|  |  |
| --- | --- |
| Name | MALLIKARJUNA. H |
| Project Name | E-commerce Website for Sporty Shoes |
| Batch | MS FSD FEB 2022 Cohort 1 |
| Phase 3 | Implement Frameworks the DevOps way |

Contents

[Implement Frameworks the DevOps way 1](#_Toc106659292)

[Implement Frameworks the DevOps way 3](#_Toc106659293)

[Project Folders Structure 3](#_Toc106659294)

[Source Code 4](#_Toc106659295)

[Tools Used 21](#_Toc106659296)

[Postman link 21](#_Toc106659297)

[GitHub URL: 21](#_Toc106659298)

|  |
| --- |
| Implement Frameworks the DevOps way |
| Concepts Used   * CRUD Operation * Control Structures * Java Collections * Exception Handling * Java Object Oriented Concepts * Annotations * Functions * Wrapper * Swagger |
| Project Folders Structure |
|  |
| Source Code |
| * package com.shoe;   import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  @SpringBootApplication  public class Sportyshoes1Application {  public static void main(String[] args) {  SpringApplication.run(Sportyshoes1Application.class, args);  System.out.println("Welcome to Shoe Shop");  }    } |
| * **package com.shoe.config;**   **import java.util.Collections;**  **import org.springframework.context.annotation.Bean;**  **import org.springframework.context.annotation.Configuration;**  **import springfox.documentation.builders.PathSelectors;**  **import springfox.documentation.builders.RequestHandlerSelectors;**  **import springfox.documentation.service.ApiInfo;**  **import springfox.documentation.spi.DocumentationType;**  **import springfox.documentation.spring.web.plugins.Docket;**  **import springfox.documentation.swagger2.annotations.EnableSwagger2;**  **@Configuration**  **@EnableSwagger2**  **public class SwaggerConfiguration {**  **@Bean**  **public Docket swaggerConfiguration() {**  **return new Docket(DocumentationType.SWAGGER\_2)**  **.select()**  **.paths(PathSelectors.any())**  **.apis(RequestHandlerSelectors.basePackage("com.shoe"))**  **.build()**  **.apiInfo(apiDetails());**  **}**    **private ApiInfo apiDetails() {**  **return new ApiInfo(**  **"Spring Boot Swagger Shoe API",**  **"Spring Boot Shoe application with Swagger",**  **"1.0",**  **"Free to use",**  **new springfox.documentation.service.Contact("Mallikarjuna.H",**  **"https://github.com/Mallikarjunahugar", "mallikarjuna.j17@gmail.com"),**  **"API License",**  **"https://github.com/Mallikarjunahugar",**  **Collections.emptyList()**  **);**  **}**  **}**   * **com.shoe.controller**   **package com.shoe.controller;**  **import java.util.Optional;**  **import org.springframework.beans.factory.annotation.Autowired;**  **import org.springframework.http.HttpStatus;**  **import org.springframework.web.bind.annotation.DeleteMapping;**  **import org.springframework.web.bind.annotation.GetMapping;**  **import org.springframework.web.bind.annotation.PathVariable;**  **import org.springframework.web.bind.annotation.PostMapping;**  **import org.springframework.web.bind.annotation.PutMapping;**  **import org.springframework.web.bind.annotation.RequestBody;**  **import org.springframework.web.bind.annotation.RequestMapping;**  **import org.springframework.web.bind.annotation.ResponseStatus;**  **import org.springframework.web.bind.annotation.RestController;**  **import com.shoe.model.AdminModel;**  **import com.shoe.repository.AdminRepository;**  **import io.swagger.annotations.ApiOperation;**  **@RestController**  **@RequestMapping("/admins")**  **public class AdminController {**  **@Autowired**  **AdminRepository adminRepository;**  **// insert new admin into database**  **@PostMapping("/add")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Admin Registration Success!")**  **public AdminModel addAdmin(@RequestBody AdminModel admin) {**  **return adminRepository.save(admin);**  **}**  **// fetch all admin list from database**  **@GetMapping("/all")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "All admin details!")**  **@ApiOperation(value = "Fetch Admins by admin\_id", notes = "Fetching admin\_id", response = AdminModel.class)**  **public Iterable<AdminModel> allProduct() {**  **return adminRepository.findAll();**  **}**  **// fetch specific admin by their ID**  **@GetMapping("/{ADMIN\_ID}")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Specific admin detail!")**  **@ApiOperation(value = "Find Admins by admin\_id", notes = "Provide an admin\_id", response = AdminModel.class)**  **public Optional<AdminModel> productById(@PathVariable("ADMIN\_ID") int ADMIN\_ID) {**  **return adminRepository.findById(ADMIN\_ID);**  **}**  **// update existing admin**  **@PutMapping("/update")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Admin details updated!")**  **public AdminModel updateOrder(@RequestBody AdminModel admin) {**  **return adminRepository.save(admin);**  **}**  **// delete admin from database**  **@DeleteMapping("/{ADMIN\_ID}")**  **@ResponseStatus(value = HttpStatus.CREATED, reason = "Admin details deleted!")**  **public void deleteAdmin(@PathVariable("ADMIN\_ID") int ADMIN\_ID) {**  **adminRepository.deleteById(ADMIN\_ID);**  **}**  **}**  **package com.shoe.controller;**  **import java.util.Optional;**  **import org.springframework.beans.factory.annotation.Autowired;**  **import org.springframework.http.HttpStatus;**  **import org.springframework.web.bind.annotation.DeleteMapping;**  **import org.springframework.web.bind.annotation.GetMapping;**  **import org.springframework.web.bind.annotation.PathVariable;**  **import org.springframework.web.bind.annotation.PostMapping;**  **import org.springframework.web.bind.annotation.PutMapping;**  **import org.springframework.web.bind.annotation.RequestBody;**  **import org.springframework.web.bind.annotation.RequestMapping;**  **import org.springframework.web.bind.annotation.ResponseStatus;**  **import org.springframework.web.bind.annotation.RestController;**  **import com.shoe.model.OrderModel;**  **import com.shoe.repository.OrderRepository;**  **import io.swagger.annotations.ApiOperation;**  **@RestController**  **@RequestMapping("/orders")**  **public class OrderController {**  **@Autowired**  **OrderRepository orderRepository;**  **// insert new order into database**  **@PostMapping("/add")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Order placed!")**  **public OrderModel addOrder(@RequestBody OrderModel order) {**  **return orderRepository.save(order);**  **}**  **// fetch all order list from database**  **@GetMapping("/all")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "All order details!")**  **@ApiOperation(value = "Fetch orders by order\_id",**  **notes = "Fetching order\_id",**  **response = OrderModel.class)**  **public Iterable<OrderModel> allProduct() {**  **return orderRepository.findAll();**  **}**  **// fetch specific order by their ID**  **@GetMapping("/{order\_id}")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Specific order detail!")**  **@ApiOperation(value = "Fetch orders by order\_id",**  **notes = "Fetching order\_id",**  **response = OrderModel.class)**  **public Optional<OrderModel> productById(@PathVariable("order\_id") int order\_id) {**  **return orderRepository.findById(order\_id);**  **}**  **// update existing order**  **@PutMapping("/update")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Order details updated!")**  **public OrderModel updateOrder(@RequestBody OrderModel order) {**  **return orderRepository.save(order);**  **}**  **// delete order from database**  **@DeleteMapping("/{order\_id}")**  **@ResponseStatus(value = HttpStatus.CREATED, reason = "Order deleted!")**  **public void deleteProduct(@PathVariable("order\_id") int order\_id) {**  **orderRepository.deleteById(order\_id);**  **}**  **}**  **package com.shoe.controller;**  **import java.util.Optional;**  **import org.springframework.beans.factory.annotation.Autowired;**  **import org.springframework.http.HttpStatus;**  **import org.springframework.web.bind.annotation.DeleteMapping;**  **import org.springframework.web.bind.annotation.GetMapping;**  **import org.springframework.web.bind.annotation.PathVariable;**  **import org.springframework.web.bind.annotation.PostMapping;**  **import org.springframework.web.bind.annotation.PutMapping;**  **import org.springframework.web.bind.annotation.RequestBody;**  **import org.springframework.web.bind.annotation.RequestMapping;**  **import org.springframework.web.bind.annotation.ResponseStatus;**  **import org.springframework.web.bind.annotation.RestController;**  **import com.shoe.model.ProductModel;**  **import com.shoe.repository.ProductRepository;**  **import io.swagger.annotations.ApiOperation;**  **@RestController**  **@RequestMapping("/products")**  **public class ProductController {**    **@Autowired**  **ProductRepository productRepository;**  **// insert new product into database**  **@PostMapping("/add")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Products Added!")**  **public ProductModel addProduct(@RequestBody ProductModel product) {**  **return productRepository.save(product);**  **}**  **// fetch all product list from database**  **@GetMapping("/all")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "All product details!")**  **@ApiOperation(value = "Fetch products by product\_id",**  **notes = "Fetching product\_id",**  **response = ProductModel.class)**  **public Iterable<ProductModel> allProduct() {**  **return productRepository.findAll();**  **}**  **// fetch specific product by their ID**  **@GetMapping("/{product\_id}")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Specific product!")**  **@ApiOperation(value = "Fetch products by product\_id",**  **notes = "Fetching product\_id",**  **response = ProductModel.class)**  **public Optional<ProductModel> productById(@PathVariable("product\_id") int product\_id) {**  **return productRepository.findById(product\_id);**  **}**  **// update existing product**  **@PutMapping("/update")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Product details updated!")**  **public ProductModel updateProduct(@RequestBody ProductModel product) {**  **return productRepository.save(product);**  **}**  **// delete product from database**  **@DeleteMapping("/{product\_id}")**  **@ResponseStatus(value = HttpStatus.CREATED, reason = "Product deleted!")**  **public void deleteProduct(@PathVariable("product\_id") int product\_id) {**  **productRepository.deleteById(product\_id);**  **}**  **}**  **package com.shoe.controller;**  **import java.util.Optional;**  **import org.springframework.beans.factory.annotation.Autowired;**  **import org.springframework.http.HttpStatus;**  **import org.springframework.web.bind.annotation.DeleteMapping;**  **import org.springframework.web.bind.annotation.GetMapping;**  **import org.springframework.web.bind.annotation.PathVariable;**  **import org.springframework.web.bind.annotation.PostMapping;**  **import org.springframework.web.bind.annotation.PutMapping;**  **import org.springframework.web.bind.annotation.RequestBody;**  **import org.springframework.web.bind.annotation.RequestMapping;**  **import org.springframework.web.bind.annotation.ResponseStatus;**  **import org.springframework.web.bind.annotation.RestController;**  **import com.shoe.model.UserModel;**  **import com.shoe.repository.UserRepository;**  **import io.swagger.annotations.ApiOperation;**  **@RestController**  **@RequestMapping("/users")**  **public class UserController {**    **@Autowired**  **UserRepository userRepository;**  **// insert new user into database**  **@PostMapping("/add")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Registration Success!")**  **public UserModel addUser(@RequestBody UserModel user) {**    **return userRepository.save(user);**    **}**  **// fetch all user list from database**  **@GetMapping("/all")**  **//@ResponseStatus(value = HttpStatus.CREATED)**  **@ApiOperation(value = "Fetch users by user\_id",**  **notes = "Fetching user\_id",**  **response = UserModel.class)**  **public Iterable<UserModel> allProduct() {**  **return userRepository.findAll();**  **}**  **// fetch specific user by their ID**  **@GetMapping("/{user\_id}")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "Specific user!")**  **@ApiOperation(value = "Fetch users by user\_id",**  **notes = "Fetching user\_id",**  **response = UserModel.class)**  **public Optional<UserModel> productById(@PathVariable("user\_id") int user\_id) {**  **return userRepository.findById((long) user\_id);**  **}**  **// update existing user**  **@PutMapping("/update")**  **//@ResponseStatus(value = HttpStatus.CREATED, reason = "User details updated!")**  **public UserModel updateUser(@RequestBody UserModel user) {**  **return userRepository.save(user);**  **}**  **// delete user from database**  **@DeleteMapping("/{user\_id}")**  **@ResponseStatus(value = HttpStatus.CREATED, reason = "User details deleted!")**  **public void deleteProduct(@PathVariable("user\_id") int user\_id) {**  **userRepository.deleteById((long) user\_id);**  **}**  **}**   * **com.shoe.exception**   **package** com.shoe.exception;  **public** **class** DatabaseOperationException **extends** Exception {  /\*\*  \*  \*/  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** DatabaseOperationException(String errorMessage) {  **super**(errorMessage);  }    **public** DatabaseOperationException(String errorMessage, Exception e) {  **super**(errorMessage, e);  }  }  **package** com.shoe.exception;  **public** **class** MyResourceNotCreatedException **extends** Exception{  /\*\*  \*  \*/  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** MyResourceNotCreatedException(String errorMessage) {  **super**(errorMessage);  }  }  **package** com.shoe.exception;  **public** **class** MyResourceNotDeletedException **extends** Exception {  /\*\*  \*  \*/  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** MyResourceNotDeletedException(String errorMessage) {  **super**(errorMessage);  }  }  **package** com.shoe.exception;  **public** **class** MyResourceNotFoundException **extends** Exception{  /\*\*  \*  \*/  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** MyResourceNotFoundException(String errorMessage) {  **super**(errorMessage);  }  }  **package** com.shoe.exception;  **public** **class** MyResourceNotUpdatedException **extends** Exception {  /\*\*  \*  \*/  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** MyResourceNotUpdatedException(String errorMessage) {  **super**(errorMessage);  }  }   * **com.shoe.model.**   **package com.shoe.model;**  **import javax.persistence.Column;**  **import javax.persistence.Entity;**  **import javax.persistence.GeneratedValue;**  **import javax.persistence.GenerationType;**  **import javax.persistence.Id;**  **import javax.persistence.Table;**  **import io.swagger.annotations.ApiModel;**  **import io.swagger.annotations.ApiModelProperty;**  **@Entity**  **@Table(name = "admins")**  **@ApiModel(description = "Details about Admin")**  **public class AdminModel {**  **@ApiModelProperty(notes = "Admin\_id")**  **private Integer ADMIN\_ID;**  **@ApiModelProperty(notes = "Admin\_name")**  **private String ADMIN\_NAME;**  **@ApiModelProperty(notes = "Admin\_password")**  **private String ADMIN\_PASSWORD;**  **public AdminModel() {**  **super();**  **// TODO Auto-generated constructor stub**  **}**  **public AdminModel(Integer aDMIN\_ID, String aDMIN\_NAME, String aDMIN\_PASSWORD) {**  **super();**  **ADMIN\_ID = aDMIN\_ID;**  **ADMIN\_NAME = aDMIN\_NAME;**  **ADMIN\_PASSWORD = aDMIN\_PASSWORD;**  **}**  **@Id**  **@GeneratedValue(strategy = GenerationType.IDENTITY)**  **@Column(name = "ADMIN\_ID")**  **public Integer getADMIN\_ID() {**  **return ADMIN\_ID;**  **}**  **public void setADMIN\_ID(Integer aDMIN\_ID) {**  **ADMIN\_ID = aDMIN\_ID;**  **}**  **@Column(name = "ADMIN\_NAME")**  **public String getADMIN\_NAME() {**  **return ADMIN\_NAME;**  **}**  **public void setADMIN\_NAME(String aDMIN\_NAME) {**  **ADMIN\_NAME = aDMIN\_NAME;**  **}**  **@Column(name = "ADMIN\_PASSWORD")**  **public String getADMIN\_PASSWORD() {**  **return ADMIN\_PASSWORD;**  **}**  **public void setADMIN\_PASSWORD(String aDMIN\_PASSWORD) {**  **ADMIN\_PASSWORD = aDMIN\_PASSWORD;**  **}**  **@Override**  **public String toString() {**  **return "AdminModel [ADMIN\_ID=" + ADMIN\_ID + ", ADMIN\_NAME=" + ADMIN\_NAME + ", ADMIN\_PASSWORD=" + ADMIN\_PASSWORD**  **+ "]";**  **}**  **}**  **package com.shoe.model;**  **import javax.persistence.Column;**  **import javax.persistence.Entity;**  **import javax.persistence.Id;**  **import javax.persistence.Table;**  **import io.swagger.annotations.ApiModel;**  **import io.swagger.annotations.ApiModelProperty;**  **@Entity**  **@Table(name = "orders")**  **@ApiModel(description = "About Order Details")**  **public class OrderModel {**  **@ApiModelProperty(notes = "order\_id")**  **private Long order\_id;**  **@ApiModelProperty(notes = "order\_date")**  **private Integer order\_date;**  **@ApiModelProperty(notes = "order\_quntity")**  **private Integer order\_quntity;**  **@ApiModelProperty(notes = "user\_id")**  **private Integer user\_id;**  **public OrderModel() {**  **super();**  **// TODO Auto-generated constructor stub**  **}**  **public OrderModel(Long order\_id, Integer order\_date, Integer order\_quntity, Integer user\_id) {**  **super();**  **this.order\_id = order\_id;**  **this.order\_date = order\_date;**  **this.order\_quntity = order\_quntity;**  **this.user\_id = user\_id;**  **}**  **@Id**    **@Column(name = "order\_id")**  **public Long getOrder\_id() {**  **return order\_id;**  **}**  **public void setOrder\_id(Long order\_id) {**  **this.order\_id = order\_id;**  **}**  **@Column(name = "order\_date")**  **public Integer getOrder\_date() {**  **return order\_date;**  **}**  **public void setOrder\_date(Integer order\_date) {**  **this.order\_date = order\_date;**  **}**  **@Column(name = "order\_quntity")**  **public Integer getOrder\_quntity() {**  **return order\_quntity;**  **}**  **public void setOrder\_quntity(Integer order\_quntity) {**  **this.order\_quntity = order\_quntity;**  **}**  **@Column(name = "user\_id")**  **public Integer getUser\_id() {**  **return user\_id;**  **}**  **public void setUser\_id(Integer user\_id) {**  **this.user\_id = user\_id;**  **}**  **@Override**  **public String toString() {**  **return "OrderModel [order\_id=" + order\_id + ", order\_date=" + order\_date + ", order\_quntity=" + order\_quntity**  **+ ", user\_id=" + user\_id + "]";**  **}**  **}**  **package com.shoe.model;**  **import javax.persistence.Column;**  **import javax.persistence.Entity;**  **import javax.persistence.GeneratedValue;**  **import javax.persistence.GenerationType;**  **import javax.persistence.Id;**  **import javax.persistence.Table;**  **import io.swagger.annotations.ApiModel;**  **import io.swagger.annotations.ApiModelProperty;**  **@Entity**  **@Table(name = "products")**  **@ApiModel(description = "Details about Products")**  **public class ProductModel {**  **@ApiModelProperty(notes = "product\_id")**  **private Integer product\_id;**  **@ApiModelProperty(notes = "product\_name")**  **private String product\_name;**  **@ApiModelProperty(notes = "product\_msrp")**  **private float product\_msrp;**  **@ApiModelProperty(notes = "product\_in\_stock")**  **private Integer product\_in\_stock;**  **@ApiModelProperty(notes = "product\_brand")**  **private String product\_brand;**  **public ProductModel() {**  **}**  **public ProductModel(Integer product\_id, String product\_name, float product\_msrp, Integer product\_in\_stock, String product\_brand) {**  **this.product\_id = product\_id;**  **this.product\_name = product\_name;**  **this.product\_msrp = product\_msrp;**  **this.product\_in\_stock = product\_in\_stock;**  **this.product\_brand = product\_brand;**  **}**  **@Id**  **@GeneratedValue(strategy = GenerationType.IDENTITY)**  **@Column(name = "product\_id")**  **public Integer getId() {**  **return product\_id;**  **}**    **@Column(name = "product\_name")**  **public String getName() {**  **return product\_name;**  **}**  **public void setName(String product\_name) {**  **this.product\_name = product\_name;**  **}**  **@Column(name = "product\_msrp")**  **public float getPrice() {**  **return product\_msrp;**  **}**  **public void setPrice(float product\_msrp) {**  **this.product\_msrp = product\_msrp;**  **}**  **@Column(name = "product\_in\_stock")**  **public Integer getStock() {**  **return product\_in\_stock;**  **}**  **public void setStock(Integer product\_in\_stock) {**  **this.product\_in\_stock = product\_in\_stock;**  **}**  **@Column(name = "product\_brand")**  **public String getBrand() {**  **return product\_brand;**  **}**  **public void setBrand(String product\_brand) {**  **this.product\_brand = product\_brand;**  **}**  **public void setId(Integer product\_id) {**  **this.product\_id = product\_id;**  **}**  **@Override**  **public String toString() {**  **return "ProductModel [id=" + product\_id + ", name=" + product\_name + ", price=" + product\_msrp + ", stock=" + product\_in\_stock + ", brand="**  **+ product\_brand + "]";**  **}**  **}**  **package com.shoe.model;**  **import javax.persistence.Column;**  **import javax.persistence.Entity;**  **import javax.persistence.GeneratedValue;**  **import javax.persistence.GenerationType;**  **import javax.persistence.Id;**  **import javax.persistence.Table;**  **import io.swagger.annotations.ApiModel;**  **import io.swagger.annotations.ApiModelProperty;**  **@Entity**  **@Table(name = "users")**  **@ApiModel(description = "Details about Users")**  **public class UserModel {**  **@ApiModelProperty(notes = "user\_id")**  **private Long user\_id;**  **@ApiModelProperty(notes = "user\_email")**  **private String user\_email;**  **@ApiModelProperty(notes = "user\_password")**  **private String user\_password;**  **@ApiModelProperty(notes = "user\_name")**  **private String user\_name;**      **public UserModel() {**  **super();**  **// TODO Auto-generated constructor stub**  **}**  **public UserModel(Long user\_id, String user\_email, String user\_password, String user\_name) {**  **super();**  **this.user\_id = user\_id;**  **this.user\_email = user\_email;**  **this.user\_password = user\_password;**  **this.user\_name = user\_name;**  **}**  **@Id**  **@GeneratedValue(strategy = GenerationType.IDENTITY)**  **@Column(name = "user\_id")**  **public Long getUser\_id() {**  **return user\_id;**  **}**  **public void setUser\_id(Long user\_id) {**  **this.user\_id = user\_id;**  **}**  **@Column(name = "user\_email")**  **public String getUser\_email() {**  **return user\_email;**  **}**  **public void setUser\_email(String user\_email) {**  **this.user\_email = user\_email;**  **}**  **@Column(name = "user\_password")**  **public String getUser\_password() {**  **return user\_password;**  **}**  **public void setUser\_password(String user\_password) {**  **this.user\_password = user\_password;**  **}**  **@Column(name = "user\_name")**  **public String getUser\_name() {**  **return user\_name;**  **}**  **public void setUser\_name(String user\_name) {**  **this.user\_name = user\_name;**  **}**  **@Override**  **public String toString() {**  **return "UserModel [user\_id=" + user\_id + ", user\_email=" + user\_email + ", user\_password=" + user\_password**  **+ ", user\_name=" + user\_name + "]";**  **}**  **}**   * **com.shoe.repository**   **package com.shoe.repository;**  **import org.springframework.data.jpa.repository.JpaRepository;**  **import com.shoe.model.AdminModel;**  **public interface AdminRepository extends JpaRepository<AdminModel, Integer> {**  **}**  **package com.shoe.repository;**  **import org.springframework.data.jpa.repository.JpaRepository;**  **import com.shoe.model.OrderModel;**  **public interface OrderRepository extends JpaRepository<OrderModel, Integer> {**  **}**  **package com.shoe.repository;**  **import org.springframework.data.jpa.repository.JpaRepository;**  **import com.shoe.model.ProductModel;**  **public interface ProductRepository extends JpaRepository<ProductModel, Integer> {**  **}**  **package com.shoe.repository;**  **import org.springframework.data.jpa.repository.JpaRepository;**  **import com.shoe.model.UserModel;**  **public interface UserRepository extends JpaRepository<UserModel, Long> {**  **}**   * **com.shoe.service**   **package com.shoe.service;**  **import javax.transaction.Transactional;**  **import org.springframework.stereotype.Service;**    **@Service**  **@Transactional**  **public class AdminService {**  **}**  **package com.shoe.service;**  **import javax.transaction.Transactional;**  **import org.springframework.stereotype.Service;**    **@Service**  **@Transactional**  **public class OrderService {**  **}**  **package com.shoe.service;**  **import javax.transaction.Transactional;**  **import org.springframework.stereotype.Service;**  **@Service**  **@Transactional**  **public class ProductService {**    **}**  **package com.shoe.service;**  **import javax.transaction.Transactional;**  **import org.springframework.stereotype.Service;**    **@Service**  **@Transactional**  **public class UserService {**  **}** |
|  |

# Tools Used

* Postman
* STS Tool
* MySQL Workbench

# Postman link

* https://www.getpostman.com/collections/4bdecb8638c19dcf9fbc

# GitHub URL:

https://github.com/Mallikarjunahugar/sportyshoes1.git