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

E-commerce Website

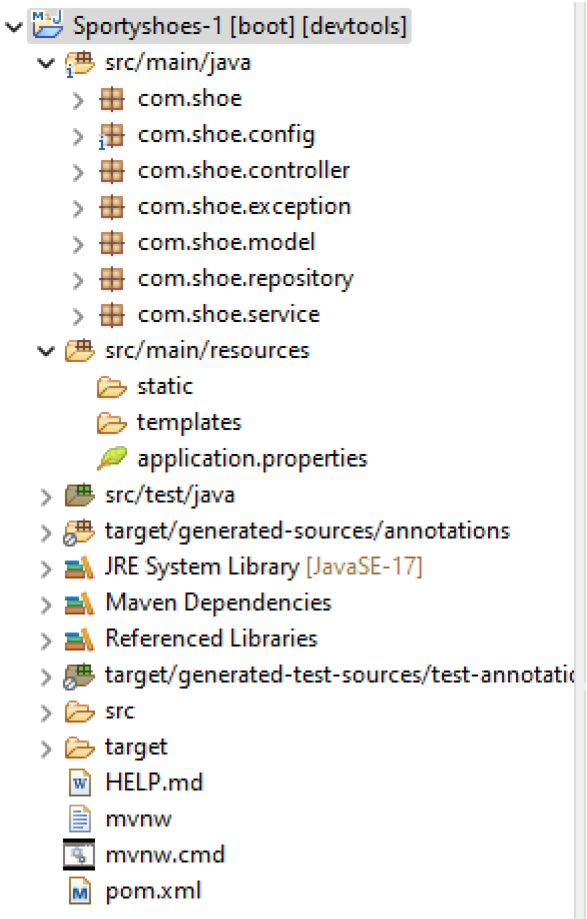
Contents

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



```
Sportyshoes-1 [boot] [devtools]
  src/main/java
    com.shoe
    com.shoe.config
    com.shoe.controller
    com.shoe.exception
    com.shoe.model
    com.shoe.repository
    com.shoe.service
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  target/generated-sources/annotations
  JRE System Library [JavaSE-17]
  Maven Dependencies
  Referenced Libraries
  target/generated-test-sources/test-annotatic
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

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) {
```

```java
                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