

WELCOME TO SPORTY SHOES

Powered by sportyshoe.com

Created a Maven Project using spring initializer called "SportyShoe" has features as below.

- Able to manage the products in the store including categorizing them
- Browse the list of users who have signed up and be able to search users
- See purchase reports filtered by date and category

Above Project is made on Spring Boot Application with maven dependency using Eclipse IDE and contains following configuration:

6 Main packages under -> Java Resource -> src/main/java as

1. com.shoe.SportyShoes
2. com.shoe.SportyShoes.model
3. com.shoe.SportyShoes.controller
4. com.shoe.SportyShoes.repository
5. com.shoe.SportyShoes.service
6. com.shoe.SportyShoes.serviceImpl
7. com.shoe.SportyShoes.exceptions

These main packages are having classes as listed below,

1. com.shoe.SportyShoes : This Package has the single Main class with Spring Run method.
2. com.shoe.SportyShoes.model: This package has 3 classes
 - AdminLogin.java
 - PurchaseReport.java
 - Shoe.java
3. com.shoe.SportyShoes.controller: This package has 3 classes

- AdminController.java
 - PurchaseReportController.java
 - ShoeController.java
4. com.shoe.SportyShoes.service: This package has 2 interfaces
 - AdminService.java
 - SportyShoesService.java
 5. com.shoe.SportyShoes.serviceImpl: This package has 2 classes
 - AdminServiceImpl.java
 - SportyShoeServiceImpl.java
 6. com.shoe.SportyShoes.exceptions: This package has 2 classes
 - ErrorConstant.java
 - SportyShoeBusinessException.java
 7. com.shoe.SportyShoes.repository: This has 3 classes
 - AdminRepository.java
 - PurchaseRepository.java
 - ShoesRepository.java

11 jsp files under src -> main -> webapp ->jsp

1. login.jsp
2. forgotpassword.jsp
3. welcome.jsp
4. addshoe.jsp
5. deleteshoe.jsp
6. find.jsp
7. findresult.jsp
8. addpurchasereport.jsp
9. deletepurchasereport.jsp
10. getpurchasereport.jsp
11. reportresult.jsp

1 Pom.xml file under Project having dependencies injected.

1 application.properties under src/main/resources folder for database and server information.

Application flow:

1. Running the project on given port number will display admin login page.
2. Admin has to login with correct credential or can reset the password if forgot.
3. Logged in successfully will bring admin on welcome page which has 7 best features supported for this application.

Add New Product
Search Available Shoes
Delete A Product
Add a Purchase Report
Delete a Purchase Report
Get Purchase Report
Log Out

4. Clicking on *Add New Product* provides an option to add new shoes in the database.
5. *Search Available Shoes* can help admin to pull the shoes already available in database.
6. Choosing *Delete A Product* option allow user to delete any existing shoe from database.
7. *Add A Purchase Report* support a feature for admin to create a purchase report in database for future reference.
8. Choosing to *Delete a Purchase Report* allows admin to delete an existing report from database by just entering its id.
9. With the help of *Get Purchase Report*, admin can pull the purchase report info available in database with the help of id, date if purchase and category. Also he can choose to see the all report.
10. A *log Out* option is present for Admin to safely logout. Also logout will bring back the admin on home page.
11. Similarly a separate custom exception class called SportyShoeBusinessException.java is present under com.shoe.SportyShoes.exceptions package with pre declared custom exceptions in ErrorConstant.java class used to through exception at required conditions.
12. Spring Boot repository features are used to communicate with SQL database and process queries depending on the request made.

13. Application Properties file is used to mention properties like server port number, create database etc.

Note: I Have tried handled most of the exception and for any unforeseen exception it will just refresh the page for user to enter the values again.