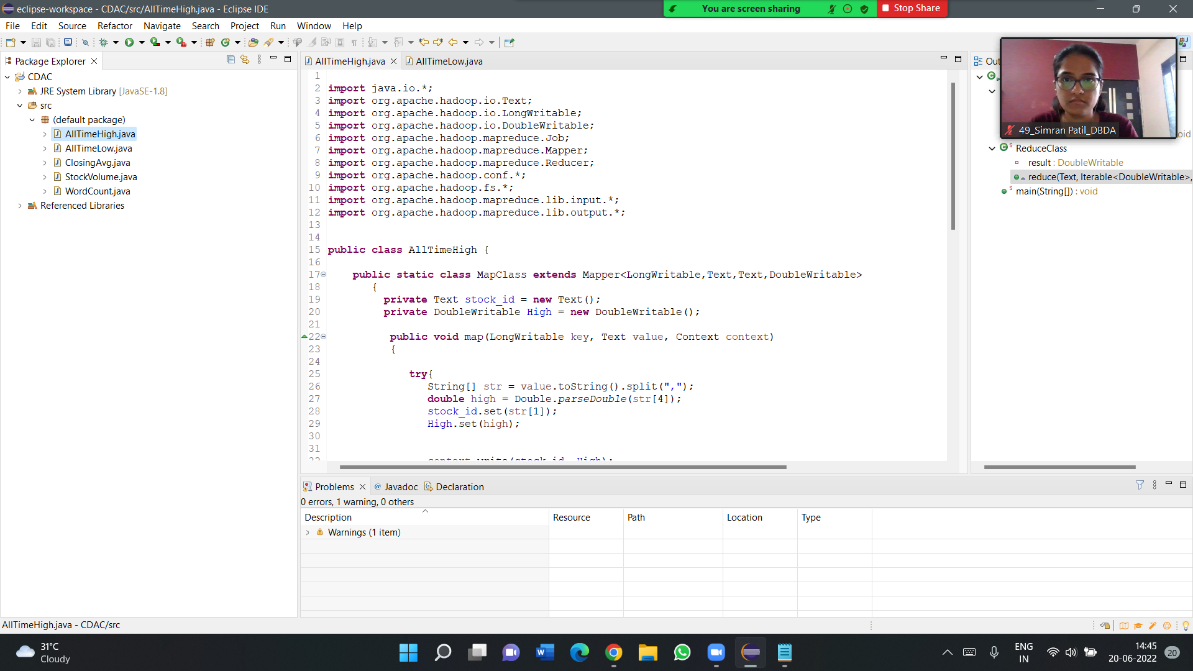
**BIGDATA MODULE END**

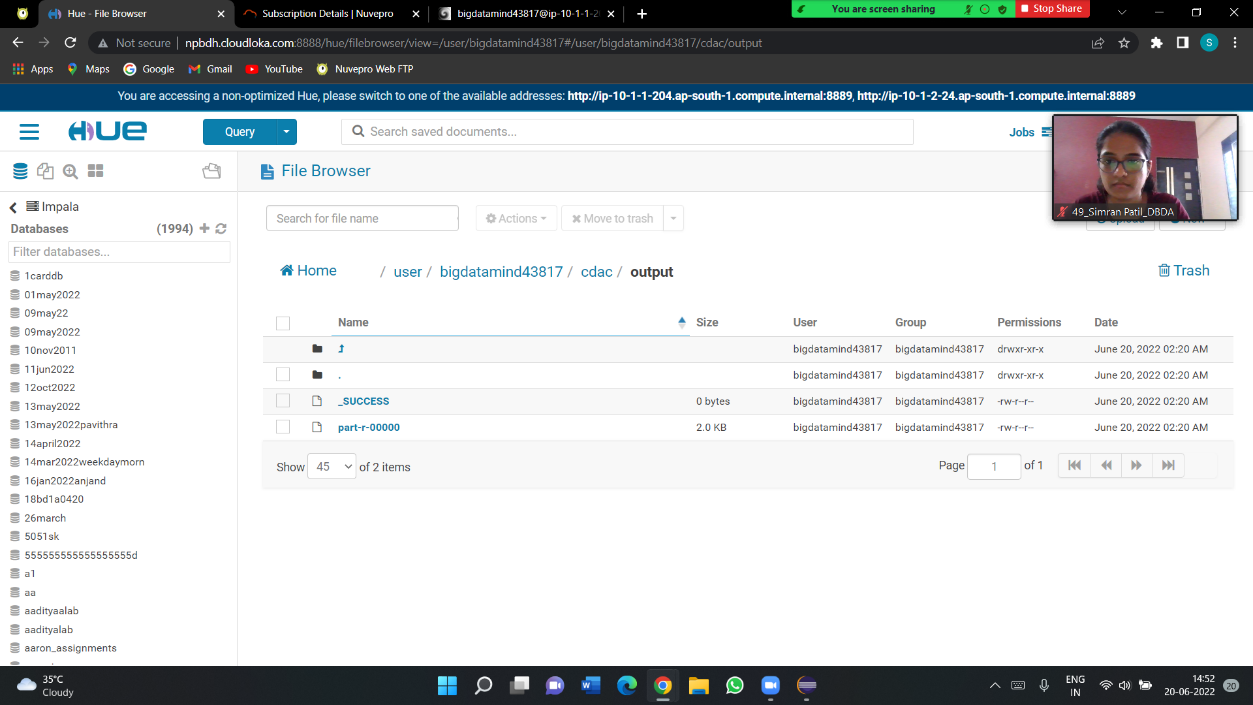
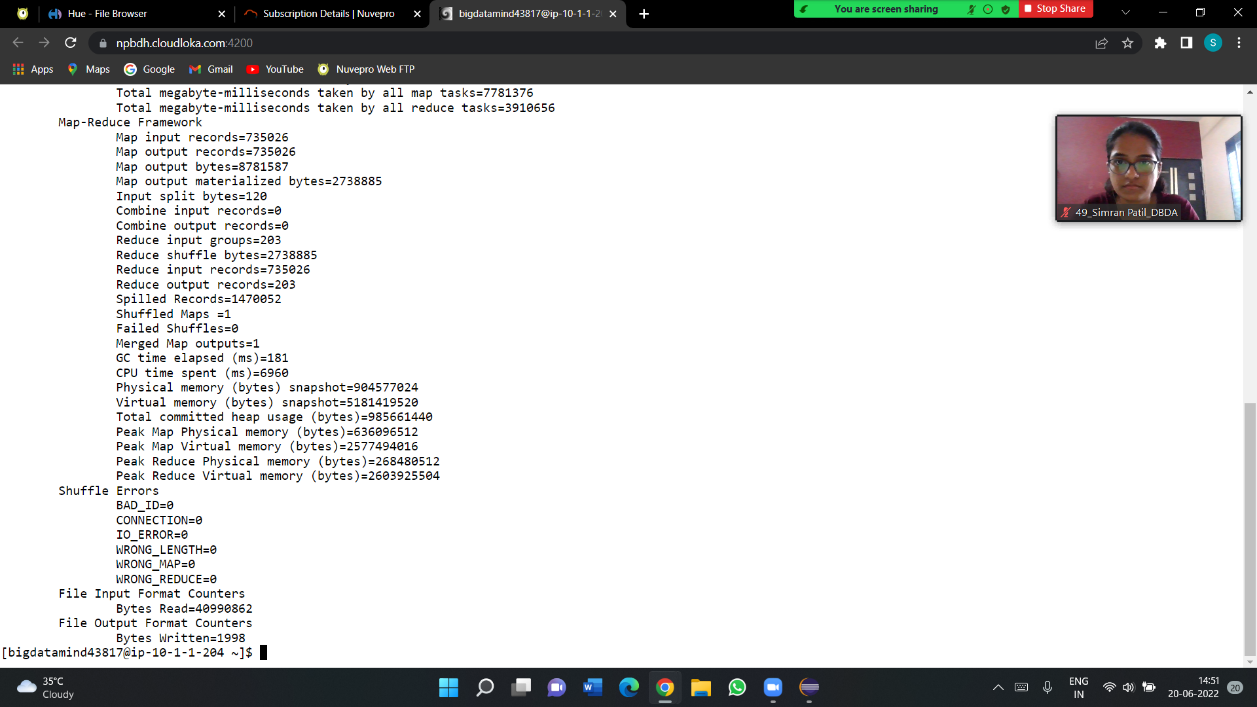
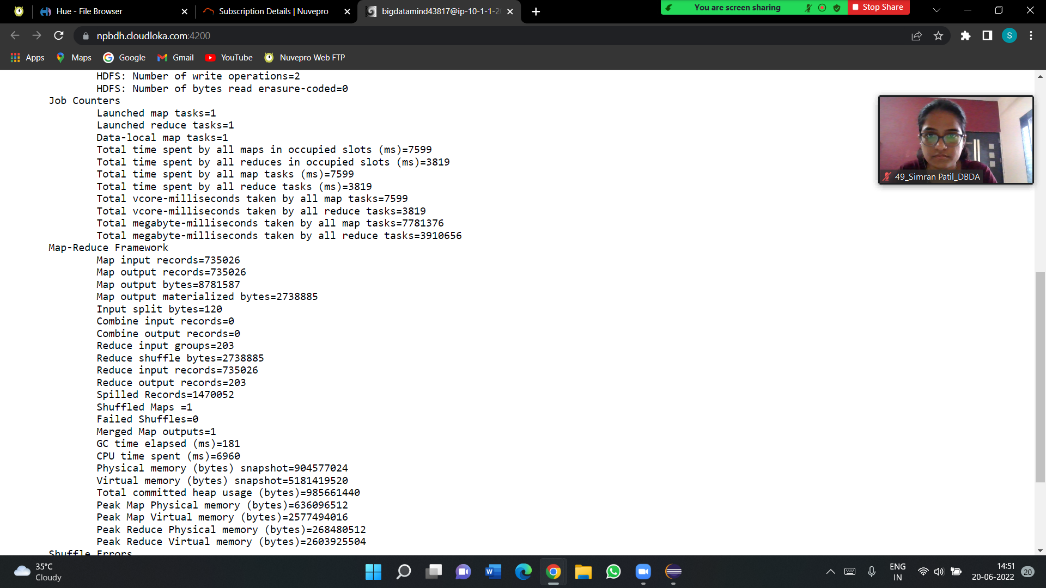
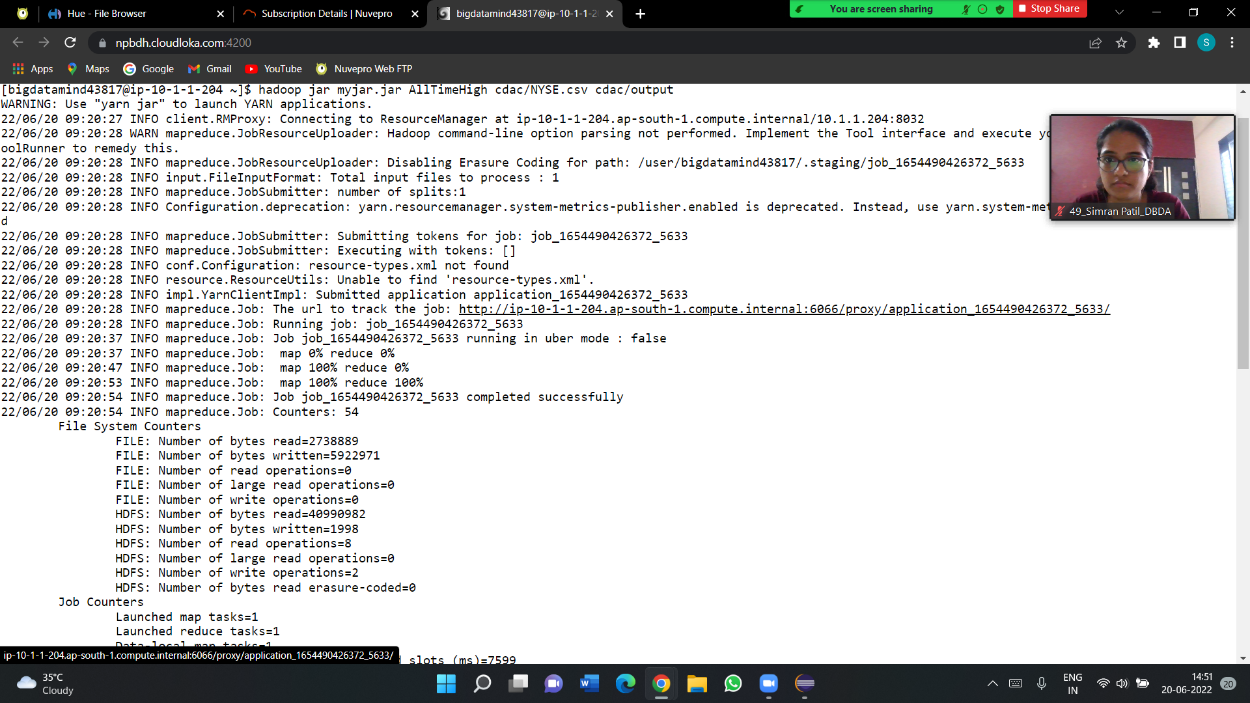
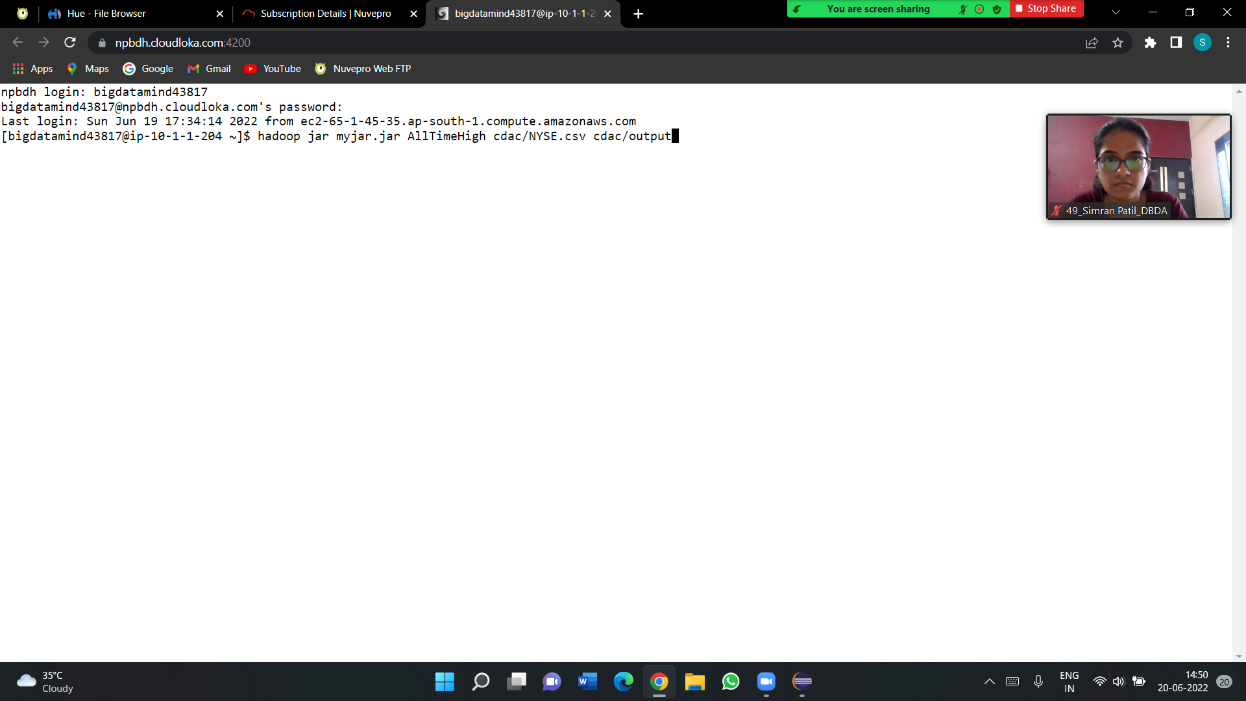
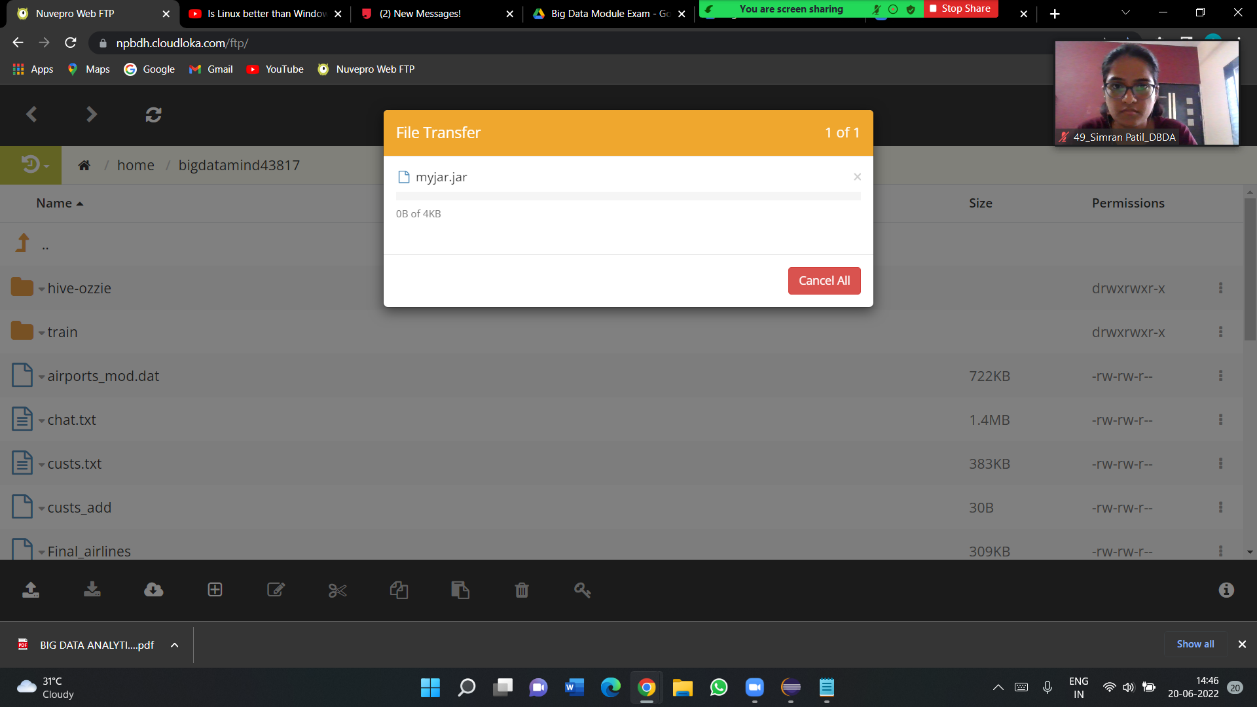
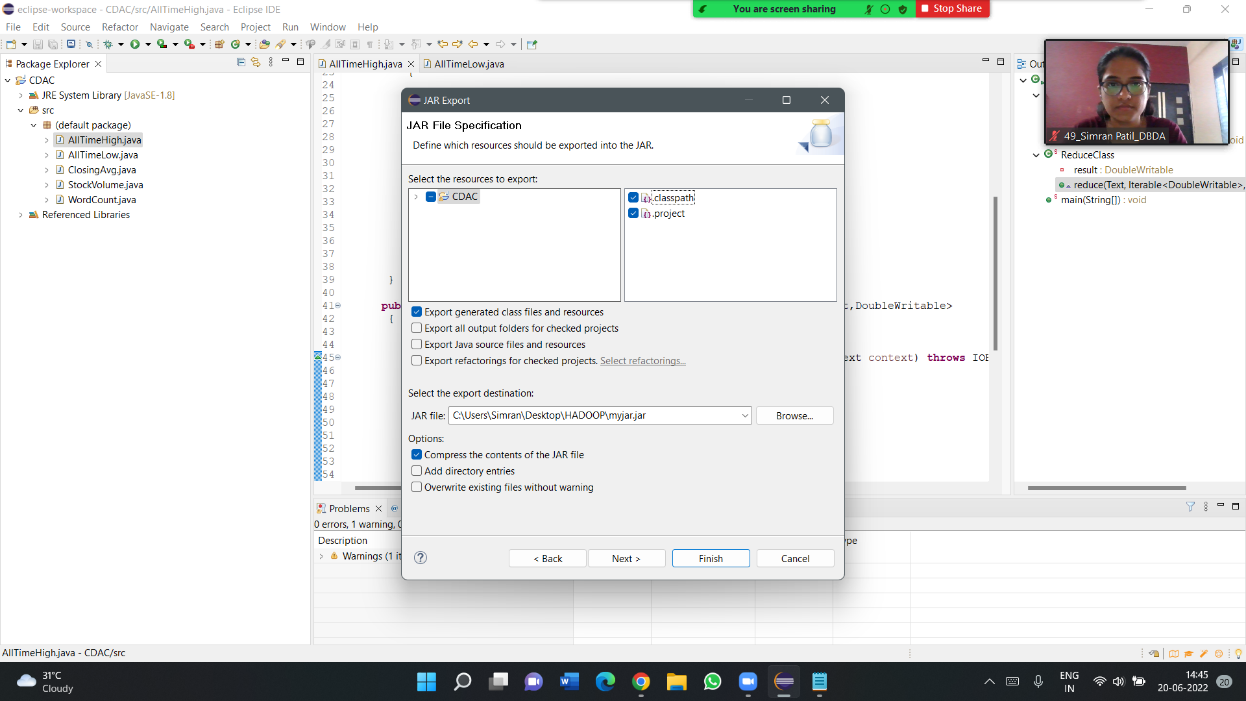
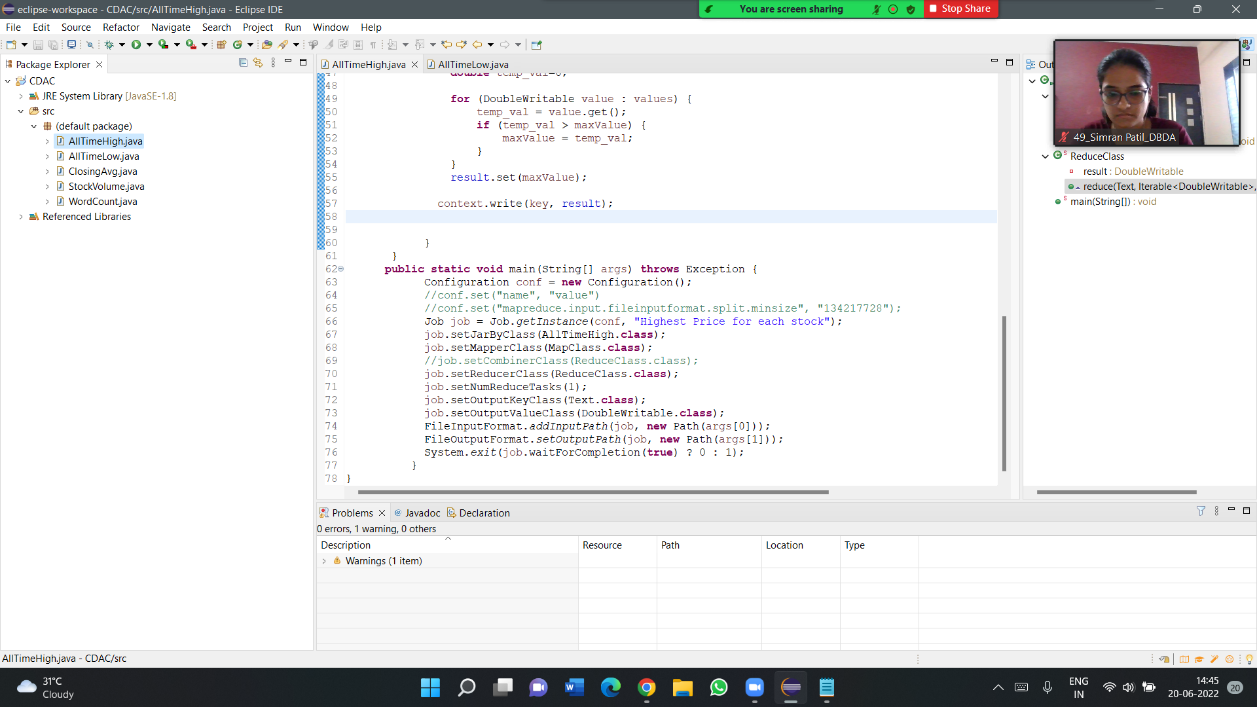
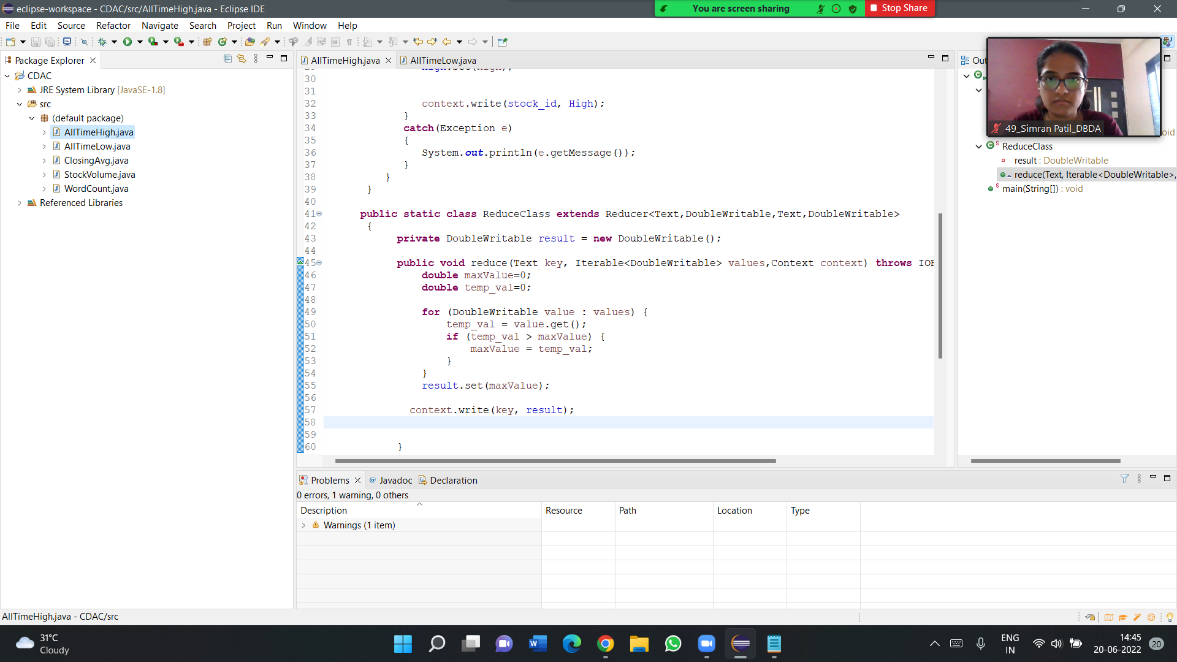
**Q1- MAPREDUCE**

Problem Statement

Here, we have chosen the stock market dataset on which we have performed map-reduce operations. Following is the structure of the data. Kindlyfind the solutions to the questions below. Data Structure 1. Exchange Name 2 Stock symbol 3. Transaction date 4. Opening price of the stock 5. Intra day high price of the stock 6. Intra day low price of the stock 7. Closing price of the stock 8. Total Volume of the stock on the particular day 9. Adjustment Closing price of the stock Field Separator – comma

Question 2 : Find all time High price for each stock

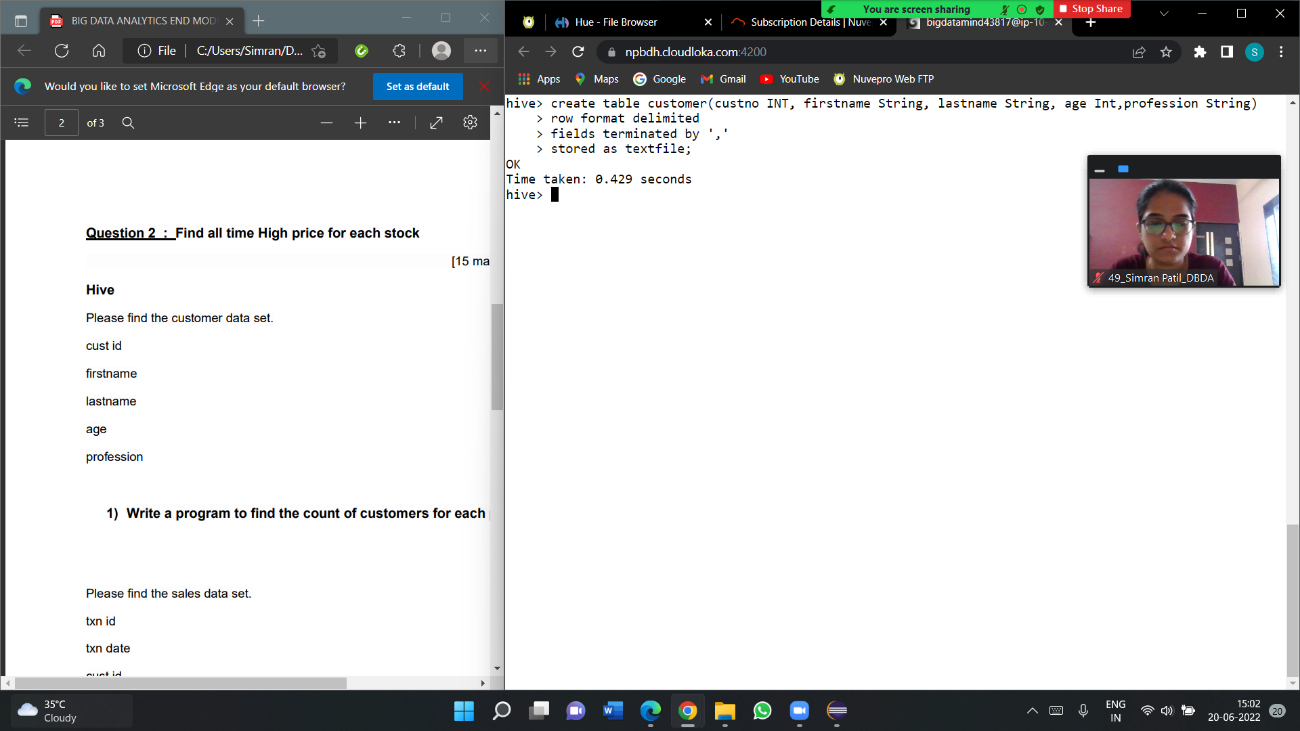




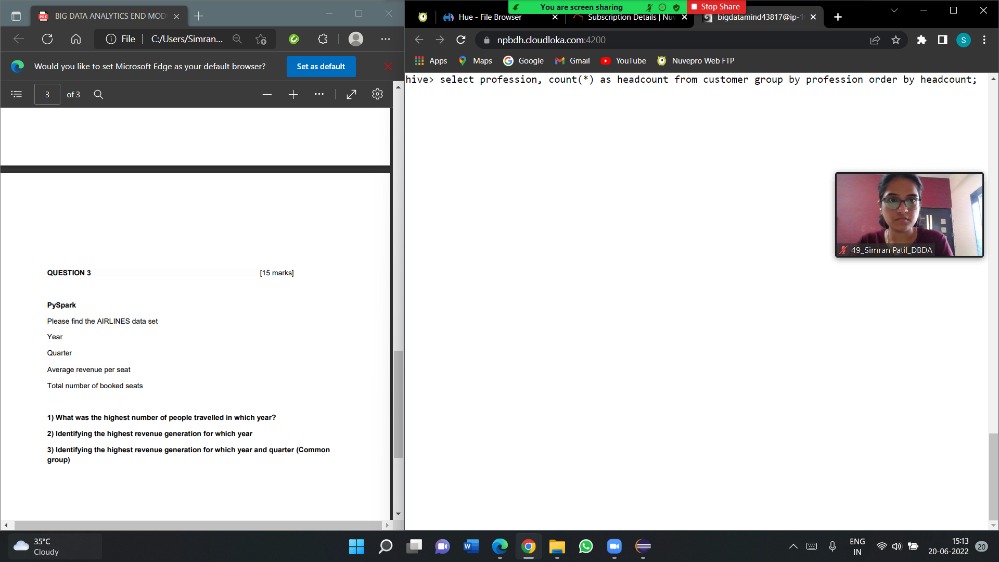


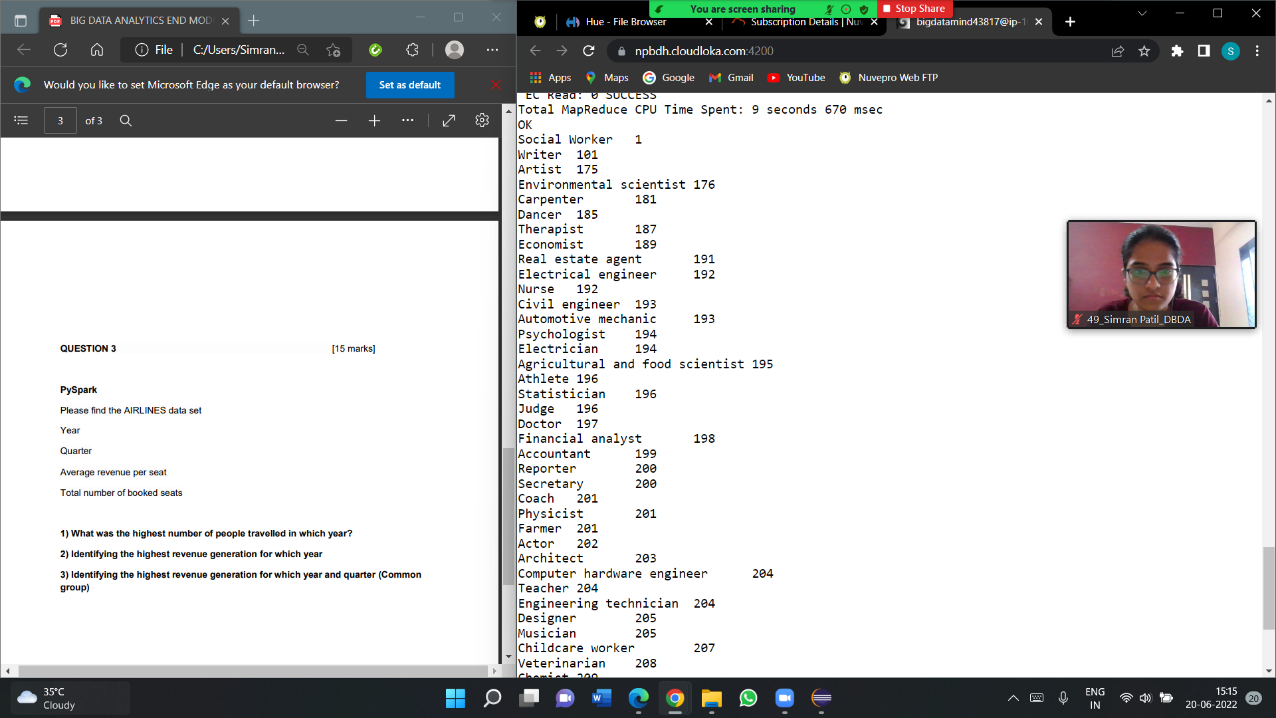
**Q2-Hive**

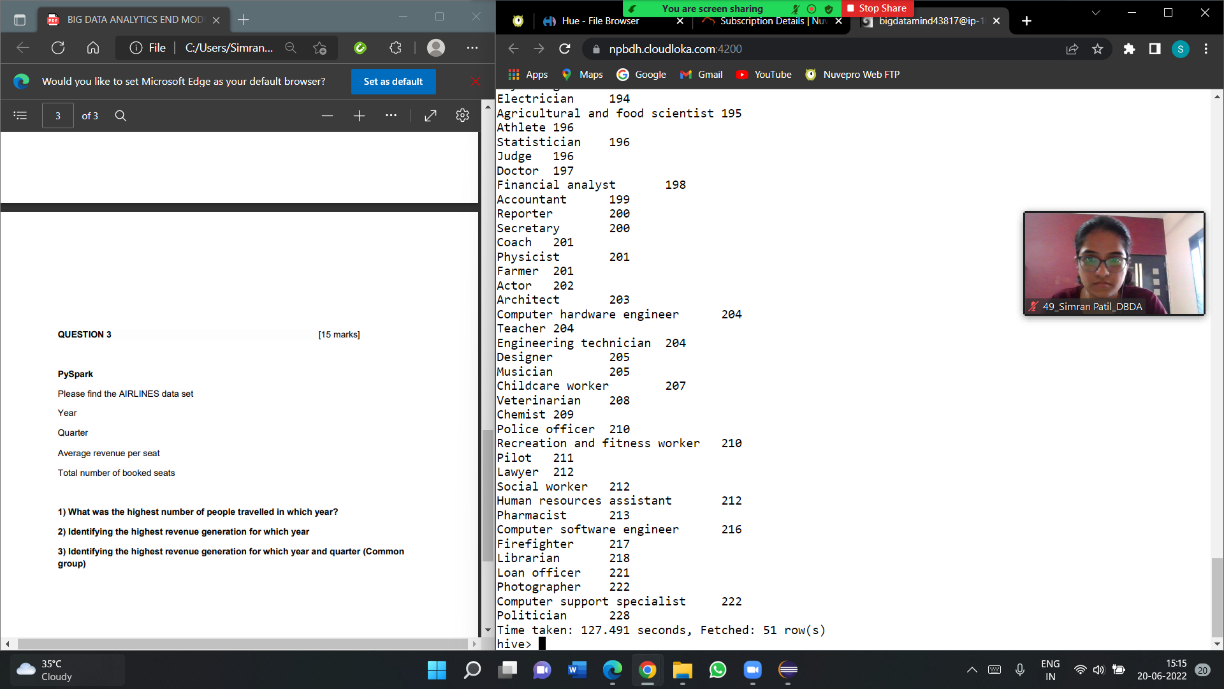
Please find the customer data set. cust id firstname lastname age profession



1. Write a program to find the count of customers for each profession.







==Please find the sales data set

txn id

txn date

cust id

amount

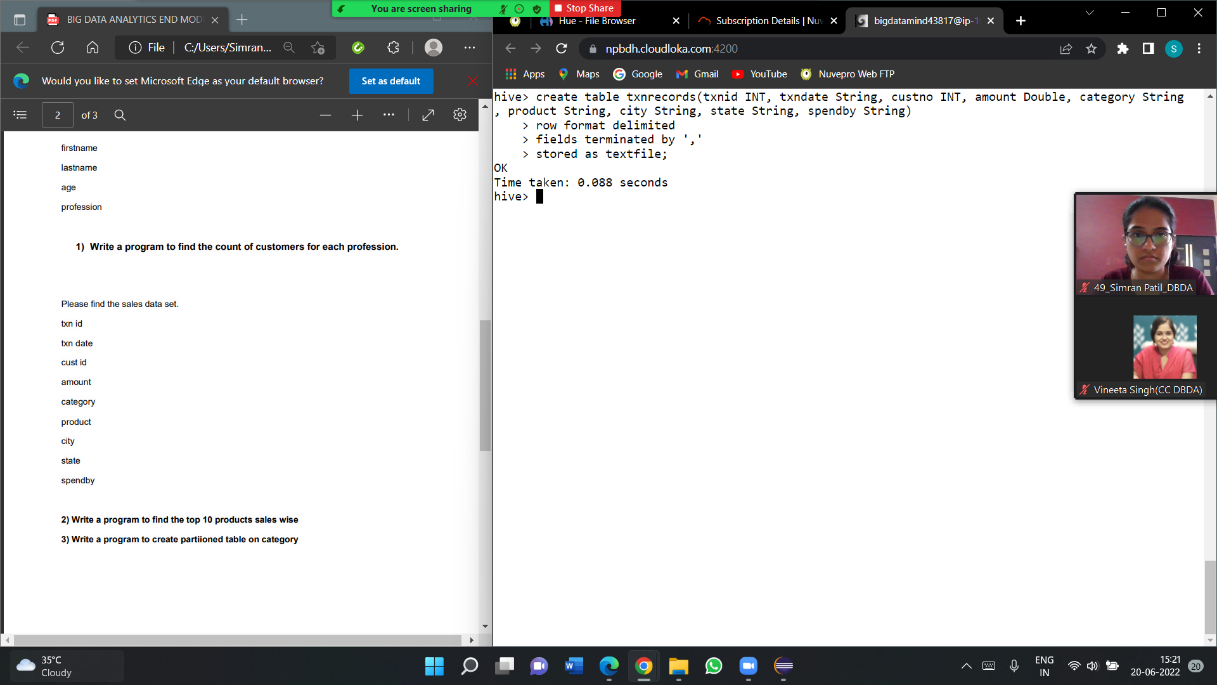
category

product

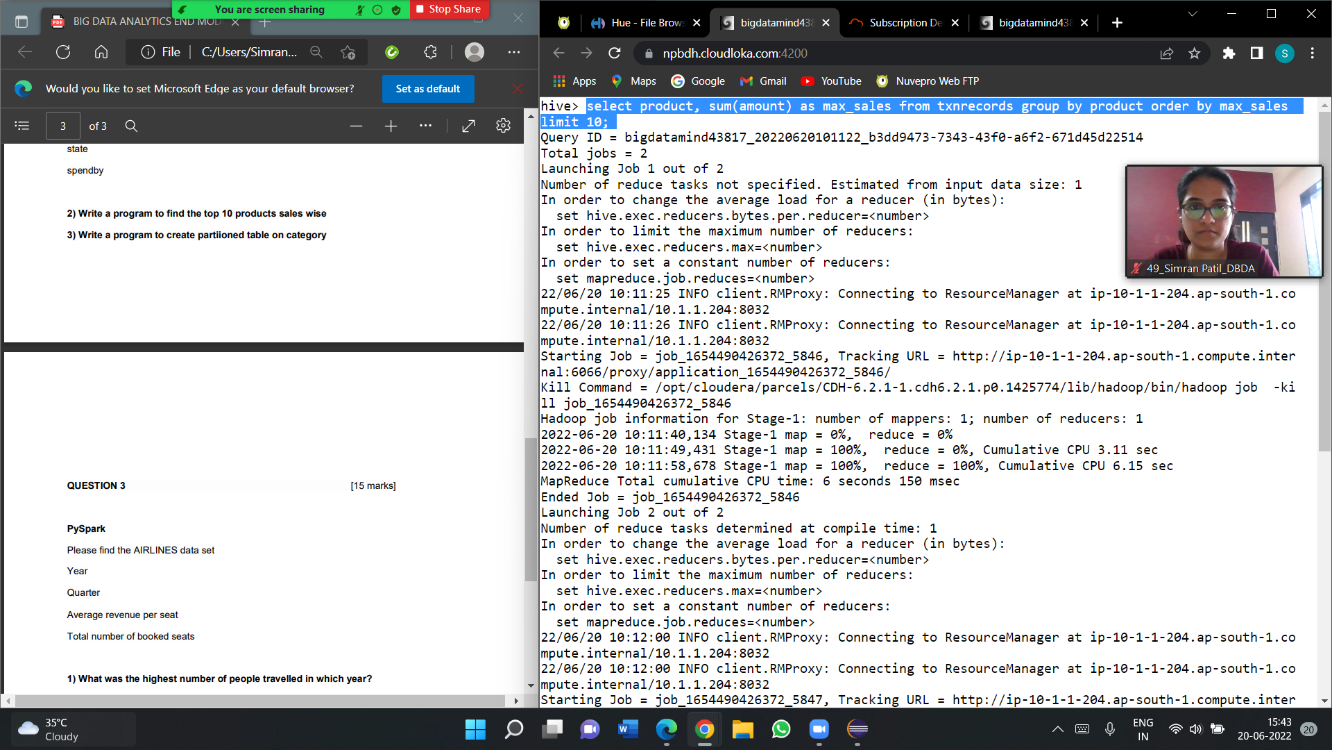
city

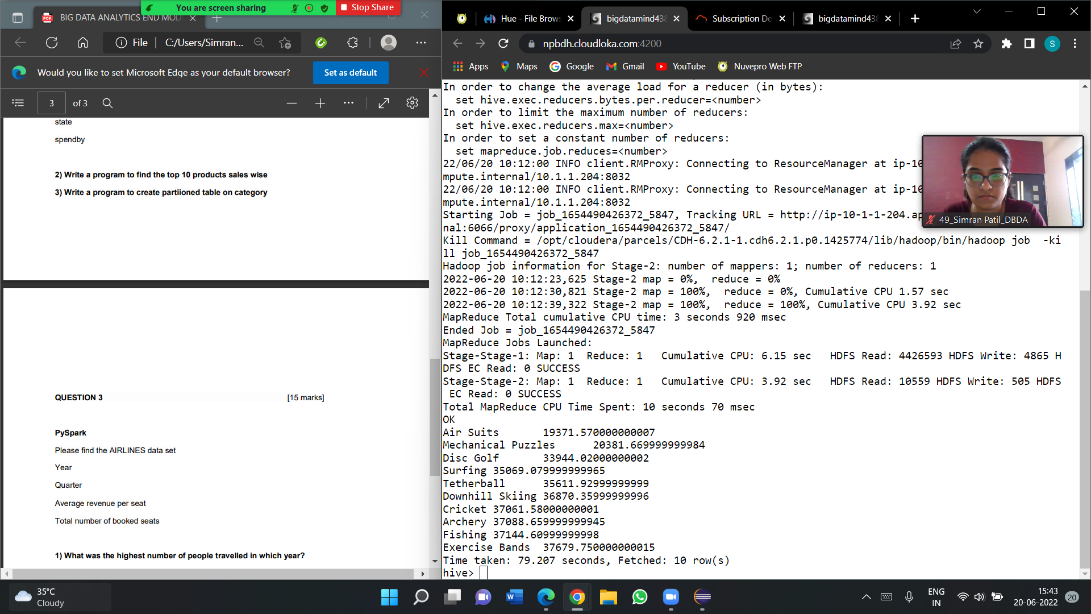
state

spendby

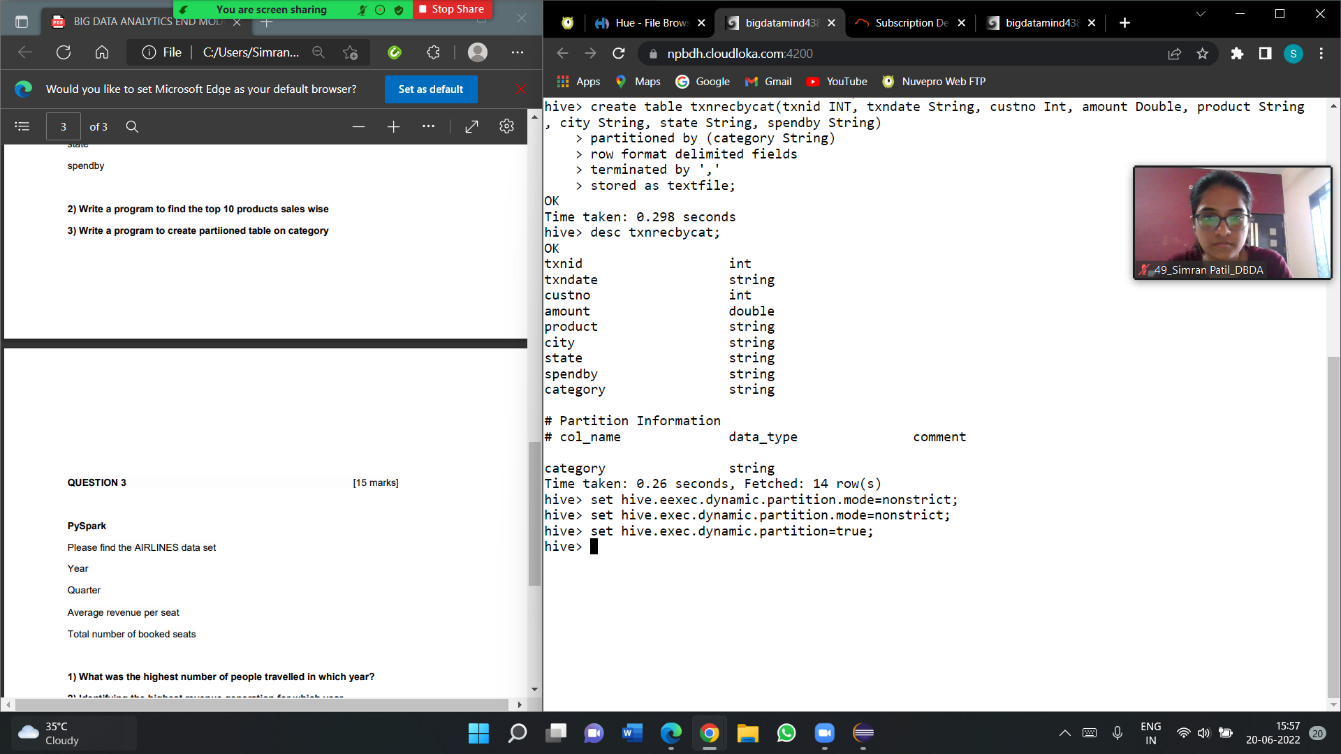
****

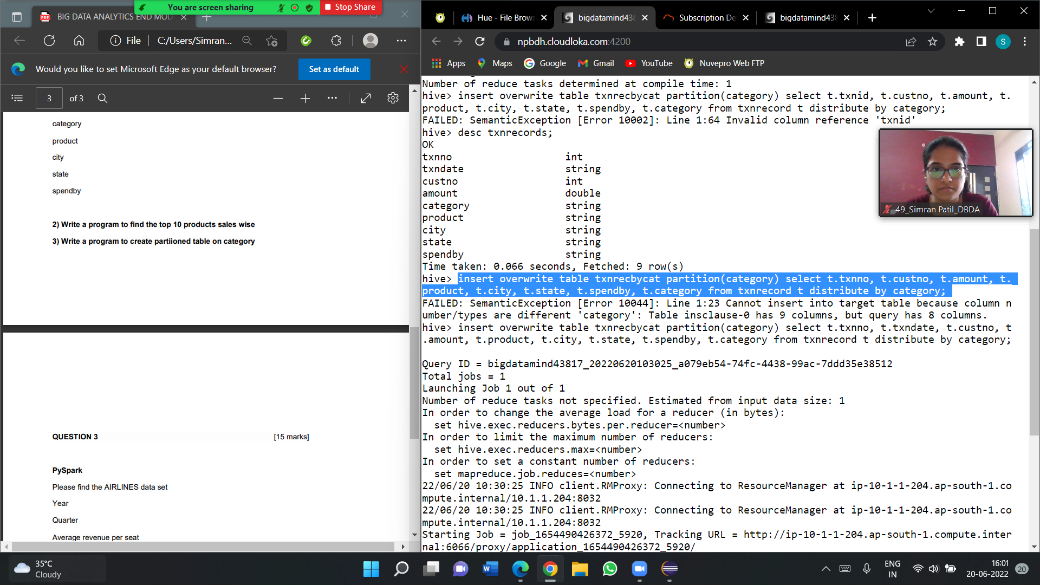
1. **Write a program to find the top 10 products sales wise**

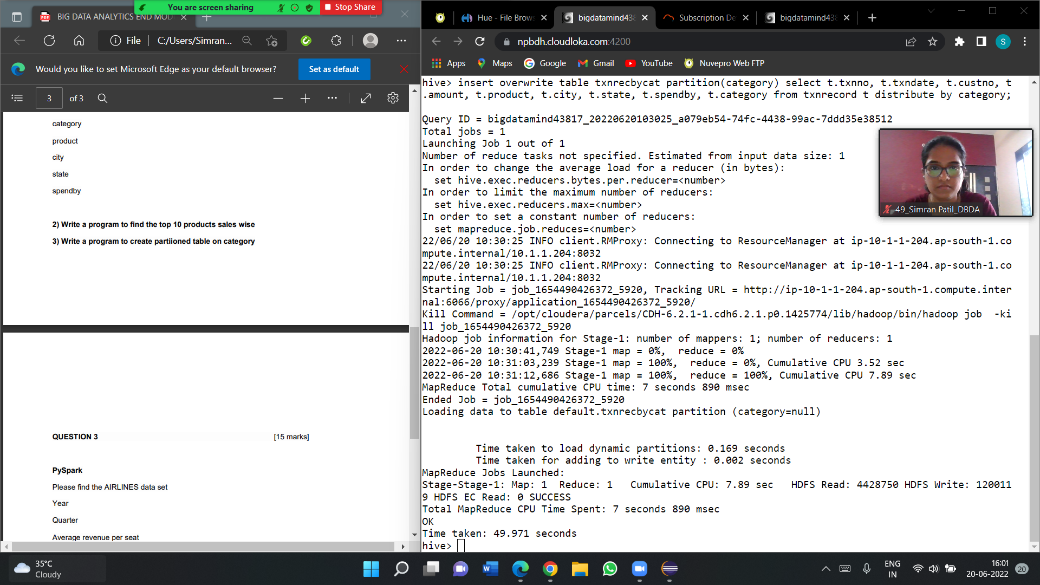
****

****

1. **Write a program to create partiioned table on category**

****

****

****

QUESTION 3

PySpark

Please find the AIRLINES data set

Year

Quarter

Average revenue per seat

Total number of booked seats

1) What was the highest number of people travelled in which year?

from pyspark.sql.types import StructType, StringType, DoubleType, IntegerType,LongType

>>> schema1 = StructType().add("year",StringType(),True).add("qtr",

StringType(),True).add("arps",DoubleType(),True).add("booked\_seats",

LongType(),True)

>>> airlineDF = spark.read.format("csv").option("header","TRUE").schema

(schema1).load("/user/bigdatamind43823/cdacexam/airlines.csv")

>>> airlineDF.registerTempTable("airlines")

>>> highpass = spark.sql("select year, sum(booked\_seats) as totalpassengers from airlines group by year order by totalpassengers desc")

>>> highpass.show()

+----+---------------+

|year|totalpassengers|

+----+---------------+

|2007| 176299|

|2013| 173676|

|2001| 173598|

|1996| 167223|

|2008| 166897|

|2012| 166076|

|2015| 165438|

|2004| 164800|

|2010| 163741|

|2014| 159823|

|1997| 157972|

|2003| 156153|

|2000| 154376|

|2006| 153789|

|2002| 152195|

|2005| 150610|

|2009| 150308|

|1999| 150000|

|1995| 148520|

|2011| 142647|

+----+---------------+

only showing top 20 rows

2) Identifying the highest revenue generation for which year

highrev = spark.sql("select year, sum(arps\*booked\_seats)/1000000 as rev\_in\_millions from airlines group by year order by rev\_in\_millions desc limit 1")

>>> highrev.show()

+----+---------------+

|year|rev\_in\_millions|

+----+---------------+

|2013| 66.36320871|

+----+---------------+

3) Identifying the highest revenue generation for which year and quarter (Common

group)

highrev = spark.sql("select (year,qtr) as period, sum(arps\*booked\_seats)/1000000 as rev\_in\_million from airlines group by period order by rev\_in\_million desc limit 1")

>>> highrev.show()

+---------+--------------+

| period|rev\_in\_million|

+---------+--------------+

|[2014, 4]| 18.81940848|

+---------+--------------+