26-Project Using Spring, Fetching Image

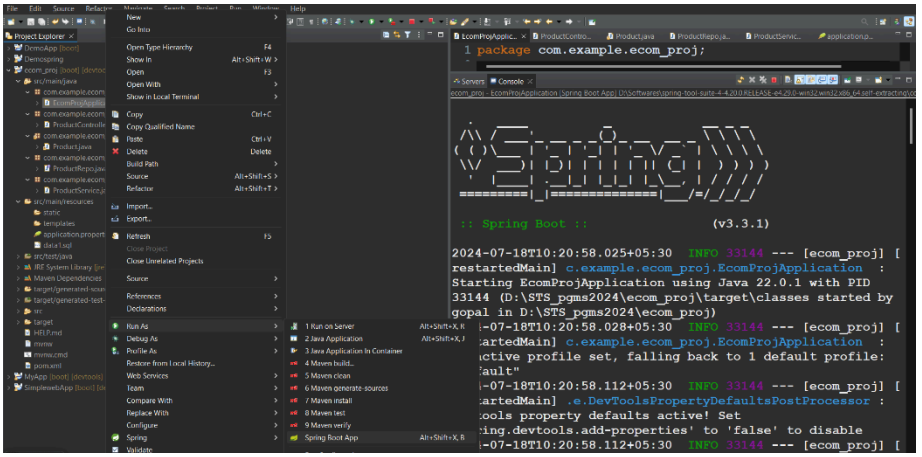
We've been facing issues with our product images not displaying correctly on both the home page and the product details page. These problems are likely due to inconsistencies in variable names between the backend and frontend code. As a result, the images aren't showing up as expected. However, we've now resolved the naming discrepancies on both the front end and back end. The next step is to focus on fixing the image URLs. It's important to note that the backend code for handling these URLs is not yet complete.

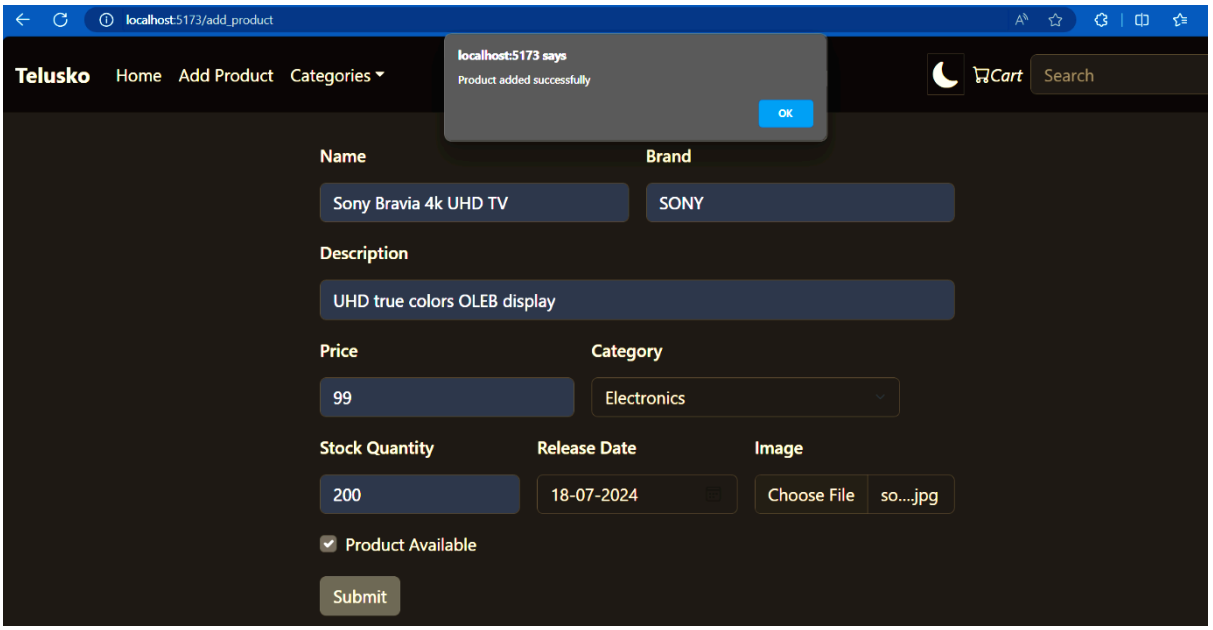
Backend Code for Image URL:

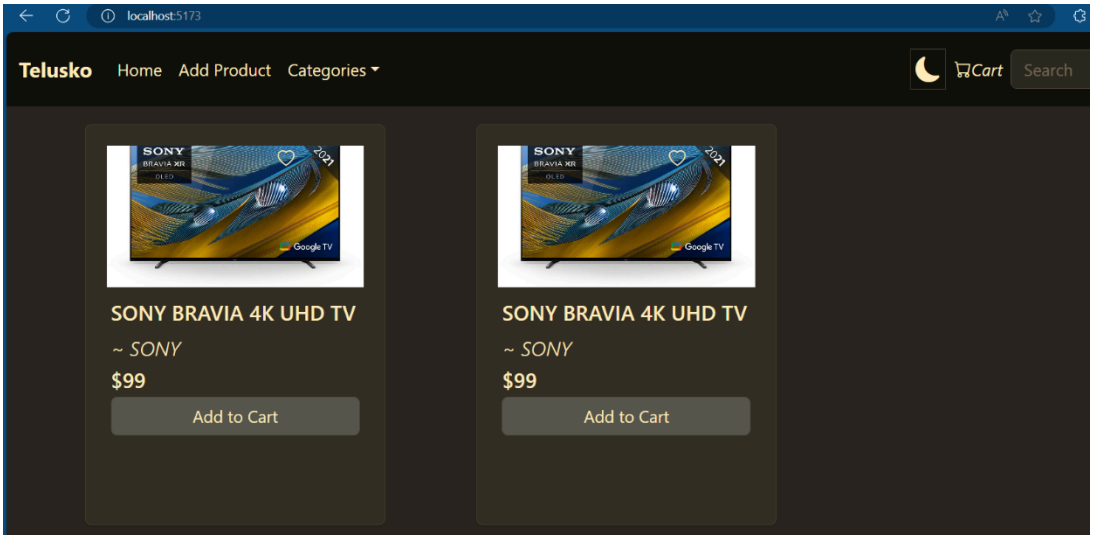
1) **Add endpoint for image URL:** Let's add a method g**etImageByProductI**d to get the image URL and the product ID in the controller.
   We will use a ResponseEntity object of type Product with the URL mapping as @GetMapping("/product/{productid}/image") accepting productId as a parameter that needs to be marked with @PathVariable. Then we will ask the service to give us the implementation. But first, we will get the image in byte format, which will give us the imageData, the imageType (which tells us what kind of content the image is), the image body, and finally the whole image.

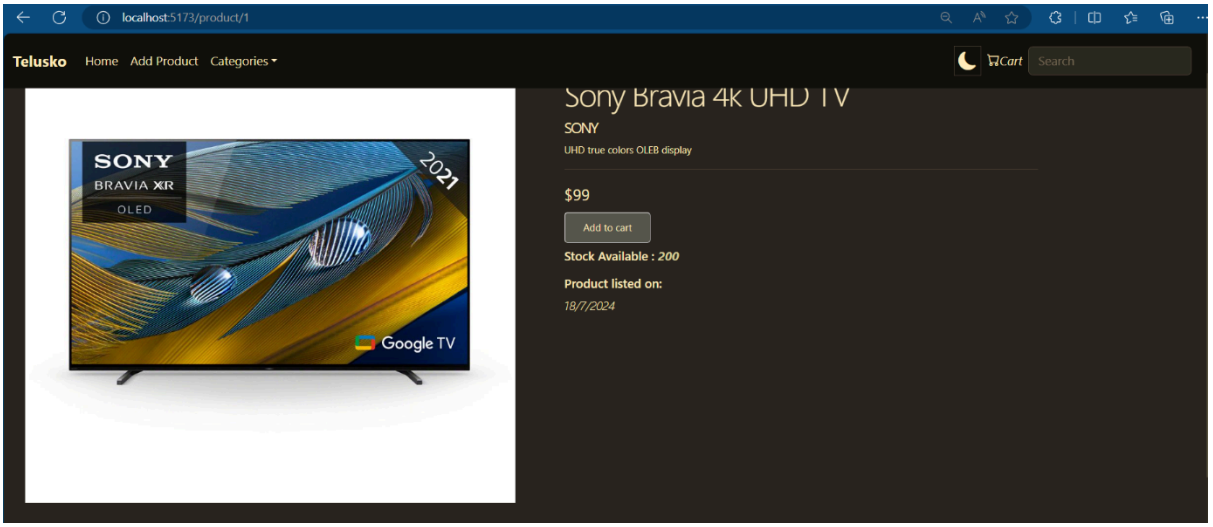
```
@GetMapping("/product/{productId}/image")
public ResponseEntity<byte[]> getImageByProductId(@PathVariable int productId)
    Product product = service.getProductById(productId);
    byte[] imageFile = product.getImageData();
    return ResponseEntity.ok().contentType(MediaType.valueOf
            (product.getImageType())).body(imageFile);

}
```

2) **Restart the project**
   Restart the project, go back to the UI's "Add Product" menu, and add an image to the product details. After clicking "Submit," a pop-up will appear to confirm that the product was successfully added. When we go back to the home page, we can now see our product with a picture. If we click on it, we can see the picture along with its details and a description. The "out of stock" message has also been removed.


package com.example.ecom_proj;
:: Spring Boot ::
(v3.3.1)
2024-07-18T10:20:58.025+05:30 INFO 33144 --- [ecom_proj] [
restartedMain] c.example.ecom_proj.EcomProjApplication :
Starting EcomProjApplication using Java 22.0.1 with PID
33144 (D:\STS_pgms2024\ecom_proj\target\classes started by
gopal in D:\STS_pgms2024\ecom_proj)


localhost:5173/add_product
Telusko
Home
Add Product
Categories
Cart
Search
localhost:5173 says
Product added successfully
OK
Name
Brand
Sony Bravia 4k UHD TV
SONY
Description
UHD true colors OLEB display
Price
Category
99
Electronics
Stock Quantity
Release Date
Image
200
18-07-2024
Choose File
so....jpg
Product Available
Submit


localhost:5173
Telusko
Home
Add Product
Categories
Cart
Search
SONY BRAVIA 4K UHD TV
~ SONY
$99
Add to Cart
SONY BRAVIA 4K UHD TV
~ SONY
$99
Add to Cart



We can try additional things by managing and handling the backend. For example, when we send the data, we can check and make sure that if an ID is sent but not present, then no data is displayed, right? In the same way, we can handle these kinds of requests with try-catch blocks or conditional statements.

Stay tuned for the next chapter, where we'll talk about the way to delete and update files. Up until then, keep learning and have fun coding!

☺ 👾