Assignment 2: Working with JSON Data (product_data.json)

Tasks:

1. Load the JSON data:

- Load the product_data.json file into a DataFrame.
- Display the first 10 rows and inspect the schema.

1 minute ago (1s) 6

```
from pyspark.sql import functions as F
# Copy the file from the local workspace to DBFS
dbutils.fs.cp("file:/Workspace/Shared/product_data.json", "dbfs:/FileStore/product_data.json")
# Load the JSON file
df = spark.read.format("json").load("dbfs:/FileStore/product_data.json",multiLine=True)

# Display the first 10 rows
display(df.limit(10))

# Inspect the schema
df.printSchema()
```

Output Terminal Debug Console

	Category	Price	ProductID	ProductName	Stock
1	Electronics	1200	101	Laptop	35
2	Electronics	800	102	Smartphone	80
3	Furniture	150	103	Desk Chair	60
4	Electronics	300	104	Monitor	45
5	Furniture	350	105	Desk	25

5 rows | 0.93 seconds runtime

```
root
 |-- Category: string (nullable = true)
 |-- Price: long (nullable = true)
 |-- ProductID: long (nullable = true)
 |-- ProductName: string (nullable = true)
 |-- Stock: long (nullable = true)
```

2. Data Cleaning:

- Remove rows where Stock is less than 30.
- Filter the products that belong to the "Electronics" category.

Just now (<1s)

```
# Remove rows where Stock is less than 30
df_cleaned = df.filter(col('Stock') >= 30)
df_cleaned.show()

# Filter products that belong to the "Electronics" category
filtered_products = df_cleaned.filter(df_cleaned['Category'] == 'Electronics')
filtered_products.show()
```

Output Terminal Debug Console

df_cleaned: pyspark.sql.dataframe.DataFrame = [Category: string, Price: long ... 3 more fields]

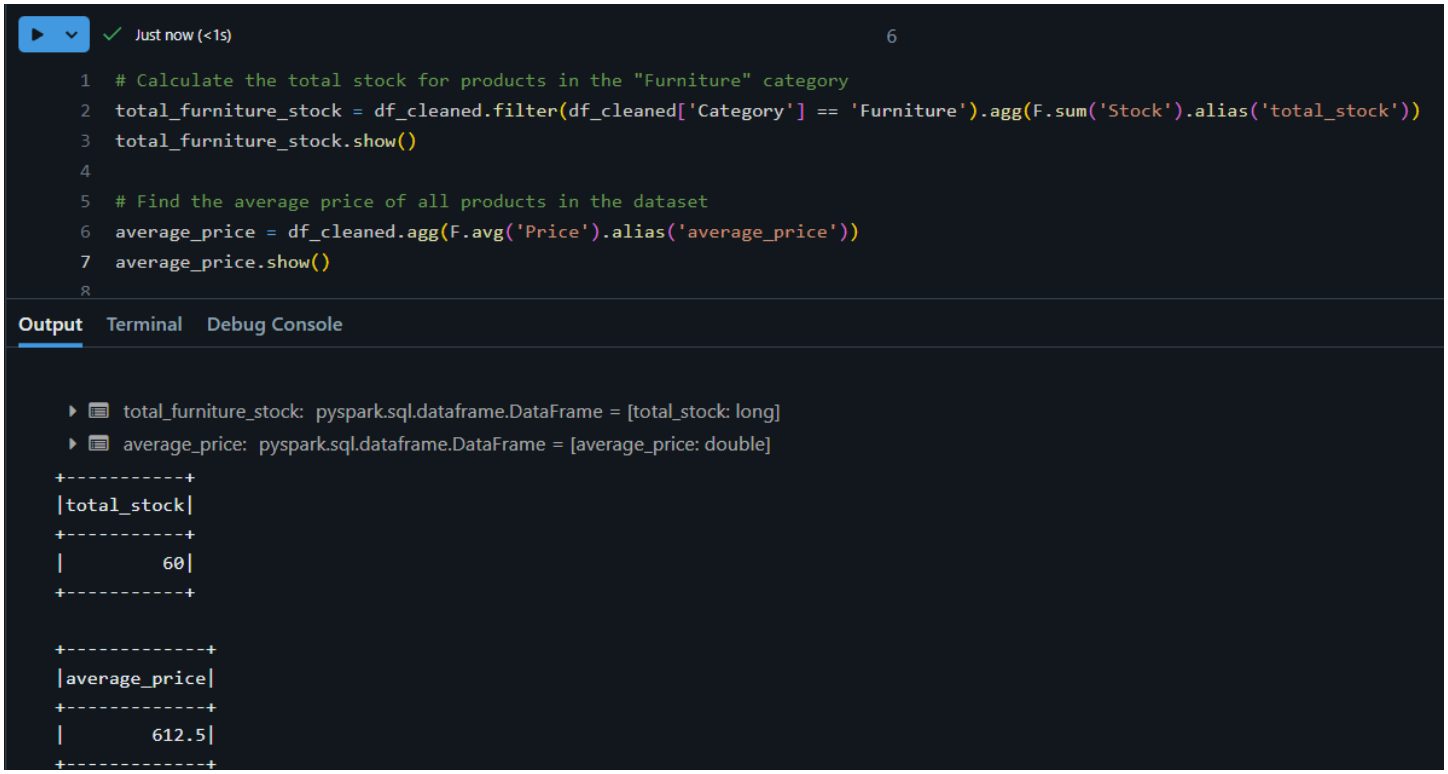
filtered_products: pyspark.sql.dataframe.DataFrame = [Category: string, Price: long ... 3 more fields]

```
+-----------+-----+---------+-----------+-----+
|   Category|Price|ProductID|ProductName|Stock|
+-----------+-----+---------+-----------+-----+
|Electronics| 1200|      101|     Laptop|   35|
|Electronics|  800|      102| Smartphone|   80|
|  Furniture|  150|      103| Desk Chair|   60|
|Electronics|  300|      104|    Monitor|   45|
+-----------+-----+---------+-----------+-----+

+-----------+-----+---------+-----------+-----+
|   Category|Price|ProductID|ProductName|Stock|
+-----------+-----+---------+-----------+-----+
|Electronics| 1200|      101|     Laptop|   35|
|Electronics|  800|      102| Smartphone|   80|
|Electronics|  300|      104|    Monitor|   45|
+-----------+-----+---------+-----------+-----+
```

3. Data Aggregation:

- Calculate the total stock for products in the "Furniture" category.
- Find the average price of all products in the dataset.



4. Write the Data to JSON:

- Save the cleaned and aggregated data into a new JSON file.