLException;

// Update Operations

Integer updateAdmin(Admin admin) throws SQLException ;

Integer updateAdminName(Admin admin) throws SQLException;

Integer updateAdminPassword(Admin admin) throws SQLException;

// Delete Operations

Integer deleteAdminByAdminId(Integer adminId) throws SQLException;

}

package com.sportyshoes.daos;

import java.sql.SQLException;

import java.util.ArrayList;

import com.sportyshoes.models.Admin;

import com.sportyshoes.models.Product;

public interface ProductDao {

// Create Operation

Integer createProduct(Product product) throws SQLException ;

// Read/Retrieve Operations

Product findProductByName(String productName) throws SQLException ;

Product findProductByProductId(Integer productId) throws SQLException ;

ArrayList<Product> findAllProduct() throws SQLException;

// Update Operations

Integer updateProductName(Product product) throws SQLException ;

Integer updateProductMSRP(Product product) throws SQLException;

Integer updateQuantityInStock(Product product) throws SQLException;

Integer updateProductVendor(Product product) throws SQLException;

// Delete Operations

Integer deleteProductByProductId(Integer productId) throws SQLException;

}

package com.sportyshoes.daos;

import java.sql.SQLException;

import java.util.ArrayList;

import com.sportyshoes.models.Order;

import com.sportyshoes.models.User;

public interface OrderDao {

Integer createOrder(Order order) throws SQLException;

ArrayList<Order> findAllOrdersByDateASC() throws SQLException;

ArrayList<Order> findAllOrdersByDateDESC() throws SQLException;

ArrayList<Order> findAllOrdersByOrderIdASC() throws SQLException;

ArrayList<Order> findAllOrdersByOrderIdDESC() throws SQLException;

Order findOrderByProductId(Integer productId) throws SQLException ;

Order findOrderByUserId(Integer userId) throws SQLException ;

}

**Repository package:**

package com.sportyshoes.repositories;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Repository;

import com.sportyshoes.daos.UserDao;

import com.sportyshoes.models.User;

import com.sportyshoes.utils.DatabaseConnection;

@Repository

public class UserRepository implements UserDao{

@Autowired

private DatabaseConnection connection;

@Override

public Integer createUser(User user) throws SQLException {

String insertUserFormat = "INSERT INTO users (USER\_ID, PASSWORD, USER\_NAME, NAME) VALUES ( ?, ?, ?. ? )";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(insertUserFormat);) {

preparedStatement.setInt(1, user.getUserId());

preparedStatement.setString(2, user.getPassword());

preparedStatement.setString(3, user.getUsername());

preparedStatement.setString(4, user.getName());

return preparedStatement.executeUpdate();

}

}

@Override

public User findUserByUserNameandPassword(String userName, String password) throws SQLException {

// Read (Retrieve) Operation using PreparedStatement

String getUserFormat = "SELECT \*FROM users WHERE USER\_NAME = ? and PASSWORD=?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getUserFormat);) {

preparedStatement.setString(1, userName);

preparedStatement.setString(2, password);

ResultSet rs = preparedStatement.executeQuery();

if (!rs.isBeforeFirst()) {

return null;

}

User user = new User();

while (rs.next()) {

user.setUserId(rs.getInt("USER\_ID"));

user.setPassword(rs.getString("PASSWORD"));

user.setUsername(rs.getString("USER\_NAME"));

user.setName(rs.getString("NAME"));

}

return user;

}

}

@Override

public User findUserByUserId(Integer userId) throws SQLException {

// Read (Retrieve) Operation using PreparedStatement

String getUserFormat = "SELECT \*FROM users WHERE USER\_ID = ? ";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getUserFormat);) {

preparedStatement.setInt(1, userId);

ResultSet rs = preparedStatement.executeQuery();

if (!rs.isBeforeFirst()) {

return null;

}

User user = new User();

while (rs.next()) {

user.setUserId(rs.getInt("USER\_ID"));

user.setPassword(rs.getString("PASSWORD"));

user.setUsername(rs.getString("USER\_NAME"));

user.setName(rs.getString("NAME"));

}

return user;

}

}

@Override

public User findUserByName(String name) throws SQLException {

String getUserFormat = "SELECT \*FROM users WHERE NAME = ? ";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getUserFormat);) {

preparedStatement.setString(1, name);

ResultSet rs = preparedStatement.executeQuery();

if (!rs.isBeforeFirst()) {

return null;

}

User user = new User();

while (rs.next()) {

user.setUserId(rs.getInt("USER\_ID"));

user.setPassword(rs.getString("PASSWORD"));

user.setUsername(rs.getString("USER\_NAME"));

user.setName(rs.getString("NAME"));

}

return user;

}

}

@Override

public ArrayList<User> findAllUsers() throws SQLException {

String getUsersFormat = "SELECT \*FROM users";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getUsersFormat);) {

ResultSet rs = preparedStatement.executeQuery();

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

if (!rs.isBeforeFirst()) {

return users;

}

while (rs.next()) {

User user = new User();

user.setUserId(rs.getInt("User\_ID"));

user.setPassword(rs.getString("PASSWORD"));

user.setUsername(rs.getString("USER\_NAME"));

user.setName(rs.getString("NAME"));

users.add(user);

}

return users;

}

}

@Override

public Integer updateUser(User user) throws SQLException {

String updateUserFormat = "UPDATE users SET PASSWORD = ?, USER\_NAME = ?, WHERE USER\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(updateUserFormat);) {

preparedStatement.setString(1, user.getPassword());

preparedStatement.setString(2, user.getUsername());

return preparedStatement.executeUpdate();

}

}

@Override

public Integer updateUserPassword(User user) throws SQLException {

String updateUserPasswodFormat = "UPDATE users SET PASSWORD = ?, WHERE USER\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(updateUserPasswodFormat);) {

preparedStatement.setString(1, user.getPassword());

return preparedStatement.executeUpdate();

}

}

@Override

public Integer updateUserName(User user) throws SQLException {

String updateUserNameFormat = "UPDATE users SET USER\_NAME = ?, WHERE USER\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(updateUserNameFormat);) {

preparedStatement.setString(1, user.getUsername());

return preparedStatement.executeUpdate();

}

}

@Override

public Integer deleteUserByUserId(Integer userId) throws SQLException {

String deleteUserFormat = " DELETE FROM users WHERE USER\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(deleteUserFormat);) {

preparedStatement.setInt(1, userId);

return preparedStatement.executeUpdate();

}

}

}

package com.sportyshoes.repositories;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Repository;

import com.sportyshoes.daos.AdminDao;

import com.sportyshoes.models.Admin;

import com.sportyshoes.utils.DatabaseConnection;

@Repository

public class AdminRepository implements AdminDao {

@Autowired

private DatabaseConnection connection;

@Override

public Integer createAdmin(Admin admin) throws SQLException {

String insertAdminFormat = "INSERT INTO admins (ADMIN\_ID, PASSWORD, ADMIN\_NAME) VALUES ( ?, ?, ? )";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(insertAdminFormat);) {

preparedStatement.setInt(1, admin.getAdminId());

preparedStatement.setString(2, admin.getPassword());

preparedStatement.setString(3, admin.getAdminName());

return preparedStatement.executeUpdate();

}

}

@Override

public Admin findAdminByAdminNameandPassword(String adminName, String password) throws SQLException {

// Read (Retrieve) Operation using PreparedStatement

String getAdminFormat = "SELECT \*FROM admins WHERE ADMIN\_NAME = ? and PASSWORD=?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getAdminFormat);) {

preparedStatement.setString(1, adminName);

preparedStatement.setString(2, password);

ResultSet rs = preparedStatement.executeQuery();

if (!rs.isBeforeFirst()) {

return null;

}

Admin admin = new Admin();

while (rs.next()) {

admin.setAdminId(rs.getInt("ADMIN\_ID"));

admin.setPassword(rs.getString("PASSWORD"));

admin.setAdminName(rs.getString("ADMIN\_NAME"));

}

return admin;

}

}

@Override

public ArrayList<Admin> findAllAdmin() throws SQLException {

String getAdminsFormat = "SELECT \*FROM admins";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getAdminsFormat);) {

ResultSet rs = preparedStatement.executeQuery();

ArrayList<Admin> admins = new ArrayList<>();

if (!rs.isBeforeFirst()) {

return admins;

}

while (rs.next()) {

Admin admin = new Admin();

admin.setAdminId(rs.getInt("ADMIN\_ID"));

admin.setPassword(rs.getString("PASSWORD"));

admin.setAdminName(rs.getString("ADMIN\_NAME"));

admins.add(admin);

}

return admins;

}

}

@Override

public Integer updateAdmin(Admin admin) throws SQLException {

String updateAdminFormat = "UPDATE admins SET PASSWORD = ?, ADMIN\_NAME = ?, WHERE ADMIN\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(updateAdminFormat);) {

preparedStatement.setString(1, admin.getPassword());

preparedStatement.setString(2, admin.getAdminName());

return preparedStatement.executeUpdate();

}

}

@Override

public Integer updateAdminName(Admin admin) throws SQLException {

String updateAdminNameFormat = "UPDATE admins SET ADMIN\_NAME = ?, WHERE ADMIN\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(updateAdminNameFormat);) {

preparedStatement.setString(1, admin.getAdminName());

return preparedStatement.executeUpdate();

}

}

@Override

public Integer updateAdminPassword(Admin admin) throws SQLException {

String updateAdminPasswodFormat = "UPDATE admins SET PASSWORD = ?, WHERE ADMIN\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(updateAdminPasswodFormat);) {

preparedStatement.setString(1, admin.getPassword());

return preparedStatement.executeUpdate();

}

}

@Override

public Integer deleteAdminByAdminId(Integer adminId) throws SQLException {

String deleteAdminFormat = " DELETE FROM admins WHERE ADMIN\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(deleteAdminFormat);) {

preparedStatement.setInt(1, adminId);

return preparedStatement.executeUpdate();

}

}

}

package com.sportyshoes.repositories;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Repository;

import com.sportyshoes.daos.ProductDao;

import com.sportyshoes.models.Product;

import com.sportyshoes.utils.DatabaseConnection;

@Repository

public class ProductRepository implements ProductDao{

@Autowired

private DatabaseConnection connection;

@Override

public Integer createProduct(Product product) throws SQLException {

String insertProductFormat = "INSERT INTO products (PRODUCT\_ID, PRODUCT\_NAME, MSRP, QUANITY\_IN\_STOCK, PRODUCT\_VENDOR) VALUES ( ?, ?, ?, ?, ? )";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(insertProductFormat);) {

preparedStatement.setInt(1, product.getProductId());

preparedStatement.setString(2, product.getProductName());

preparedStatement.setDouble(3, product.getMSRP());

preparedStatement.setInt(4, product.getQuantityInStock());

preparedStatement.setString(5, product.getProductVendor());

return preparedStatement.executeUpdate();

}

}

@Override

public Product findProductByName(String productName) throws SQLException {

// Read (Retrieve) Operation using PreparedStatement

String getProductFormat = "SELECT \*FROM products WHERE PRODUCT\_NAME = ? ";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getProductFormat);) {

preparedStatement.setString(1, productName);

ResultSet rs = preparedStatement.executeQuery();

if (!rs.isBeforeFirst()) {

return null;

}

Product product = new Product();

while (rs.next()) {

product.setProductId(rs.getInt("PRODUCT\_ID"));

product.setProductName(rs.getString("PRODUCT\_NAME"));

product.setMSRP(rs.getDouble("MSRP"));

product.setQuantityInStock(rs.getInt("QUANTITY\_IN\_STOCK"));

product.setProductVendor(rs.getString("PRODUCT\_VENDOR"));

}

return product;

}

}

@Override

public Product findProductByProductId(Integer productId) throws SQLException {

String getProductFormat = "SELECT \*FROM products WHERE PRODUCT\_ID = ? ";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getProductFormat);) {

preparedStatement.setInt(1, productId);

ResultSet rs = preparedStatement.executeQuery();

if (!rs.isBeforeFirst()) {

return null;

}

Product product = new Product();

while (rs.next()) {

product.setProductId(rs.getInt("PRODUCT\_ID"));

product.setProductName(rs.getString("PRODUCT\_NAME"));

product.setMSRP(rs.getDouble("MSRP"));

product.setQuantityInStock(rs.getInt("QUANTITY\_IN\_STOCK"));

product.setProductVendor(rs.getString("PRODUCT\_VENDOR"));

}

return product;

}

}

@Override

public ArrayList<Product> findAllProduct() throws SQLException {

String getProductsFormat = "SELECT \*FROM products";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getProductsFormat);) {

ResultSet rs = preparedStatement.executeQuery();

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

if (!rs.isBeforeFirst()) {

return products;

}

while (rs.next()) {

Product product = new Product();

product.setProductId(rs.getInt("PRODUCT\_ID"));

product.setProductName(rs.getString("PRODUCT\_NAME"));

product.setMSRP(rs.getDouble("MSRP"));

product.setQuantityInStock(rs.getInt("QUANTITY\_IN\_STOCK"));

product.setProductVendor(rs.getString("PRODUCT\_VENDOR"));

products.add(product);

}

return products;

}

}

@Override

public Integer updateProductName(Product product) throws SQLException {

String updateProductNameFormat = "UPDATE products SET PRODUCT\_NAME = ?, WHERE PRODUCT\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(updateProductNameFormat);) {

preparedStatement.setString(1, product.getProductName());

return preparedStatement.executeUpdate();

}

}

@Override

public Integer updateProductMSRP(Product product) throws SQLException {

String updateMSRPFormat = "UPDATE products SET MSRP = ?, WHERE PRODUCT\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(updateMSRPFormat);) {

preparedStatement.setDouble(1, product.getMSRP());

return preparedStatement.executeUpdate();

}

}

@Override

public Integer updateQuantityInStock(Product product) throws SQLException {

String updateuantityInStockFormat = "UPDATE products SET QUANITY\_IN\_STOCK = ?, WHERE PRODUCT\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(updateuantityInStockFormat);) {

preparedStatement.setInt(1, product.getQuantityInStock());

return preparedStatement.executeUpdate();

}

}

@Override

public Integer updateProductVendor(Product product) throws SQLException {

String updateProductVendorFormat = "UPDATE products SET PRODUCT\_VENDOR = ?, WHERE PRODUCT\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(updateProductVendorFormat);) {

preparedStatement.setString(1, product.getProductVendor());

return preparedStatement.executeUpdate();

}

}

@Override

public Integer deleteProductByProductId(Integer productId) throws SQLException {

String deleteProductFormat = " DELETE FROM products WHERE PRODUCT\_ID = ?";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(deleteProductFormat);) {

preparedStatement.setInt(1, productId);

return preparedStatement.executeUpdate();

}

}

}

package com.sportyshoes.repositories;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Repository;

import com.sportyshoes.daos.OrderDao;

import com.sportyshoes.models.Order;

import com.sportyshoes.models.Product;

import com.sportyshoes.utils.DatabaseConnection;

@Repository

public class OrderRepository implements OrderDao{

@Autowired

private DatabaseConnection connection;

@Override

public Integer createOrder(Order order) throws SQLException {

String insertOrderFormat = "INSERT INTO orders (ORDER\_ID, REF\_PRODUCT\_ID, DATE, REF\_USER\_ID) VALUES ( ?, ?, ?, ?)";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(insertOrderFormat);) {

preparedStatement.setInt(1, order.getOrderId());

preparedStatement.setInt(2, order.getProductId());

preparedStatement.setDate(3, order.getDate());

preparedStatement.setInt(4, order.getUserId());

return preparedStatement.executeUpdate();

}

}

@Override

public ArrayList<Order> findAllOrdersByDateASC() throws SQLException {

String getOrdersFormat = "SELECT \*FROM orders ORDER BY DATE ASC";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getOrdersFormat);) {

ResultSet rs = preparedStatement.executeQuery();

ArrayList<Order> orders = new ArrayList<>();

if (!rs.isBeforeFirst()) {

return orders;

}

while (rs.next()) {

Order order = new Order();

order.setOrderId(rs.getInt("ORDER\_ID"));

order.setProductId(rs.getInt("REF\_PRODUCT\_ID"));

order.setDate(rs.getDate("DATE"));

order.setUserId(rs.getInt("REF\_USER\_ID"));

orders.add(order);

}

return orders;

}

}

@Override

public ArrayList<Order> findAllOrdersByDateDESC() throws SQLException {

String getOrdersFormat = "SELECT \*FROM orders ORDER BY DATE DESC";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getOrdersFormat);) {

ResultSet rs = preparedStatement.executeQuery();

ArrayList<Order> orders = new ArrayList<>();

if (!rs.isBeforeFirst()) {

return orders;

}

while (rs.next()) {

Order order = new Order();

order.setOrderId(rs.getInt("ORDER\_ID"));

order.setProductId(rs.getInt("REF\_PRODUCT\_ID"));

order.setDate(rs.getDate("DATE"));

order.setUserId(rs.getInt("REF\_USER\_ID"));

orders.add(order);

}

return orders;

}

}

@Override

public ArrayList<Order> findAllOrdersByOrderIdASC() throws SQLException {

String getOrdersFormat = "SELECT \*FROM orders ORDER BY ORDER\_ID ASC";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getOrdersFormat);) {

ResultSet rs = preparedStatement.executeQuery();

ArrayList<Order> orders = new ArrayList<>();

if (!rs.isBeforeFirst()) {

return orders;

}

while (rs.next()) {

Order order = new Order();

order.setOrderId(rs.getInt("ORDER\_ID"));

order.setProductId(rs.getInt("REF\_PRODUCT\_ID"));

order.setDate(rs.getDate("DATE"));

order.setUserId(rs.getInt("REF\_USER\_ID"));

orders.add(order);

}

return orders;

}

}

@Override

public ArrayList<Order> findAllOrdersByOrderIdDESC() throws SQLException {

String getOrdersFormat = "SELECT \*FROM orders ORDER BY ORDER\_ID DESC";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getOrdersFormat);) {

ResultSet rs = preparedStatement.executeQuery();

ArrayList<Order> orders = new ArrayList<>();

if (!rs.isBeforeFirst()) {

return orders;

}

while (rs.next()) {

Order order = new Order();

order.setOrderId(rs.getInt("ORDER\_ID"));

order.setProductId(rs.getInt("REF\_PRODUCT\_ID"));

order.setDate(rs.getDate("DATE"));

order.setUserId(rs.getInt("REF\_USER\_ID"));

orders.add(order);

}

return orders;

}

}

@Override

public Order findOrderByProductId(Integer productId) throws SQLException {

String getOrderFormat = "SELECT \*FROM orders WHERE REF\_PRODUCT\_ID = ? ";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getOrderFormat);) {

preparedStatement.setInt(1, productId);

ResultSet rs = preparedStatement.executeQuery();

if (!rs.isBeforeFirst()) {

return null;

}

Order order = new Order();

while (rs.next()) {

order.setOrderId(rs.getInt("ORDER\_ID"));

order.setProductId(rs.getInt("REF\_PRODUCT\_ID"));

order.setDate(rs.getDate("DATE"));

order.setUserId(rs.getInt("REF\_USER\_ID"));

}

return order;

}

}

@Override

public Order findOrderByUserId(Integer userId) throws SQLException {

String getOrderFormat = "SELECT \*FROM orders WHERE REF\_USER\_ID = ? ";

try (PreparedStatement preparedStatement = connection.getConnection().prepareStatement(getOrderFormat);) {

preparedStatement.setInt(1, userId);

ResultSet rs = preparedStatement.executeQuery();

if (!rs.isBeforeFirst()) {

return null;

}

Order order = new Order();

while (rs.next()) {

order.setOrderId(rs.getInt("ORDER\_ID"));

order.setProductId(rs.getInt("REF\_PRODUCT\_ID"));

order.setDate(rs.getDate("DATE"));

order.setUserId(rs.getInt("REF\_USER\_ID"));

}

return order;

}

}

}

**Service package:**

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Admin;

import com.sportyshoes.repositories.AdminRepository;

@Service

public class AdminAuthService {

@Autowired

private AdminRepository adminRepository;

public Admin SignInAdmin(String adminName,String password) {

Admin admin= new Admin(adminName, password);

try {

return adminRepository.findAdminByAdminNameandPassword(adminName, password);

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Admin;

import com.sportyshoes.repositories.AdminRepository;

@Service

public class AdminService {

@Autowired

private AdminRepository adminRepository;

public Boolean AddAdmin(Integer adminId,String adminName,String password) {

Admin admin= new Admin(adminId, adminName, password);

try {

return adminRepository.createAdmin(admin)>0;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Admin;

import com.sportyshoes.repositories.AdminRepository;

@Service

public class DeleteAdminService {

@Autowired

private AdminRepository adminRepository;

public Boolean DeleteAdmin(Integer adminId) {

Admin admin= new Admin(adminId);

try {

return adminRepository.deleteAdminByAdminId(adminId)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Product;

import com.sportyshoes.repositories.ProductRepository;

@Service

public class DeleteProductService {

@Autowired

private ProductRepository productRepository;

public Boolean DeleteProduct(Integer productId) {

Product product= new Product(productId);

try {

return productRepository.deleteProductByProductId(productId)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.User;

import com.sportyshoes.repositories.UserRepository;

@Service

public class DeleteUserService {

@Autowired

private UserRepository userRepository;

public Boolean DeleteUser(Integer userId) {

User user= new User(userId);

try {

return userRepository.deleteUserByUserId(userId)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Order;

import com.sportyshoes.repositories.OrderRepository;

;

@Service

public class FindAllOrdersByDateASCService {

@Autowired

private OrderRepository orderRepository;

public ArrayList<Order> GetallOrderByDateASC() {

Order order= new Order();

try {

return orderRepository.findAllOrdersByDateASC();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Order;

import com.sportyshoes.repositories.OrderRepository;

@Service

public class FindAllOrdersByDateDESCService {

@Autowired

private OrderRepository orderRepository;

public ArrayList<Order> GetallOrderByDateDESC() {

Order order= new Order();

try {

return orderRepository.findAllOrdersByDateDESC();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Order;

import com.sportyshoes.repositories.OrderRepository;

@Service

public class FindAllOrdersByOrderIdASCService {

@Autowired

private OrderRepository orderRepository;

public ArrayList<Order> GetallOrderByIdASC() {

Order order= new Order();

try {

return orderRepository.findAllOrdersByOrderIdASC();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Order;

import com.sportyshoes.repositories.OrderRepository;

@Service

public class FindAllOrdersByOrderIdDESCService {

@Autowired

private OrderRepository orderRepository;

public ArrayList<Order> GetallOrderByIdDESC() {

Order order= new Order();

try {

return orderRepository.findAllOrdersByOrderIdDESC();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Order;

import com.sportyshoes.repositories.OrderRepository;

@Service

public class FindOrderByProductIdService {

@Autowired

private OrderRepository orderRepository;

public Order findOrderByProductId(Integer productId) {

Order order= new Order(productId);

try {

return orderRepository.findOrderByProductId(productId);

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Order;

import com.sportyshoes.repositories.OrderRepository;

@Service

public class FindOrderByUserIdService {

@Autowired

private OrderRepository orderRepository;

public Order findOrderByUserId(Integer userId) {

Order order= new Order(userId);

try {

return orderRepository.findOrderByUserId(userId);

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Product;

import com.sportyshoes.repositories.ProductRepository;

@Service

public class FindProductByNameService {

@Autowired

private ProductRepository productRepository;

public Product findProductByName(String productName) {

Product product= new Product(productName);

try {

return productRepository.findProductByName(productName);

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Product;

import com.sportyshoes.repositories.ProductRepository;

@Service

public class FindProductByProductIdService {

@Autowired

private ProductRepository productRepository;

public Product findProductByProductId(Integer productId) {

Product product= new Product(productId);

try {

return productRepository.findProductByProductId(productId);

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.User;

import com.sportyshoes.repositories.UserRepository;

@Service

public class FindUserByNameService {

@Autowired

private UserRepository userRepository;

public User findUserByName(String name) {

User user= new User(name);

try {

return userRepository.findUserByName(name);

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Admin;

import com.sportyshoes.repositories.AdminRepository;

@Service

public class GetAllAdminService {

@Autowired

private AdminRepository adminRepository;

public ArrayList<Admin> GetallAdmin() {

Admin admin= new Admin();

try {

return adminRepository.findAllAdmin();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Product;

import com.sportyshoes.repositories.ProductRepository;

@Service

public class GetAllProductService {

@Autowired

private ProductRepository productRepository;

public ArrayList<Product> GetallProduct() {

Product product= new Product();

try {

return productRepository.findAllProduct();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.User;

import com.sportyshoes.repositories.UserRepository;

@Service

public class GetAllUserService {

@Autowired

private UserRepository userRepository;

public ArrayList<User> GetallUser() {

User user= new User();

try {

return userRepository.findAllUsers();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

import java.util.Date;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Order;

import com.sportyshoes.repositories.OrderRepository;

@Service

public class OrderService {

@Autowired

private OrderRepository orderRepository;

public Boolean AddOrder(Integer orderId,Integer productId,Date date, Integer userId) {

Order order= new Order(orderId, productId, (java.sql.Date) date, userId);

try {

return orderRepository.createOrder(order)>0;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Product;

import com.sportyshoes.repositories.ProductRepository;

@Service

public class ProductService {

@Autowired

private ProductRepository productRepository;

public Boolean AddProduct(Integer productId,String productName,Double MSRP, Integer quantityInStock,String productVendor ) {

Product product= new Product(productId, productName, MSRP,quantityInStock,productVendor);

try {

return productRepository.createProduct(product)>0;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Admin;

import com.sportyshoes.repositories.AdminRepository;

@Service

public class UpdateAdminNameService {

@Autowired

private AdminRepository adminRepository;

public Boolean UpdateAdminName(Integer adminId) {

Admin admin= new Admin(adminId);

try {

return adminRepository.updateAdminName(admin)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Admin;

import com.sportyshoes.repositories.AdminRepository;

@Service

public class UpdateAdminPasswordService {

@Autowired

private AdminRepository adminRepository;

public Boolean UpdateAdminPassword(Integer adminId) {

Admin admin= new Admin(adminId);

try {

return adminRepository.updateAdminPassword(admin)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Admin;

import com.sportyshoes.repositories.AdminRepository;

@Service

public class UpdateAdminService {

@Autowired

private AdminRepository adminRepository;

public Boolean UpdateAdmin(Integer adminId) {

Admin admin= new Admin(adminId);

try {

return adminRepository.updateAdmin(admin)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Product;

import com.sportyshoes.repositories.ProductRepository;

@Service

public class UpdateProductMSRPService {

@Autowired

private ProductRepository productRepository;

public Boolean UpdateProductMSRP(Integer productId) {

Product product= new Product(productId);

try {

return productRepository.updateProductMSRP(product)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Product;

import com.sportyshoes.repositories.ProductRepository;

@Service

public class UpdateProductNameService {

@Autowired

private ProductRepository productRepository;

public Boolean UpdateProductName(Integer productId) {

Product product= new Product(productId);

try {

return productRepository.updateProductName(product)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Product;

import com.sportyshoes.repositories.ProductRepository;

@Service

public class UpdateProductVendorService {

@Autowired

private ProductRepository productRepository;

public Boolean UpdateProductVendor(Integer productId) {

Product product= new Product(productId);

try {

return productRepository.updateProductVendor(product)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.Product;

import com.sportyshoes.repositories.ProductRepository;

@Service

public class UpdateQuantityInStockService {

@Autowired

private ProductRepository productRepository;

public Boolean UpdateQuantityInStock(Integer productId) {

Product product= new Product(productId);

try {

return productRepository.updateQuantityInStock(product)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.User;

import com.sportyshoes.repositories.UserRepository;

@Service

public class UpdateUserNameService {

@Autowired

private UserRepository userRepository;

public Boolean UpdateUserName(Integer userId) {

User user= new User(userId);

try {

return userRepository.updateUserName(user)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.User;

import com.sportyshoes.repositories.UserRepository;

@Service

public class UpdateUserPasswordService {

@Autowired

private UserRepository userRepository;

public Boolean UpdateUserPassword(Integer userId) {

User user= new User(userId);

try {

return userRepository.updateUserPassword(user)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.User;

import com.sportyshoes.repositories.UserRepository;

@Service

public class UpdateUserService {

@Autowired

private UserRepository userRepository;

public Boolean UpdateUser(Integer userId) {

User user= new User(userId);

try {

return userRepository.updateUser(user)>0;

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.User;

import com.sportyshoes.repositories.UserRepository;

@Service

public class UserAccountDetailService {

@Autowired

private UserRepository userRepository;

public User AccountDetail(Integer userId) {

User user= new User(userId);

try {

return userRepository.findUserByUserId(userId);

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.User;

import com.sportyshoes.repositories.UserRepository;

@Service

public class UserAuthService {

@Autowired

private UserRepository userRepository;

public User SignInUser(String userName,String password) {

User user= new User(userName, password);

try {

return userRepository.findUserByUserNameandPassword(userName, password);

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

}

package com.sportyshoes.services;

import java.sql.SQLException;

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

import org.springframework.stereotype.Service;

import com.sportyshoes.models.User;

import com.sportyshoes.repositories.UserRepository;

@Service

public class UserService {

@Autowired

private UserRepository userRepository;

public Boolean AddUser(Integer userId,String userName,String password, String name) {

User user= new User(userId, userName, password, name);

try {

return userRepository.createUser(user)>0;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return false;

}

}

**User controller package:**

package com.sportyshoes.controllers.user;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

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

import com.sportyshoes.models.User;

import com.sportyshoes.services.UserAccountDetailService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class AccountDetailController {

@Autowired

private UserAccountDetailService accountDetailService;

@GetMapping("/{userid}")

public Map <String, String> userDetail1(@RequestBody MultiValueMap<String, String> ACDetail1) {

Map <String, String> DetailResponse1=new HashMap<>();

User userfound1= accountDetailService.AccountDetail(Integer.parseInt(ACDetail1.get("userId").get(0)));

if(userfound1 != null) {

DetailResponse1.put("status", "true");

DetailResponse1.put("massege", "User found");

}else {

DetailResponse1.put("status", "false");

DetailResponse1.put("massege", "Invalid userId");

}

return DetailResponse1;

}

}

package com.sportyshoes.controllers.user;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.time.LocalDate;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.OrderService;

import com.sportyshoes.services.ProductService;

import io.swagger.models.properties.StringProperty.Format;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user/{userId}")

public class AddOrderController {

@Autowired

private OrderService orderService;

@PostMapping("/product/{productId}/order")

public Map <String, String> AddOrderByProductId(@RequestBody MultiValueMap<String, String> newOrderDetails) {

Map <String, String> addOrderResponse=new HashMap<>();

@SuppressWarnings("deprecation")

SimpleDateFormat formatter = new SimpleDateFormat("dd-MM-yyyy");

Boolean orderAdded;

try {

orderAdded = orderService.AddOrder(Integer.parseInt(newOrderDetails.get("orderId").get(0)),Integer.parseInt(newOrderDetails.get("productId").get(0)),formatter.parse(newOrderDetails.get("date").get(0)), Integer.parseInt(newOrderDetails.get("userId").get(0)));

if(orderAdded) {

addOrderResponse.put("status", "True");

addOrderResponse.put("massege", "Order added successfully");

}else {

addOrderResponse.put("status", "False");

addOrderResponse.put("massege", "Order not added");

}

} catch (NumberFormatException | ParseException e) {

e.printStackTrace();

} return addOrderResponse;

}}

package com.sportyshoes.controllers.user;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.models.User;

import com.sportyshoes.services.UserAuthService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class AuthController {

@Autowired

private UserAuthService authService;

@PostMapping("/signIn")

public Map <String, String> signIn(@RequestBody MultiValueMap<String, String> credentials) {

Map <String, String> signInResponse=new HashMap<>();

User userAuthenticated= authService.SignInUser(credentials.get("userName").get(0), credentials.get("password").get(0));

if(userAuthenticated != null) {

signInResponse.put("status", "true");

signInResponse.put("massege", "The user has been authenticated successfully");

}else {

signInResponse.put("status", "false");

signInResponse.put("massege", "Invalid credential!");

}

return signInResponse;

}

}

package com.sportyshoes.controllers.user;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.DeleteUserService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class DeleteUserController {

@Autowired

private DeleteUserService deleteService;

@DeleteMapping("{userId}/delete")

public Map <String, String> Delete(@RequestBody MultiValueMap<String, String> delete) {

Map <String, String> userDeleteResponse=new HashMap<>();

Boolean userDeleted= deleteService.DeleteUser(Integer.parseInt(delete.get("userId").get(0)));

if(userDeleted) {

userDeleteResponse.put("status", "True");

userDeleteResponse.put("massege", "The user has been deleted successfully");

}else {

userDeleteResponse.put("status", "False");

userDeleteResponse.put("massege", "Invalid Id!");

}

return userDeleteResponse;

}}

package com.sportyshoes.controllers.user;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.models.Product;

import com.sportyshoes.models.User;

import com.sportyshoes.services.FindProductByNameService;

import com.sportyshoes.services.GetAllProductService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/product")

public class FindProductByNameController {

@Autowired

private FindProductByNameService findProductByNameService;

@GetMapping("/search?name=<product>")

public Map <String, String> ProductDetail(@RequestBody MultiValueMap<String, String> searchProductbyName) {

Map <String, String> searchProductResponse=new HashMap<>();

Product productfound= findProductByNameService.findProductByName(searchProductbyName.get("productName").get(0));

if(productfound != null) {

searchProductResponse.put("status", "true");

searchProductResponse.put("massege", "Product found");

}else {

searchProductResponse.put("status", "false");

searchProductResponse.put("massege", "Product not found");

}

return searchProductResponse;

}

}

package com.sportyshoes.controllers.user;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.models.Product;

import com.sportyshoes.services.FindProductByProductIdService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/product")

public class FindProductByProductIdController {

@Autowired

private FindProductByProductIdService findProductByProductIdService;

@GetMapping("/{productId}")

public Map <String, String> FindProduct(@RequestBody MultiValueMap<String, String> searchProductbyProductId) {

Map <String, String> searchProductResponse=new HashMap<>();

Product productfound= findProductByProductIdService.findProductByProductId(Integer.parseInt(searchProductbyProductId.get("productId").get(0)));

if(productfound != null) {

searchProductResponse.put("status", "true");

searchProductResponse.put("massege", "Product found");

}else {

searchProductResponse.put("status", "false");

searchProductResponse.put("massege", "Product not found");

}

return searchProductResponse;

}

}

package com.sportyshoes.controllers.user;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

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

import com.sportyshoes.models.Order;

import com.sportyshoes.models.Product;

import com.sportyshoes.services.GetAllProductService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/product")

public class GetAllProductController {

@Autowired

private GetAllProductService getAllProductService;

@GetMapping("/all")

@ResponseBody

public List ProductDetail() {

List<Product> allProduct=getAllProductService.GetallProduct();

return allProduct;

}

// public Map <String, String> ProductDetail(@RequestBody MultiValueMap<String, String> AllProductDetail) {

// Map <String, String> AllProductResponse=new HashMap<>();

// ArrayList<Product> allProduct= getAllProductService.GetallProduct();

// if(allProduct != null) {

// AllProductResponse.put("status", "true");

// AllProductResponse.put("massege", "All product found ");

// }else {

// AllProductResponse.put("status", "false");

// AllProductResponse.put("massege", "Product not found");

// }

// return AllProductResponse;

//

//

// }

}

package com.sportyshoes.controllers.user;

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

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

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

@RestController

@RequestMapping("/")

public class HomeController {

@GetMapping("")

public String index() {

return "Hello visitor! Welcome to Sportyshoes. Get the best sports shoes";

}

}

package com.sportyshoes.controllers.user;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.UpdateUserService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class UpdateUserController {

@Autowired

private UpdateUserService updateService;

@PostMapping("{userid}/update")

public Map <String, String> signUp(@RequestBody MultiValueMap<String, String> updateDetails) {

Map <String, String> userUpdateResponse=new HashMap<>();

Boolean userUpdated= updateService.UpdateUser(Integer.parseInt(updateDetails.get("userId").get(0)));

if(userUpdated) {

userUpdateResponse.put("status", "True");

userUpdateResponse.put("massege", "The user has been updated successfully");

}else {

userUpdateResponse.put("status", "False");

userUpdateResponse.put("massege", "Invalid Info!");

}

return userUpdateResponse;

}}

package com.sportyshoes.controllers.user;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

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

import com.sportyshoes.services.UpdateUserService;

import com.sportyshoes.services.UpdateUserNameService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class UpdateUsernameController {

@Autowired

private UpdateUserNameService updateUserNameService ;

@PatchMapping("{userId}/update/username")

public Map <String, String> Updateusername(@RequestBody MultiValueMap<String, String> updateUsername) {

Map <String, String> usernameUpdateResponse=new HashMap<>();

Boolean usernameUpdated= updateUserNameService.UpdateUserName(Integer.parseInt(updateUsername.get("userId").get(0)));

if(usernameUpdated) {

usernameUpdateResponse.put("status", "True");

usernameUpdateResponse.put("massege", "The user username has been updated successfully");

}else {

usernameUpdateResponse.put("status", "False");

usernameUpdateResponse.put("massege", "Invalid Id!");

}

return usernameUpdateResponse;

}}

package com.sportyshoes.controllers.user;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.UpdateUserPasswordService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class UpdateUserPasswordController {

@Autowired

private UpdateUserPasswordService updatePasswordService ;

@PatchMapping("{userid}/update/password")

public Map <String, String> Updateusername(@RequestBody MultiValueMap<String, String> updatePassword) {

Map <String, String> passwordUpdateResponse=new HashMap<>();

Boolean passwordUpdated= updatePasswordService.UpdateUserPassword(Integer.parseInt(updatePassword.get("userId").get(0)));

if(passwordUpdated) {

passwordUpdateResponse.put("status", "True");

passwordUpdateResponse.put("massege", "The user password has been updated successfully");

}else {

passwordUpdateResponse.put("status", "False");

passwordUpdateResponse.put("massege", "Invalid Id!");

}

return passwordUpdateResponse;

}

}

package com.sportyshoes.controllers.user;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.UserService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class UserController {

@Autowired

private UserService userService;

@PostMapping("/signUp")

public Map <String, String> signUp(@RequestBody MultiValueMap<String, String> newUserDetails) {

Map <String, String> signUpResponse=new HashMap<>();

Boolean userRegistered= userService.AddUser(Integer.parseInt(newUserDetails.get("userId").get(0)),newUserDetails.get("userName").get(0), newUserDetails.get("password").get(0),newUserDetails.get("name").get(0));

if(userRegistered) {

signUpResponse.put("status", "True");

signUpResponse.put("massege", "The user has been registered successfully");

}else {

signUpResponse.put("status", "False");

signUpResponse.put("massege", "Invalid Info!");

}

return signUpResponse;

}

}

package com.sportyshoes.controllers.user;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.models.Order;

import com.sportyshoes.models.Product;

import com.sportyshoes.services.FindOrderByProductIdService;

import com.sportyshoes.services.FindOrderByUserIdService;

import com.sportyshoes.services.FindProductByProductIdService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class ViewUserAllOrdersController {

@Autowired

private FindOrderByUserIdService findOrderByUserIdService;

@GetMapping("/{userId}/order/all")

public Map <String, String> FindProduct(@RequestBody MultiValueMap<String, String> searchOrdertbyUserId) {

Map <String, String> searchOrderResponse=new HashMap<>();

Order orderfound= findOrderByUserIdService.findOrderByUserId(Integer.parseInt(searchOrdertbyUserId.get("userId").get(0)));

if(orderfound != null) {

searchOrderResponse.put("status", "true");

searchOrderResponse.put("massege", "Order found");

}else {

searchOrderResponse.put("status", "false");

searchOrderResponse.put("massege", "Order not found");

}

return searchOrderResponse;

}

}

**Admin controller package:**

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.AdminService;

import com.sportyshoes.services.ProductService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/product")

public class AddProductController {

@Autowired

private ProductService productService;

@PostMapping("/add")

public Map <String, String> signUp(@RequestBody MultiValueMap<String, String> newProductDetails) {

Map <String, String> addProductResponse=new HashMap<>();

Boolean productAdded= productService.AddProduct(Integer.parseInt(newProductDetails.get("productId").get(0)),newProductDetails.get("productName").get(0), Double.parseDouble(newProductDetails.get("MSRP").get(0)),Integer.parseInt(newProductDetails.get("quantityInStock").get(0)),newProductDetails.get("productVendor").get(0));

if(productAdded) {

addProductResponse.put("status", "True");

addProductResponse.put("massege", "Product added successfully");

}else {

addProductResponse.put("status", "False");

addProductResponse.put("massege", "Product not added");

}

return addProductResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.boot.autoconfigure.kafka.KafkaProperties.Admin;

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.AdminAuthService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class AdminAuthController {

@Autowired

private AdminAuthService authService;

@PostMapping("/signIn")

public Map <String, String> signIn(@RequestBody MultiValueMap<String, String> credentials) {

Map <String, String> signInResponse=new HashMap<>();

com.sportyshoes.models.Admin adminAuthenticated= authService.SignInAdmin(credentials.get("adminName").get(0), credentials.get("password").get(0));

if(adminAuthenticated != null) {

signInResponse.put("status", "True");

signInResponse.put("massege", "The admin has been authenticated successfully");

}else {

signInResponse.put("status", "False");

signInResponse.put("massege", "Invalid credential!");

}

return signInResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.AdminService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class AdminController {

@Autowired

private AdminService adminService;

@PostMapping("/signUp")

public Map <String, String> signUp(@RequestBody MultiValueMap<String, String> newAdminDetails) {

Map <String, String> signUpResponse=new HashMap<>();

Boolean adminRegistered= adminService.AddAdmin(Integer.parseInt(newAdminDetails.get("adminId").get(0)),newAdminDetails.get("adminName").get(0), newAdminDetails.get("password").get(0));

if(adminRegistered) {

signUpResponse.put("status", "True");

signUpResponse.put("massege", "The admin has been registered successfully");

}else {

signUpResponse.put("status", "False");

signUpResponse.put("massege", "Invalid Info!");

}

return signUpResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.DeleteAdminService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class DeleteAdminController {

@Autowired

private DeleteAdminService deleteService;

@DeleteMapping("{adminid}/delete")

public Map <String, String> Delete(@RequestBody MultiValueMap<String, String> delete) {

Map <String, String> adminDeleteResponse=new HashMap<>();

Boolean adminDeleted= deleteService.DeleteAdmin(Integer.parseInt(delete.get("adminId").get(0)));

if(adminDeleted) {

adminDeleteResponse.put("status", "True");

adminDeleteResponse.put("massege", "The admin has been deleted successfully");

}else {

adminDeleteResponse.put("status", "False");

adminDeleteResponse.put("massege", "Invalid Id!");

}

return adminDeleteResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.DeleteProductService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class DeleteProductController {

@Autowired

private DeleteProductService deleteProductService;

@DeleteMapping("{productId}/delete")

public Map <String, String> DeleteProduct(@RequestBody MultiValueMap<String, String> deleteProduct) {

Map <String, String> productDeleteResponse=new HashMap<>();

Boolean productDeleted= deleteProductService.DeleteProduct(Integer.parseInt(deleteProduct.get("userId").get(0)));

if(productDeleted) {

productDeleteResponse.put("status", "True");

productDeleteResponse.put("massege", "The user has been deleted successfully");

}else {

productDeleteResponse.put("status", "False");

productDeleteResponse.put("massege", "Invalid Id!");

}

return productDeleteResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

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

import com.sportyshoes.models.Order;

import com.sportyshoes.models.User;

import com.sportyshoes.services.FindAllOrdersByDateASCService;

import com.sportyshoes.services.GetAllUserService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/order")

public class FindAllOrdersByDateASCController {

@Autowired

private FindAllOrdersByDateASCService FindAllOrdersByDateASCService;

@GetMapping("/all/orderDate&orderingASC")

//@ResponseBody

public List orderDetail() {

List<Order> allOrder=FindAllOrdersByDateASCService.GetallOrderByDateASC();

return allOrder;

}

// public Map <String, String> orderDetail(@RequestBody MultiValueMap<String, String> AllOrderDetail) {

// Map <String, String> AllOrderResponse=new HashMap<>();

// ArrayList<Order> allOrder= FindAllOrdersByDateASCService.GetallOrderByDateASC();

// if(allOrder != null) {

// AllOrderResponse.put("status", "true");

// AllOrderResponse.put("massege", "All order found ");

// }else {

// AllOrderResponse.put("status", "False");

// AllOrderResponse.put("massege", "order not found");

// }

// return AllOrderResponse;

// }

//

}

package com.sportyshoes.controllers.admin;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

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

import com.sportyshoes.models.Order;

import com.sportyshoes.services.FindAllOrdersByDateDESCService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/order")

public class FindAllOrdersByDateDESCController {

@Autowired

private FindAllOrdersByDateDESCService FindAllOrdersByDateDESCService;

@GetMapping("/all/orderDate&orderingDESC")

// @ResponseBody

public List orderDetail() {

List<Order> allOrder=FindAllOrdersByDateDESCService.GetallOrderByDateDESC();

return allOrder;

}

// public Map <String, String> orderDetail(@RequestBody MultiValueMap<String, String> AllOrderDetail) {

// Map <String, String> AllOrderResponse=new HashMap<>();

// ArrayList<Order> allOrder= FindAllOrdersByDateDESCService.GetallOrderByDateDESC();

// if(allOrder != null) {

// AllOrderResponse.put("status", "true");

// AllOrderResponse.put("massege", "All order found ");

// }else {

// AllOrderResponse.put("status", "False");

// AllOrderResponse.put("massege", "order not found");

// }

// return AllOrderResponse;

// }

}

package com.sportyshoes.controllers.admin;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

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

import com.sportyshoes.models.Order;

import com.sportyshoes.services.FindAllOrdersByOrderIdASCService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/order")

public class FindAllOrdersByOrderIdASCController {

@Autowired

private FindAllOrdersByOrderIdASCService findAllOrdersByOrderIdASCService;

@GetMapping("/all/orderId&orderingASC")

//@ResponseBody

public List orderDetail() {

List<Order> allOrder=findAllOrdersByOrderIdASCService.GetallOrderByIdASC();

return allOrder;

}

// public Map <String, String> orderDetail(@RequestBody MultiValueMap<String, String> AllOrderDetail) {

// Map <String, String> AllOrderResponse=new HashMap<>();

// ArrayList<Order> allOrder= findAllOrdersByOrderIdASCService.GetallOrderByIdASC();

// if(allOrder != null) {

// AllOrderResponse.put("status", "true");

// AllOrderResponse.put("massege", "All order found ");

// }else {

// AllOrderResponse.put("status", "False");

// AllOrderResponse.put("massege", "order not found");

// }

// return AllOrderResponse;

// }

}

package com.sportyshoes.controllers.admin;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

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

import com.sportyshoes.models.Order;

import com.sportyshoes.services.FindAllOrdersByOrderIdDESCService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/order")

public class FindAllOrdersByOrderIdDESCController {

@Autowired

private FindAllOrdersByOrderIdDESCService findAllOrdersByOrderIdDESCService;

@GetMapping("/all/orderId&orderingDESC")

//@ResponseBody

public List orderDetail() {

List<Order> allOrder=findAllOrdersByOrderIdDESCService.GetallOrderByIdDESC();

return allOrder;

}

// public Map <String, String> orderDetail(@RequestBody MultiValueMap<String, String> AllOrderDetail) {

// Map <String, String> AllOrderResponse=new HashMap<>();

// ArrayList<Order> allOrder= findAllOrdersByOrderIdDESCService.GetallOrderByIdDESC();

// if(allOrder != null) {

// AllOrderResponse.put("status", "true");

// AllOrderResponse.put("massege", "All order found ");

// }else {

// AllOrderResponse.put("status", "False");

// AllOrderResponse.put("massege", "order not found");

// }

// return AllOrderResponse;

// }

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.models.Order;

import com.sportyshoes.models.Product;

import com.sportyshoes.services.FindOrderByProductIdService;

import com.sportyshoes.services.FindOrderByUserIdService;

import com.sportyshoes.services.FindProductByProductIdService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class FindOrdersByUserIdController {

@Autowired

private FindOrderByUserIdService findOrderByUserIdService;

@GetMapping("/{userId}/order/all")

public Map <String, String> FindProduct(@RequestBody MultiValueMap<String, String> searchOrdertbyUserId) {

Map <String, String> searchOrderResponse=new HashMap<>();

Order orderfound= findOrderByUserIdService.findOrderByUserId(Integer.parseInt(searchOrdertbyUserId.get("userId").get(0)));

if(orderfound != null) {

searchOrderResponse.put("status", "true");

searchOrderResponse.put("massege", "Order found");

}else {

searchOrderResponse.put("status", "false");

searchOrderResponse.put("massege", "Order not found");

}

return searchOrderResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.models.Product;

import com.sportyshoes.services.FindProductByProductIdService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class FindProductByProductIdController1 {

@Autowired

private FindProductByProductIdService findProductByProductIdService;

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

public Map <String, String> FindProduct(@RequestBody MultiValueMap<String, String> searchProductbyProductId) {

Map <String, String> searchProductResponse=new HashMap<>();

Product productfound= findProductByProductIdService.findProductByProductId(Integer.parseInt(searchProductbyProductId.get("productId").get(0)));

if(productfound != null) {

searchProductResponse.put("status", "true");

searchProductResponse.put("massege", "Product found");

}else {

searchProductResponse.put("status", "false");

searchProductResponse.put("massege", "Product not found");

}

return searchProductResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.FindUserByNameService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class FindUserByNameController {

@Autowired

private FindUserByNameService findUserByNameService;

@GetMapping("/search?name=<user>")

public Map <String, String> ProductDetail(@RequestBody MultiValueMap<String, String> searchUserbyName) {

Map <String, String> searchUserResponse=new HashMap<>();

com.sportyshoes.models.User userfound= findUserByNameService.findUserByName(searchUserbyName.get("name").get(0));

if(userfound != null) {

searchUserResponse.put("status", "true");

searchUserResponse.put("massege", "User found");

}else {

searchUserResponse.put("status", "false");

searchUserResponse.put("massege", "User not found");

}

return searchUserResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.models.Admin;

import com.sportyshoes.models.Product;

import com.sportyshoes.services.GetAllAdminService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class GetAllAdminController {

@Autowired

private GetAllAdminService getAllAdminService;

@GetMapping("/getAllAdmin")

public List AdminDetail() {

List<Admin> allAdmin=getAllAdminService.GetallAdmin();

return allAdmin;

}

// public Map <String, String> adminDetail(@RequestBody MultiValueMap<String, String> AllAdminDetail) {

// Map <String, String> AllAdminResponse=new HashMap<>();

// ArrayList<com.sportyshoes.models.Admin> Alladmin= getAllAdminService.GetallAdmin();

//

// return AllAdminResponse;

//

//

// }

}

package com.sportyshoes.controllers.admin;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.models.Product;

import com.sportyshoes.services.GetAllProductService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class GetAllProductController1 {

@Autowired

private GetAllProductService getAllProductService;

@GetMapping("/product/all")

public List ProductDetail() {

List<Product> allProduct=getAllProductService.GetallProduct();

return allProduct;

}

// public Map <String, String> ProductDetail(@RequestBody MultiValueMap<String, String> AllProductDetail) {

// Map <String, String> AllProductResponse=new HashMap<>();

// ArrayList<Product> allProduct= getAllProductService.GetallProduct();

//

// return AllProductResponse;

//

//

// }

}

package com.sportyshoes.controllers.admin;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

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

import com.sportyshoes.models.Admin;

import com.sportyshoes.models.User;

import com.sportyshoes.services.GetAllUserService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/user")

public class GetAllUserController {

@Autowired

private GetAllUserService getAllUserService;

@GetMapping("/getAllUser")

@ResponseBody

public List userDetail() {

List<User> allUser=getAllUserService.GetallUser();

return allUser;

}

// public Map <String, String> userDetail(@RequestBody MultiValueMap<String, String> AllUserDetail) {

// Map <String, String> AllUserResponse=new HashMap<>();

// ArrayList<User> allUser= getAllUserService.GetallUser();

// if(allUser != null) {

// AllUserResponse.put("status", "true");

// AllUserResponse.put("massege", "All user found ");

// }else {

// AllUserResponse.put("status", "False");

// AllUserResponse.put("massege", "user not found");

// }

// return AllUserResponse;

//

//

// }

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

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

import com.sportyshoes.services.UpdateAdminService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class UpdateAdminController {

@Autowired

private UpdateAdminService updateService;

@PatchMapping("{adminid}/update")

public Map <String, String> signUp(@RequestBody MultiValueMap<String, String> updateDetails) {

Map <String, String> adminUpdateResponse=new HashMap<>();

Boolean adminUpdated= updateService.UpdateAdmin(Integer.parseInt(updateDetails.get("adminId").get(0)));

if(adminUpdated) {

adminUpdateResponse.put("status", "True");

adminUpdateResponse.put("massege", "The admin has been updated successfully");

}else {

adminUpdateResponse.put("status", "False");

adminUpdateResponse.put("massege", "Invalid Info!");

}

return adminUpdateResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.UpdateAdminNameService;

import com.sportyshoes.services.UpdateUserNameService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class UpdateAdminNameController {

@Autowired

private UpdateAdminNameService updateAdminNameService ;

@PatchMapping("{adminId}/update/adminName")

public Map <String, String> UpdateadminName(@RequestBody MultiValueMap<String, String> updateAdminname) {

Map <String, String> adminnameUpdateResponse=new HashMap<>();

Boolean adminnameUpdated= updateAdminNameService.UpdateAdminName(Integer.parseInt(updateAdminname.get("adminId").get(0)));

if(adminnameUpdated) {

adminnameUpdateResponse.put("status", "True");

adminnameUpdateResponse.put("massege", "The admin adminname has been updated successfully");

}else {

adminnameUpdateResponse.put("status", "False");

adminnameUpdateResponse.put("massege", "Invalid Id!");

}

return adminnameUpdateResponse;

}}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.UpdateAdminPasswordService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class UpdateAdminPasswordController {

@Autowired

private UpdateAdminPasswordService updatePasswordService ;

@PatchMapping("{adminid}/update/password")

public Map <String, String> Updateusername(@RequestBody MultiValueMap<String, String> updatePassword) {

Map <String, String> passwordUpdateResponse=new HashMap<>();

Boolean passwordUpdated= updatePasswordService.UpdateAdminPassword(Integer.parseInt(updatePassword.get("adminId").get(0)));

if(passwordUpdated) {

passwordUpdateResponse.put("status", "True");

passwordUpdateResponse.put("massege", "The admin password has been updated successfully");

}else {

passwordUpdateResponse.put("status", "False");

passwordUpdateResponse.put("massege", "Invalid Id!");

}

return passwordUpdateResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.UpdateProductMSRPService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/product")

public class UpdateProductMSRPController {

@Autowired

private UpdateProductMSRPService updateProductMSRPService ;

@PatchMapping("{productId}/update/msrp")

public Map <String, String> UpdateMSRP(@RequestBody MultiValueMap<String, String> updateProductMSRP) {

Map <String, String> MSRPUpdateResponse=new HashMap<>();

Boolean MSRPUpdated= updateProductMSRPService.UpdateProductMSRP(Integer.parseInt(updateProductMSRP.get("productId").get(0)));

if( MSRPUpdated) {

MSRPUpdateResponse.put("status", "True");

MSRPUpdateResponse.put("massege", "The product MSRP has been updated successfully");

}else {

MSRPUpdateResponse.put("status", "False");

MSRPUpdateResponse.put("massege", "Invalid ProductId!");

}

return MSRPUpdateResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.UpdateProductNameService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/product")

public class UpdateProductNameController {

@Autowired

private UpdateProductNameService updateProductNameService ;

@PatchMapping("{productId}/update/name")

public Map <String, String> Updateusername(@RequestBody MultiValueMap<String, String> updateProductname) {

Map <String, String> productNameUpdateResponse=new HashMap<>();

Boolean productNameUpdated= updateProductNameService.UpdateProductName(Integer.parseInt(updateProductname.get("productId").get(0)));

if( productNameUpdated) {

productNameUpdateResponse.put("status", "True");

productNameUpdateResponse.put("massege", "The product name has been updated successfully");

}else {

productNameUpdateResponse.put("status", "False");

productNameUpdateResponse.put("massege", "Invalid Id!");

}

return productNameUpdateResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.UpdateProductVendorService;

import com.sportyshoes.services.UpdateQuantityInStockService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/product")

public class UpdateProductVendorController {

@Autowired

private UpdateProductVendorService updateProductVendorService ;

@PatchMapping("{productId}/update/productVendor")

public Map <String, String> UpdateMSRP(@RequestBody MultiValueMap<String, String> updateProductVendor) {

Map <String, String> productVendorUpdateResponse=new HashMap<>();

Boolean productVendorUpdated= updateProductVendorService.UpdateProductVendor(Integer.parseInt(updateProductVendor.get("productId").get(0)));

if( productVendorUpdated) {

productVendorUpdateResponse.put("status", "True");

productVendorUpdateResponse.put("massege", "The product quantity in stock has been updated successfully");

}else {

productVendorUpdateResponse.put("status", "False");

productVendorUpdateResponse.put("massege", "Invalid ProductId!");

}

return productVendorUpdateResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.services.UpdateProductMSRPService;

import com.sportyshoes.services.UpdateQuantityInStockService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/product")

public class UpdateQuantityInStockController {

@Autowired

private UpdateQuantityInStockService updateQuantityInStockService ;

@PatchMapping("{productId}/update/quantityInStock")

public Map <String, String> UpdateMSRP(@RequestBody MultiValueMap<String, String> updateQuantityInStock) {

Map <String, String> quantityInStockUpdateResponse=new HashMap<>();

Boolean quantityInStockUpdated= updateQuantityInStockService.UpdateQuantityInStock(Integer.parseInt(updateQuantityInStock.get("productId").get(0)));

if( quantityInStockUpdated) {

quantityInStockUpdateResponse.put("status", "True");

quantityInStockUpdateResponse.put("massege", "The product quantity in stock has been updated successfully");

}else {

quantityInStockUpdateResponse.put("status", "False");

quantityInStockUpdateResponse.put("massege", "Invalid ProductId!");

}

return quantityInStockUpdateResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

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

import com.sportyshoes.models.User;

import com.sportyshoes.services.UserAccountDetailService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/admin")

public class UserAccountDetailController {

@Autowired

private UserAccountDetailService accountDetailService;

@GetMapping("/{userid}")

public Map <String, String> userDetail(@RequestBody MultiValueMap<String, String> ACDetail) {

Map <String, String> DetailResponse=new HashMap<>();

User userfound= accountDetailService.AccountDetail(Integer.parseInt(ACDetail.get("userId").get(0)));

if( userfound != null) {

DetailResponse.put("status", "True");

DetailResponse.put("massege", "User account detail found");

}else {

DetailResponse.put("status", "False");

DetailResponse.put("massege", "Invalid UserId!");

}

return DetailResponse;

}

}

package com.sportyshoes.controllers.admin;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.util.MultiValueMap;

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

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

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

import com.sportyshoes.models.Order;

import com.sportyshoes.services.FindOrderByProductIdService;

import io.swagger.v3.oas.annotations.parameters.RequestBody;

@RestController

@RequestMapping("/api/product")

public class ViewOrderByProductIdController {

@Autowired

private FindOrderByProductIdService findOrderByProductIdService;

@GetMapping("/{productId}/order/all")

public Map <String, String> FindProduct(@RequestBody MultiValueMap<String, String> searchOrdertbyProductId) {

Map <String, String> viewOrderResponse=new HashMap<>();

Order orderfound= findOrderByProductIdService.findOrderByProductId(Integer.parseInt(searchOrdertbyProductId.get("productId").get(0)));

if(orderfound != null) {

viewOrderResponse.put("status", "true");

viewOrderResponse.put("massege", "Order found");

}else {

viewOrderResponse.put("status", "false");

viewOrderResponse.put("massege", "Order not found");

}

return viewOrderResponse;

}

}

**Pom.xml**

<?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>2.7.0</version>

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

</parent>

<groupId>com.spring.boot.sportyshoes.api</groupId>

<artifactId>sportyshoes-app-api</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>sportyshoes-app-api</name>

<description>Spring boot e-commerce project application-sportyshoes.com</description>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<scope>runtime</scope>

</dependency>

<dependency>

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

<artifactId>spring-boot-configuration-processor</artifactId>

<optional>true</optional>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-boot-starter</artifactId>

<version>3.0.0</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**DATABASE COMMANDS**

create database sportyshoes;

create table users(USER\_ID int PRIMARY KEY AUTO\_INCREMENT,

PASSWORD varchar(30), USER\_NAME varchar(30), NAME varchar(30));

insert into users values(1, ‘ram123’, ‘ram1990’,’ramlal’);

insert into users values(2, ‘shyam123’, ‘shyam1998’,’shyamlal’);

create table admins(ADMIN\_ID int PRIMARY KEY AUTO\_INCREMENT,

PASSWORD varchar(30), ADMIN\_NAME varchar(30));

insert into admins values(1, ‘kalpana123’, ‘kalpana’);

insert into admins values(2, ‘ADMIN123’, ‘ADMIN’);

create table products(PRODUCT\_ID int PRIMARY KEY AUTO\_INCREMENT,

PRODUCT\_NAME varchar(30), MSRP double, QUANTITY\_IN\_STOCK int , PRODUCT\_VENDOR varchar(30));

insert into products values(1, ‘PUMA’, 1583, 23,’ADIDAS’);

insert into products values(2, ‘CAMPUS’,5090, 20,’ADIDAS’);

create table orders(ORDER\_ID int PRIMARY KEY AUTO\_INCREMENT,

REF\_PRODUCT\_ID int, DATE date, REF\_USER\_ID int);

insert into orders values(1, 1, ‘2022-06-20’ 1);

insert into orders values(2, 1, ‘2022-06-23’ 2);

insert into orders values(3, 2, ‘2022-07-23’ 3);