**NAME :- ANIKET SANJAYKUMAR BIYANI**

**DATA ENGINEERING BATCH -1**

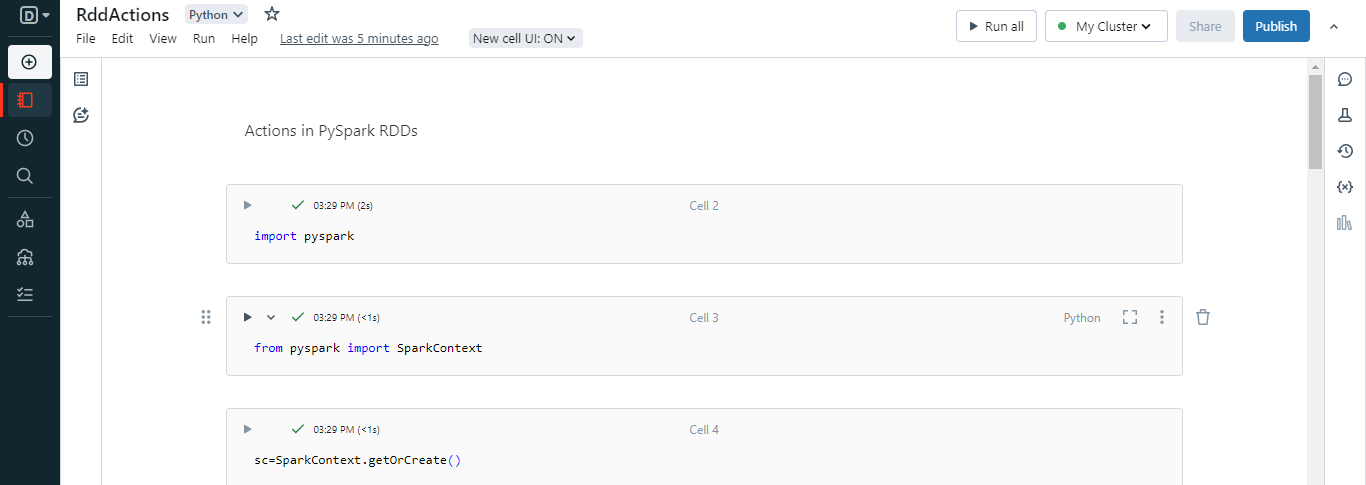
**PYTHON DAY-13 ASSIGNMENT**

**TRANSFORMATIONS AND ACTIONS:-**

* PySpark RDD Operations Resilient Distributed Dataset or RDD in a PySpark is a core data structure of PySpark.
* PySpark RDD’s is a low-level object and are highly efficient in performing distributed tasks
* PySpark RDD has a set of operations to accomplish any task. These operations are of two types: 1. Transformations 2. Actions

**ACTIONS:-->**

* Actions are a kind of operation which are applied on an RDD to produce a single value.
* These methods are applied on a resultant RDD and produces a non-RDD value, thus removing the laziness of the transformation of RDD.
* **we will initialize a SparkContext to perform the operations**

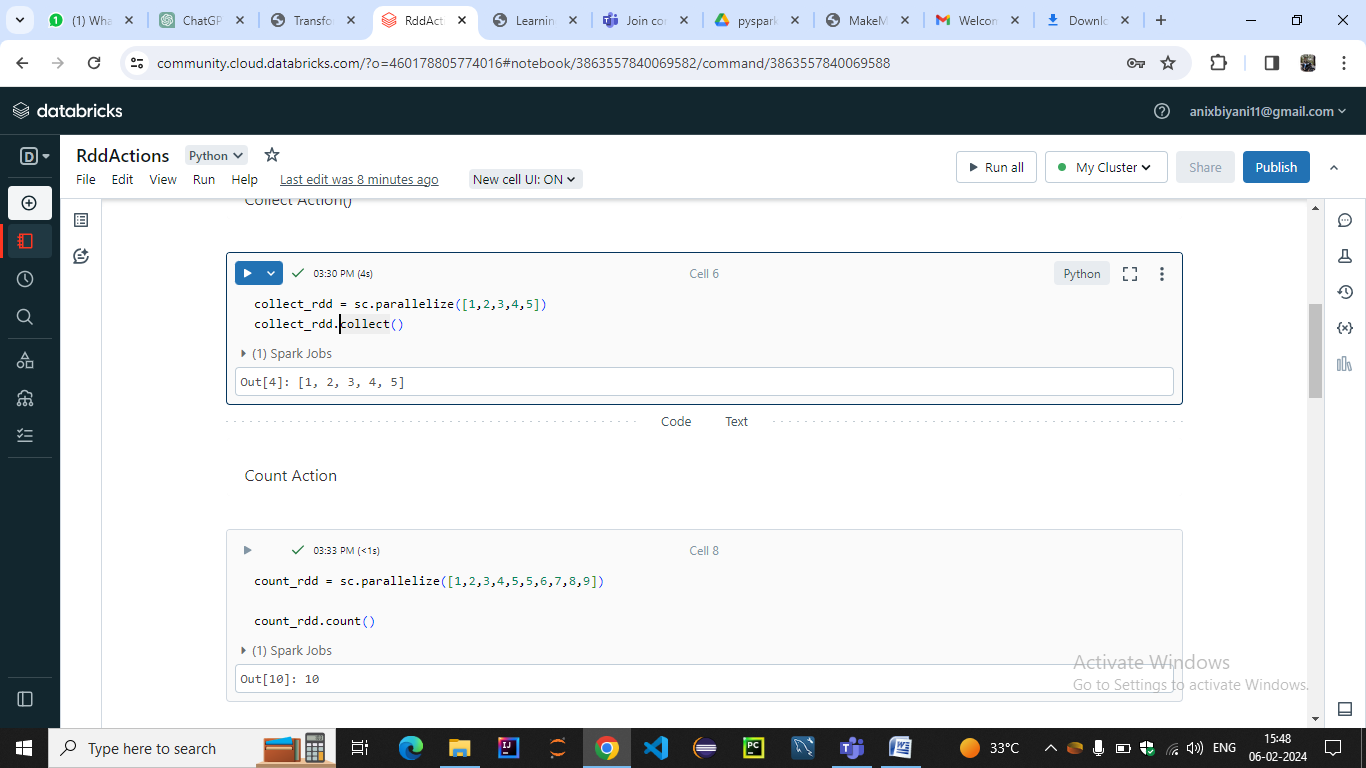
****

* **Collect Action():-**

The .collect() action on an RDD returns a list of all the elements of the RDD. It’s a great asset for displaying all the contents of our RDD.

* **Count Action():-**

The .count() action on an RDD is an operation that returns the number of elements of our RDD.

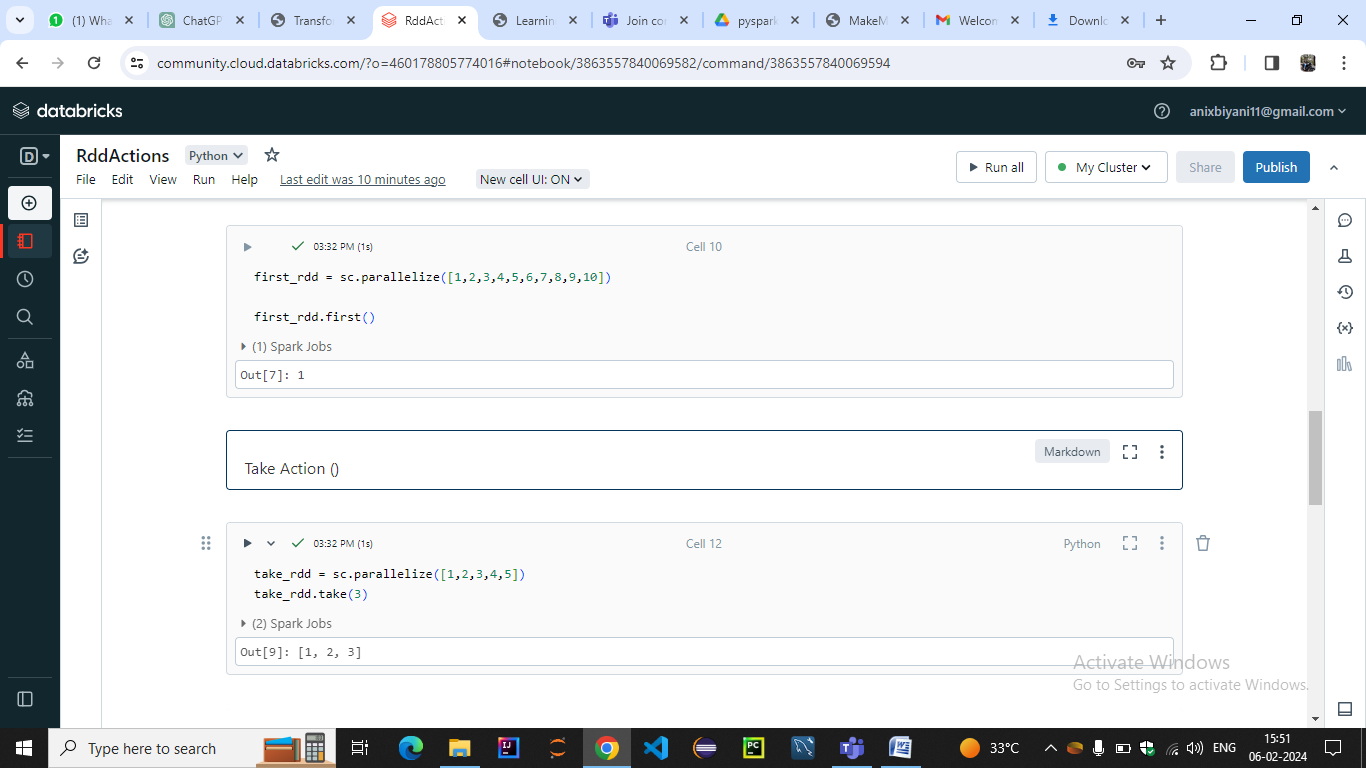


* **The .first() Action ():-**

The .first() action on an RDD returns the first element from our RDD.

* **. The .take() Action** :-

The .take(n) action on an RDD returns n number of elements from the RDD



* **The .reduce() Action :-**

The .reduce() Actiontakes two elements from the given RDD and operates. This operation is performed using an anonymous function or lambda

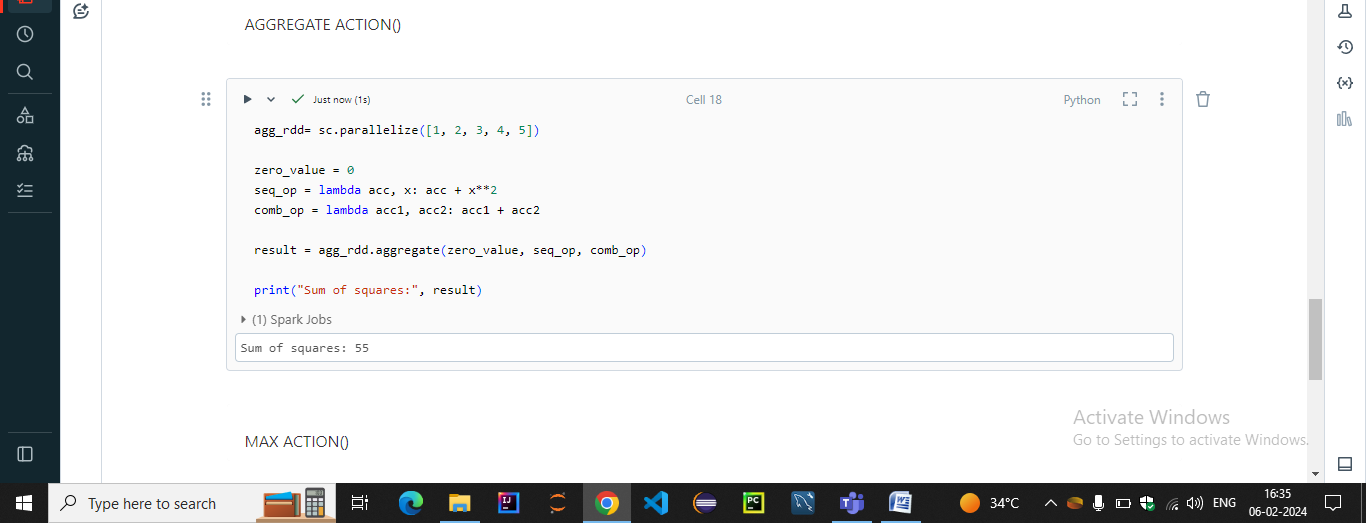
* **The .saveAsTextFile() Action:-**

It is used to serve the resultant RDD as a text file. We can also specify the path to which file needed to be saved. This helps in saving our results especially when we are working with a large amount of data



* AGGREGATE Action():-

It provides more flexibility compared to **reduce** or **fold** actions by allowing you to specify both an initial value and a combination function to aggregate the elements

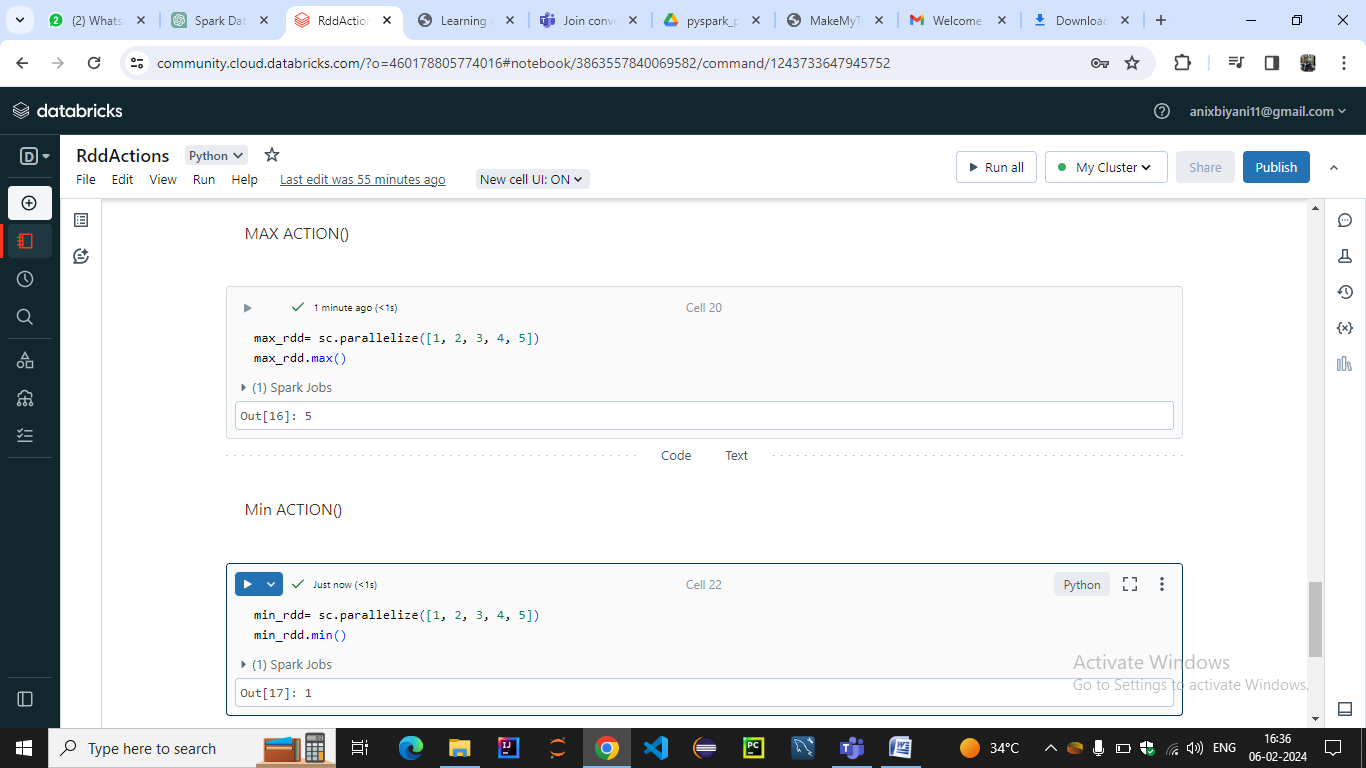


* MAX ()ACTION:-

We use the **max** method to find the maximum value in the RDD and store it in the variable **max\_value**.

* MIN() ACTION:-

Similarly, we use the **min** method to find the minimum value in the RDD and store it in the variable **min\_value.**

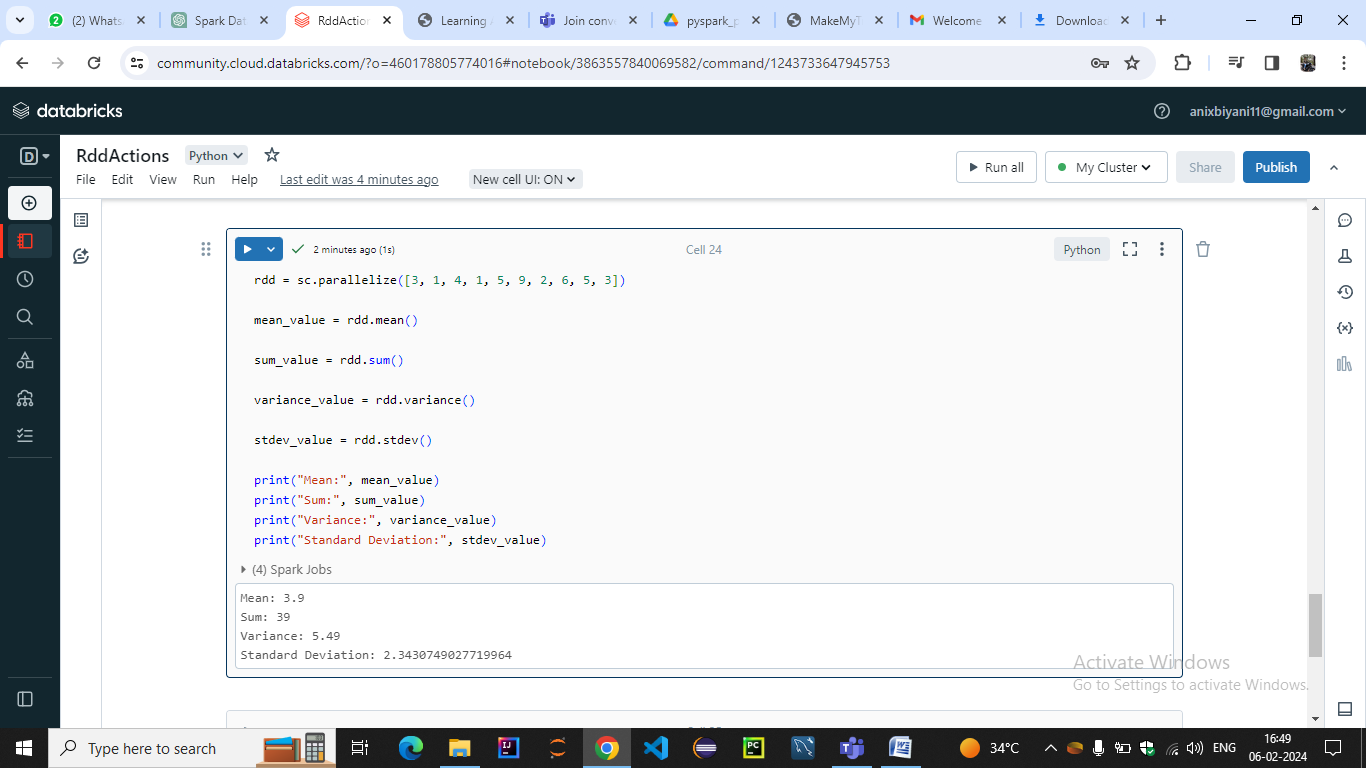


**MEAN ACTION():-**

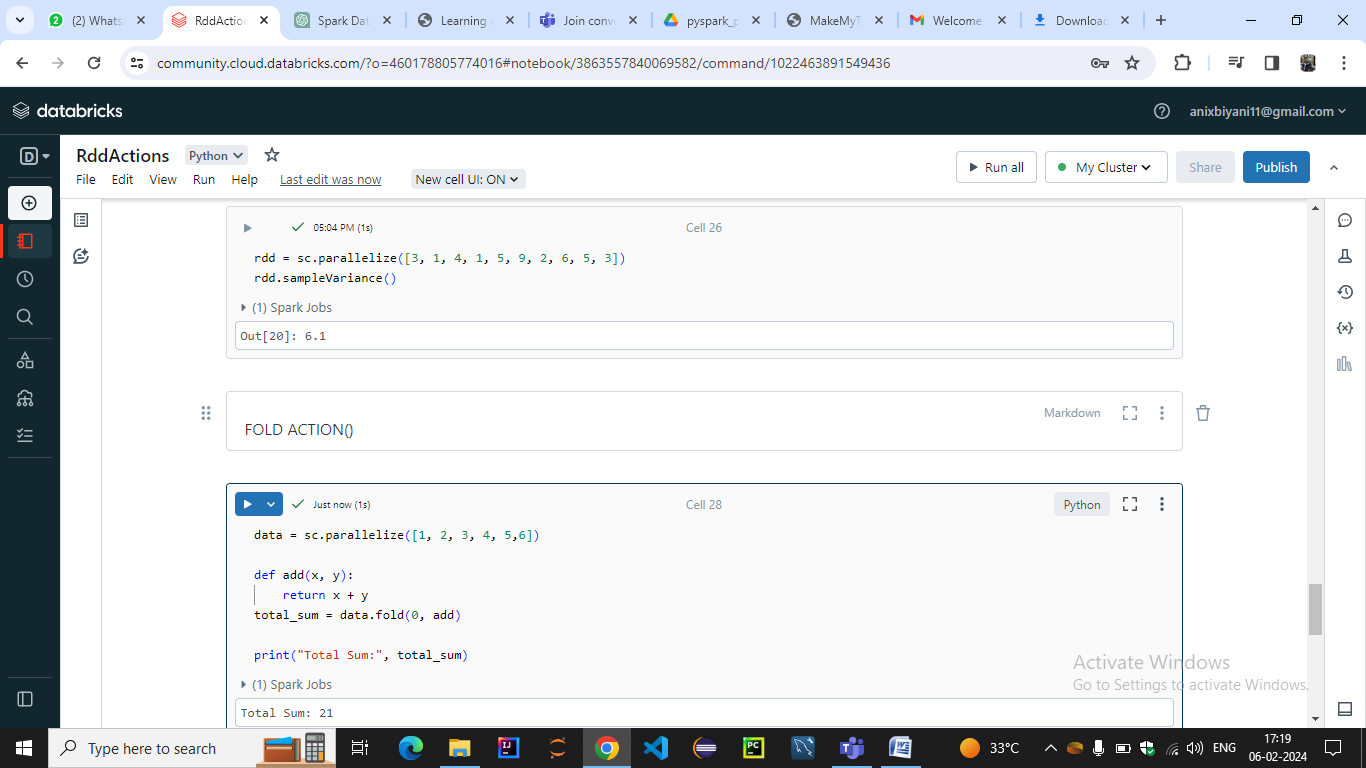
**SUM ACTION():-**

**VARIANCE ACTION():-**

**STDEV ACTION():-**



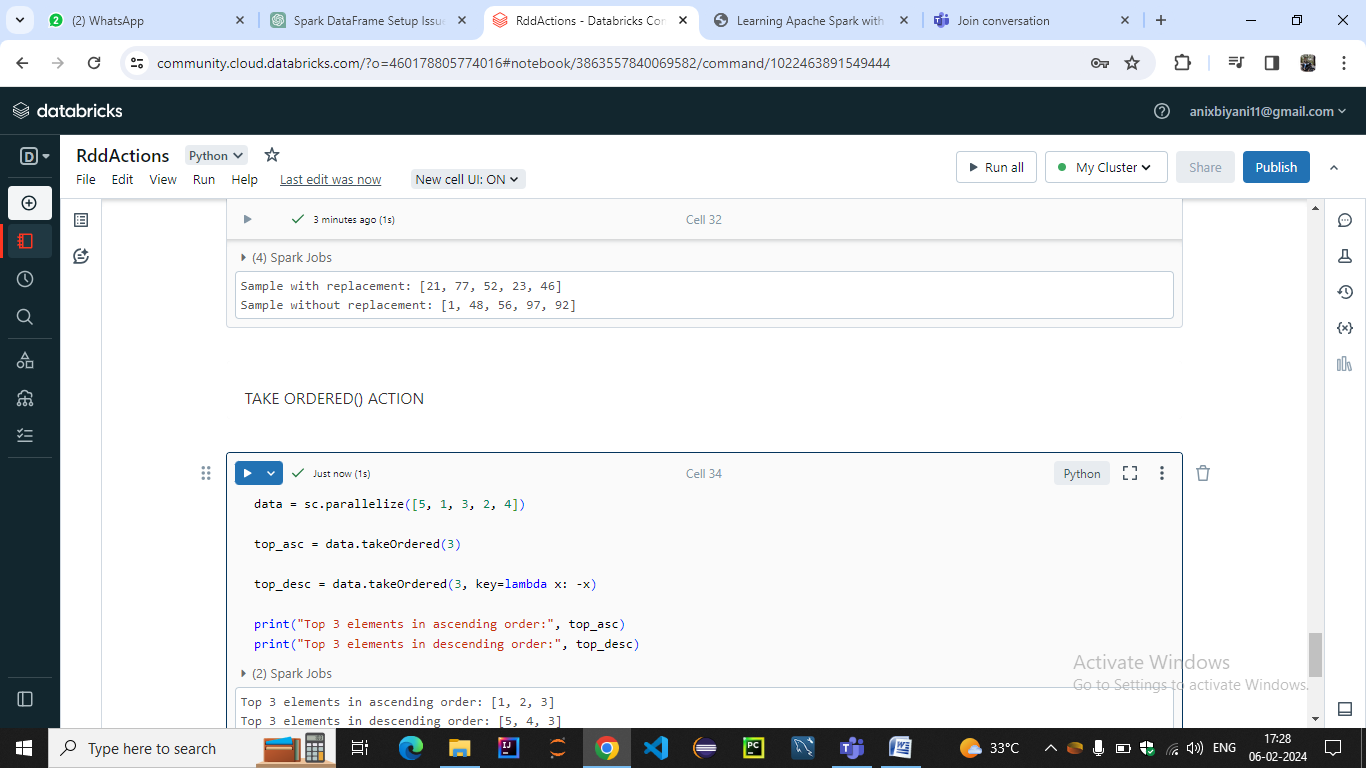
SAMPLE VARIANCE()

FOLD()

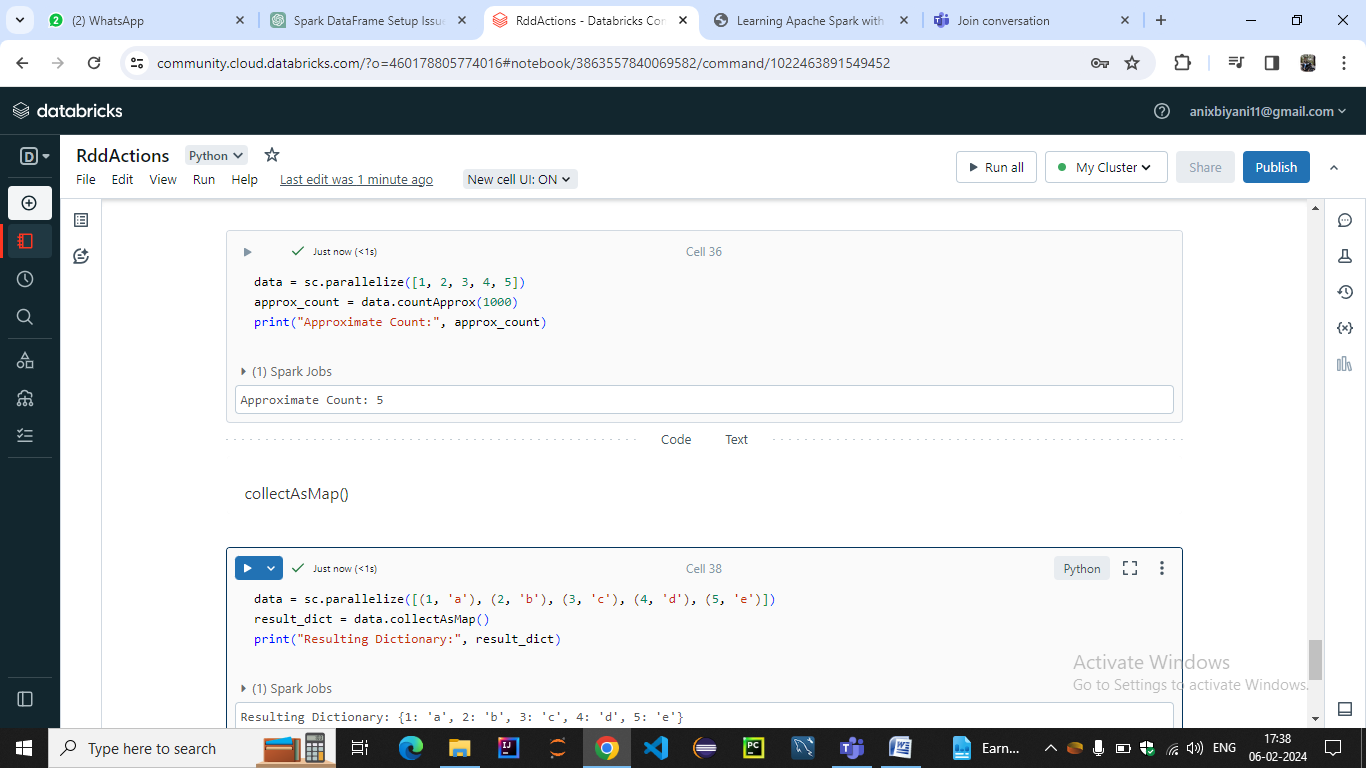
TOP ACTION():-



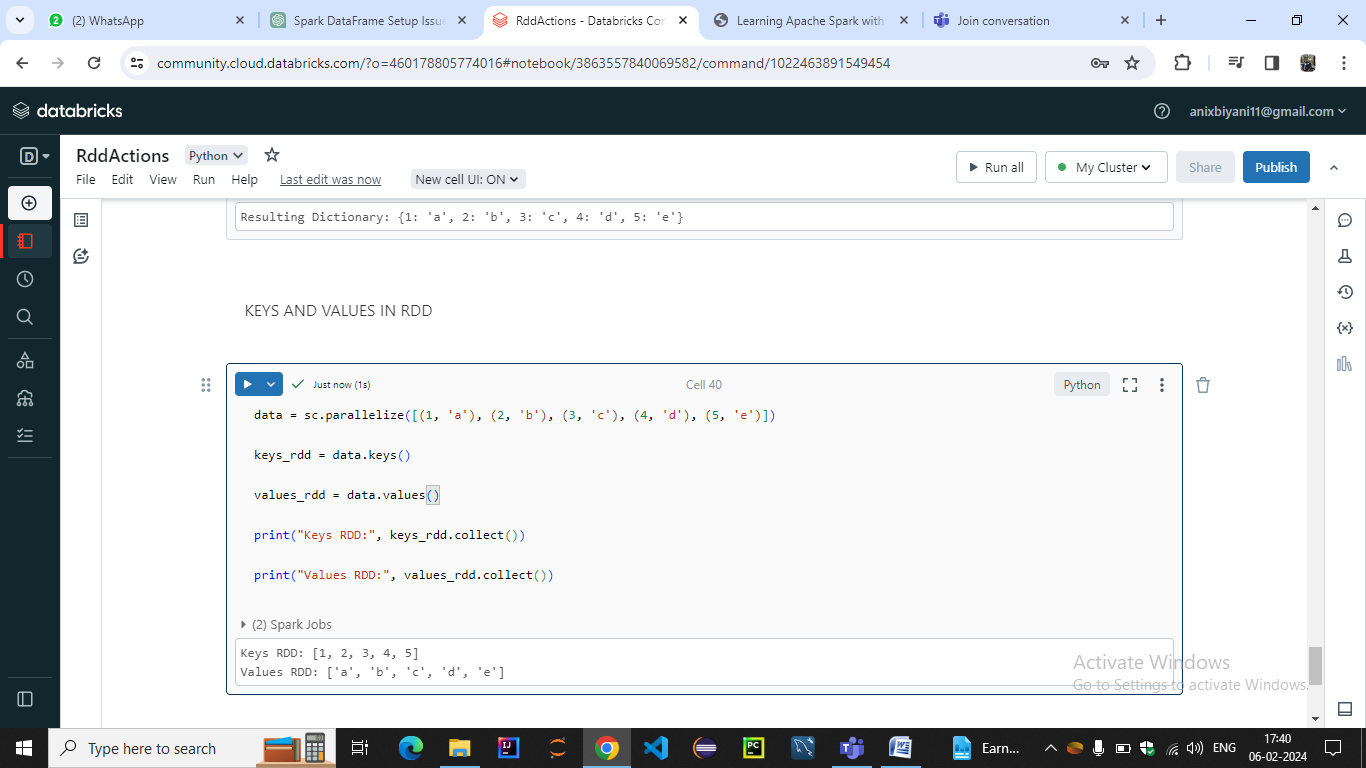
takeOrdered():- takeSample Action():-



COUNT APPROX() ,COLLECTASMAP()

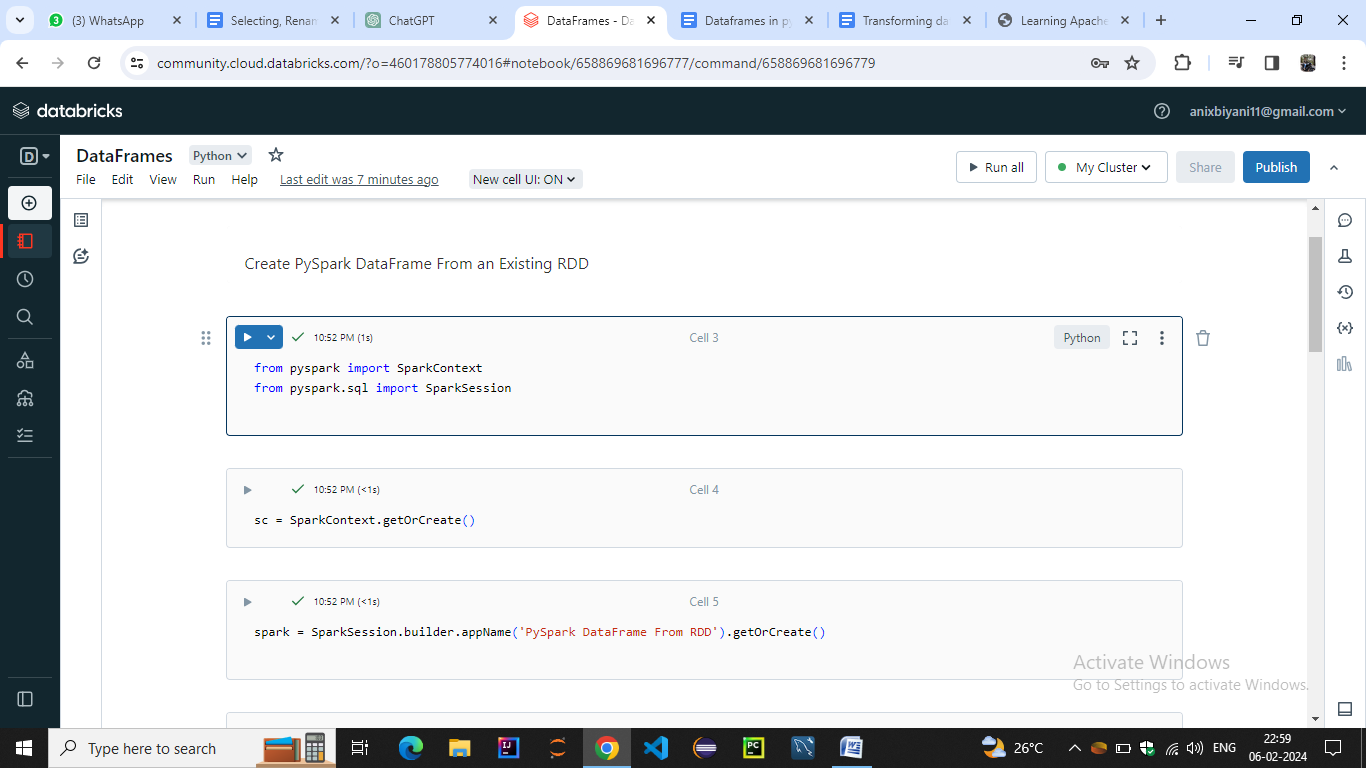


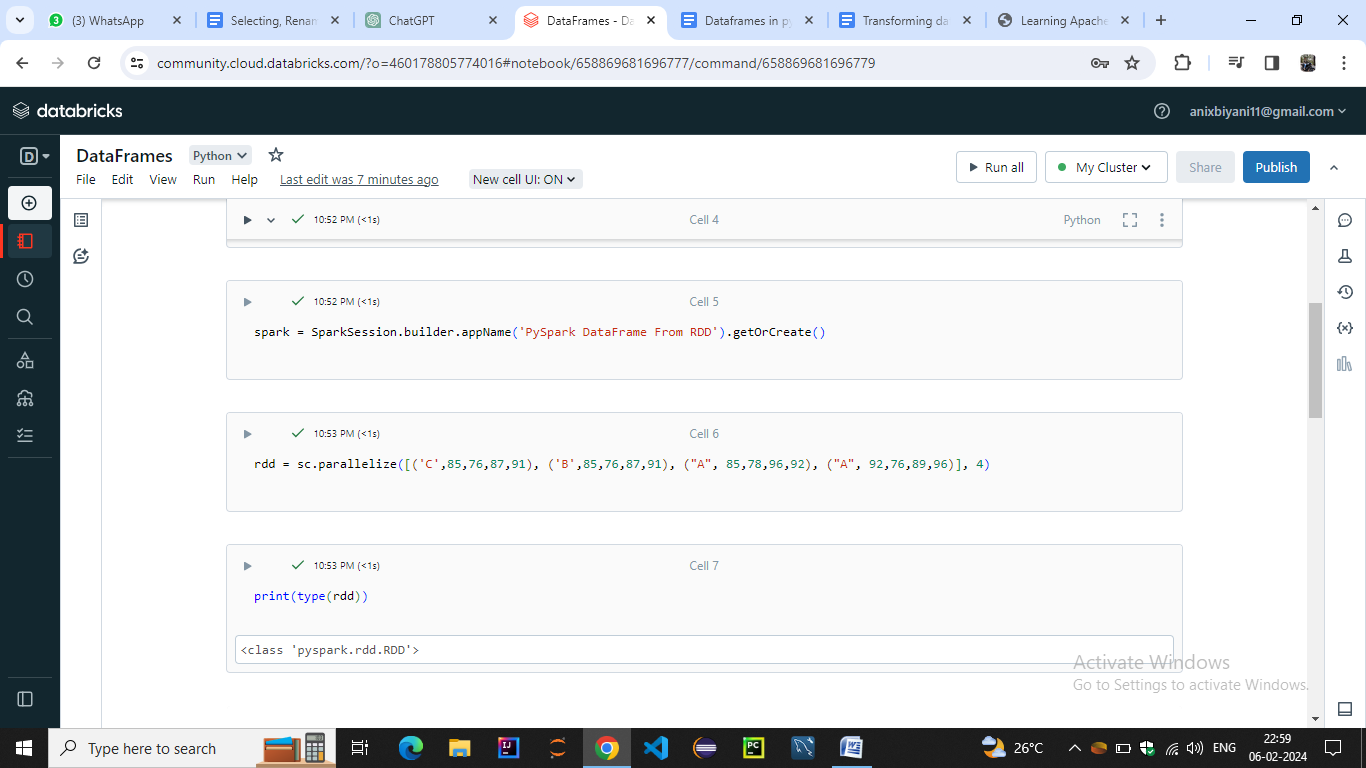
KEYS AND VALUES IN RDD

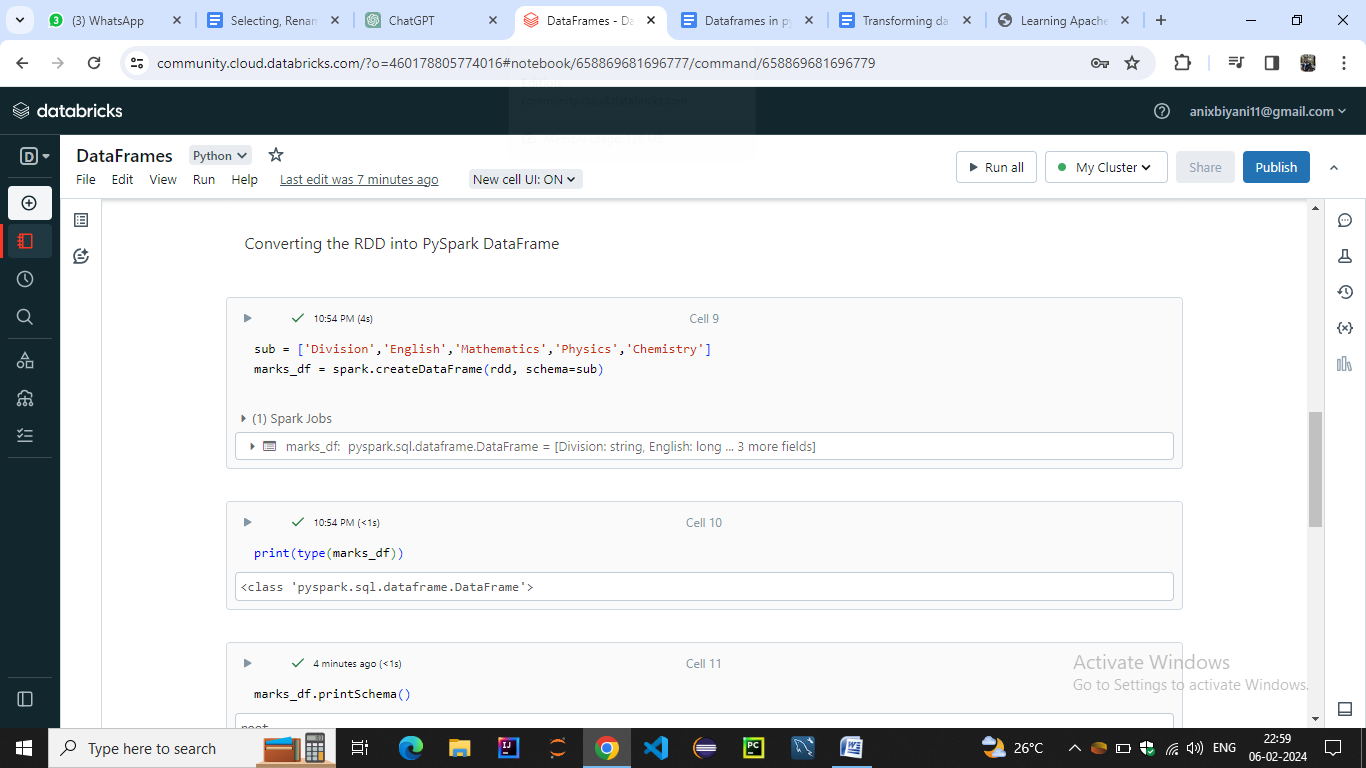


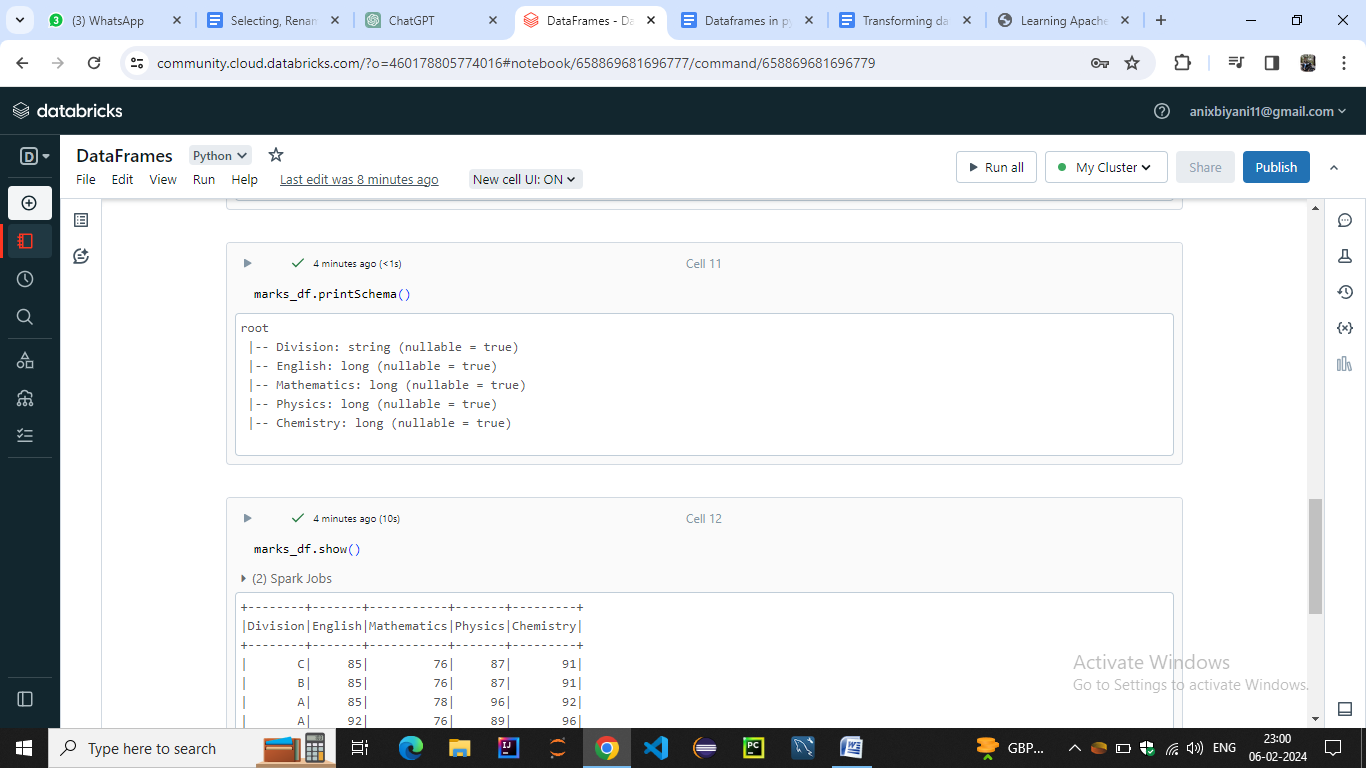
* Dataframes in pyspark

**Create PySpark DataFrame From an Existing RDD**



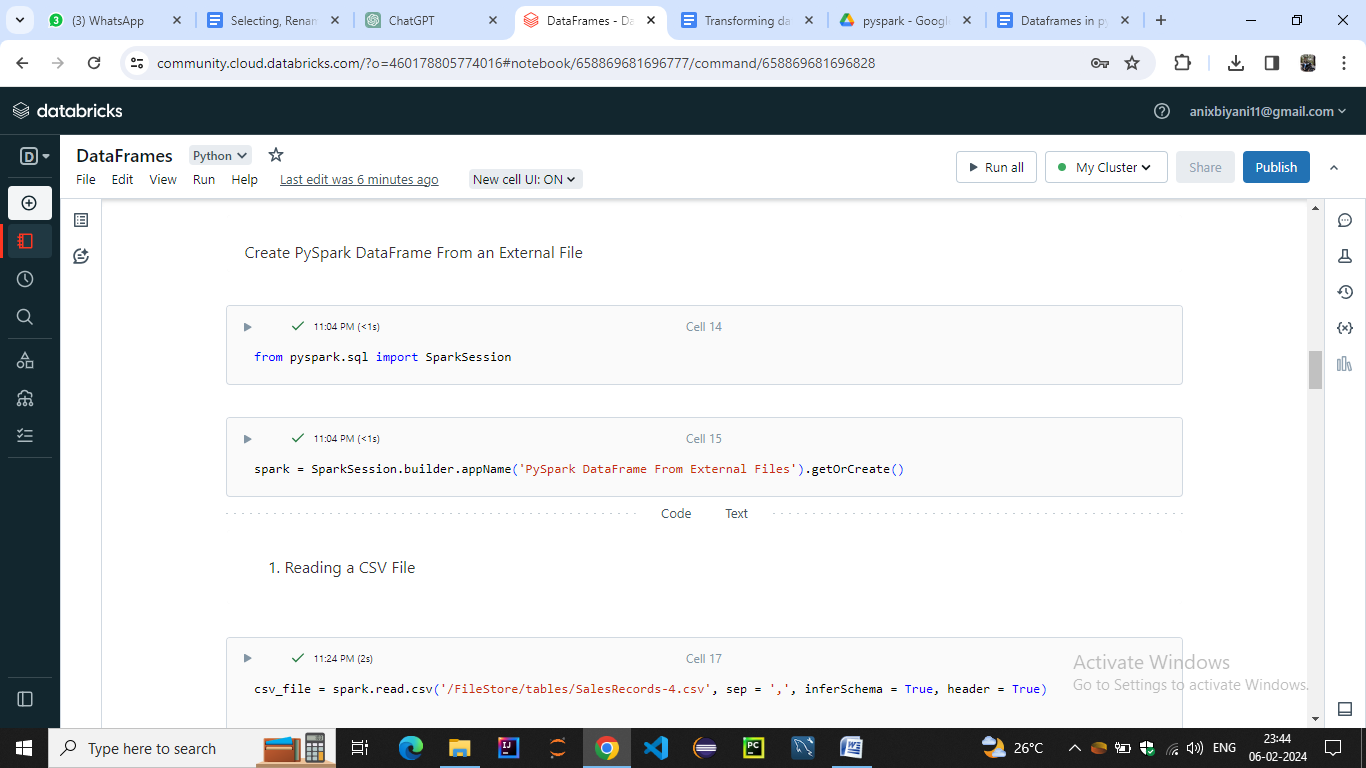


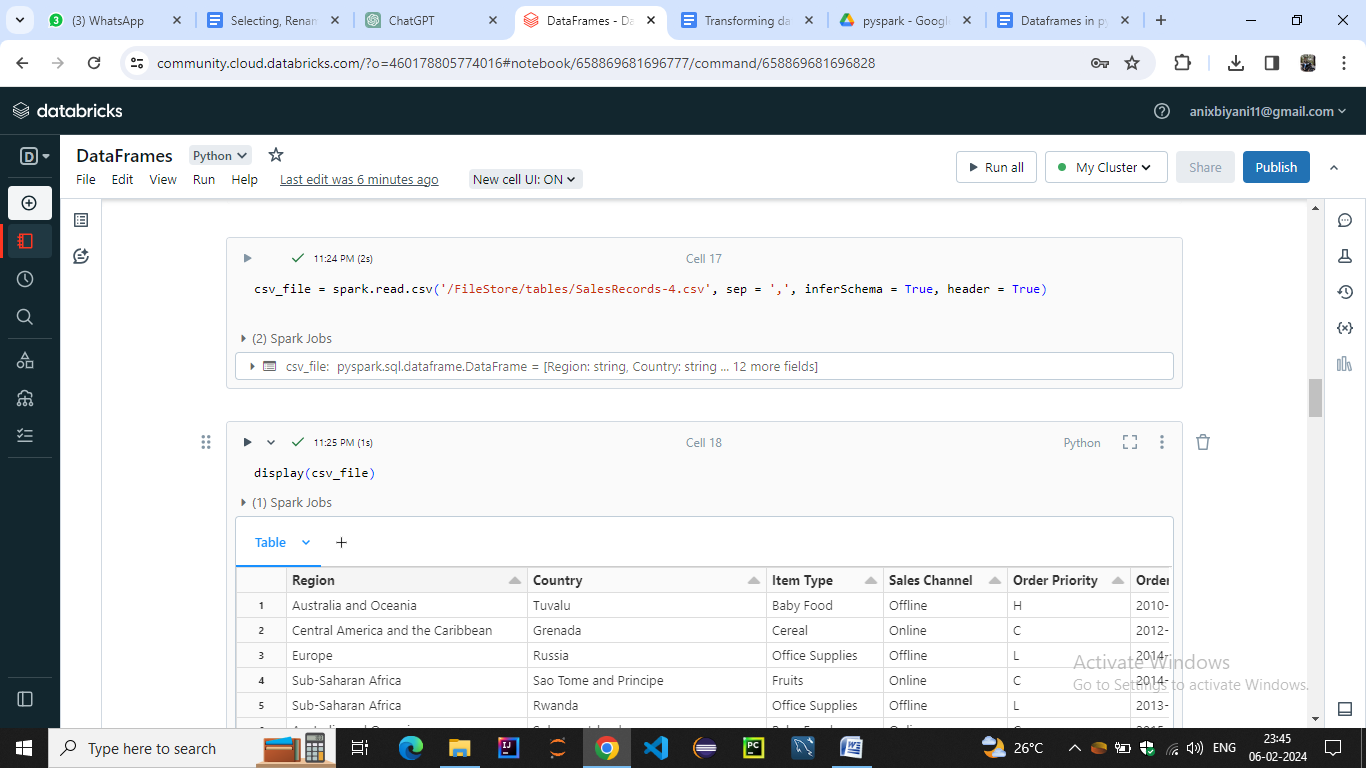




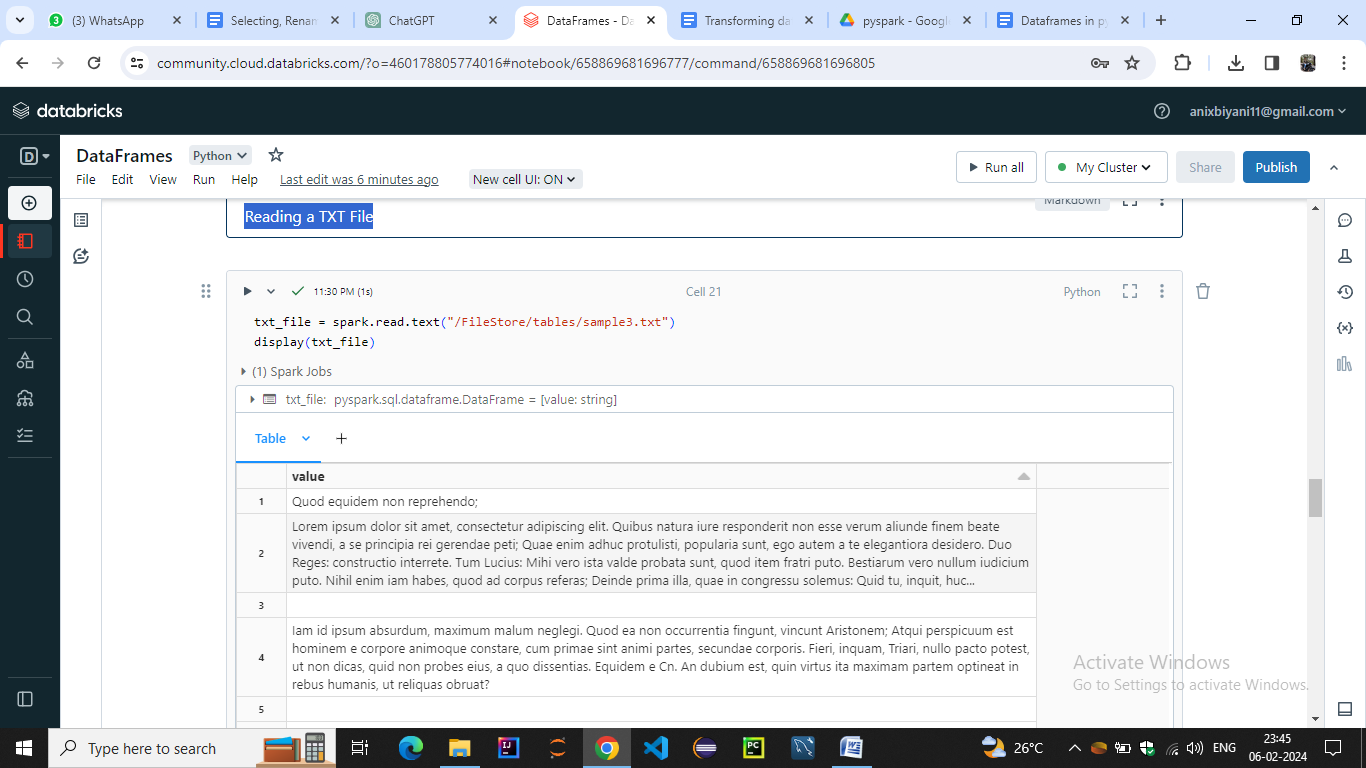
**Create PySpark DataFrame From an External File**

1. Reading a CSV File

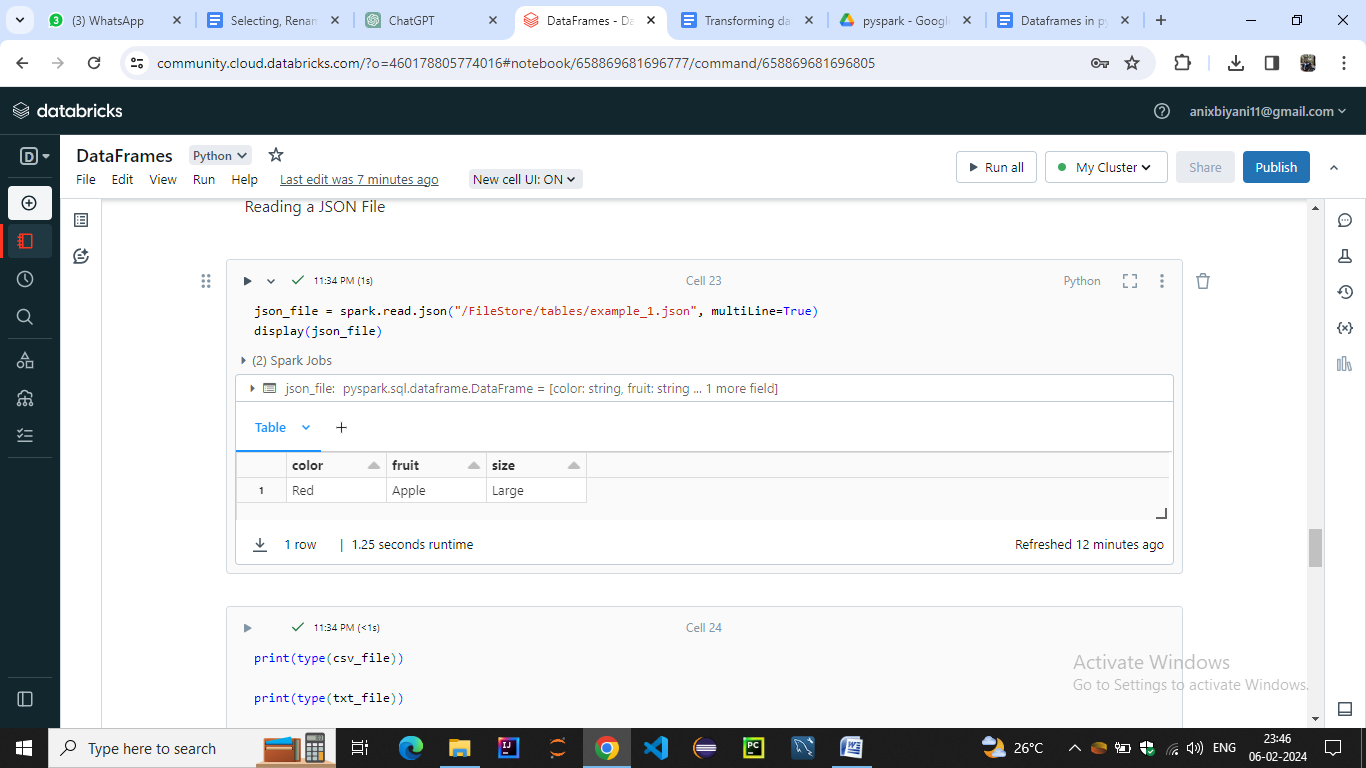




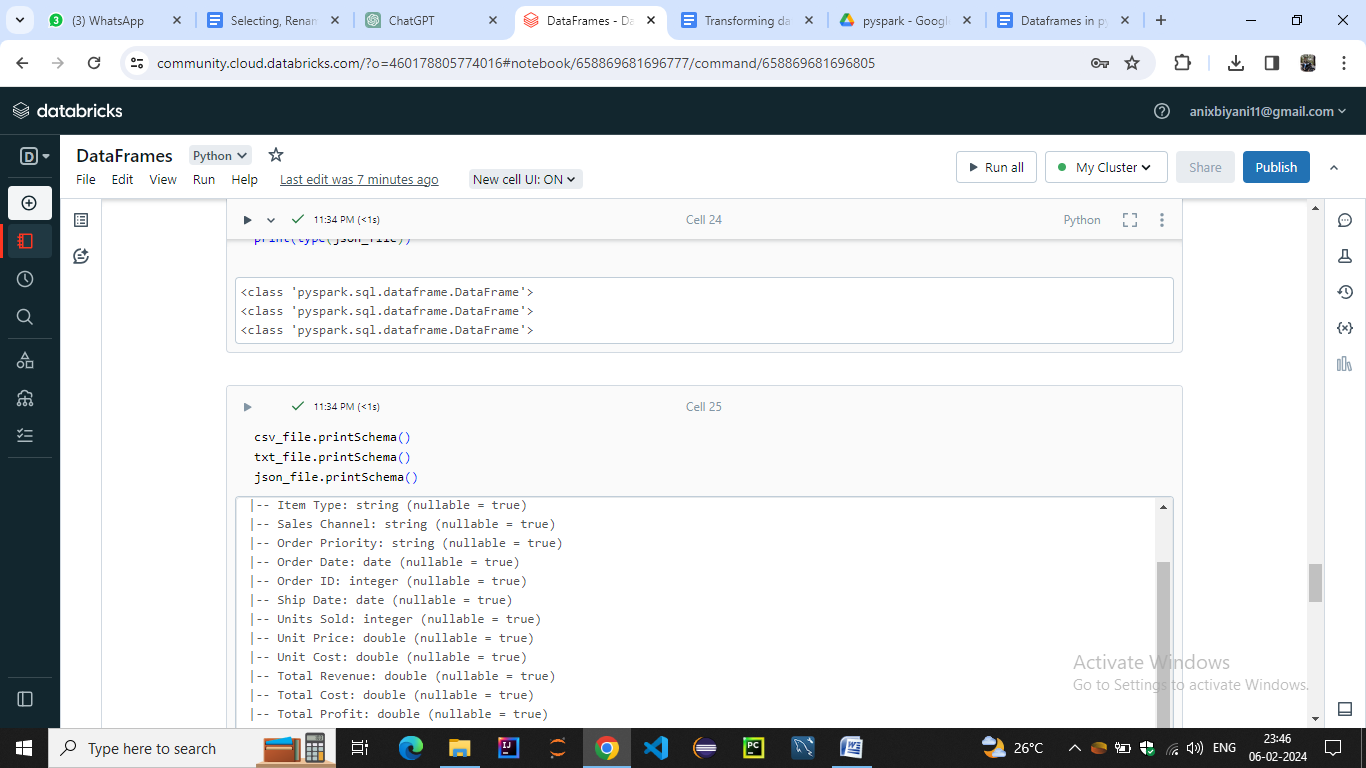
Reading a TXT File



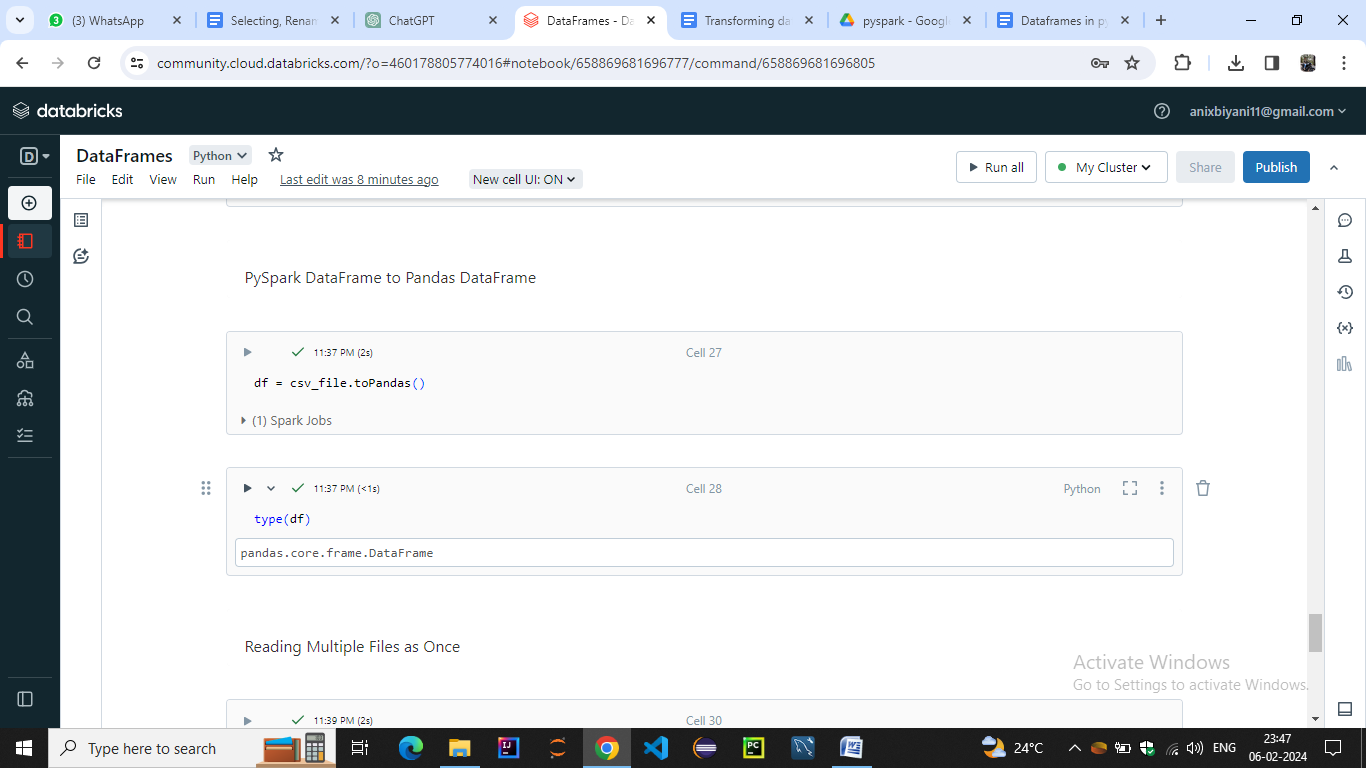
Reading a JSON File



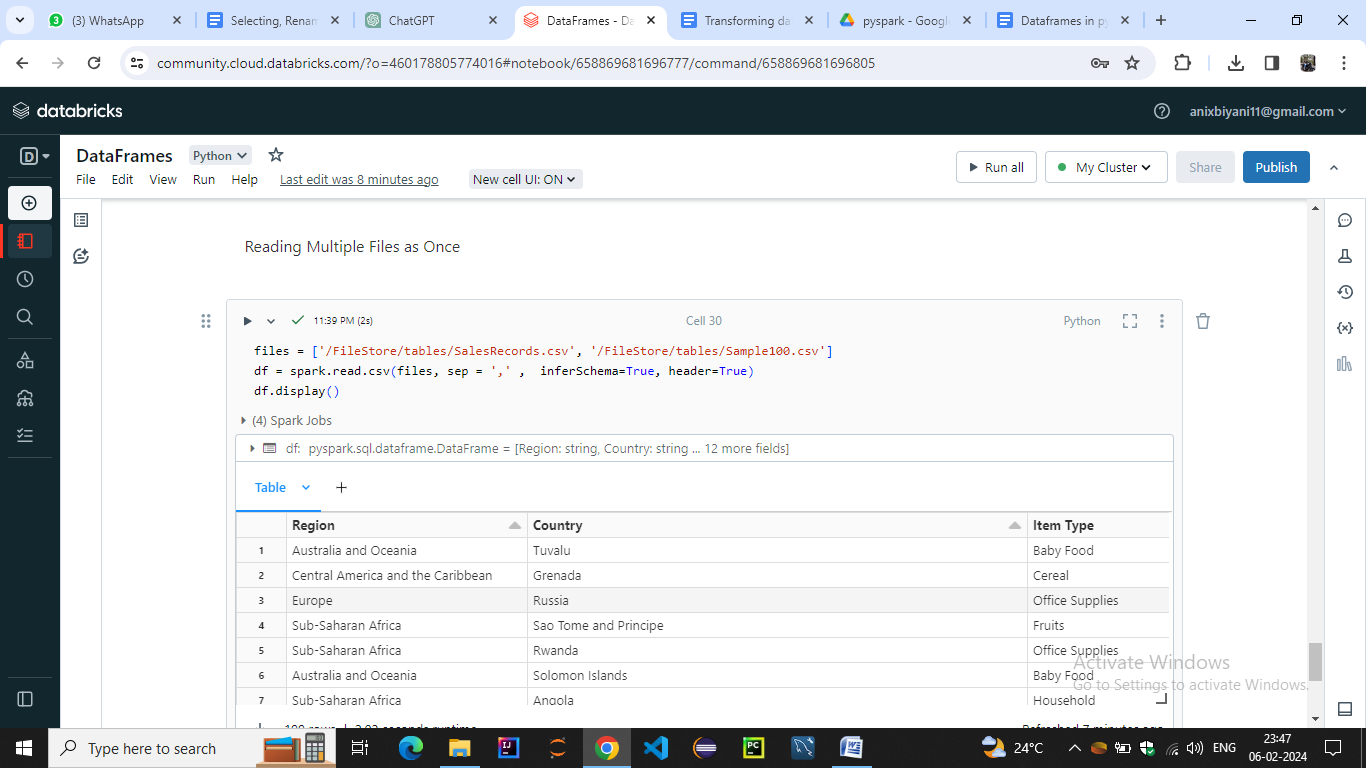
TYPE AND PRINTSCHEMA



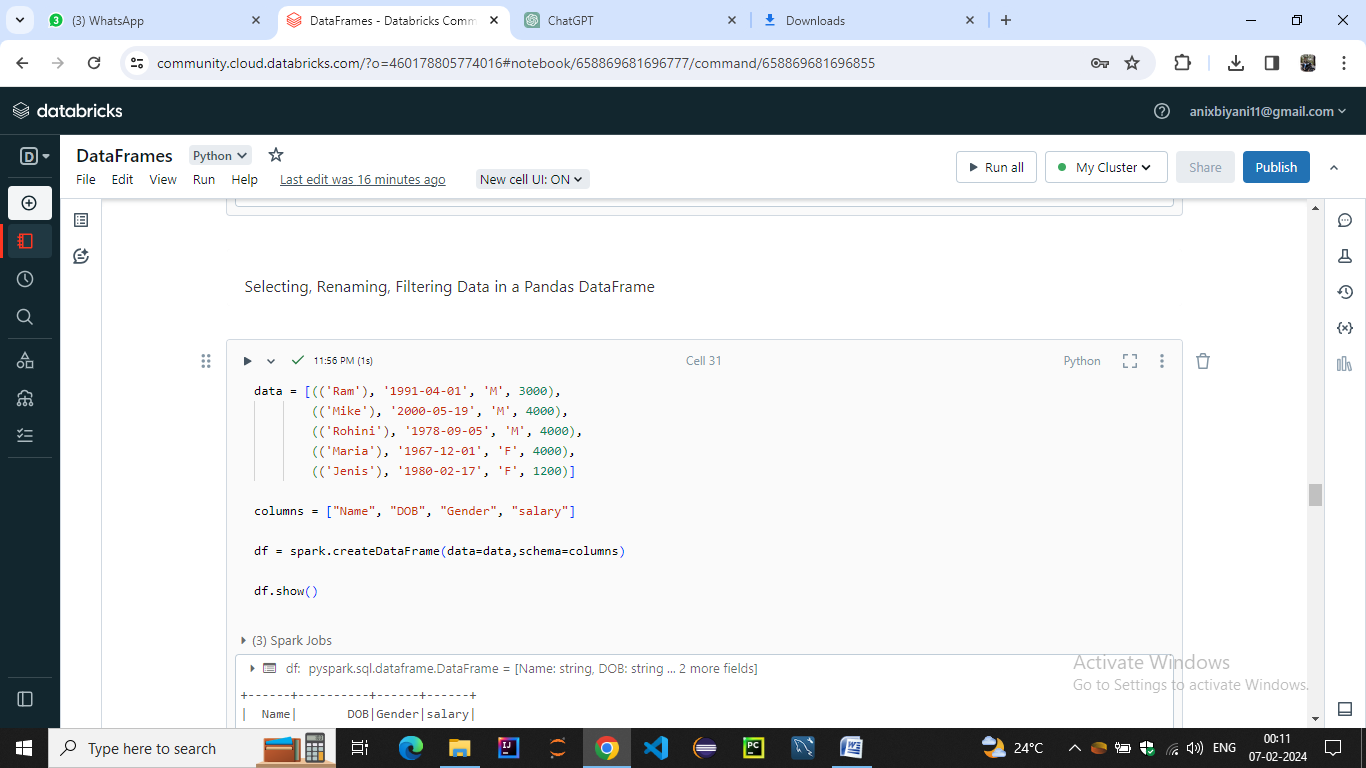
**PySpark DataFrame to Pandas DataFrame**

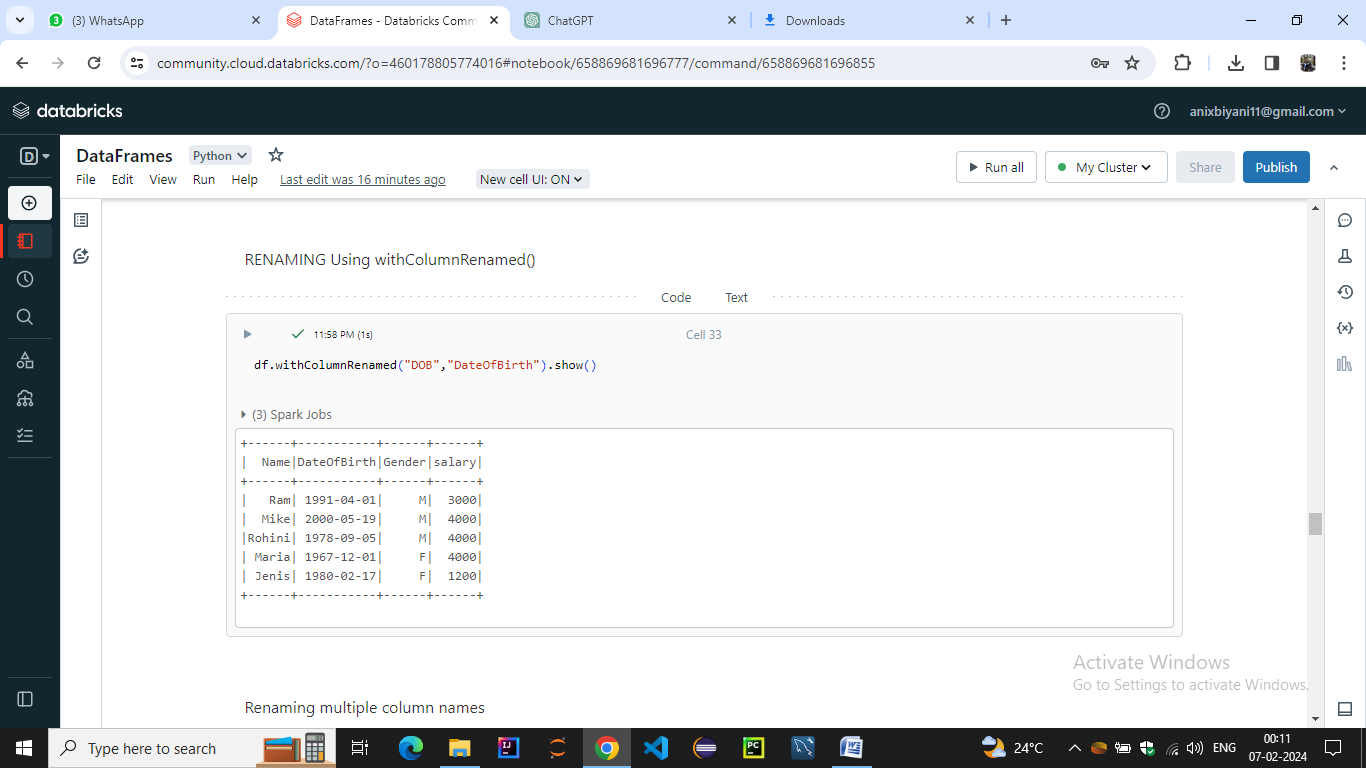


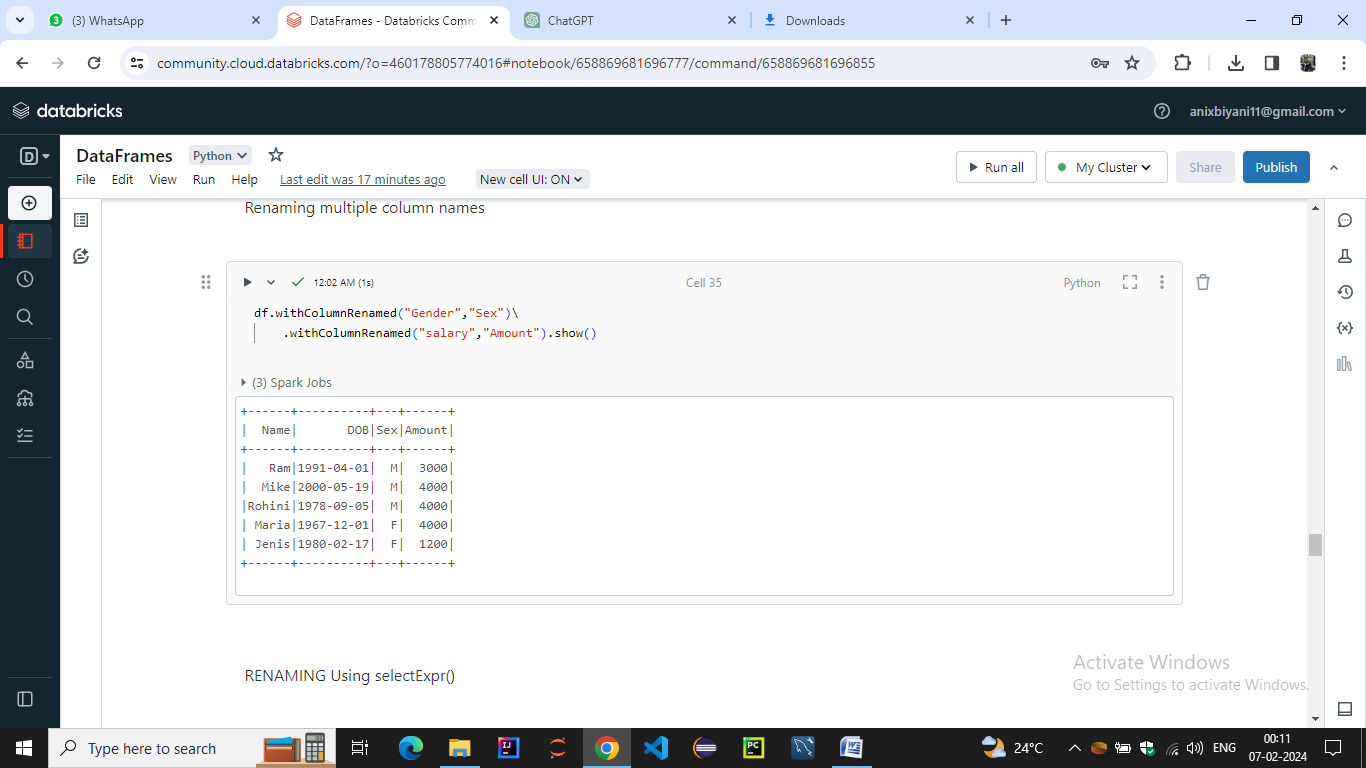
**Reading Multiple Files as Once**

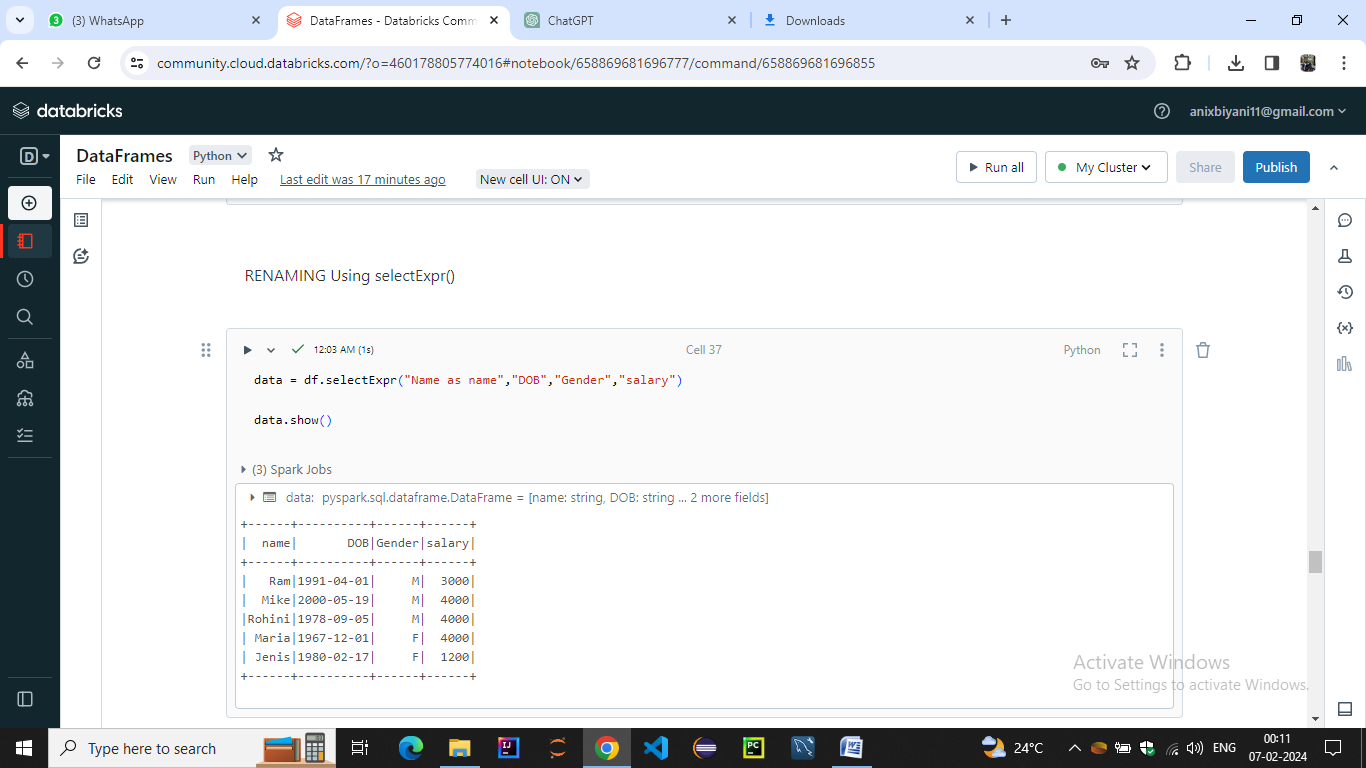


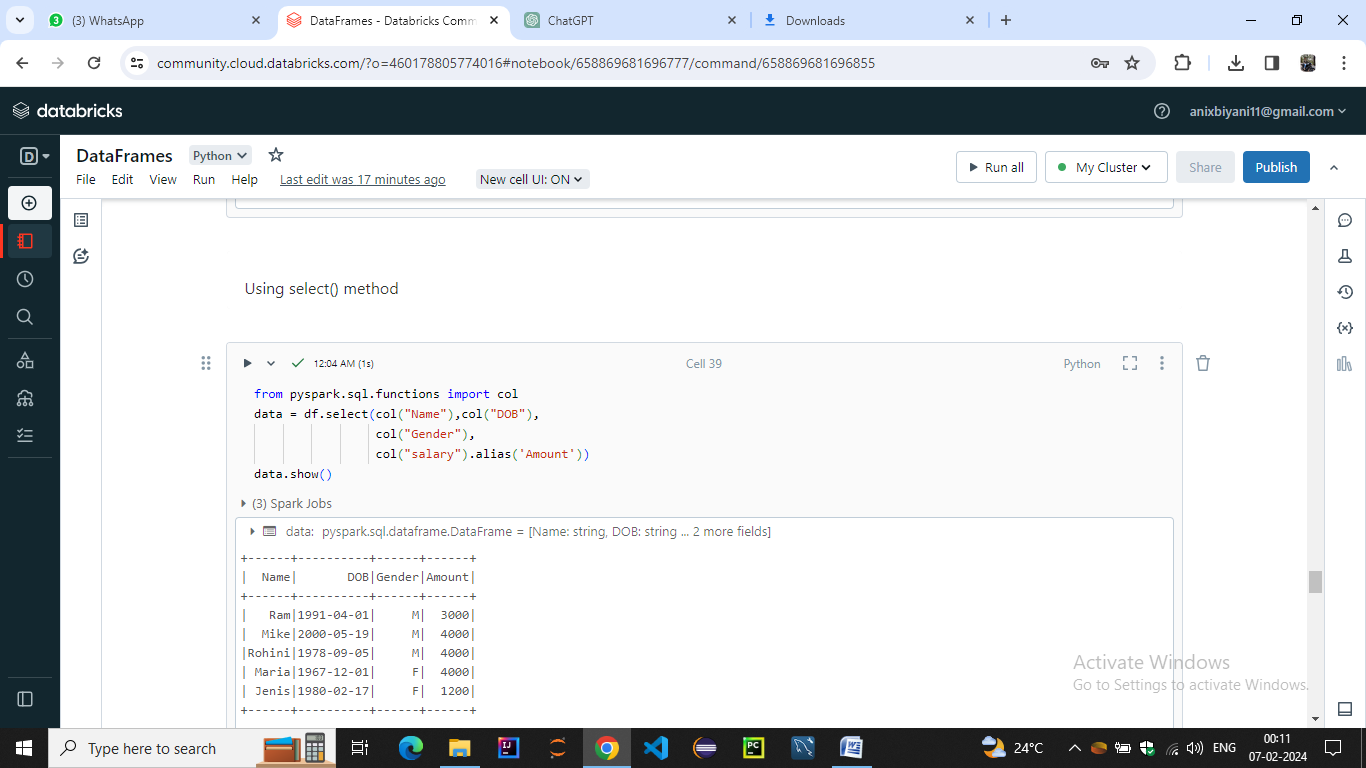
Selecting, Renaming, Filtering Data in a Pandas DataFrame

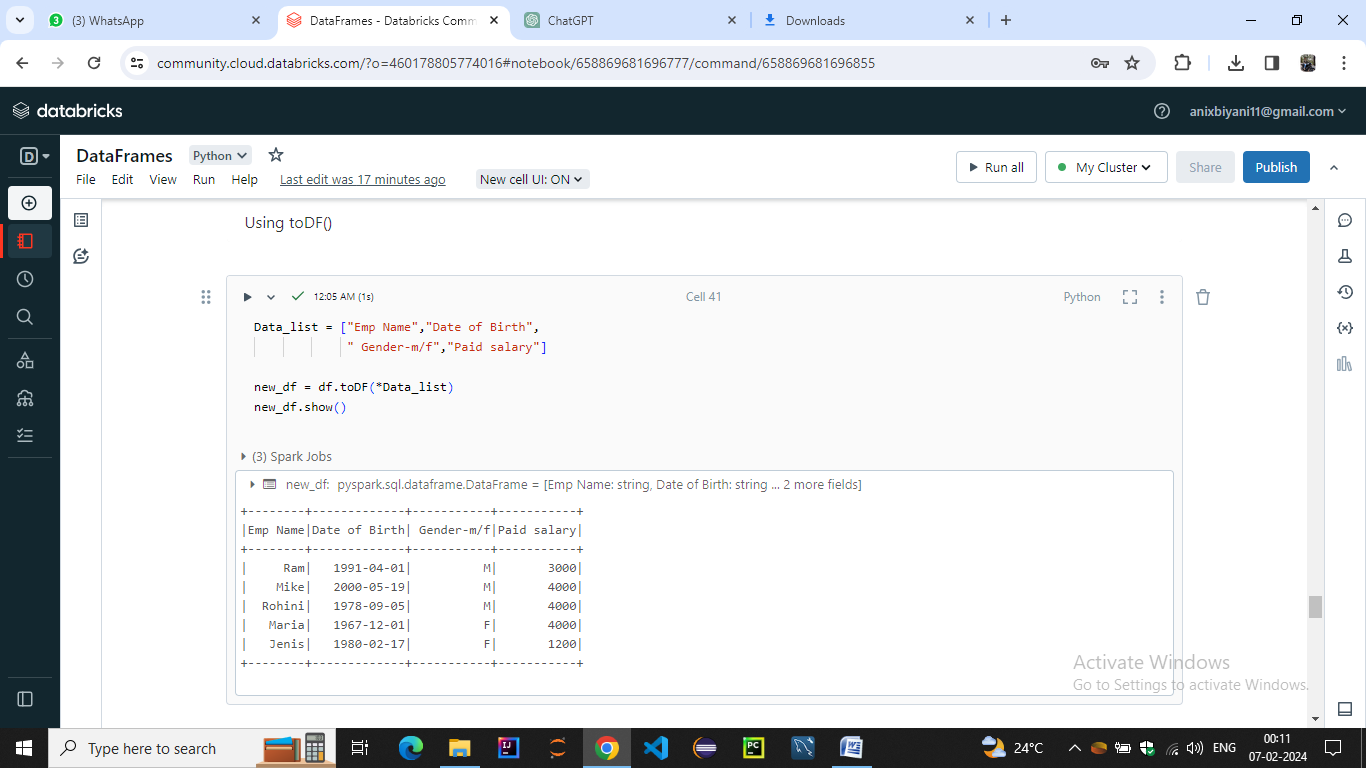




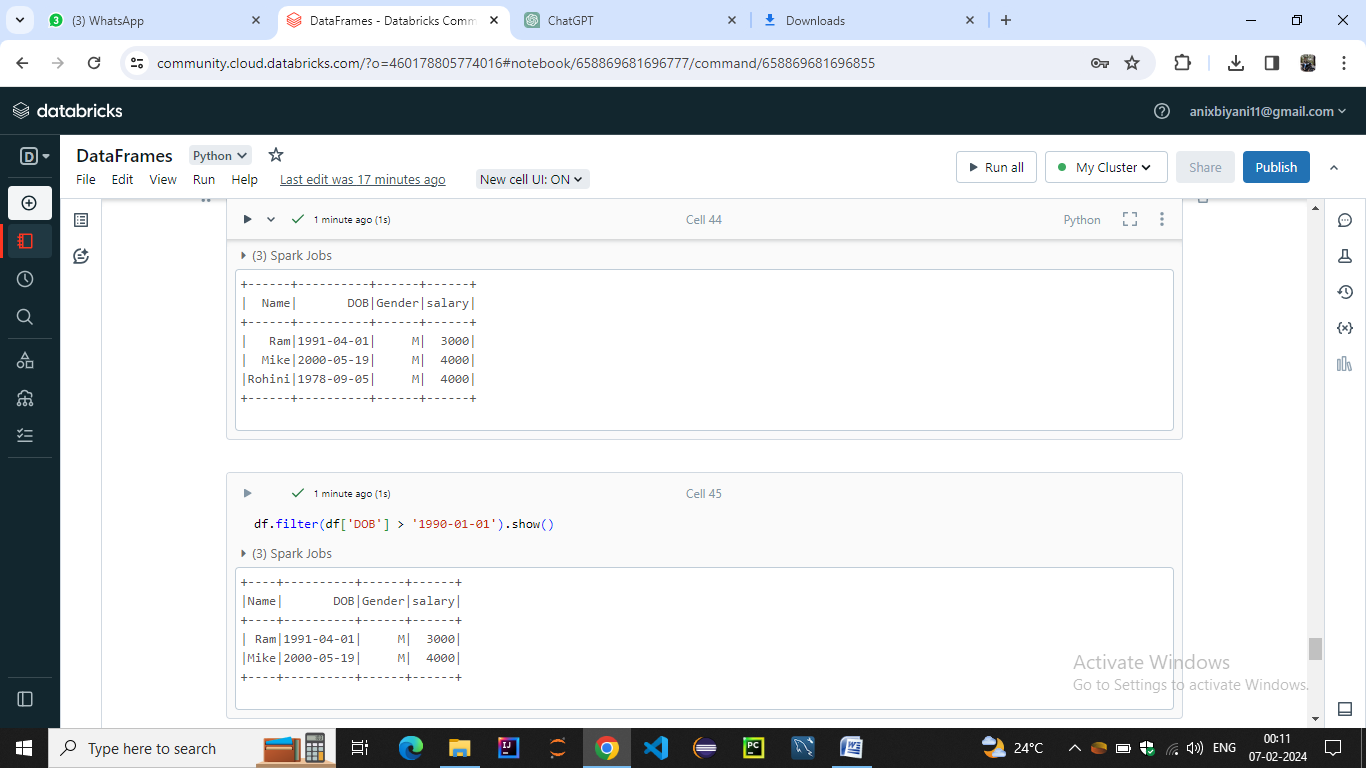


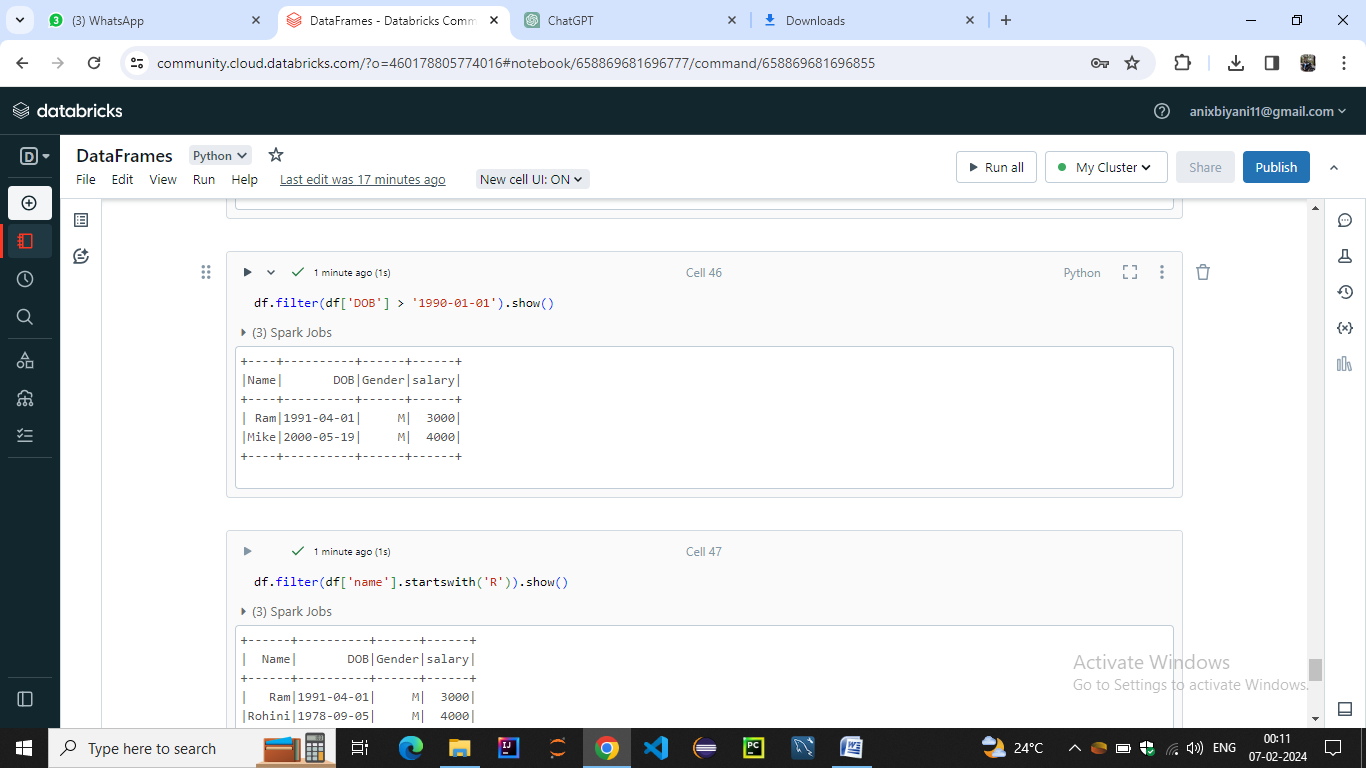


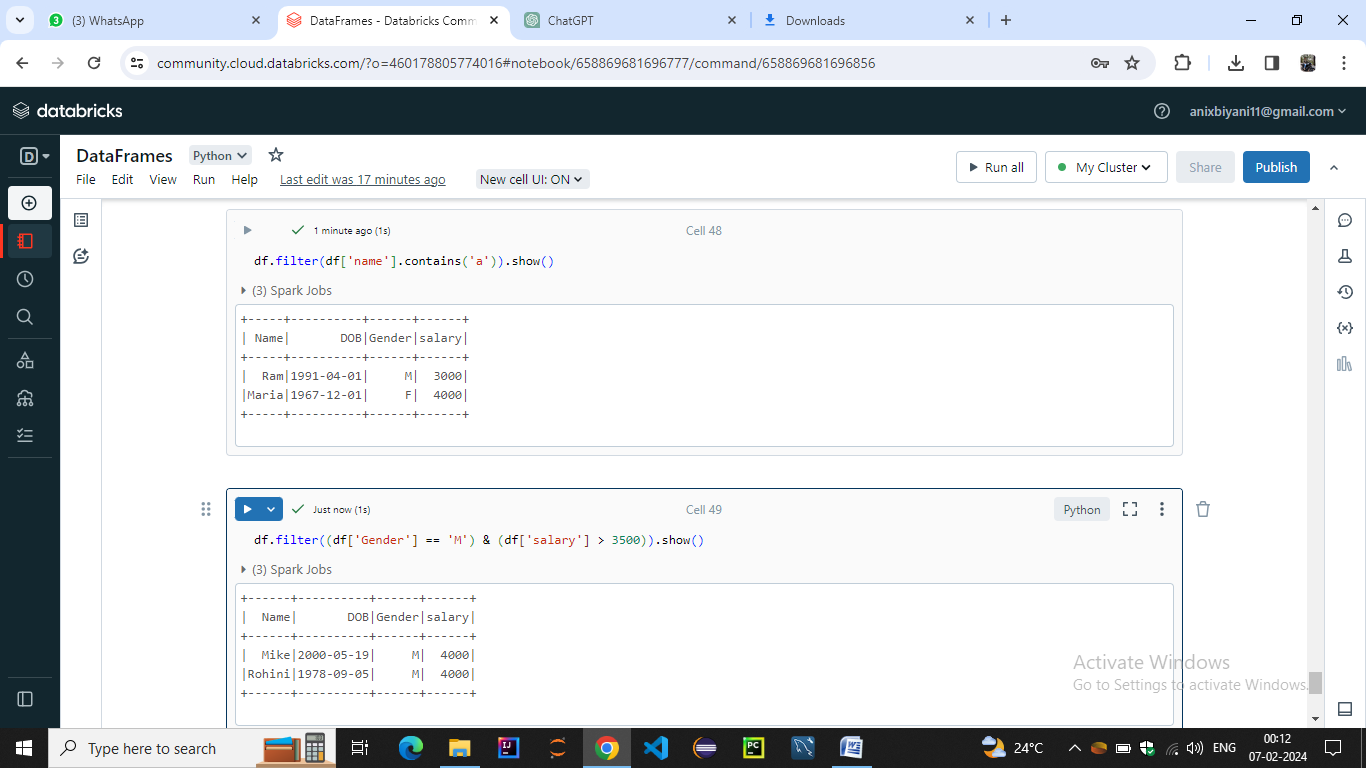












RDD TRANSFORMATIONS:-

