

# Sporty Shoes ↗

Phase 3 Project  
Project Documentation  
Arsha R Unnithan



# CLICKABLE TABLE OF CONTENTS

CLICK ON ANY OF THE ICONS TO JUMP TO A PARTICULAR SECTION OF THIS PRESENTATION



PROJECT  
OVERVIEW



APPLICATION  
LOGOS



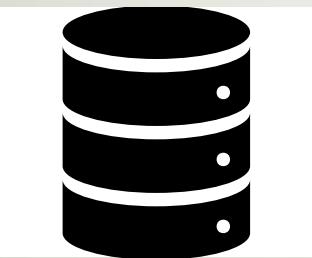
PROJECT  
MANAGEMENT



USER STORIES



DESIGN: API & DATABASE



SOURCE CODE



SCREENSHOTS &  
TESTING



PROJECT  
CONCLUSION

# Project Overview

- **Sporty Shoes** is a company that manufactures and sells sports shoes. They have a walk-in store, and now, they wish to launch their e-commerce portal [sportyshoes.com](http://sportyshoes.com). They decided to hire a Backend Developer to develop a prototype of the application.
- This project focuses on developing a REST API on the backend. It will be tested by using POSTMAN. There is **no frontend requirement** for this project. You should use dummy data for users, products, and order history.
- **Required Technical Components – Backend & Database Management**



IDE: Eclipse



Framework: Spring Boot



Programming Language: Java



Database: MySQL



# Core Concepts Demonstrated

- Create the Spring Boot application
- Leverage Spring Security features
- Establish the REST API
- Enable Swagger Documentation
- Database design and scripting
  - Create the MySQL database
  - Product information
  - User data
  - Initialization script
  - Populated tables via script and POSTMAN



# Features: Admin User Role

## Admin with the following features:

- There will be an admin to manage the website. An administrator login will be required to access the admin page. Therefore, the application should have a Login/Logout Feature
- Reset password
- Manage the products in the store including categorizing them
- Browse the list of users who have signed up and be able to search users

## Features that were not implemented

(ran out of time)

- See purchase reports filtered by date and category
- Add API endpoints for supplemental tables (i.e. brands, styles, occasions, etc.)





Sporty  
*shoes*



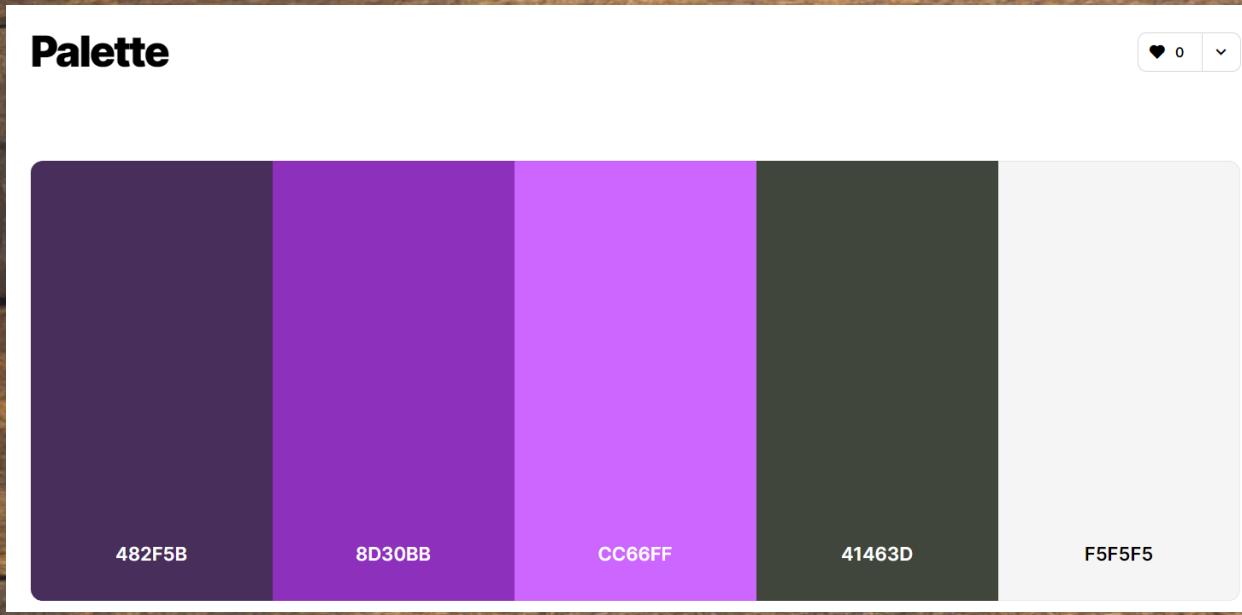
# Sporty Shoes App Logos



# Application Color Palette & Fonts

## Colors

### Palette



<https://colors.co/palette/482f5b-8d30bb-96998c-41463d-f5f5f5>

Original Base Color  
Original Accent Color

Color model:	RGB
Red:	204
Green:	102
Blue:	255
Hex:	#CC66FF

New  
Current

Color model:	RGB
Red:	112
Green:	48
Blue:	160
Hex:	#7030A0

New  
Current

## Fonts

Fox & Cat

Dreaming Outloud Script Pro



# Project Management (page 1:2)

Project Overview	
Client	Simplilearn
Consultant	Nia Kelley Jester Full Stack Java Developer
Application Name	Sporty Shoes
Application Phase	Prototype
Phase 3 Project Deliverable	Sporty Shoes is a company that manufactures and sells sports shoes. The goal of this project is to develop a REST API using the Spring framework. There is no frontend requirement for this project. You will also learn to test the API using POSTMAN.

Requirements Development			
Planned Start Date	Tuesday, December 20, 2022	Actual Start Date	Tuesday, December 20, 2022
Planned Finish Date	Tuesday, December 20, 2022	Actual Finish Date	Wednesday, December 21, 2022



# Project Management (page 2:2)

Code Development			
Holiday/Unplanned Outage (sick)	Saturday, December 24, 2022 – Wednesday, December 28, 2022		
Planned Project Code Duration	1 Week (3 working days)	Actual Project Code Duration	1.4 Weeks (7 working days)
Planned Start Date	Monday, October 31, 2022	Actual Start Date	Wednesday, December 21, 2022
Planned Finish Date	Friday, November 18, 2022	Actual Finish Date	Saturday, December 31, 2022
Planned Number of Sprints	3	Actual Number of Sprints	3
Planned Sprint Duration	1 Day	Actual Sprint Duration	2 Days
Total Number of Product Backlog Items	68		

Project Documentation			
Planned Start Date	Monday, December 19, 2022	Actual Start Date	Monday, December 19, 2022
Planned Finish Date	Monday, December 26, 2022	Actual Finish Date	Sunday, January 1, 2023

Project Summary		
	Planned	Actual
Total Project Weeks	1.2 Weeks	1.6 Weeks
Total Project Days	6 Days	8 Days
Total Project Hours	30 Hours	43 Hours



# Breakdown: Product Backlog by Sprint

Total Product Backlog	68 Items
User Scenarios	4
Sprint 1	33
Sprint 2	13
Sprint 3	4
Moved to next planning cycle	18



# Breakdown: Product Backlog by Scenario

Product Backlog ID	Status	Role	Scenario	Backlog Item (User Story)	Story Points	Assigned Sprint
1			Database Development			Header
1.1	Done	Database	Create	Create User Roles Table	Small	1
1.2	Done	Database	Create	Create Genders Table	Small	1
1.3	Done	Database	Create	Create Styles Table	Small	1
1.4	Done	Database	Create	Create Features Table	Small	1
1.5	Done	Database	Create	Create Occasions Table	Small	1
1.6	Done	Database	Create	Create Brands Table	Small	1
1.7	Done	Database	Create	Create Sizes Table	Medium	1
1.8	Done	Database	Create	Create Users Table	Medium	1
1.9	Done	Database	Create	Create Products Table	Medium	1
1.10	Done	Database	Create	Create Orders Table	Medium	1
1.11	Done	Database	Create	Create Order Items Table	Medium	1
1.12	Done	Database	Populate	Populate User Roles Table	Small	1
1.13	Done	Database	Populate	Populate Genders Table	Small	1
1.14	Done	Database	Populate	Populate Styles Table	Small	1
1.15	Done	Database	Populate	Populate Features Table	Small	1
1.16	Done	Database	Populate	Populate Occasions Table	Small	1
1.17	Done	Database	Populate	Populate Brands Table	Small	1
1.18	Done	Database	Populate	Populate Sizes Table	Small	1
1.19	Done	Database	Populate	Populate Users Table	Small	1
1.20	Done	Database	Populate	Populate Products Table	Small	1
1.21	Done	Database	Populate	Populate Orders Table	Small	1
1.22	Done	Database	Populate	Populate Order Items Table	Small	1



# Breakdown: Product Backlog by Scenario

Product Backlog ID	Status	Role	Scenario	Backlog Item (User Story)	Story Points	Assigned Sprint
2			User Management			
2.1	Done	Customer/Admin	Users	All Users will be stored in the backend database.	Medium	1
2.2	Done	Admin	Admin Login	Admin can login to the application	Small	1
2.3	Done	Admin	Admin Logout	Admin can logout of the application	Small	1
2.4	Done	Customer	Registered User Login	A registered user can login into the application	Small	1
2.5	Done	Customer	Registered User Logout	A registered user can logout into the application	Small	1
2.6	Done	Unregistered	Login	An unregistered user can not login to the application	Small	1
2.7	Done	Admin	List all users	Admin User can view a list of users stored in the database (HTTP GET)	Small	1
2.8	Done	Admin	Add a new User	Admin User can add a new user: first name, last name, user name, email address, phone number, and role type. (HTTP POST)	Medium	1
2.9	Done	Admin	Update User Details	Admin User can update the details for any user: first name, last name, user name, email address, phone number, and role type. (HTTP PUT)	Medium	1
2.10	Done	Admin	Change Password	Admin User can change reset passwords for any user. (HTTP PUT)	Medium	1
2.11	Not Working	Admin	Delete User	Admin User can delete a User from the database. (HTTP DELETE)	Small	1



# Breakdown: Product Backlog by Scenario

Product Backlog ID	Status	Role	Scenario	Backlog Item (User Story)	Story Points	Assigned Sprint		
3			<strong>Product Management</strong>					
3	3.1	Done	Admin	List all products	Admin User can view a list of products stored in the database (HTTP GET)	Small	2	
	3.2	Done	Admin	List all brands	Admin User can view a list of brands stored in the database (database only)	Small	2	
	3.3	Done	Admin	List all styles	Admin User can view a list of styles stored in the database (database only)	Small	2	
	3.4	Done	Admin	List all occasions	Admin User can view a list of occasions stored in the database (database only)	Small	2	
	3.5	Done	Admin	List all genders	Admin User can view a list of genders stored in the database (database only)	Small	2	
	3.6	Done	Admin	List all features	Admin User can view a list of features stored in the database (database only)	Small	2	
	3.7	Done	Admin	Add new product	Admin User can add a new product (HTTP POST)	Small	2	
	3.8	Done	Admin	Add new brand	Admin User can add a new brand (database only)	Small	2	
	3.9	Done	Admin	Add new style	Admin User can add a new style (database only)	Small	2	
	3.10	Done	Admin	Add new occasion	Admin User can add a new occasion (database only)	Small	2	
	3.11	Done	Admin	Add new feature	Admin User can add a new feature (database only)	Small	2	
	3.12	Done	Admin	Update Product Details	Admin User can update the details for any product. (HTTP PUT)	Medium	2	
	3.13	Not Working	Admin	Delete Product	Admin User can delete a User from the database. (HTTP DELETE)	Medium	2	
	3.14	Not Started	Admin	Add API Endpoint - Add Brand	Admin User can add a new brand (HTTP POST)	Small	Next Release	
	3.15	Not Started	Admin	Add API Endpoint - Update Brand	Admin User can update the details for any brand. (HTTP PUT)	Small	Next Release	



# Breakdown: Product Backlog by Scenario

Product Backlog ID	Status	Role	Scenario	Backlog Item (User Story)	Story Points	Assigned Sprint		
3			<b>Product Management</b>					
3	3.16	Not Started	Admin	Add API Endpoint - Delete Brand	Admin User can delete a brand from the database. (HTTP DELETE)	Medium	Next Release	
	3.17	Not Started	Admin	Add API Endpoint - Add Style	Admin User can add a new style (HTTP POST)	Small	Next Release	
	3.18	Not Started	Admin	Add API Endpoint - Update Style	Admin User can update the details for any style. (HTTP PUT)	Small	Next Release	
	3.19	Not Started	Admin	Add API Endpoint - Delete Style	Admin User can delete a style from the database. (HTTP DELETE)	Medium	Next Release	
	3.2	Not Started	Admin	Add API Endpoint - Add Occasion	Admin User can add a new occasion (HTTP POST)	Small	Next Release	
	3.21	Not Started	Admin	Add API Endpoint - Update Occasion	Admin User can update the details for any occasion. (HTTP PUT)	Small	Next Release	
	3.22	Not Started	Admin	Add API Endpoint - Delete Occasion	Admin User can delete a occasion from the database. (HTTP DELETE)	Medium	Next Release	
	3.23	Not Started	Admin	Add API Endpoint - Add Feature	Admin User can add a new feature (HTTP POST)	Small	Next Release	
	3.24	Not Started	Admin	Add API Endpoint - Update Feature	Admin User can update the details for any feature. (HTTP PUT)	Small	Next Release	
	3.25	Not Started	Admin	Add API Endpoint - Delete Feature	Admin User can delete a feature from the database. (HTTP DELETE)	Medium	Next Release	



# Breakdown: Product Backlog by Scenario

Product Backlog ID	Status	Role	Scenario	Backlog Item (User Story)	Story Points	Assigned Sprint		
4			Order Management					
4	4.1	Done	Admin	List orders	Admin User can view a list of orders stored in the database (database only)	Small	3	
	4.2	Done	Admin	List order items	Admin User can view a list of order items stored in the database (database only)	Small	3	
	4.3	Done	Admin	Update Order Details	Admin User can update orders stored in the database (database only)	Small	3	
	4.4	Done	Admin	Update Order Item Details	Admin User can update order items stored in the database (database only)	Small	3	
	4.5	Not Started	Admin	Add API Endpoint - Add Order	Admin User can add a new order (HTTP POST)	Small	Next Release	
	4.6	Not Started	Admin	Add API Endpoint - Update Order	Admin User can update the details for any order. (HTTP PUT)	Small	Next Release	
	4.7	Not Started	Admin	Add API Endpoint - Delete Order	Admin User can delete an order from the database. (HTTP DELETE)	Medium	Next Release	
	4.8	Not Started	Admin	Add API Endpoint - Add Order Item	Admin User can add a new order item (HTTP POST)	Small	Next Release	
	4.9	Not Started	Admin	Add API Endpoint - Delete Order Item	Admin User can update the details for any order item. (HTTP PUT)	Small	Next Release	
	4.10	Not Started	Admin	Add API Endpoint - Update Order Item	Admin User can delete a order item from the database. (HTTP DELETE)	Medium	Next Release	





# High Level Design

Sporty Shoes

# API DESIGN

## Api Documentation

Api Documentation

[Apache 2.0](#)

**home-controller** : Home Controller

		Show/Hide	List Operations	Expand Operations
GET	/admin		admin	
GET	/home		home	
GET	/user		user	

**login-controller** : Login Controller

		Show/Hide	List Operations	Expand Operations
GET	/login		login	

**product-controller** : Product Controller

		Show/Hide	List Operations	Expand Operations
POST	/api/admin/products/add		addProduct	
GET	/api/admin/products/all		getAllProducts	
DELETE	/api/admin/products/product/delete/{id}		deleteProduct	
PUT	/api/admin/products/product/update/{id}		updateProduct	
GET	/api/admin/products/product/{id}		getProductById	

**user-controller** : User Controller

		Show/Hide	List Operations	Expand Operations
POST	/api/admin/users/add		addUser	
GET	/api/admin/users/all		getAllUsers	
DELETE	/api/admin/users/user/delete/{id}		deleteUser	
PUT	/api/admin/users/user/update/{id}		updateUser	
GET	/api/admin/users/user/{id}		getUserById	

[ BASE URL: / , API VERSION: 1.0 ]



## BACKEND SERVER



**REST  
CONTROLLER**

**REST SERVICES**

**REST  
REPOSITORIES**

**REST MODELS**



- Built the server, controller, access to the database, and the relevant endpoints
- The backend is library agnostic, meaning it can communicate with any tool that has the capability of sending HTTP requests (i.e. POSTMAN, Angular, etc.)
- Reminder: the REST Controller is the part of the backend (server) that will be exposed to the outside world.



# Database Design

Sporty Shoes

# Database Design

## Tables with no foreign key association

### styles

**PK:** styleId  
styleName

### genders

**PK:** genderId  
genderName

### features

**PK:** featureId  
featureName

### occasions

**PK:** occasionId  
occasionName

### brands (1:M products)

**PK:** brandId  
brandName

**PRIMARY KEY (PK)**  
**FOREIGN KEY (FK)**

## Tables WITH foreign key associations or indexes

### users (1:M orders)

**PK:** username  
roleType  
firstname  
lastname  
userid  
email  
password  
phone\_number  
date\_created  
enabled

### orders

**(1:M order\_items)**  
**PK:** orderId  
**FK:** username  
orderNum  
shippingCost  
orderTotal  
orderDate  
date\_created

### products

**PK:** productId  
**FK:** genderId  
**FK:** brandId  
**FK:** styleId  
**FK:** featureId  
**FK:** occasionId  
productName  
productDescription  
unitPrice  
date\_created

### order\_items

**PK:** itemId  
**FK:** orderId  
**FK:** productId  
**FK:** sizeId  
quantity  
subtotal

### sizes

**PK:** sizeId  
**FK:** genderId  
sizeName

### authorities

**PK:** username  
authority



Had to use a script to create this table, due to an index.

**Note:** USERS and AUTHORITIES tables are default table names in Spring JDBC. The default JDBC authentication provided by Spring Security uses these table names also. Wound up not using this in implementation.

# Database Design: Data Tables (1 of 5)

```
mysql> show tables;
+-----+
| Tables_in_sportyshoesdb |
+-----+
| authorities
| brands
| features
| genders
| occasions
| order_items
| orders
| products
| sizes
| styles
| users
+-----+
11 rows in set (0.00 sec)
```

```
mysql> select * from brands;
+-----+
| brand_id | brand_name |
+-----+
| 1 | New Balance
| 2 | Nike
| 3 | Michael Kors
| 4 | Coach
| 5 | VANS
| 6 | Reebok
| 7 | Adidas
| 8 | Cole Haan
| 9 | Kelly & Katie
| 10 | Franco Sarto
| 11 | Steve Madden
| 12 | Anne Klein
| 13 | Sketchers
| 14 | Timberland
| 15 | Vince Camuto
| 16 | Brooks
+-----+
16 rows in set (0.00 sec)
```

```
mysql> select * from features;
+-----+
| feature_id | feature_name |
+-----+
| 1 | track running
| 2 | comfort
| 3 | insulated
| 4 | lightweight
| 5 | memory foam
| 6 | road running
| 7 | trail hiking/running
| 8 | water-resistant
| 9 | washable
| 10 | padded footbed
+-----+
10 rows in set (0.00 sec)
```

```
mysql> select * from occasions;
+-----+
| occasion_id | occasion_name |
+-----+
| 1 | athleisure
| 2 | beach/pool
| 3 | casual
| 4 | cold weather
| 5 | fitness & active
| 6 | fashion & street
| 7 | special occasion
| 8 | sport
| 9 | travel
| 10 | business
+-----+
10 rows in set (0.00 sec)
```

```
mysql> select * from styles;
+-----+
| style_id | style_name |
+-----+
| 1 | sneaker
| 2 | running
| 3 | slide
| 4 | slingback
| 5 | wedge
| 6 | boots
| 7 | flats
| 8 | pump
| 9 | loafers
| 10 | slip on
+-----+
10 rows in set (0.00 sec)
```

```
mysql> select * from genders;
+-----+
| gender_id | gender_name |
+-----+
| 1 | women
| 2 | men
| 3 | boys
| 4 | girls
+-----+
4 rows in set (0.00 sec)
```

# Database Design: Data Tables (2 of 5)

```
mysql> select * from users;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| username | date_created | email | enabled | firstname | userid | lastname | password | phone_number | role_type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| christine | 2022-12-31 17:56:00.000000 | cjohnson@home.com | 0x01 | Christine | 3 | Johnson | $2a$12$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtfKK | 9858851975 | USER |
| clementine | 2022-12-31 17:56:00.000000 | bauchc@home.com | 0x01 | Clementine | 4 | Bauch | $2a$12$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtfKK | 6059902842 | USER |
| jasen | 2022-12-31 17:56:00.000000 | jasen@home.com | 0x01 | Jasen | 2 | Jester | $2a$12$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtfKK | 8783019215 | USER |
| monica | 2022-12-31 17:56:00.000000 | msharp@home.com | 0x01 | Monica | 6 | Sharp | $2a$12$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtfKK | 4062991259 | USER |
| msmith | 2022-12-31 17:56:00.000000 | michael@home.com | 0x01 | Michael | 8 | Smith | $2a$12$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtfKK | 7604129733 | USER |
| niajester | 2022-12-31 17:54:28.000000 | nia@home.com | 0x01 | Nia | 1 | Kelley Jester | $2a$12$P21wlkhXzMRD3t3UHTnx..Y3sninurN3Q4EH04F7LU8vD/zChj0S | 7450495098 | ADMIN |
| ozbrown | 2022-12-31 17:56:00.000000 | ozbrown@home.com | 0x01 | Oswald | 7 | Brown | $2a$12$P21wlkhXzMRD3t3UHTnx..Y3sninurN3Q4EH04F7LU8vD/zChj0S | 9015753955 | ADMIN |
| patw | 2022-12-31 17:56:00.000000 | patweissnat@home.com | 0x01 | Patricia | 5 | Weissnat | $2a$12$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtfKK | 4316974631 | USER |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

- Used [bCrypt-Generator.com](https://www.bcrypt-generator.com) to hashed passwords
- Admin password = admin123
- User password = abc123

```
mysql> select * from authorities;
+-----+-----+
| username | authority |
+-----+-----+
| christine | USER |
| clementine | USER |
| jasen | USER |
| monica | USER |
| msmith | USER |
| niajester | ADMIN |
| ozbrown | ADMIN |
| patw | USER |
+-----+-----+
8 rows in set (0.00 sec)
```

# Database Design: Data Tables (3 of 5)

```
mysql> select * from orders;
+-----+-----+-----+-----+-----+-----+
| order_id | date_created | order_date | order_num | order_total | shipping_cost | username |
+-----+-----+-----+-----+-----+-----+
| 1 | 2022-12-31 17:56:22.000000 | 2022-12-20 13:23:44.000000 | WI1001NK | 567.00 | 0.00 | niajester |
| 2 | 2022-12-31 17:56:22.000000 | 2022-09-09 10:06:20.000000 | FA5030MS | 317.50 | 12.50 | msmith |
| 3 | 2022-12-31 17:56:22.000000 | 2022-07-22 09:34:19.000000 | SU2039MS | 439.00 | 0.00 | monica |
| 4 | 2022-12-31 17:56:22.000000 | 2022-04-04 20:15:31.000000 | SP2108CB | 125.00 | 20.00 | clementine |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> select * from order_items;
+-----+-----+-----+-----+-----+-----+
| item_id | quantity | subtotal | order_id | product_id | size_id |
+-----+-----+-----+-----+-----+-----+
| 1 | 2 | 458.00 | 1 | 2 | 10 |
| 2 | 1 | 79.00 | 1 | 4 | 10 |
| 3 | 1 | 30.00 | 1 | 10 | 60 |
| 4 | 1 | 65.00 | 2 | 5 | 32 |
| 5 | 1 | 240.00 | 2 | 7 | 32 |
| 6 | 1 | 60.00 | 3 | 1 | 6 |
| 7 | 2 | 300.00 | 3 | 3 | 6 |
| 8 | 1 | 79.00 | 3 | 4 | 6 |
| 9 | 1 | 50.00 | 4 | 9 | 63 |
| 10 | 1 | 55.00 | 4 | 8 | 63 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

# Database Design: Data Tables (4 of 5)

```
mysql> select * from products;
+-----+-----+
| product_id | date_created | product_description
+-----+-----+
| 1 | 2022-12-31 17:56:22.000000 | Surround your foot with the casual care-free style of the Domani flat sandal from Anne Klein. Designed with iFlex technology for 90-degree sole bendability, this slide sandal boasts of an elegant knot detail at the front that expresses chill vibes.
| 2 | 2022-12-31 17:56:22.000000 | You cannot go wrong with a classic look, and the Arendie boot by Vince Camuto proves it with its suede construction and sleek silhouette that bring timeless flair to your favorite ensembles. While its look draws you in, the padded footbed and mesh lining ensure a comfortable fit to keep you going.
| 3 | 2022-12-31 17:56:22.000000 | Bring a menswear-inspired vibe to your look with the Knit OG Grand wingtip oxford from Cole Haan. These knitted lace-ups are completed with GRAND.ØS cushioning and a flexible rubber sole to accompany your active lifestyle.
| 4 | 2022-12-31 17:56:22.000000 | Balanced on a mid-height heel that feels just right, the Misha pump from Kelly & Katie serves up a timeless silhouette in of-the-moment uppers that matches the needs of your rotating closet.
| 5 | 2022-12-31 17:56:22.000000 | Hit the track with ease in the retro-inspired Run 70s running shoe from adidas. This mixed suede & textile design proves sturdy, while the foam midsole and full-rubber sole with traction pods provides padding and grip to keep you moving.
| 6 | 2022-12-31 17:56:22.000000 | The Ethen Oxford from Vince Camuto can effortlessly be dressed up or down thanks to its versatile silhouette. The sneaker-inspired sole complements the smooth leather upper and lace-up design for your new favorite pair.
| 7 | 2022-12-31 17:56:22.000000 | Looking for the right mix of style and functionality? Then choose the Cole Haan Grand Atlantic chukka boot. The water-resistant leather, lace-up front, comfort-enhancing fully padded sock lining, EVA sole with ultimate cushioning technology and rubber pods in heel and toe complete this ankle boot.
| 8 | 2022-12-31 17:56:22.000000 |
| 9 | 2022-12-31 17:56:22.000000 | Keep the little foot warm and cozy in the Timber Tykes snow boot from Timberland. The Splash Blaster protective coating and OrthoLite footbed elevate the comfort and offer protection.
| 10 | 2022-12-31 17:56:22.000000 | Classy and a little bit sassy, the Fiona flat from Kelly & Katie keeps your little one looking charming. Glittery accents on the cap toe and slim heel complement the subtle bow accent for their new favorite for dances and parties.
+-----+-----+
| product_name | unit_price | brand_id | feature_id | gender_id | occasion_id | style_id |
+-----+-----+
| Domani Sandal | 60.00 | 12 | 2 | 1 | 3 | 7 |
| ARENDIE BOOT | 229.00 | 15 | 10 | 1 | 6 | 6 |
| Knit OG Grand Wingtip Oxford | 150.00 | 8 | 2 | 1 | 1 | 9 |
| Misha Pump | 79.00 | 9 | 2 | 1 | 10 | 8 |
| 70s Running Shoe | 65.00 | 7 | 10 | 2 | 5 | 2 |
| Ethen Oxford | 139.00 | 15 | 2 | 2 | 10 | 9 |
| Grand Atlantic Chukka Boot | 240.00 | 8 | 2 | 2 | 3 | 6 |
| Timber Tykes Snow Boot - Kids | 55.00 | 14 | 8 | 3 | 4 | 6 |
| Ward Sneaker - Kids | 50.00 | 5 | 10 | 3 | 3 | 1 |
| Fiona Flat - Kids | 30.00 | 9 | 2 | 4 | 6 | 7 |
+-----+-----+
10 rows in set (0.00 sec)
```

# Database Design: Data Tables (5 of 5)

```
mysql> select * from sizes;
+-----+-----+-----+
| size_id | size_name | gender_id |
+-----+-----+-----+
| 1      | 4          | 1        |
| 2      | 5          | 1        |
| 3      | 5.5        | 1        |
| 4      | 6          | 1        |
| 5      | 6.5        | 1        |
| 6      | 7          | 1        |
| 7      | 7.5        | 1        |
| 8      | 8          | 1        |
| 9      | 8.5        | 1        |
| 10     | 9          | 1        |
| 11     | 9.5        | 1        |
| 12     | 10         | 1        |
| 13     | 10.5       | 1        |
| 14     | 11         | 1        |
| 15     | 12         | 1        |
| 16     | 13         | 1        |
| 17     | 14         | 1        |
| 18     | 5          | 2        |
| 19     | 6          | 2        |
| 20     | 6.5        | 2        |
| 21     | 7          | 2        |
| 22     | 7.5        | 2        |
| 23     | 8          | 2        |
| 24     | 8.5        | 2        |
| 25     | 9          | 2        |
| 26     | 9.5        | 2        |
| 27     | 10         | 2        |
| 28     | 10.5       | 2        |
| 29     | 11         | 2        |
| 30     | 12         | 2        |
| 31     | 12.5       | 2        |
| 32     | 13         | 2        |
| 33     | 13.5       | 2        |
| 34     | 14         | 2        |
| 35     | 15         | 2        |
| 36     | 4T         | 3        |
| 37     | 5T         | 3        |
| 38     | 6T         | 3        |
| 39     | 7T         | 3        |
+-----+-----+-----+
```

40	8T	3
41	9T	3
42	10T	3
43	11Y	3
44	12Y	3
45	2Y	3
46	3Y	3
47	3.5Y	3
48	4Y	3
49	4.5Y	3
50	5Y	3
51	5.5Y	3
52	6Y	3
53	6.5Y	3
54	4T	4
55	5T	4
56	6T	4
57	7T	4
58	8T	4
59	9T	4
60	10T	4
61	11Y	4
62	12Y	4
63	2Y	4
64	3Y	4
65	3.5Y	4
66	4Y	4
67	4.5Y	4
68	5Y	4
69	5.5Y	4
70	6Y	4
71	6.5Y	4
72	7Y	4
73	One Size	4

73 rows in set (0.00 sec)

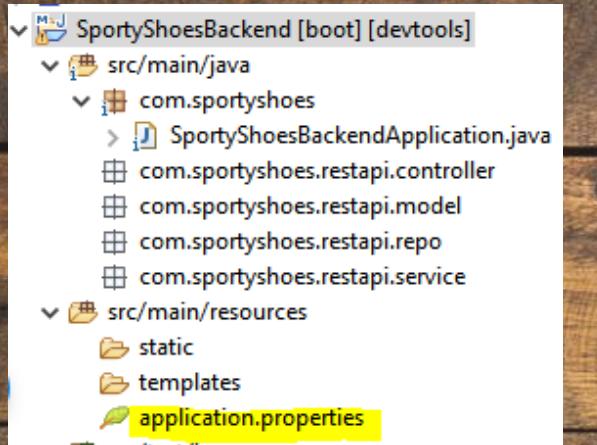


# **DATABASE & SWAGGER2 CONFIGURATION**



# Database Connectivity

**Database Connectivity:** Create the application.properties file to setup connectivity to the database



```
1 #MySQL Configuration
2 spring.datasource.username=root
3 spring.datasource.password=DesignerNia27!
4 spring.datasource.url=jdbc:mysql://localhost:3306/sportyshoesdb
5 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
6
7 spring.sql.init.platform = mysql
8 spring.devtools.restart.enabled = true
9
10 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
11 spring.jpa.hibernate.ddl-auto=update
12 spring.jpa.show-sql=true
13
14 server.port=8080
15
16 #For Swagger Documentation
17 spring.mvc.pathmatch.matching-strategy=ANT_PATH_MATCHER
```

**Database Creation:** Create the database (sql -> sportyshoesdb)

- 1 • CREATE DATABASE IF NOT EXISTS sportyshoesdb;
- 2 • USE sportyshoesdb;



# Update pom.xml

**Swagger Dependencies:** Update pom.xml to include the dependencies for Swagger2 and SwaggerUI

```
https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>2.7.5</version> <!-- 3.0.0 -->
9     <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>com.sportyshoes</groupId>
12    <artifactId>SportyShoesBackend</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>SportyShoesBackend</name>
15    <description>Project for RESTful API with Spring Boot</description>
16    <properties>
17      <java.version>17</java.version>
18    </properties>
54
55    <!-- https://mvnrepository.com/artifact/io.springfox/springfox-swagger2 -->
56    <dependency>
57      <groupId>io.springfox</groupId>
58      <artifactId>springfox-swagger2</artifactId>
59      <version>2.7.0</version>
60    </dependency>
61
62    <!-- https://mvnrepository.com/artifact/io.springfox/springfox-swagger-ui -->
63    <dependency>
64      <groupId>io.springfox</groupId>
65      <artifactId>springfox-swagger-ui</artifactId>
66      <version>2.7.0</version>
67    </dependency>
```

## Notes:

1. I couldn't get it to work with the 3.0.0 version of Swagger2 and Swagger-UI. I had to switch to the 2.7.0 version (which Ms. Sonam showed us during class). Also, I had to use 2.7.5 of spring-boot-starter-parent.
2. I found the following article helpful: <https://www.baeldung.com/swagger-2-documentation-for-spring-rest-api>



It's important to mention that the latest version of Swagger specification, now known as OpenAPI 3.0, is better supported by the Springdoc project and should be used for documenting Spring REST API.



# Error using 3.0.0 spring-boot-starter-parent

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Markers Properties Servers Snippets Terminal Console X Progress JUnit

<terminated> SportyShoesBackend - SportyShoesBackendApplication [Spring Boot App] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Dec 21, 2022, 8:26:29 PM – 8:26:34 PM) [pid: 18540]

```
2022-12-21T20:26:34.358-06:00  WARN 18540 --- [ restartedMain] ocalVariableTableParameterNameDiscoverer : Using deprecated '-debug' fallback for parameter name resolution
2022-12-21T20:26:34.359-06:00  WARN 18540 --- [ restartedMain] ocalVariableTableParameterNameDiscoverer : Using deprecated '-debug' fallback for parameter name resolution
2022-12-21T20:26:34.376-06:00  WARN 18540 --- [ restartedMain] ConfigServletWebServerApplicationContext : Exception encountered during context initialization - cancelling refresh attempt: org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with name 'apiDescriptionReader' defined in URL [jar:file:/C:/Users/niake/.m2/repository/org/springframework/boot/spring-boot-starter-data-rest/2.7.12/spring-boot-starter-data-rest-2.7.12.jar!/:spring-boot-starter-data-rest-2.7.12.jar]: Unsatisfied dependency expressed through field 'apiDescriptionReader'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'org.springframework.boot.autoconfigure.condition.ConditionEvaluationReportLogger': Bean instantiation via factory method failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'org.springframework.boot.SpringApplication': Bean instantiation via factory method failed; nested exception is org.springframework.context.ApplicationContextException: Application run failed
2022-12-21T20:26:34.377-06:00  INFO 18540 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2022-12-21T20:26:34.381-06:00  INFO 18540 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2022-12-21T20:26:34.392-06:00  INFO 18540 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
2022-12-21T20:26:34.393-06:00  INFO 18540 --- [ restartedMain] o.apache.catalina.core.StandardService : Stopping service [Tomcat]
2022-12-21T20:26:34.408-06:00  INFO 18540 --- [ restartedMain] .s.b.a.l.ConditionEvaluationReportLogger : 
```

Error starting ApplicationContext. To display the condition evaluation report re-run your application with 'debug' enabled.

```
2022-12-21T20:26:34.429-06:00 ERROR 18540 --- [ restartedMain] o.s.boot.SpringApplication : Application run failed
```

[org.springframework.beans.factory.UnsatisfiedDependencyException](#): Error creating bean with name 'apiDescriptionReader' defined in URL [jar:file:/C:/Users/niake/.m2/repository/org/springframework/boot/spring-boot-starter-data-rest/2.7.12/spring-boot-starter-data-rest-2.7.12.jar!/:spring-boot-starter-data-rest-2.7.12.jar]: Unsatisfied dependency expressed through field 'apiDescriptionReader'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'org.springframework.boot.autoconfigure.condition.ConditionEvaluationReportLogger': Bean instantiation via factory method failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'org.springframework.boot.SpringApplication': Bean instantiation via factory method failed; nested exception is org.springframework.context.ApplicationContextException: Application run failed

```
at org.springframework.beans.factory.support.ConstructorResolver.createArgumentArray(ConstructorResolver.java:793) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.beans.factory.support.ConstructorResolver.autowireConstructor(ConstructorResolver.java:242) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.autowireConstructor(AbstractAutowireCapableBeanFactory.java:1344) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBeanInstance(AbstractAutowireCapableBeanFactory.java:1188) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:561) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:521) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.beans.factory.support.AbstractBeanFactory.lambda$doGetBean$0(AbstractBeanFactory.java:326) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:234) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.beans.factory.support.AbstractBeanFactory doGetBean(AbstractBeanFactory.java:324) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:200) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:961) ~[spring-beans-6.0.2.jar:6.0.2]
at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:915) ~[spring-context-6.0.2.jar:6.0.2]
at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:582) ~[spring-context-6.0.2.jar:6.0.2]
at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:734) ~[spring-boot-2.7.12.jar:2.7.12]
at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:438) ~[spring-boot-2.7.12.jar:2.7.12]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:330) ~[spring-boot-2.7.12.jar:2.7.12]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1311) ~[spring-boot-2.7.12.jar:2.7.12]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1300) ~[spring-boot-2.7.12.jar:2.7.12]
```

## Notes:

1. I couldn't get it to work with the 3.0.0 version of Swagger2 and Swagger-UI. I had to switch to the 2.7.0 version (which Ms. Sonam showed us during class). Also, I had to use 2.7.5 of spring-boot-starter-parent.
  2. I found the following article helpful: <https://www.baeldung.com/swagger-2-documentation-for-spring-rest-api>

[It's important to mention that the latest version of Swagger specification, now known as OpenAPI 3.0, is better supported by the Springdoc project and should be used for documenting Spring REST API.](#)



# Updated main application

**Swagger Dependencies:** Update SportyShoesBackendApplication.java to include annotations and an explicit Bean for Swagger2

```
1 package com.sportyshoes;
2
3+import org.springframework.boot.SpringApplication;
11
12 @SpringBootApplication
13 @EnableSwagger2
14 public class SportyShoesBackendApplication {
15
16@    public static void main(String[] args) {
17        SpringApplication.run(SportyShoesBackendApplication.class, args);
18    }
19
20    //For Swagger documentation
21@    @Bean
22    public Docket allApis() {
23        return new Docket(DocumentationType.SWAGGER_2)
24            .select()
25            .apis(RequestHandlerSelectors.basePackage("com.sportyshoes.restapi"))
26            .build();
27    }
28
29 } //end class
```



# At this point, you can start the Spring Boot application.

```
SportyShoesBackend - SportyShoesBackendApplication [Spring Boot App] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Dec 21, 2022, 8:30:28 PM) [pid: 3420]
2022-12-21 20:30:30.142 INFO 3420 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties=true' to see them.
2022-12-21 20:30:30.142 INFO 3420 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level' property to 'INFO'.
2022-12-21 20:30:30.918 INFO 3420 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-12-21 20:30:30.933 INFO 3420 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 6 ms. Found 0 JPA repositories.
2022-12-21 20:30:31.558 INFO 3420 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http).
2022-12-21 20:30:31.569 INFO 3420 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-12-21 20:30:31.569 INFO 3420 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.68]
2022-12-21 20:30:31.684 INFO 3420 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-12-21 20:30:31.684 INFO 3420 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1541 ms.
2022-12-21 20:30:31.875 INFO 3420 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2022-12-21 20:30:31.927 INFO 3420 --- [ restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.12.Final
2022-12-21 20:30:32.089 INFO 3420 --- [ restartedMain] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2022-12-21 20:30:32.204 INFO 3420 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-12-21 20:30:32.499 INFO 3420 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-12-21 20:30:32.516 INFO 3420 --- [ restartedMain] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
2022-12-21 20:30:32.782 INFO 3420 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2022-12-21 20:30:32.792 INFO 3420 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2022-12-21 20:30:32.998 WARN 3420 --- [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during a view's rendering. Explicitly disable this feature if needed.
2022-12-21 20:30:33.132 INFO 3420 --- [ restartedMain] pertySourcedRequestMappingHandlerMapping : Mapped URL path [/v2/api-docs] onto method [springfox.documentation.swagger2.annotations.EnableSwagger2].
2022-12-21 20:30:33.271 INFO 3420 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2022-12-21 20:30:33.419 WARN 3420 --- [ restartedMain] .s.s.UserDetailsServiceAutoConfiguration : 

Using generated security password: d3f3ba76-a293-4a01-9329-5841cea29753 For Spring Security. Username is user

This generated password is for development use only. Your security configuration must be updated before running your application in production.

2022-12-21 20:30:33.540 INFO 3420 --- [ restartedMain] o.s.s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.session.SessionManagementFilter].
2022-12-21 20:30:33.609 INFO 3420 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''.
2022-12-21 20:30:33.610 INFO 3420 --- [ restartedMain] d.s.w.p.DocumentationPluginsBootstrapper : Context refreshed.
2022-12-21 20:30:33.627 INFO 3420 --- [ restartedMain] d.s.w.p.DocumentationPluginsBootstrapper : Found 1 custom documentation plugin(s).
2022-12-21 20:30:33.654 INFO 3420 --- [ restartedMain] s.d.s.w.s.ApiListingReferenceScanner : Scanning for api listing references.
2022-12-21 20:30:33.673 INFO 3420 --- [ restartedMain] c.s.SportyShoesBackendApplication : Started SportyShoesBackendApplication in 3.951 seconds (JVM running for 3.951 seconds).
```



# Running API Documentation

<http://localhost:8080/swagger-ui.html>

The screenshot shows the Swagger UI interface running at <http://localhost:8080/swagger-ui.html#/home45controller>. The top navigation bar includes back, forward, refresh, and search icons, along with the URL and a dropdown for selecting the API version (default /v2/api-docs). The main content area is titled "Api Documentation" and "Api Documentation" under "Apache 2.0".

**home-controller : Home Controller**

Method	Path	Role
GET	/admin	admin
GET	/home	home
GET	/user	user

**login-controller : Login Controller**

Method	Path	Role
GET	/login	login

**product-controller : Product Controller**

Method	Path	Role
POST	/api/admin/products/add	addProduct
GET	/api/admin/products/all	getAllProducts
DELETE	/api/admin/products/product/delete/{id}	deleteProduct
PUT	/api/admin/products/product/update/{id}	updateProduct
GET	/api/admin/products/product/{id}	getProductById

**user-controller : User Controller**

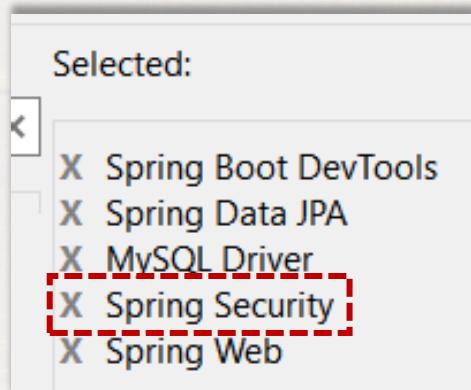
Method	Path	Role
POST	/api/admin/users/add	addUser
GET	/api/admin/users/all	getAllUsers
DELETE	/api/admin/users/user/delete/{id}	deleteUser
PUT	/api/admin/users/user/update/{id}	updateUser
GET	/api/admin/users/user/{id}	getUserById

[ BASE URL: / , API VERSION: 1.0 ]

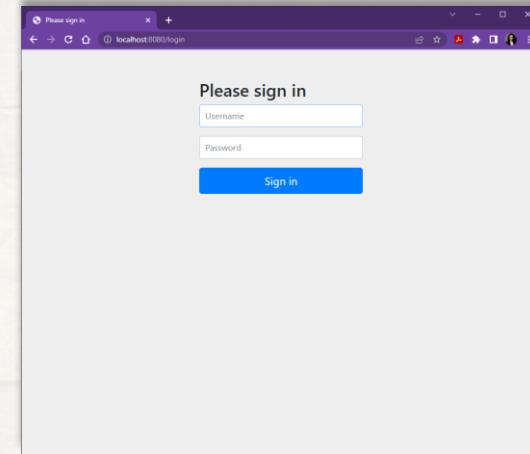


# Spring Security Dependency

- Simplest way to provide security in a Spring Boot application (`spring-boot-starter-security`)
- It brings all the necessary libraries and enables Spring Security in the Spring Boot application.



- Spring Security generates a default login page for user login and allows the user to log out from the application.



- Spring Security provides a default user named `user` and generates a default password (printed in the console log) for form-based login.

# Sporty Shoes: Spring Security Tasks

Configure Spring Security to do the following:

1. Provide a custom login page that replaces the default login page that comes with Spring Security
2. Provide for multiple users stored in a database
3. Apply different security rules for different request paths. The login page, for example, shouldn't require authentication at all. Created various endpoints for testing purposes.

# Source Code

All code for this project can be found at the following link:  
<https://github.com/arsharunnithan/sportyShoes>



Type your text





# Application Screenshots & Testing

Phase 3 Project

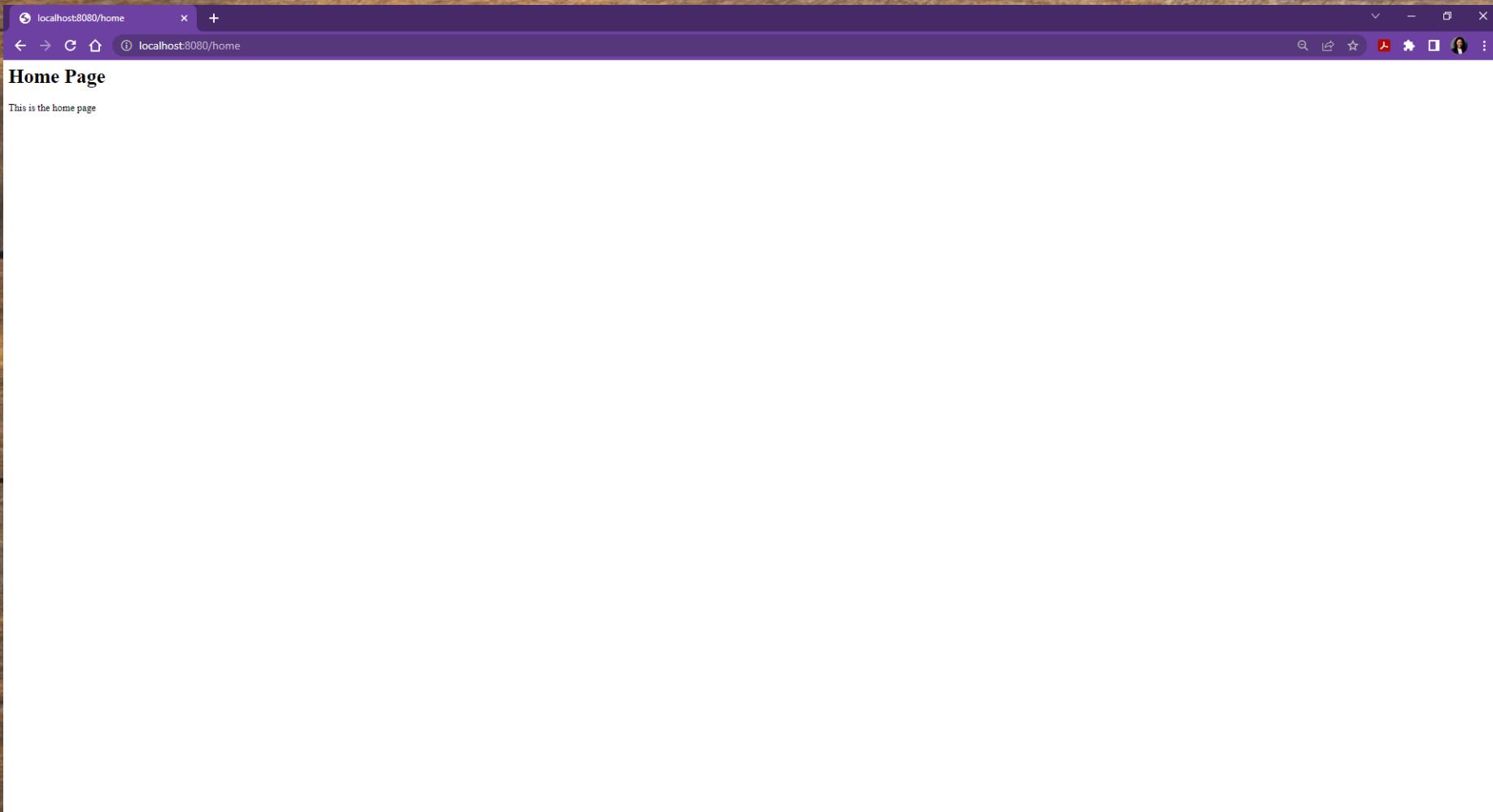
Nia Kelley Jester

December 2022

Instructor: Sonam Soni

# Home Page

No restrictions





# TESTING THE BACKEND

*with BROWSER as  
an ADMIN User*



# Note about testing with POSTMAN

## 1. Provide Basic Auth credentials

The screenshot shows the Postman interface for a GET request to `localhost:8080/api/admin/users/all`. The 'Authorization' tab is selected, and the 'Type' dropdown is set to 'Basic A...'. The 'Username' field contains 'niajester' and the 'Password' field contains 'admin123'. A red arrow points from the 'Authorization' tab to the 'Type' dropdown.

## 2. Find the JSESSIONID token from localhost (after login successful)

The screenshot shows the Chrome DevTools Network tab for a request to `localhost:8080`. The 'Application' section is selected, and the 'Cookies' table shows a row for 'JSESSIONID' with the value 'D123A86EB6BD1397190E28FFE2FD12...'. A red arrow points from the 'Application' tab to the 'Cookies' table.

## 3. Add the JSESSIONID to POSTMAN

The screenshot shows the Postman interface for the same GET request. The 'Headers' tab is selected, and the 'Cookie' header is added with the value 'JSESSIONID=D123A86EB6BD1397190E28...'. A red arrow points from the 'Cookie' row in the Headers table to the 'Cookie' field in the Body tab.



# Custom Login Page

Logged in as Admin Role

The screenshot shows a web browser window titled "Sporty Shoes Login Form" with the URL "localhost:8080/login". The page features a large, stylized purple logo on the left that reads "Sporty Shoes". The main content area has a light gray background with the text "Please sign in" centered. It contains two input fields: one for the username with the value "ozbrown" and another for the password with the value ".....". Below the inputs is a blue "Sign in" button.

## Spring Console

The screenshot shows the "Spring Console" interface with the title "SportyShoesBackend - SportyShoeBackendApplication [Spring Boot App] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Jan 1, 2023, 6:57:23 PM) [pid: 32016]". The console output is as follows:

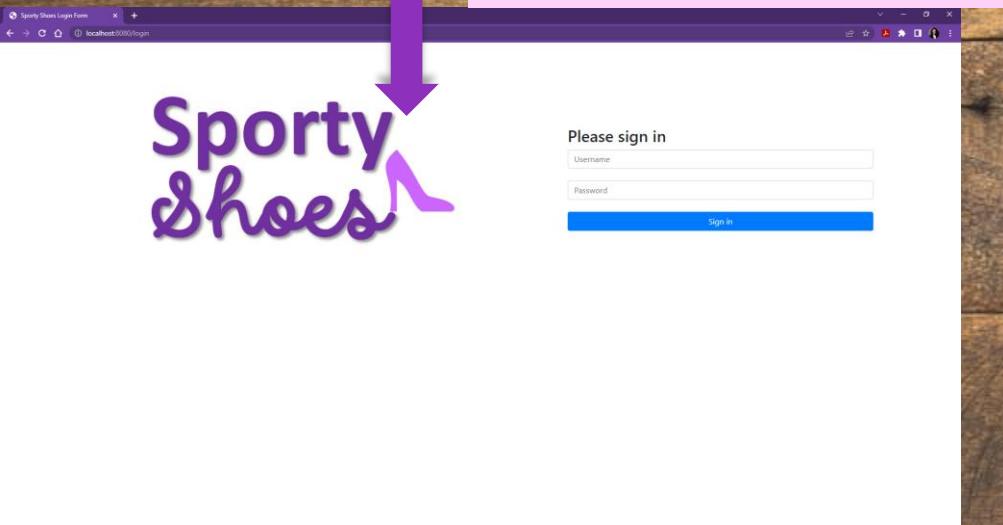
```
2023-01-01 18:57:32.091  INFO 32016 --- [ restartedMain] d.s.w.p.DocumentationPluginsBootstrapper : Context refreshed
2023-01-01 18:57:32.107  INFO 32016 --- [ restartedMain] d.s.w.p.DocumentationPluginsBootstrapper : Found 1 custom documentation plugin(s)
2023-01-01 18:57:32.134  INFO 32016 --- [ restartedMain] s.d.s.w.s.ApiListingReferenceScanner : Scanning for api listing references
2023-01-01 18:57:32.326  INFO 32016 --- [ restartedMain] c.s.SportyShoesBackendApplication : Started SportyShoesBackendApplication in 6.287 s
2023-01-01 18:57:50.671  INFO 32016 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-01-01 18:57:50.671  INFO 32016 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2023-01-01 18:57:50.673  INFO 32016 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
username = ozbrown
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, use
Found User: User [username=ozbrown, id=7, roleType=ADMIN, firstName=Oswald, lastName=Brown, email=ozbrown@home.com, password=$2a$12$P21wlkhXzMRD3t3U
```



# Note about Custom Login Page

```
>LoginController.java x
1 package com.sportyshoes.restapi.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5
6 @Controller
7 public class LoginController {
8     /* This endpoint ensures whenever there is an invocation to the login URL (e.g.,
9      http://localhost:8080/login), the login.html page is presented to the user. */
10    @GetMapping("/login")
11    public String login() {
12        return ("login");
13    }
14
15 } //end class
16
```

**@Controller** annotation allowed for the login.html page to be presented with the /login endpoint



Sporty  
Shoes

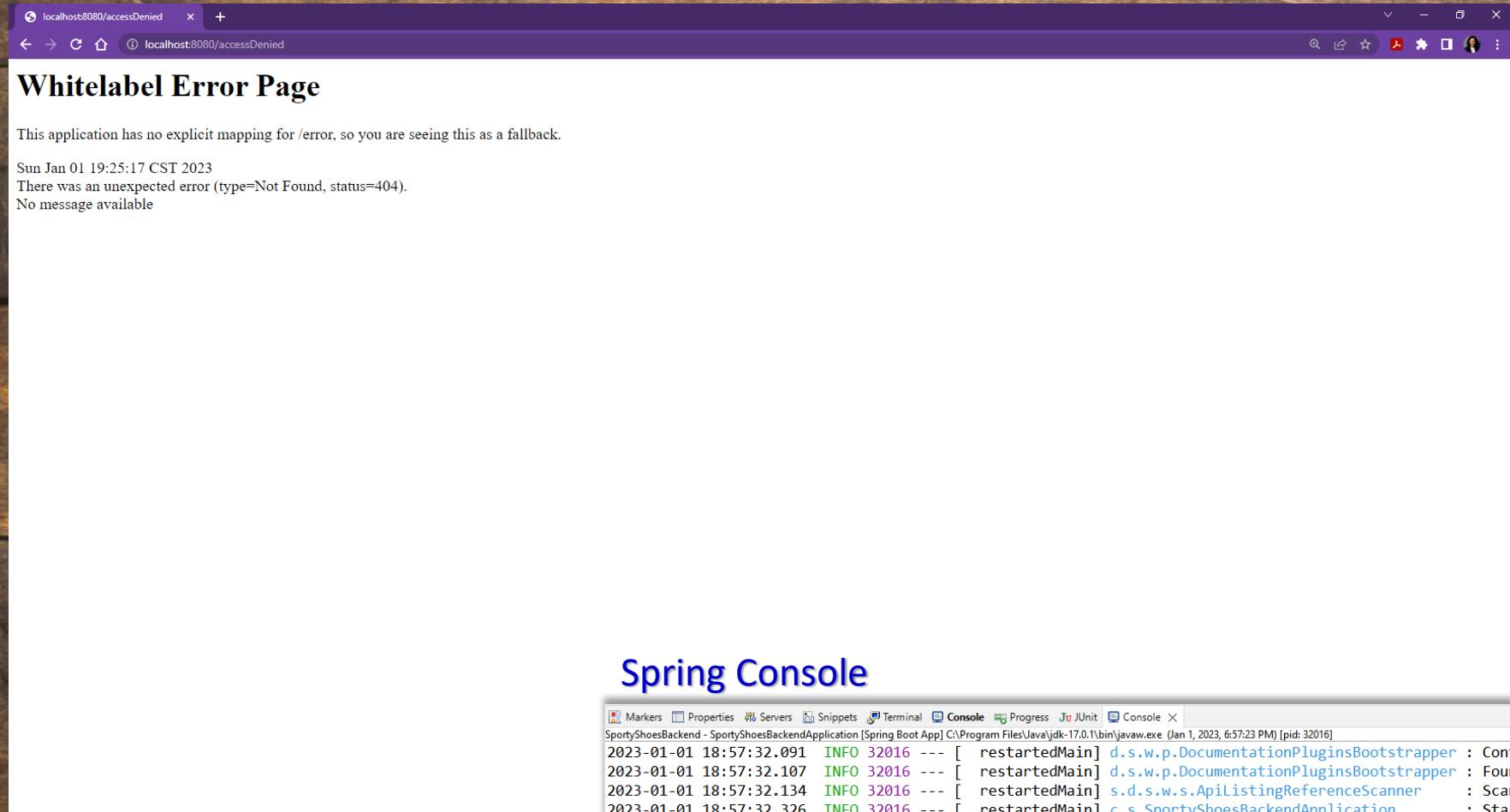
```
(LoginController.java x
1 package com.sportyshoes.restapi.controller;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6
7 @RestController
8 public class LoginController {
9     /* This endpoint ensures whenever there is an invocation to the login URL (e.g.,
10      http://localhost:8080/login), the login.html page is presented to the user. */
11    @GetMapping("/login")
12    public String login() {
13        return ("login");
14    }
15
16 } //end class
17
```

**@RestController** annotation allowed for the method String to be printed on the screen and bypassed the login.html page with the /login endpoint



# User Page

Logged in as Admin Role



## Spring Console

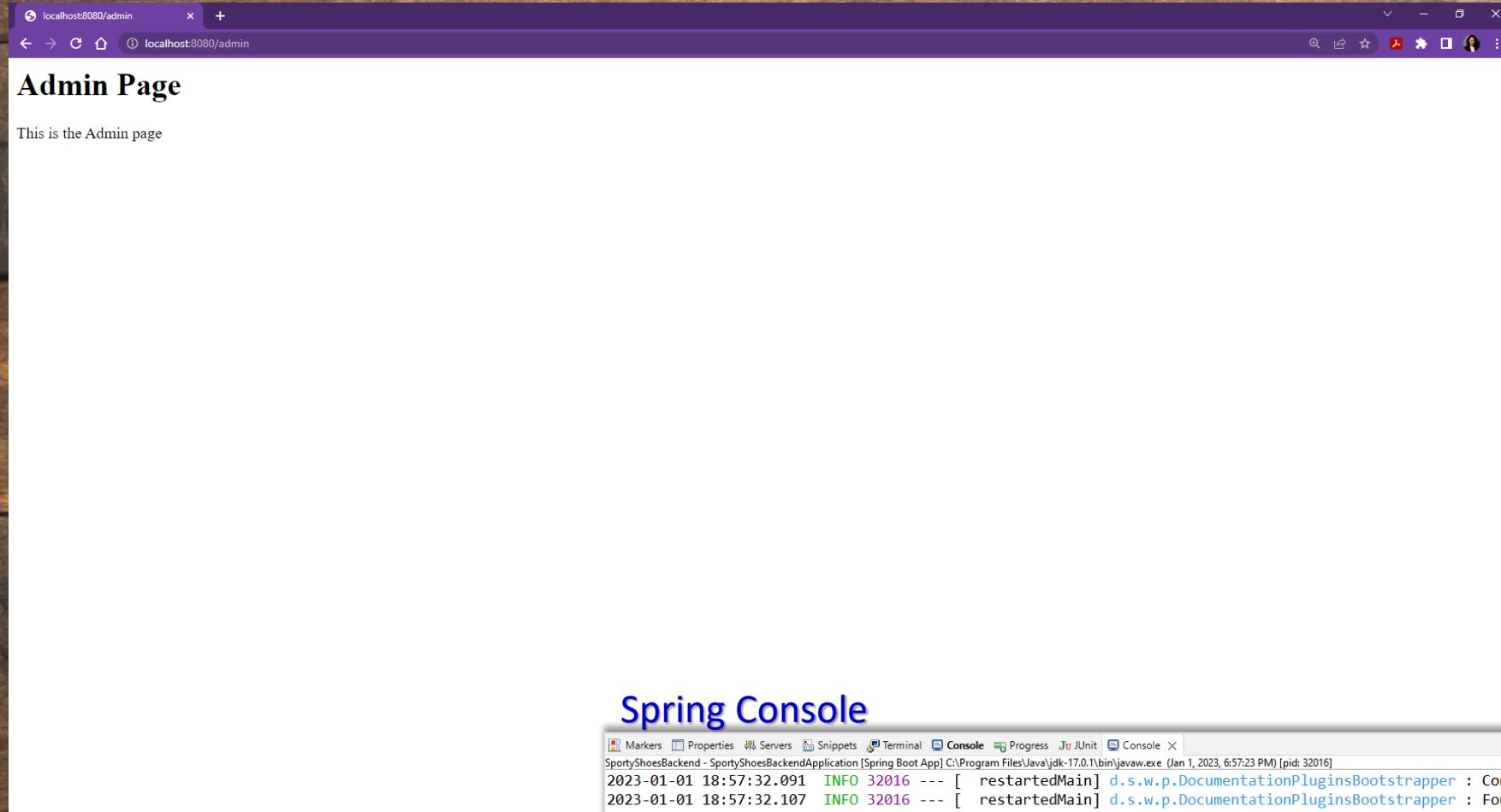
The Spring Console window displays the following log output:

```
Markers Properties Servers Snippets Terminal Console Progress JUnit Console
SportyShoesBackend - SportyShoeBackendApplication [Spring Boot App] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Jan 1, 2023, 6:57:23 PM) [pid: 32016]
2023-01-01 18:57:32.091  INFO 32016 --- [ restartedMain] d.s.w.p.DocumentationPluginsBootstrapper : Context refreshed
2023-01-01 18:57:32.107  INFO 32016 --- [ restartedMain] d.s.w.p.DocumentationPluginsBootstrapper : Found 1 custom documentation plugin(s)
2023-01-01 18:57:32.134  INFO 32016 --- [ restartedMain] s.d.s.w.s.ApiListingReferenceScanner : Scanning for api listing references
2023-01-01 18:57:32.326  INFO 32016 --- [ restartedMain] c.s.SportyShoesBackendApplication : Started SportyShoesBackendApplication in 6.287 s
2023-01-01 18:57:50.671  INFO 32016 --- [nio-8080-exec-1] o.a.c.c.c.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-01-01 18:57:50.671  INFO 32016 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2023-01-01 18:57:50.673  INFO 32016 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
inside loadUserByUsername()...
username = ozbrown
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, use
Found User: User [username=ozbrown, id=7, roleType=ADMIN, firstName=Oswald, lastName=Brown, email=ozbrown@home.com, password=$2a$12$P21wlkhXzMRD3t3U
You are not authorized to perform this action
You are not authorized to perform this action
```



# Admin Page

Logged in as Admin Role



## Spring Console

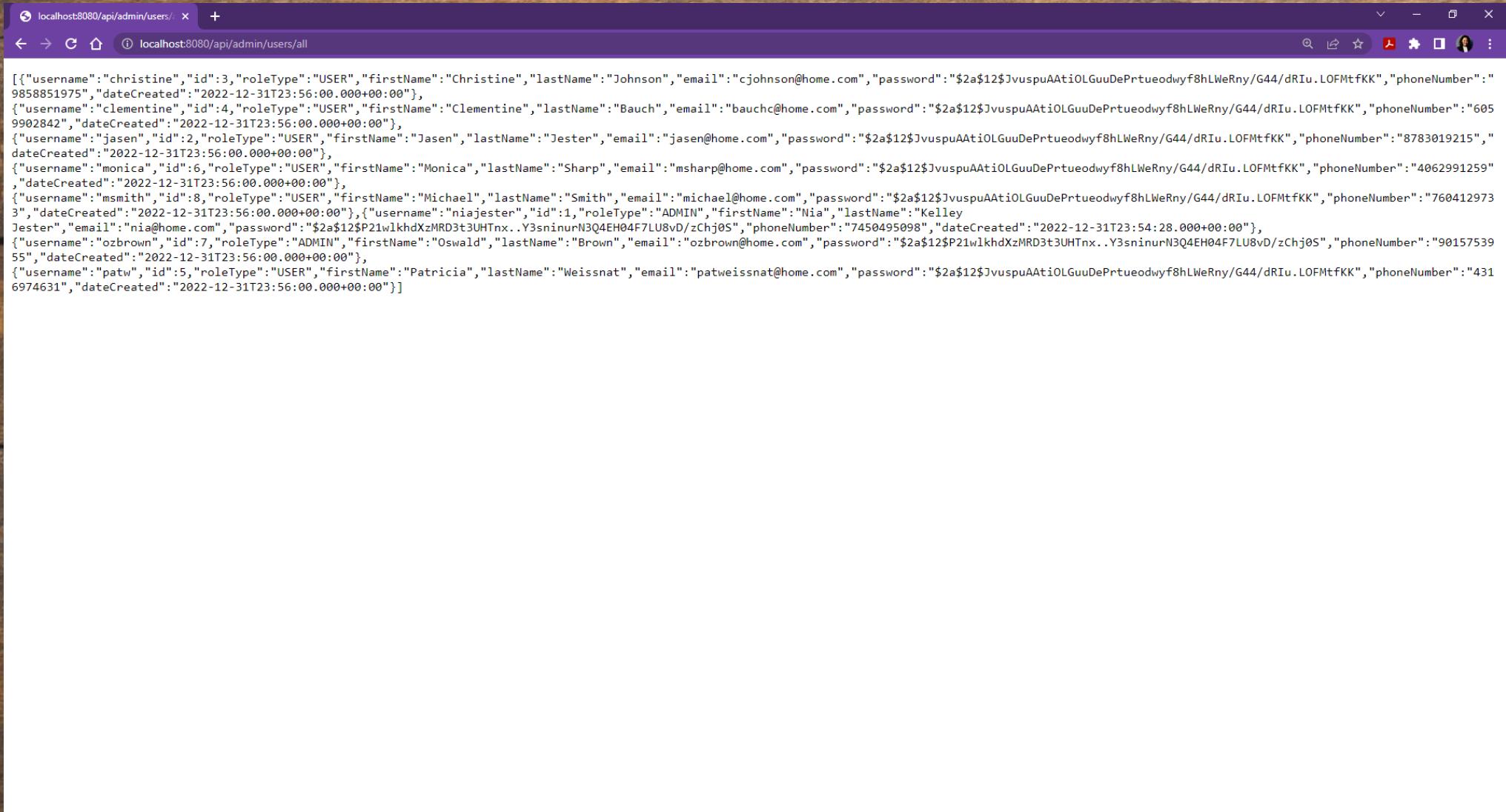
The Spring Console window displays the application logs for 'SportyShoesBackend - SportyShoesBackendApplication [Spring Boot App] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Jan 1, 2023, 6:57:23 PM) [pid: 32016]'. The logs show the application starting up, initializing documentation plugins, scanning for API listing references, and starting the DispatcherServlet. It also shows a query for a user named 'ozbrown' and a found user object.

```
2023-01-01 18:57:32.091  INFO 32016 --- [ restartedMain] d.s.w.p.DocumentationPluginsBootstrapper : Context refreshed
2023-01-01 18:57:32.107 INFO 32016 --- [ restartedMain] d.s.w.p.DocumentationPluginsBootstrapper : Found 1 custom documentation plugin(s)
2023-01-01 18:57:32.134 INFO 32016 --- [ restartedMain] s.d.s.w.s.ApiListingReferenceScanner : Scanning for api listing references
2023-01-01 18:57:32.326 INFO 32016 --- [ restartedMain] c.s.SportyShoesBackendApplication : Started SportyShoesBackendApplication in 6.287 s
2023-01-01 18:57:50.671 INFO 32016 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-01-01 18:57:50.671 INFO 32016 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2023-01-01 18:57:50.673 INFO 32016 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
inside loadUserByUsername()
username = ozbrown
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, use
Found User: User [username=ozbrown, id=7, roleType=ADMIN, firstName=Oswald, lastName=Brown, email=ozbrown@home.com, password=$2a$12$P21wlkhdxzMRD3t3U
```



# Users List: api/admin/users/all

Logged in as Admin Role



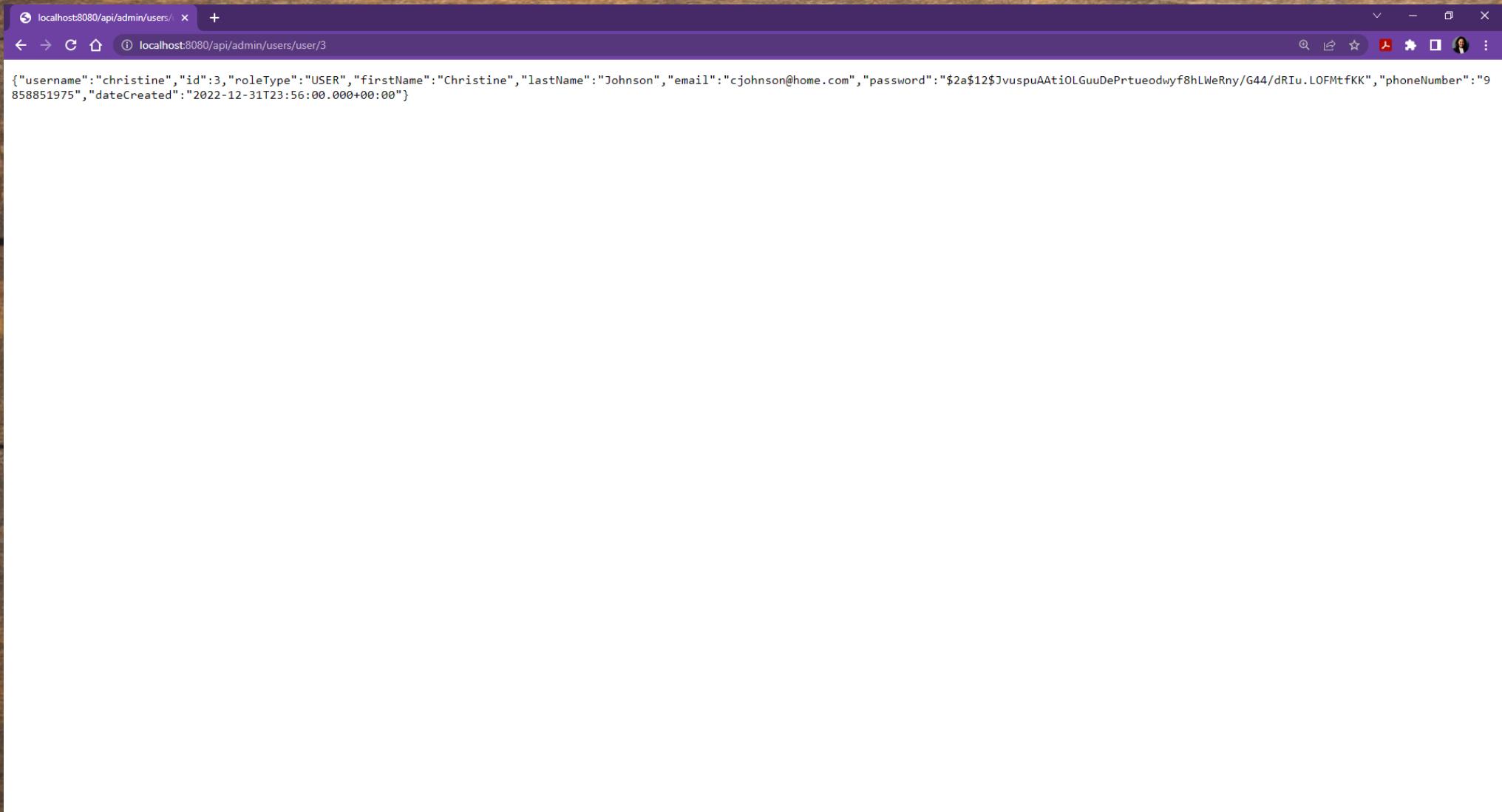
A screenshot of a Microsoft Edge browser window displaying a JSON response from the endpoint `localhost:8080/api/admin/users/all`. The JSON array contains 10 user objects, each with the following fields: username, id, roleType, firstName, lastName, email, password, and phoneNumber. The users listed are: christine, clementine, jasen, monica, msmith, niajester, ozbrown, patw, and patweissnat.

```
[{"username": "christine", "id": 3, "roleType": "USER", "firstName": "Christine", "lastName": "Johnson", "email": "cjohnson@home.com", "password": "$2a$12$JvuspuAAtiOLGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtFKK", "phoneNumber": "9858851975", "dateCreated": "2022-12-31T23:56:00.000+00:00"}, {"username": "clementine", "id": 4, "roleType": "USER", "firstName": "Clementine", "lastName": "Bauch", "email": "bauchc@home.com", "password": "$2a$12$JvuspuAAtiOLGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtFKK", "phoneNumber": "6059902842", "dateCreated": "2022-12-31T23:56:00.000+00:00"}, {"username": "jasen", "id": 2, "roleType": "USER", "firstName": "Jasen", "lastName": "Jester", "email": "jasen@home.com", "password": "$2a$12$JvuspuAAtiOLGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtFKK", "phoneNumber": "8783019215", "dateCreated": "2022-12-31T23:56:00.000+00:00"}, {"username": "monica", "id": 6, "roleType": "USER", "firstName": "Monica", "lastName": "Sharp", "email": "msharp@home.com", "password": "$2a$12$JvuspuAAtiOLGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtFKK", "phoneNumber": "4062991259", "dateCreated": "2022-12-31T23:56:00.000+00:00"}, {"username": "msmith", "id": 8, "roleType": "USER", "firstName": "Michael", "lastName": "Smith", "email": "michael@home.com", "password": "$2a$12$JvuspuAAtiOLGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtFKK", "phoneNumber": "7604129733", "dateCreated": "2022-12-31T23:56:00.000+00:00"}, {"username": "niajester", "id": 1, "roleType": "ADMIN", "firstName": "Nia", "lastName": "KelleyJester", "email": "nia@home.com", "password": "$2a$12$P21wlkhXzMRD3t3UHTnx..Y3sninurN3Q4EH04F7LU8vD/zChj0S", "phoneNumber": "7450495098", "dateCreated": "2022-12-31T23:54:28.000+00:00"}, {"username": "ozbrown", "id": 7, "roleType": "ADMIN", "firstName": "Oswald", "lastName": "Brown", "email": "ozbrown@home.com", "password": "$2a$12$P21wlkhXzMRD3t3UHTnx..Y3sninurN3Q4EH04F7LU8vD/zChj0S", "phoneNumber": "9015753955", "dateCreated": "2022-12-31T23:56:00.000+00:00"}, {"username": "patw", "id": 5, "roleType": "USER", "firstName": "Patricia", "lastName": "Weissnat", "email": "patweissnat@home.com", "password": "$2a$12$JvuspuAAtiOLGuuDePrtueodwyf8hLWeRny/G44/dRIu.LOFMtFKK", "phoneNumber": "4316974631", "dateCreated": "2022-12-31T23:56:00.000+00:00"}]
```



# Single User: api/admin/users/user/3

Logged in as Admin Role



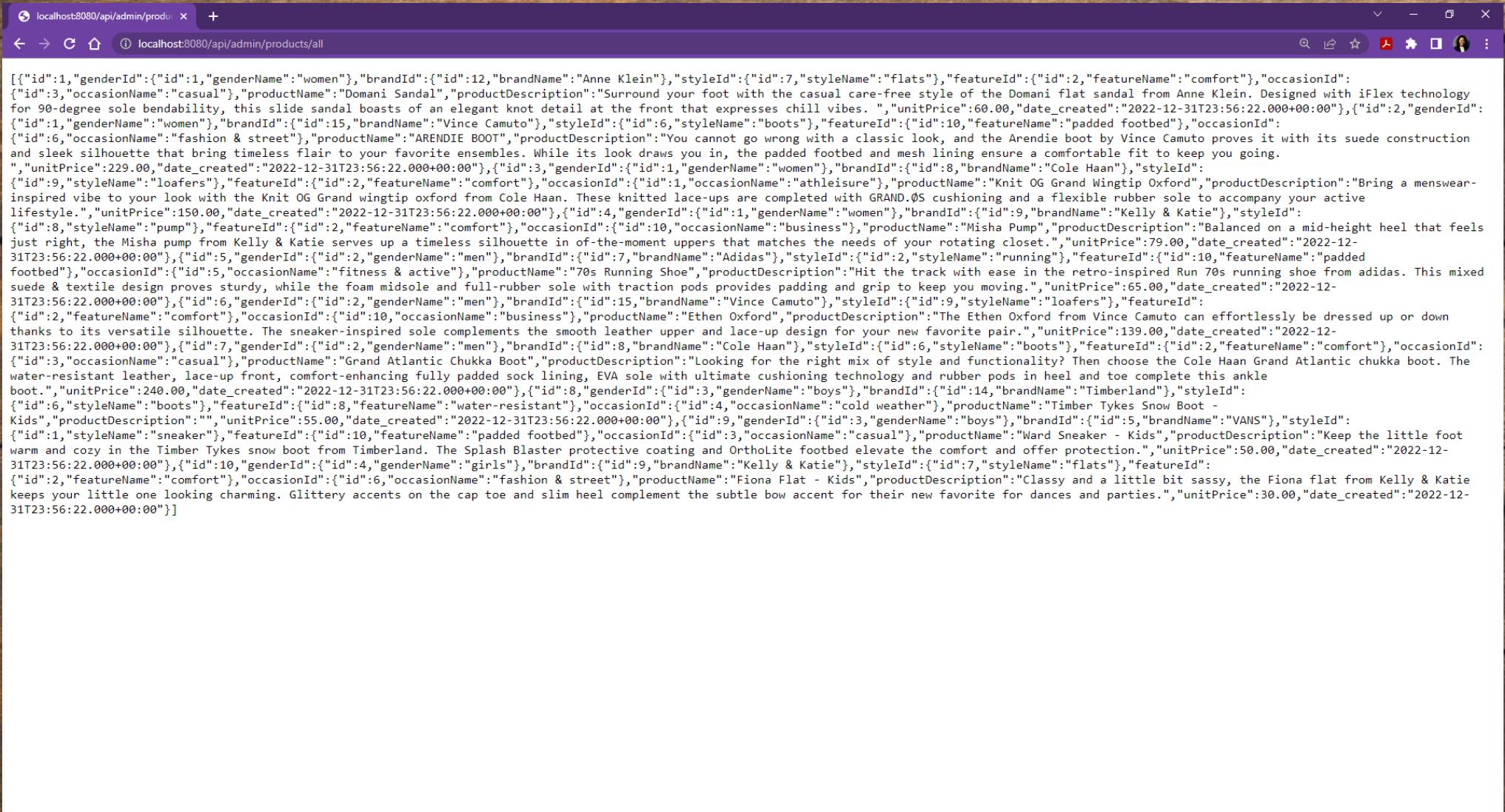
A screenshot of a Microsoft Edge browser window. The address bar shows "localhost:8080/api/admin/users/user/3". The main content area displays the following JSON data:

```
{"username": "christine", "id": 3, "roleType": "USER", "firstName": "Christine", "lastName": "Johnson", "email": "cjohnson@home.com", "password": "$2a$12$JvuspAAtiOLGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtFKK", "phoneNumber": "9858851975", "dateCreated": "2022-12-31T23:56:00.000+00:00"}
```



# Products List: api/admin/products/all

Logged in as Admin Role

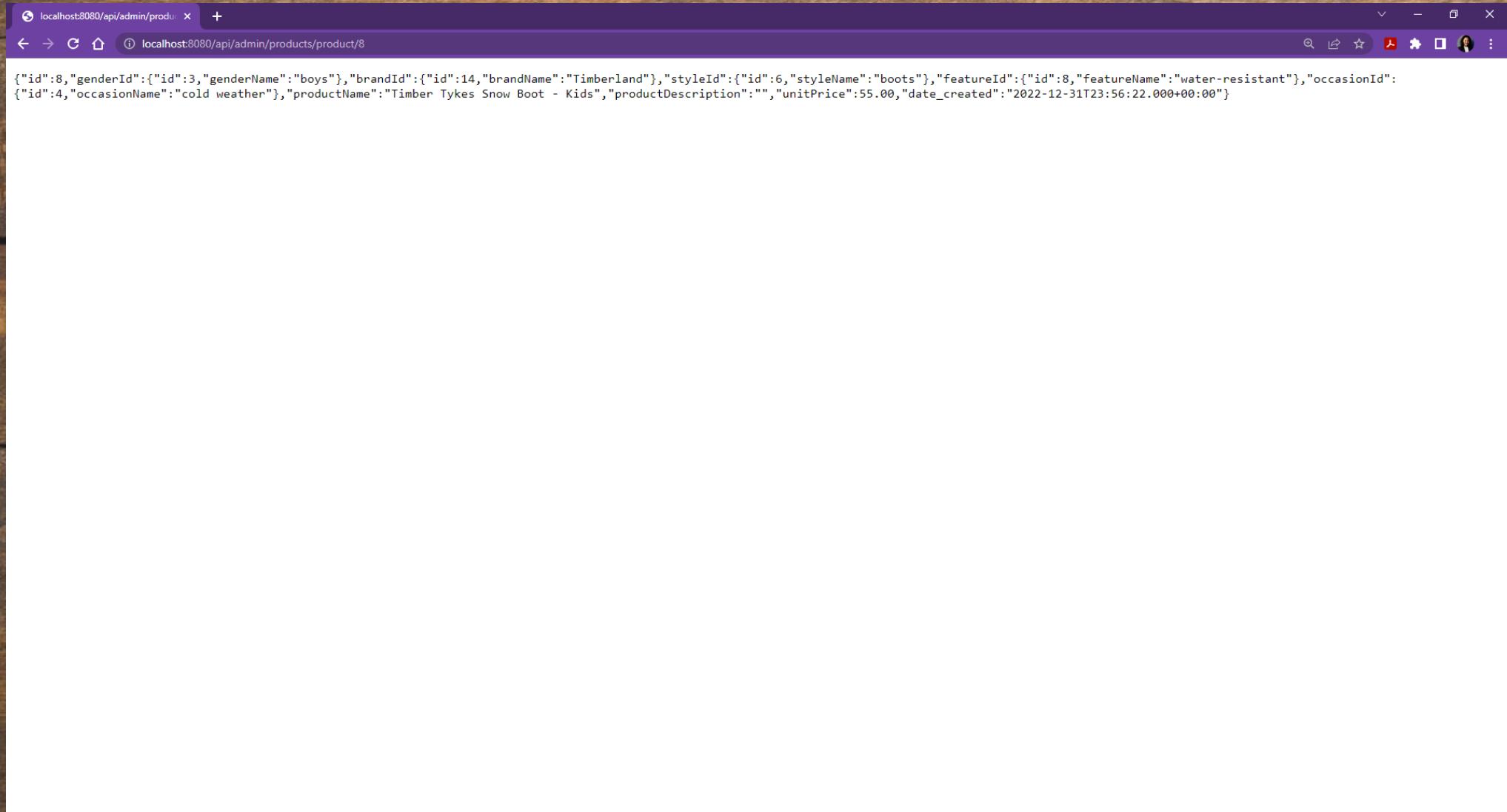


The screenshot shows a Microsoft Edge browser window with the URL `localhost:8080/api/admin/products/all`. The page displays a large block of JSON data representing a list of products. The JSON structure includes fields such as `id`, `genderId`, `brandId`, `styleId`, `featureId`, `occasionId`, `productName`, `productDescription`, `unitPrice`, and `date_created`. The data describes various shoe models like the Domani Sandal, Arendie Boot, Knit OG Grand Wingtip Oxford, Misha Pump, Ethen Oxford, Grand Atlantic Chukka Boot, and Fiona Flat - Kids, each with its unique features and pricing.

```
[{"id":1,"genderId":{"id":1,"genderName":"women"},"brandId":{"id":12,"brandName":"Anne Klein"},"styleId":{"id":7,"styleName":"flats"},"featureId":{"id":2,"featureName":"comfort"},"occasionId": {"id":3,"occasionName":"casual"}, "productName":"Domani Sandal","productDescription":"Surround your foot with the casual care-free style of the Domani flat sandal from Anne Klein. Designed with iFlex technology for 90-degree sole bendability, this slide sandal boasts of an elegant knot detail at the front that expresses chill vibes.", "unitPrice":60.00,"date_created":"2022-12-31T23:56:22.000+00:00"}, {"id":2,"genderId": {"id":1,"genderName":"women"}, "brandId": {"id":15,"brandName": "Vince Camuto"}, "styleId": {"id":6,"styleName": "boots"}, "featureId": {"id":10,"featureName": "padded footbed"}, "occasionId": {"id":6,"occasionName": "fashion & street"}, "productName": "ARENDIE BOOT", "productDescription": "You cannot go wrong with a classic look, and the Arendie boot by Vince Camuto proves it with its suede construction and sleek silhouette that bring timeless flair to your favorite ensembles. While its look draws you in, the padded footbed and mesh lining ensure a comfortable fit to keep you going.", "unitPrice":229.00,"date_created":"2022-12-31T23:56:22.000+00:00"}, {"id":3,"genderId": {"id":1,"genderName": "women"}, "brandId": {"id":8,"brandName": "Cole Haan"}, "styleId": {"id":9,"styleName": "loafers"}, "featureId": {"id":2,"featureName": "comfort"}, "occasionId": {"id":1,"occasionName": "athleisure"}, "productName": "Knit OG Grand Wingtip Oxford", "productDescription": "Bring a menswear-inspired vibe to your look with the Knit OG Grand wingtip oxford from Cole Haan. These knitted lace-ups are completed with GRAND.ØS cushioning and a flexible rubber sole to accompany your active lifestyle.", "unitPrice":150.00,"date_created":"2022-12-31T23:56:22.000+00:00"}, {"id":4,"genderId": {"id":1,"genderName": "women"}, "brandId": {"id":9,"brandName": "Kelly & Katie"}, "styleId": {"id":8,"styleName": "pump"}, "featureId": {"id":2,"featureName": "comfort"}, "occasionId": {"id":10,"occasionName": "business"}, "productName": "Misha Pump", "productDescription": "Balanced on a mid-height heel that feels just right, the Misha pump from Kelly & Katie serves up a timeless silhouette in-of-the-moment uppers that matches the needs of your rotating closet.", "unitPrice":79.00,"date_created":"2022-12-31T23:56:22.000+00:00"}, {"id":5,"genderId": {"id":2,"genderName": "men"}, "brandId": {"id":7,"brandName": "Adidas"}, "styleId": {"id":2,"styleName": "running"}, "featureId": {"id":10,"featureName": "padded footbed"}, "occasionId": {"id":5,"occasionName": "fitness & active"}, "productName": "70s Running Shoe", "productDescription": "Hit the track with ease in the retro-inspired Run 70s running shoe from adidas. This mixed suede & textile design proves sturdy, while the foam midsole and full-rubber sole with traction pods provides padding and grip to keep you moving.", "unitPrice":65.00,"date_created":"2022-12-31T23:56:22.000+00:00"}, {"id":6,"genderId": {"id":2,"genderName": "men"}, "brandId": {"id":15,"brandName": "Vince Camuto"}, "styleId": {"id":9,"styleName": "loafers"}, "featureId": {"id":2,"featureName": "comfort"}, "occasionId": {"id":10,"occasionName": "business"}, "productName": "Ethen Oxford", "productDescription": "The Ethen Oxford from Vince Camuto can effortlessly be dressed up or down thanks to its versatile silhouette. The sneaker-inspired sole complements the smooth leather upper and lace-up design for your new favorite pair.", "unitPrice":139.00,"date_created":"2022-12-31T23:56:22.000+00:00"}, {"id":7,"genderId": {"id":2,"genderName": "men"}, "brandId": {"id":8,"brandName": "Cole Haan"}, "styleId": {"id":6,"styleName": "boots"}, "featureId": {"id":2,"featureName": "comfort"}, "occasionId": {"id":3,"occasionName": "casual"}, "productName": "Grand Atlantic Chukka Boot", "productDescription": "Looking for the right mix of style and functionality? Then choose the Cole Haan Grand Atlantic chukka boot. The water-resistant leather, lace-up front, comfort-enhancing fully padded sock lining, EVA sole with ultimate cushioning technology and rubber pods in heel and toe complete this ankle boot.", "unitPrice":240.00,"date_created":"2022-12-31T23:56:22.000+00:00"}, {"id":8,"genderId": {"id":3,"genderName": "boys"}, "brandId": {"id":14,"brandName": "Timberland"}, "styleId": {"id":6,"styleName": "boots"}, "featureId": {"id":8,"featureName": "water-resistant"}, "occasionId": {"id":4,"occasionName": "cold weather"}, "productName": "Timber Tykes Snow Boot - Kids", "productDescription": "", "unitPrice":55.00,"date_created":"2022-12-31T23:56:22.000+00:00"}, {"id":9,"genderId": {"id":3,"genderName": "boys"}, "brandId": {"id":5,"brandName": "VANS"}, "styleId": {"id":1,"styleName": "sneaker"}, "featureId": {"id":10,"featureName": "padded footbed"}, "occasionId": {"id":3,"occasionName": "casual"}, "productName": "Ward Sneaker - Kids", "productDescription": "Keep the little foot warm and cozy in the Timber Tykes snow boot from Timberland. The Splash Blaster protective coating and OrthoLite footbed elevate the comfort and offer protection.", "unitPrice":50.00,"date_created":"2022-12-31T23:56:22.000+00:00"}, {"id":10,"genderId": {"id":4,"genderName": "girls"}, "brandId": {"id":9,"brandName": "Kelly & Katie"}, "styleId": {"id":7,"styleName": "flats"}, "featureId": {"id":2,"featureName": "comfort"}, "occasionId": {"id":6,"occasionName": "fashion & street"}, "productName": "Fiona Flat - Kids", "productDescription": "Classy and a little bit sassy, the Fiona flat from Kelly & Katie keeps your little one looking charming. Glittery accents on the cap toe and slim heel complement the subtle bow accent for their new favorite for dances and parties.", "unitPrice":30.00,"date_created":"2022-12-31T23:56:22.000+00:00"}]
```

# Single Product: api/admin/products/product/8

Logged in as Admin Role



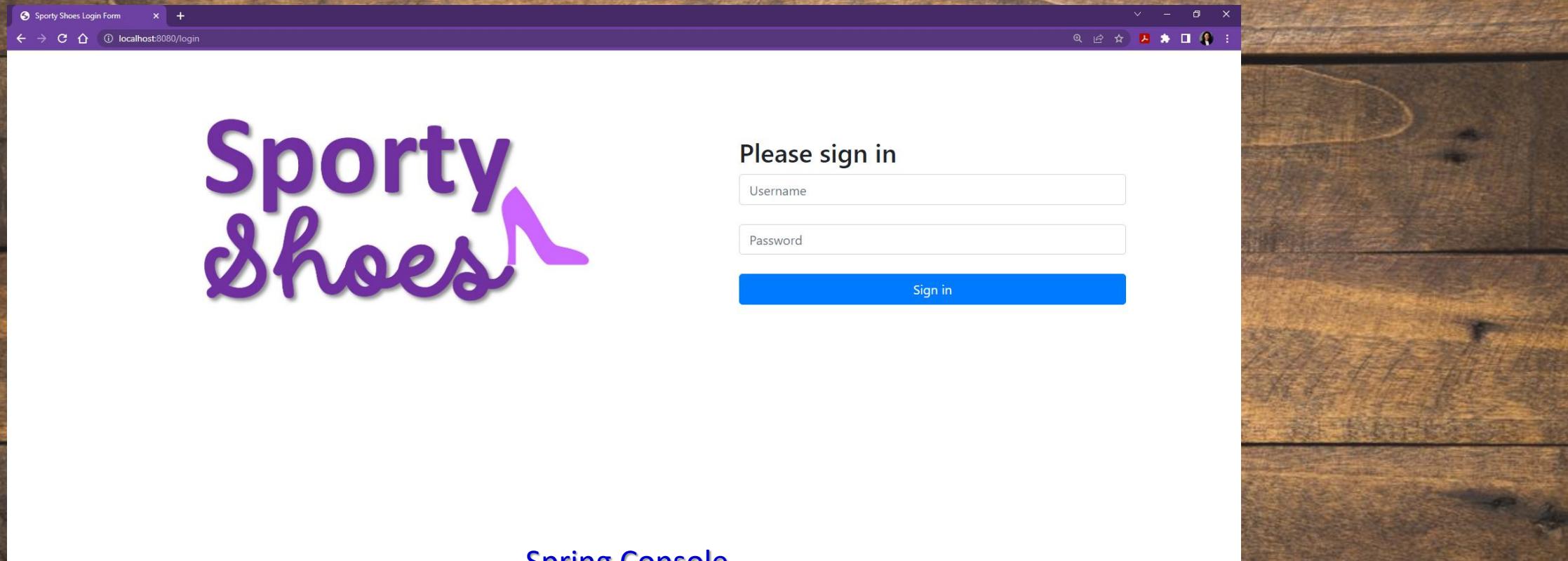
A screenshot of a Microsoft Edge browser window. The address bar shows the URL `localhost:8080/api/admin/products/product/8`. The main content area displays a JSON object representing a product:

```
{"id":8,"genderId":{"id":3,"genderName":"boys"},"brandId":{"id":14,"brandName":"Timberland"},"styleId":{"id":6,"styleName":"boots"},"featureId":{"id":8,"featureName":"water-resistant"},"occasionId":{"id":4,"occasionName":"cold weather"},"productName":"Timber Tykes Snow Boot - Kids","productDescription":"","unitPrice":55.00,"date_created":"2022-12-31T23:56:22.000+00:00"}
```



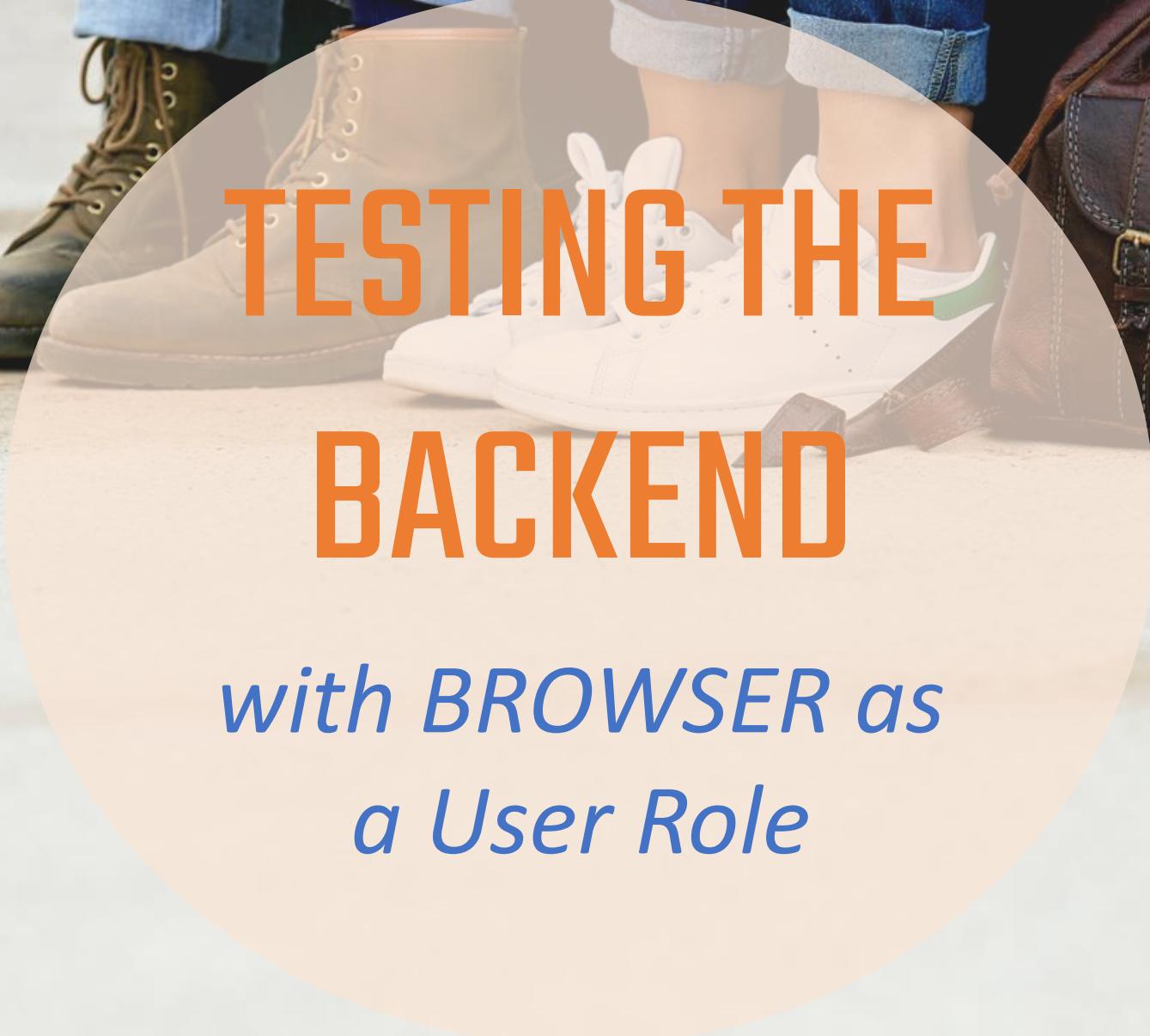
# Logout Page (redirects to Login Page)

Logged out as Admin Role



## Spring Console

A screenshot of the Spring Boot application's console output in a terminal window. The window title is "SportyShoesBackend - SportyShoesBackendApplication [Spring Boot App] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Jan 1, 2023, 6:57:23 PM) [pid: 32016]". The console output shows several Hibernate SQL queries being executed. At the bottom of the list, the message "This user logged out: ozbrown" is highlighted in yellow.



# TESTING THE BACKEND

*with BROWSER as  
a User Role*



# Custom Login Page

Logged in as User Role

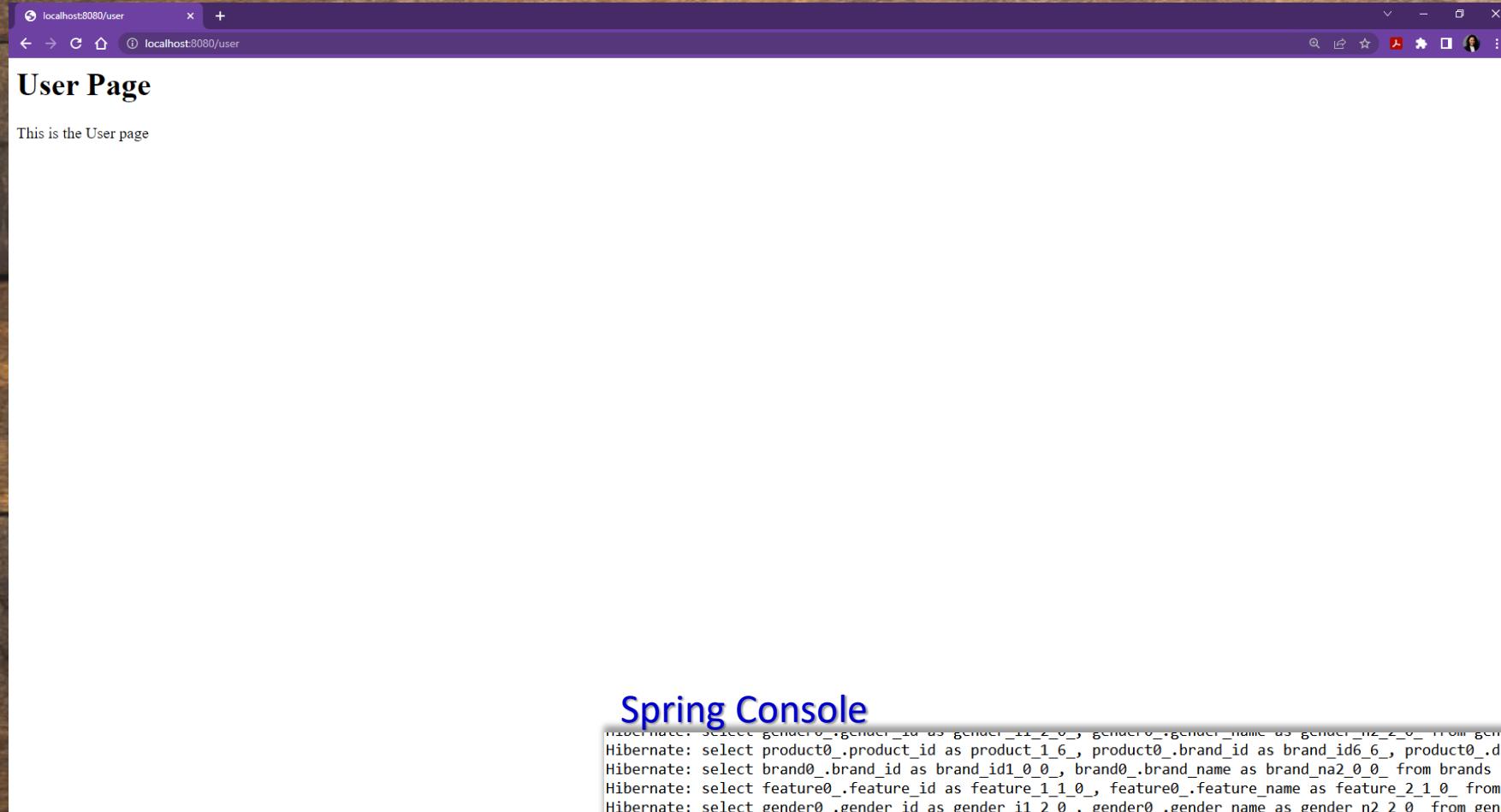
The screenshot shows a custom login interface for a website named "Sporty Shoes". The logo is displayed prominently on the left. The main form has a purple header bar with the title "Sporty Shoes Login Form" and the URL "localhost:8080/login". The form itself has a light gray background with a dark blue header that reads "Please sign in". It contains two input fields: one for "msmith" and another for a password represented by five dots. A blue "Sign in" button is at the bottom. Below the login form, there is a "Spring Console" section showing several lines of Hibernate SQL queries. One query highlights the variable "username = msmith". The entire interface is set against a wooden background.

Spring Console

```
Hibernate: select gender0_.gender_id as gender_i1_2_0_, gender0_.gender_name as gender_n2_2_0_ from genders gender0_ where gender0_.gender_id=?  
Hibernate: select product0_.product_id as product_1_6_, product0_.brand_id as brand_id6_6_, product0_.date_created as date_cre2_6_, product0_.feature  
Hibernate: select brand0_.brand_id as brand_id1_0_0_, brand0_.brand_name as brand_na2_0_0_ from brands brand0_ where brand0_.brand_id=?  
Hibernate: select feature0_.feature_id as feature_1_1_0_, feature0_.feature_name as feature_2_1_0_ from features feature0_ where feature0_.feature_id=?  
Hibernate: select gender0_.gender_id as gender_i1_2_0_, gender0_.gender_name as gender_n2_2_0_ from genders gender0_ where gender0_.gender_id=?  
Hibernate: select occasion0_.occasion_id as occasion1_3_0_, occasion0_.occasion_name as occasion2_3_0_ from occasions occasion0_ where occasion0_.occ  
Hibernate: select style0_.style_id as style_id1_8_0_, style0_.style_name as style_na2_8_0_ from styles style0_ where style0_.style_id=?  
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, use  
This user logged out: ozbrown  
inside loadUserByUserName()...  
username = msmith  
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, use  
Found User: User [username=msmith, id=8, roleType=USER, firstName=Michael, lastName=Smith, email=michael@home.com, password=$2a$12$JvuspuAAti0LGuuDeP
```

# User Page

Logged in as User Role



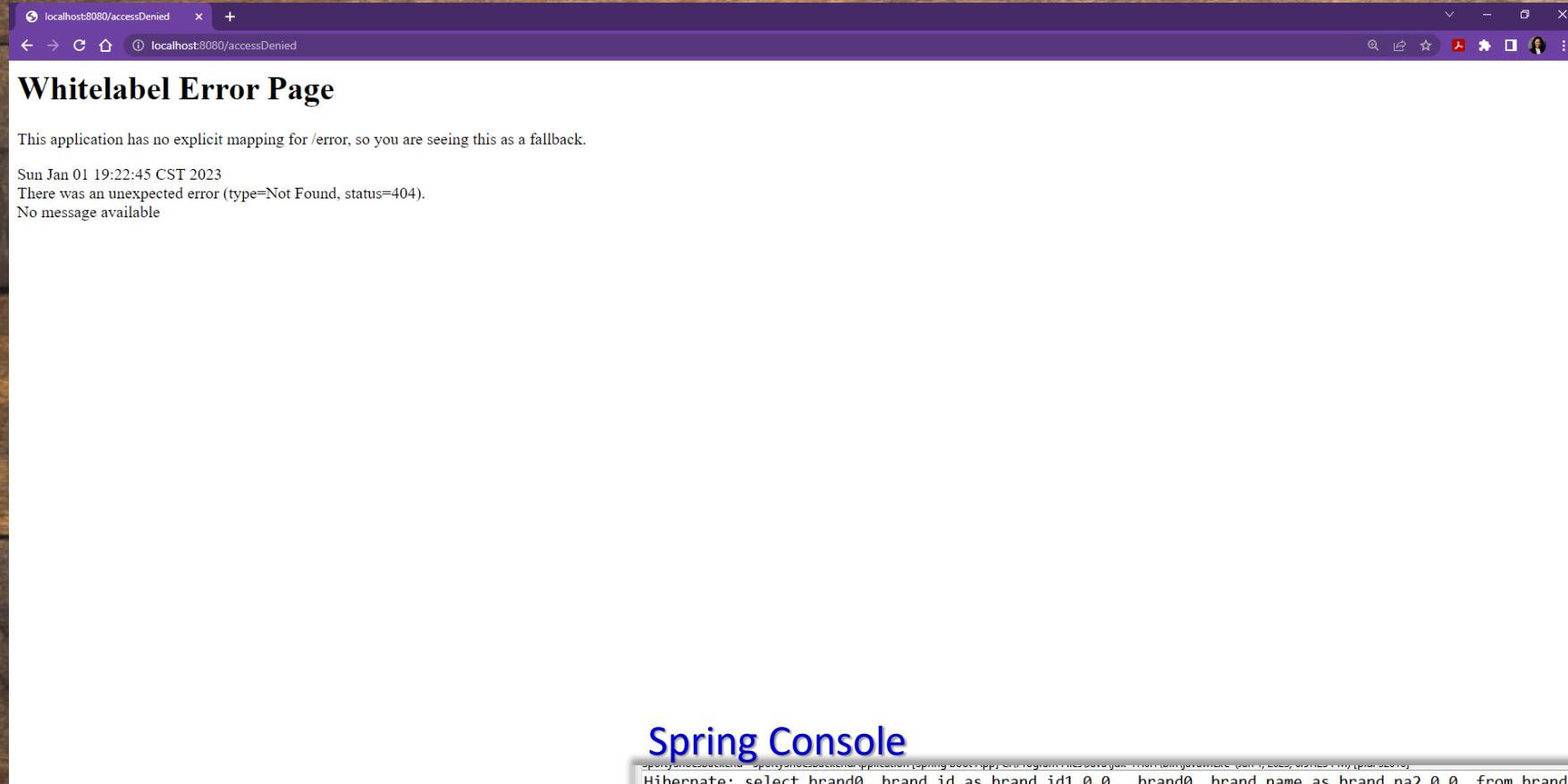
## Spring Console

```
Hibernate: select gender0_.gender_id as gender_11_2_0_, gender0_.gender_name as gender_n2_2_0_ from genders gender0_ where gender0_.gender_id=?  
Hibernate: select product0_.product_id as product_1_6_, product0_.brand_id as brand_id6_6_, product0_.date_created as date_cre2_6_, product0_.feature_id as feature_id1_6_0_0_ from products product0_ where product0_.brand_id=?  
Hibernate: select brand0_.brand_id as brand_id1_0_0_, brand0_.brand_name as brand_na2_0_0_ from brands brand0_ where brand0_.brand_id=?  
Hibernate: select feature0_.feature_id as feature_1_1_0_, feature0_.feature_name as feature_2_1_0_ from features feature0_ where feature0_.feature_id=?  
Hibernate: select gender0_.gender_id as gender_i1_2_0_, gender0_.gender_name as gender_n2_2_0_ from genders gender0_ where gender0_.gender_id=?  
Hibernate: select occasion0_.occasion_id as occasion_1_3_0_, occasion0_.occasion_name as occasion2_3_0_ from occasions occasion0_ where occasion0_.occurred_at=?  
Hibernate: select style0_.style_id as style_id1_8_0_, style0_.style_name as style_na2_8_0_ from styles style0_ where style0_.style_id=?  
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, user0_.password as password5_9_ from users user0_ where user0_.username=?  
This user logged out: ozbrown  
inside loadUserByUserName()...  
username = msmith  
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, user0_.password as password5_9_ from users user0_ where user0_.username=?  
Found User: User [username=msmith, id=8, roleType=USER, firstName=Michael, lastName=Smith, email=michael@home.com, password=$2a$12$JvuspuAAtiOLGuuDeP]
```



# Admin Page

Logged in as User Role



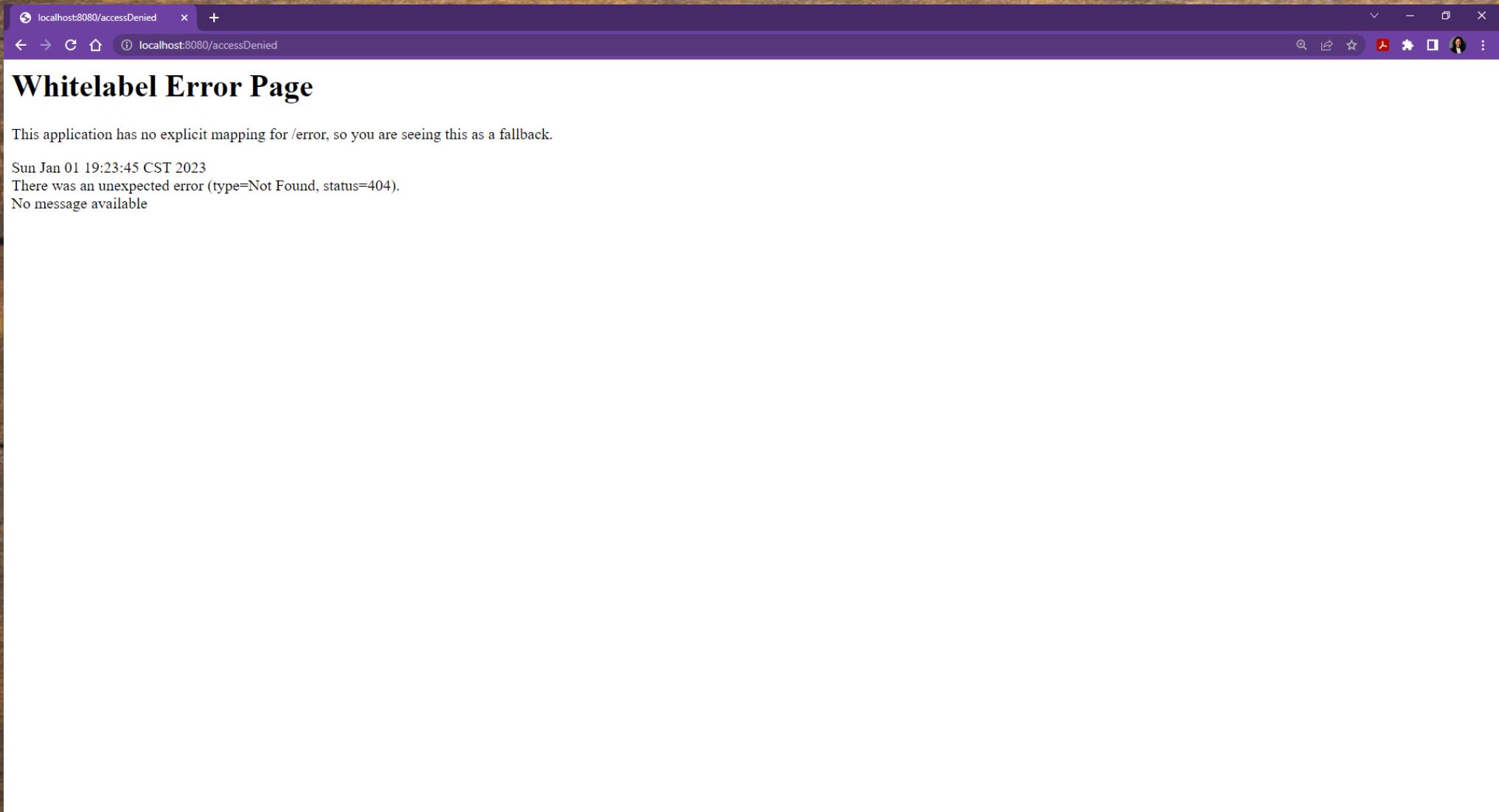
## Spring Console

```
Hibernate: select brand0_.brand_id as brand_id1_0_0_, brand0_.brand_name as brand_na2_0_0_ from brands brand0_ where brand0_.brand_id=?  
Hibernate: select feature0_.feature_id as feature_1_1_0_, feature0_.feature_name as feature_2_1_0_ from features feature0_ where feature0_.feature_id=?  
Hibernate: select gender0_.gender_id as gender_i1_2_0_, gender0_.gender_name as gender_n2_2_0_ from genders gender0_ where gender0_.gender_id=?  
Hibernate: select occasion0_.occasion_id as occasion1_3_0_, occasion0_.occasion_name as occasion2_3_0_ from occasions occasion0_ where occasion0_.occasion_id=?  
Hibernate: select style0_.style_id as style_id1_8_0_, style0_.style_name as style_na2_8_0_ from styles style0_ where style0_.style_id=?  
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, user0_.last_name as last_name5_9_, user0_.first_name as first_name6_9_, user0_.password as password7_9_ from users user0_ where user0_.username=?  
This user logged out: ozbrown  
inside loadUserByUsername()...  
username = msmith  
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, user0_.last_name as last_name5_9_, user0_.first_name as first_name6_9_, user0_.password as password7_9_ from users user0_ where user0_.username=?  
Found User: User [username=msmith, id=8, roleType=USER, firstName=Michael, lastName=Smith, email=michael@home.com, password=$2a$12$JvuspuAAtiOLGuuLcXfZuPQ]  
You are not authorized to perform this action  
Caused by: org.springframework.security.access.AccessDeniedException: Access is denied
```



# Users List: api/admin/users/all

Logged in as User Role



# Logout Page (redirects to Login Page)

Logged out as User Role

The screenshot shows a desktop environment with two windows open. The top window is a web browser titled "Sporty Shoes Login Form" with the URL "localhost:8080/login". It displays the "Sporty Shoes" logo on the left and a "Please sign in" form with fields for "Username" and "Password", and a blue "Sign in" button. The bottom window is a "Spring Console" (IntelliJ IDEA) showing Java code and a terminal window. The terminal window displays Hibernate SQL queries and logs related to user authentication. A yellow box highlights the line "This user logged out: msmith".

Sporty Shoes Login Form

localhost:8080/login

Please sign in

Username

Password

Sign in

Spring Console

```
Markers Properties Servers Snippets Terminal Console Progress JUnit Console
SportyShoesBackend - SportyShoesBackendApplication [Spring Boot App] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (Jan 1, 2023, 6:57:23 PM) [pid: 32016]
Hibernate: select occasion0_.occasion_id as occasion1_3_0_, occasion0_.occasion_name as occasion2_3_0_ from occasions occasion0_ where occasion0_.occ
Hibernate: select style0_.style_id as style_id1_8_0_, style0_.style_name as style_na2_8_0_ from styles style0_ where style0_.style_id=?
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, user
This user logged out: ozbrown
inside loadUserByUserName()...
username = msmith
Hibernate: select user0_.username as username1_9_, user0_.date_created as date_cre2_9_, user0_.email as email3_9_, user0_.enabled as enabled4_9_, user
Found User: User [username=msmith, id=8, roleType=USER, firstName=Michael, lastName=Smith, email=michael@home.com, password=$2a$12$JvuspuAAti0LGuuDef
You are not authorized to perform this action
This user logged out: msmith
```





# TESTING THE BACKEND

*with POSTMAN*



# Testing Endpoints:

@PutMapping("/user/update/3")

PUT localhost:8080/api/admin/users/user/update/3

Params Authorization Headers (11) Body Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 1
2 2   "username": "christine",
3 3   "id": 3,
4 4   "roleType": "USER",
5 5   "firstName": "Christine",
6 6   "lastName": "Johnson-Smith",
7 7   "email": "cjohnson@home.com",
8 8   "password": "$2a$12$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK",
9 9   "phoneNumber": "9858851975",
10 10  "dateCreated": "2022-12-31T23:56:00.000+00:00"
11 11
```

Body Cookies (2) Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```
mysql> select * from users;
```

username	date_created	email	enabled	firstname	userid	lastname	password
christine	2022-12-31 17:56:00.000000	cjohnson@home.com	0x01	Christine	3	Johnson	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK
clementine	2022-12-31 17:56:00.000000	bauchc@home.com	0x01	Clementine	4	Bauch	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK

```
mysql> select * from users;
```

username	date_created	email	enabled	firstname	userid	lastname	password
christine	2022-12-31 17:56:00.000000	cjohnson@home.com	0x00	Christine	3	Johnson-Smith	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK
clementine	2022-12-31 17:56:00.000000	bauchc@home.com	0x01	Clementine	4	Bauch	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK

BEFORE

username	date_created	email	enabled	firstname	userid	lastname	password	phone_number	role_type
christine	2022-12-31 17:56:00.000000	cjohnson@home.com	0x01	Christine	3	Johnson	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK	9858851975	USER
clementine	2022-12-31 17:56:00.000000	bauchc@home.com	0x01	Clementine	4	Bauch	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK	6059902842	USER

AFTER



# Testing Endpoints:

@PostMapping("/add")

POST <localhost:8080/api/admin/users/add> Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1
2     "username": "melvin",
3     "id": "8",
4     "roleType": "USER",
5     "firstName": "Melvin",
6     "lastName": "Whistle",
7     "email": "melvin@home.com",
8     "password": "$2a$12$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK",
9     "phoneNumber": "98523451975"
10
11 mysql> select * from users;
```

username	date_created	email	enabled	firstname	userid	lastname	password	phone_number	role_type
christine	2022-12-31 17:56:00.000000	cjohnson@home.com	0x00	Christine	3	Johnson-Smith	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK	9858851975	USER
clementine	2022-12-31 17:56:00.000000	bauchc@home.com	0x01	Clementine	4	Bauch	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK	6059902842	USER
jasen	2022-12-31 17:56:00.000000	jasen@home.com	0x01	Jasen	2	Jester	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK	8783019215	USER
melvin	2023-01-01 20:24:12.747313	melvin@home.com	0x00	Melvin	8	Whistle	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK	98523451975	USER
monica	2022-12-31 17:56:00.000000	msharp@home.com	0x01	Monica	6	Sharp	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK	4062991259	USER
msmith	2022-12-31 17:56:00.000000	michael@home.com	0x01	Michael	8	Smith	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK	7604129733	USER
niajester	2022-12-31 17:54:28.000000	nia@home.com	0x01	Nia	1	Kelley Jester	\$2a\$12\$P21wlkhdxzMRD3t3UHTnx..Y3sninurN3Q4EH04F7LU8vD/zChj0S	7450495098	ADMIN
ozbrown	2022-12-31 17:56:00.000000	ozbrown@home.com	0x01	Oswald	7	Brown	\$2a\$12\$P21wlkhdxzMRD3t3UHTnx..Y3sninurN3Q4EH04F7LU8vD/zChj0S	9015753955	ADMIN
patw	2022-12-31 17:56:00.000000	patweissnat@home.com	0x01	Patricia	5	Weissnat	\$2a\$12\$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK	4316974631	USER

AFTER

Body Cookies (2) Headers

Pretty Raw Prev

```
1
2     "username": "melvin",
3     "id": 8,
4     "roleType": "USER",
5     "firstName": "Melvin",
6     "lastName": "Whistle",
7     "email": "melvin@home.com",
8     "password": "$2a$12$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtfKK",
9     "phoneNumber": "98523451975",
10    "dateCreated": "2023-01-02T02:24:12.747+00:00"
```

9 rows in set (0.00 sec)

# Testing Endpoints:

@DeleteMapping("/user/delete/{id}")

The screenshot shows a Postman request to `localhost:8080/api/admin/users/user/delete/8`. The Body tab is selected, displaying a JSON object representing a user:

```
1
2   ...
3     "username": "melvin",
4     "id": "8",
5     ...
6       "roleType": "USER",
7       "firstName": "Melvin",
8       "lastName": "Whistle",
9       "email": "melvin@home.com",
10      "password": "$2a$12$JvuspuAAti0LGuuDePrtueodwyf8hLWeRny/G44/dRIu.L0FMtffKK",
11      "phoneNumber": "98523451975"
```

The response status is 500 Internal Server Error, with a size of 16.02 KB.

# Testing Endpoints:

@PutMapping("/product/update/5")

PUT localhost:8080/api/admin/products/product/update/5

Params Authorization Headers (11) Body Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
11 |     "styleId": 1,
12 |     "id": 2,
13 |     "styleName": "running"
14 |   },
15 |   "featureId": {
16 |     "id": 10,
17 |     "featureName": "padded footbed"
18 |   },
19 |   "occasionId": {
20 |     "id": 5,
21 |     "occasionName": "fitness & active"
22 |   },
23 |   "productName": "70s Running Shoe",
24 |   "productDescription": "Hit the track with ease in the retro-inspired Run 70s running shoe from adidas. This mixed suede & textile design proves sturdy, while the foam midsole and full-rubber sole with traction pods provides padding and grip to keep you moving.",
25 |   "unitPrice": 75.00,
26 |   "dateCreated": "2022-12-31T23:56:22.000+00:00"
27 | }
```

product_id	date_created	product_description	product_name	unit_price	brand_id	feature_id	gender_id	occasion_id	style_id
5	2022-12-31 17:56:22.000000	Hit the track with ease in the retro-inspired Run 70s running shoe from adidas. This mixed suede & textile design proves sturdy, while the foam midsole and full-rubber sole with traction pods p	70s Running Shoe	65.00	7	10	2	5	2



BEFORE

product_id	date_created	product_description	product_name	unit_price	brand_id	feature_id	gender_id	occasion_id	style_id
5	2022-12-31 17:56:22.000000	Hit the track with ease in the retro-inspired Run 70s running shoe from adidas. This mixed suede & textile design proves sturdy, while the foam midsole and full-rubber sole with traction pods p	70s Running Shoe	75.00	7	10	2	5	2

AFTER

localhost:8080/api/admin/products/product/5

Dimensions: Responsive 753 x 873 100% No throttling

```
{"id":5,"genderId":{"id":2,"genderName":"men"},"brandId":{"id":7,"brandName":"Adidas"},"styleId": {"id":2,"styleName":"running"}, "featureId":{"id":10,"featureName":"padded footbed"}, "occasionId": {"id":5,"occasionName":"fitness & active"}, "productName": "70s Running Shoe", "productDescription": "Hit the track with ease in the retro-inspired Run 70s running shoe from adidas. This mixed suede & textile design proves sturdy, while the foam midsole and full-rubber sole with traction pods provides padding and grip to keep you moving.", "unitPrice":75.00,"dateCreated": "2022-12-31T23:56:22.000+00:00"}
```



# Testing Endpoints:

@PostMapping("/add")

POST localhost:8080/api/admin/products/add

Params Authorization Headers (11) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1
2     "id": 11,
3     "genderId": {
4         "id": 2,
5         "genderName": "men"
6     },
7     "brandId": {
8         "id": 7,
9         "brandName": "Adidas"
10
11    mysql> select * from products where product_id=12;
12
13 +-----+-----+
14 | product_id | date_created           | product_description
15 | n_id | style_id |
16
17 +-----+
18 | 12 | 2023-01-01 20:43:19.842218 | The runner-inspired profile of the Adidas Lite Racer 2.0 shoe helps you comfortably hustle everyday. Its Cloudfoam midsole lends plush feel and daylong cushioning while the rubber sole of this lace-up shoe offers reliable traction on uneven and slippery surfaces. The upper is made from at least 50% recycled content for a sustainable finish. | Lite Racer BYD 2.0 Sneaker | 54.99 | 7 | 10 | 2 |
19 | 5 | 2 |
20
21 +-----+
22 1 row in set (0.00 sec)
23
24     "productName": "Lite Racer BYD 2.0 Sneaker",
25     "productDescription": "The runner-inspired profile of the Adidas Lite Racer 2.0 shoe helps you comfortably hustle everyday. Its Cloudfoam midsole lends plush feel and daylong cushioning while the rubber sole of this lace-up shoe offers reliable traction on uneven and slippery surfaces. The upper is made from at least 50% recycled content for a sustainable finish.",
26     "unitPrice": 54.99,
27     "date_created": "2023-01-31T23:56:22.000+00:00"
```

AFTER

localhost:8080/api/admin/products/product/12

Dimensions: Responsive 753 x 873 100% No throttling

```
{"id":12,"genderId":{"id":2,"genderName":"men"},"brandId":{"id":7,"brandName":"Adidas"},"styleId": {"id":2,"styleName":"running"}, "featureId":{"id":10,"featureName":"padded footbed"}, "occasionId": {"id":5,"occasionName":"fitness & active"}, "productName":"Lite Racer BYD 2.0 Sneaker", "productDescription":"The runner-inspired profile of the Adidas Lite Racer 2.0 shoe helps you comfortably hustle everyday. Its Cloudfoam midsole lends plush feel and daylong cushioning while the rubber sole of this lace-up shoe offers reliable traction on uneven and slippery surfaces. The upper is made from at least 50% recycled content for a sustainable finish.", "unitPrice":54.99, "date_created":"2023-01-02T02:43:19.842+00:00"}
```

# Testing Endpoints:

@DeleteMapping("/product/delete/{id}")

The screenshot shows a Postman request to the endpoint `localhost:8080/api/admin/products/product/delete/12`. The method is set to `DELETE`. The `Body` tab is selected, showing a JSON payload with various nested objects and their IDs. The `JSON` dropdown is open.

```
1
2   ...
3     "id": 11,
4     "genderId": {
5       ...
6       "id": 2,
7       ...
8       "genderName": "men"
9     },
10    "brandId": {
11      ...
12      "id": 7,
13      ...
14      "brandName": "Adidas"
15    },
16    "styleId": {
17      ...
18      "id": 2,
19      ...
20      "styleName": "running"
21    },
22    "featureId": {
23      ...
24      "id": 10,
25      ...
26      "featureName": "padded footbed"
27    },
28  },
```

Below the body, there are tabs for `Body`, `Cookies (2)`, `Headers (10)`, and `Test Results`. The `JSON` dropdown is also open, showing options: `Pretty`, `Raw`, `Preview`, `Visualize`, and `JSON`.

The screenshot shows a Java code editor with a red box highlighting a section of code. The code is annotated with **BUG TO FIX**. The code is part of a `deleteProduct` method in a `ProductService` class.

```
63
64  @DeleteMapping("/product/delete/{id}")
65  public void deleteProduct(@PathVariable("id") Long id){
66    productService.deleteProduct(id);
67    ResponseEntity.status(HttpStatus.OK);
68  }
69 } //end class
70
```

Below the code, the terminal window shows several Hibernate SQL queries and MySQL error messages indicating a foreign key constraint violation. The MySQL error message is:

```
java.sql.IntegrityConstraintViolationException: Cannot delete or update a parent row: a foreign key constraint fails ('sportyshoesdb`.`product', CONSTRAINT `FK_product_gender` FOREIGN KEY(`gender_id`) REFERENCES `gender`(`id`))
```

A small blue icon in the bottom right corner represents a document or file.

# Conclusion

Sporty Shoes Application



# Lessons Learned

## Proudest Moment

- Navigating the Spring Security architecture
- Designing a highly normalized database

## Most Challenging Design Moment

- Navigating the Spring Security architecture. Getting to verification of the database information was challenging.

## Would I Do This Again?

- Resounding yes!
- There is so much to learn about Spring Security, Spring JPA, and Spring JDBC. I want to keep working on projects to further understand the vast nature of the Spring Architecture.



# **Learning Never Stops...**

## **Items to focus on the Next Release**

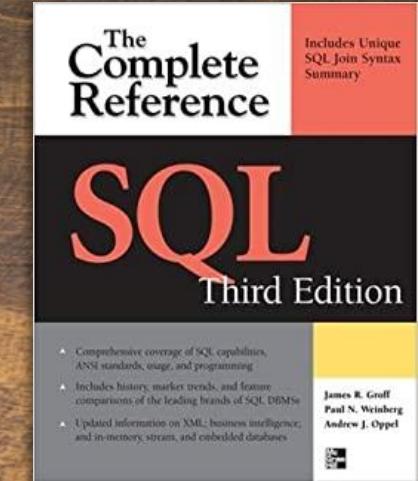
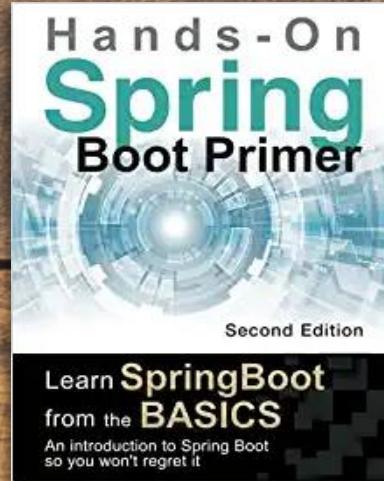
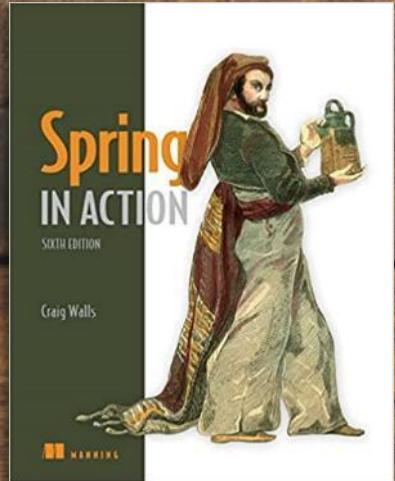
- Implement a robust JUnit Test Suite
- Fix Bug in User & Product Delete Functionality
- Add API endpoints for supplemental tables (styles, brands, occasions, features, etc)



# REFERENCES



# Technical Books



Includes Unique  
SQL Join Syntax  
Summary

James R. Groff  
Paul N. Weinberg  
Andrew J. Oppel