

First, I have installed java, python, spark in my local system.

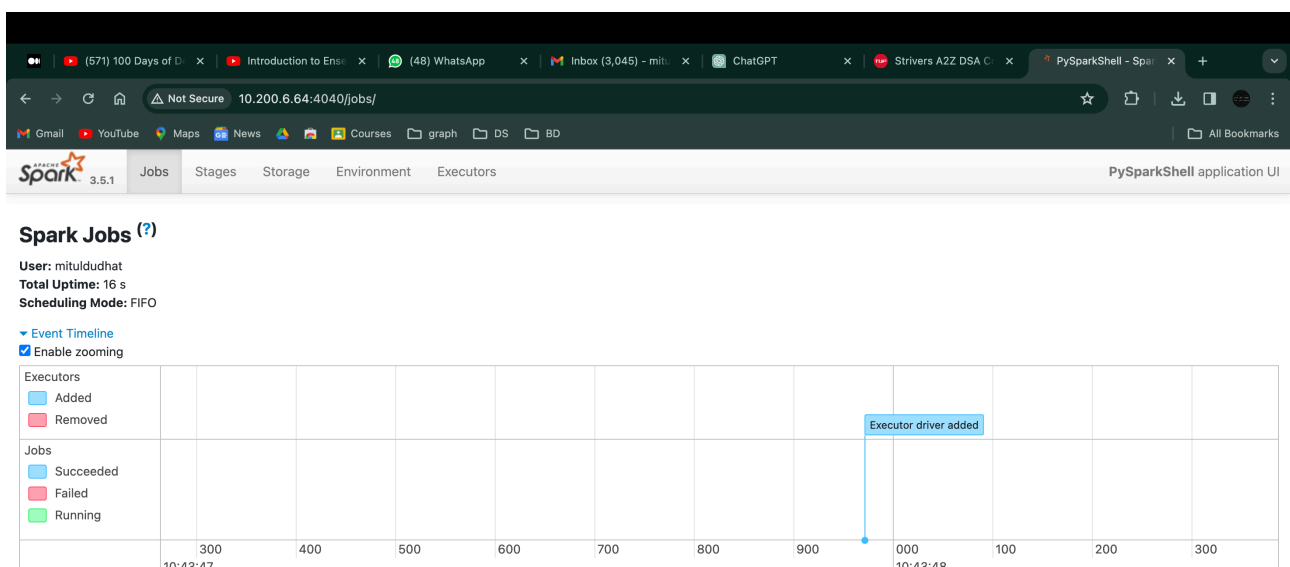
```
mituldudhat@Mituls-MacBook-Air ~ % java -version
java version "21.0.2" 2024-01-16 LTS
Java(TM) SE Runtime Environment (build 21.0.2+13-LTS-58)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.2+13-LTS-58, mixed mode, sharing)
mituldudhat@Mituls-MacBook-Air ~ % python3 --version
/Library/Frameworks/Python.framework/Versions/3.12/Resources/Python.app/Contents/MacOS/Python: can't open file '/Users/mituldudhat/---version': [Errno 2] No such file or directory
mituldudhat@Mituls-MacBook-Air ~ % pyspark
Python 3.12.1 (v3.12.1:2305ca5144, Dec 7 2023, 17:23:38) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
24/03/07 10:33:48 WARN Utils: Your hostname, Mituls-MacBook-Air.local resolves to a loopback address: 127.0.0.1; using 10.200.6.64 instead (on interface en0)
24/03/07 10:33:48 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/03/07 10:33:48 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to

  ____      _
 / ___|    / \
| |  | |  / _ \
| |  | | / ___ \
| |  | |/_/   \_\
|_|  |____/___/

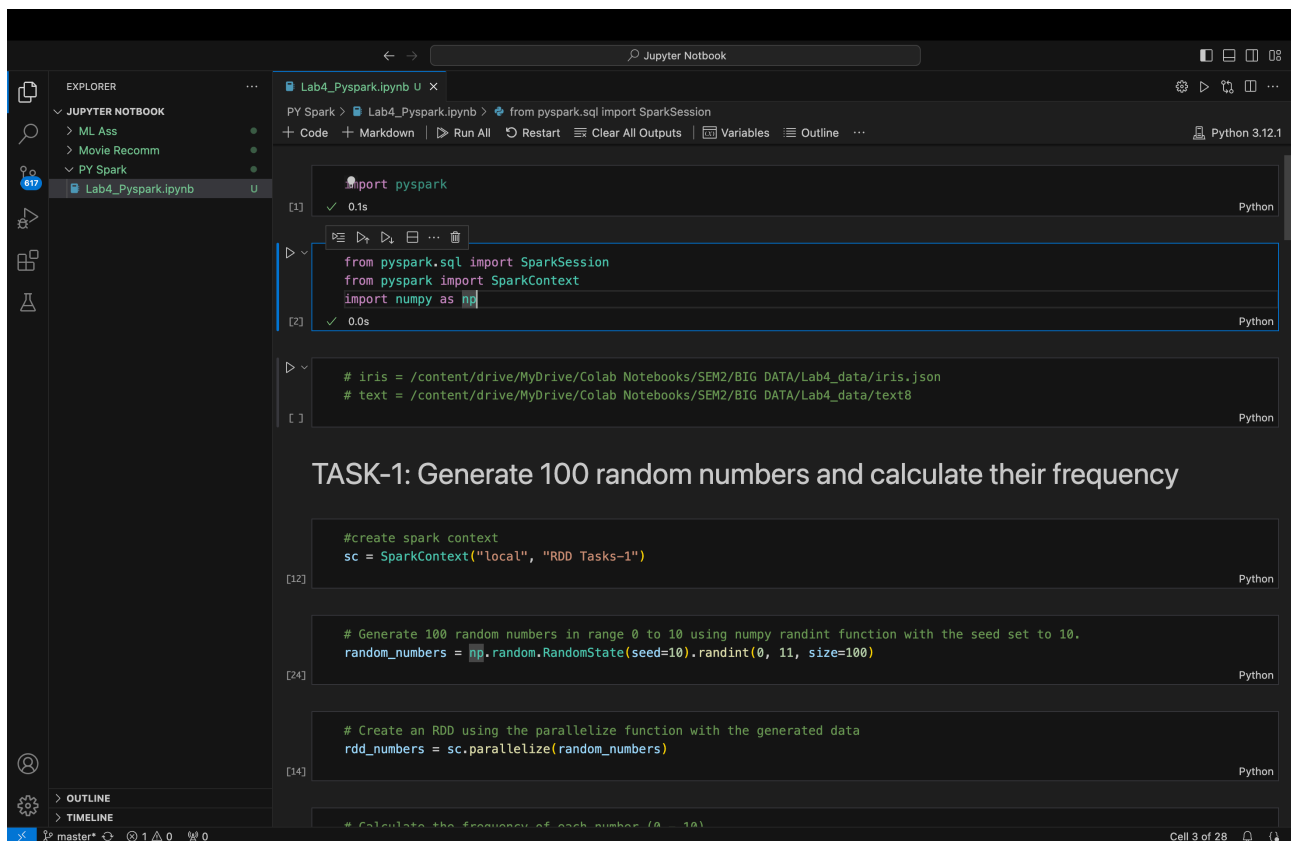
version 3.5.1

Using Python version 3.12.1 (v3.12.1:2305ca5144, Dec 7 2023 17:23:38)
Spark context Web UI available at http://10.200.6.64:4040
Spark context available as 'sc' (master = local[*], app id = local-1709787829215).
SparkSession available as 'spark'.
>>> 24/03/07 10:34:00 WARN GarbageCollectionMetrics: To enable non-built-in garbage collector(s) List(G1 Concurrent GC), users should configure it(them) to spark.eventLog.gcMetrics.youngGenerationGarbageCollectors or spark.eventLog.gcMetrics.oldGenerationGarbageCollectors
```

Then open local spark website in Chrome...



## And start Assignment's First Question in VS Code



The screenshot shows a VS Code editor with a Jupyter Notebook open. The notebook is titled "Lab4\_Pyspark.ipynb" and is in the "Code" view. The code is written in Python and is part of a Jupyter Notebook. The code is as follows:

```
import pyspark

from pyspark.sql import SparkSession
from pyspark import SparkContext
import numpy as np

# iris = /content/drive/MyDrive/Colab Notebooks/SEM2/BIG DATA/Lab4_data/iris.json
# text = /content/drive/MyDrive/Colab Notebooks/SEM2/BIG DATA/Lab4_data/text8

TASK-1: Generate 100 random numbers and calculate their frequency

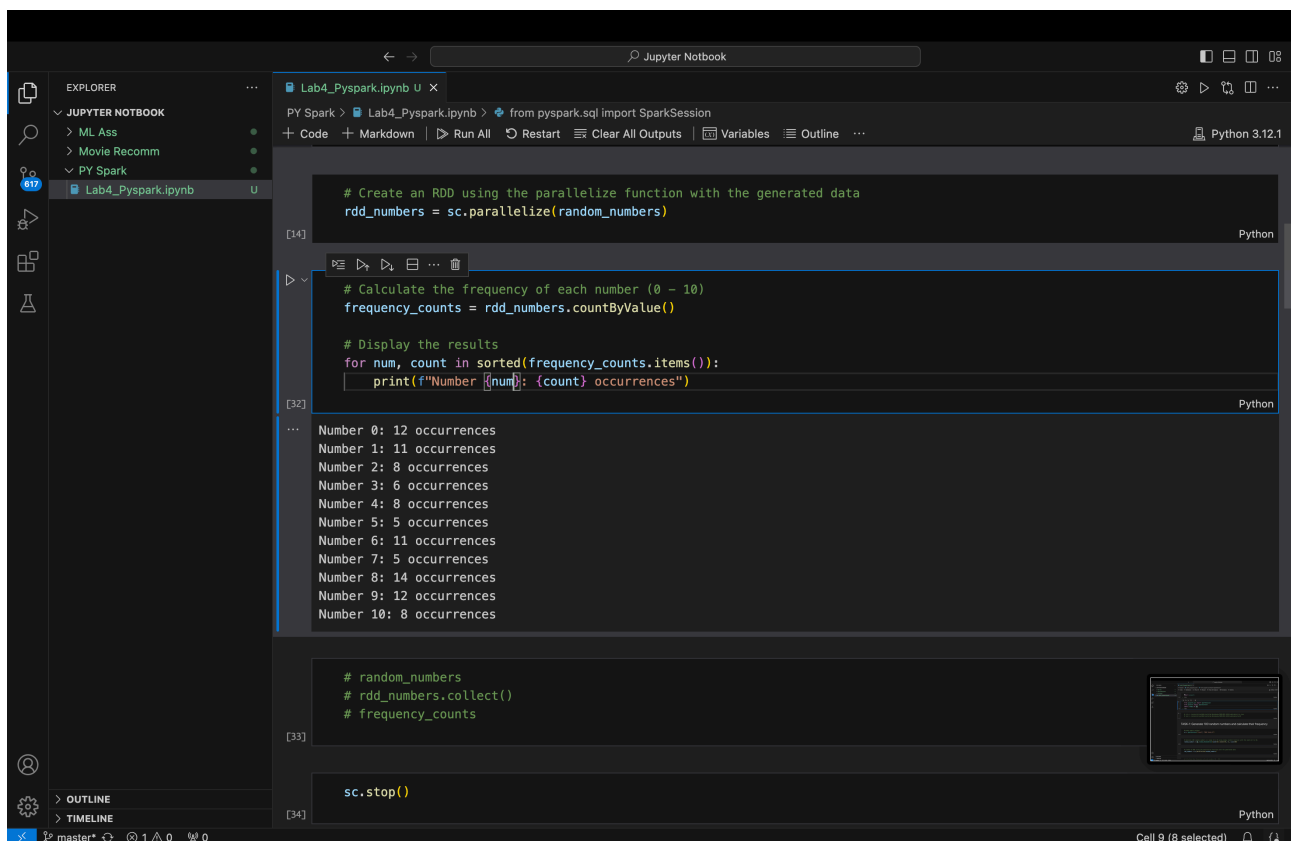
# create spark context
sc = SparkContext("local", "RDD Tasks-1")

# Generate 100 random numbers in range 0 to 10 using numpy randint function with the seed set to 10.
random_numbers = np.random.RandomState(seed=10).randint(0, 11, size=100)

# Create an RDD using the parallelize function with the generated data
rdd_numbers = sc.parallelize(random_numbers)

# Calculate the frequency of each number (0 - 10)
```

The code is executed in a Jupyter Notebook cell, and the output shows the execution of the code. The code is written in Python and is part of a Jupyter Notebook. The code is as follows:



The screenshot shows a VS Code editor with a Jupyter Notebook open. The notebook is titled "Lab4\_Pyspark.ipynb" and is in the "Code" view. The code is written in Python and is part of a Jupyter Notebook. The code is as follows:

```
# Create an RDD using the parallelize function with the generated data
rdd_numbers = sc.parallelize(random_numbers)

# Calculate the frequency of each number (0 - 10)
frequency_counts = rdd_numbers.countByValue()

# Display the results
for num, count in sorted(frequency_counts.items()):
    print(f"Number {num}: {count} occurrences")

# random_numbers
# rdd_numbers.collect()
# frequency_counts

sc.stop()
```

The code is executed in a Jupyter Notebook cell, and the output shows the execution of the code. The output is as follows:

```
Number 0: 12 occurrences
Number 1: 11 occurrences
Number 2: 8 occurrences
Number 3: 6 occurrences
Number 4: 8 occurrences
Number 5: 5 occurrences
Number 6: 11 occurrences
Number 7: 5 occurrences
Number 8: 14 occurrences
Number 9: 12 occurrences
Number 10: 8 occurrences
```

The code is written in Python and is part of a Jupyter Notebook. The code is as follows:

