PG-DBDA MARCH 2022 BATCH KHARGHAR

MODULE: BIG DATA ANALYTICS

DATE : 20TH JUNE 2022

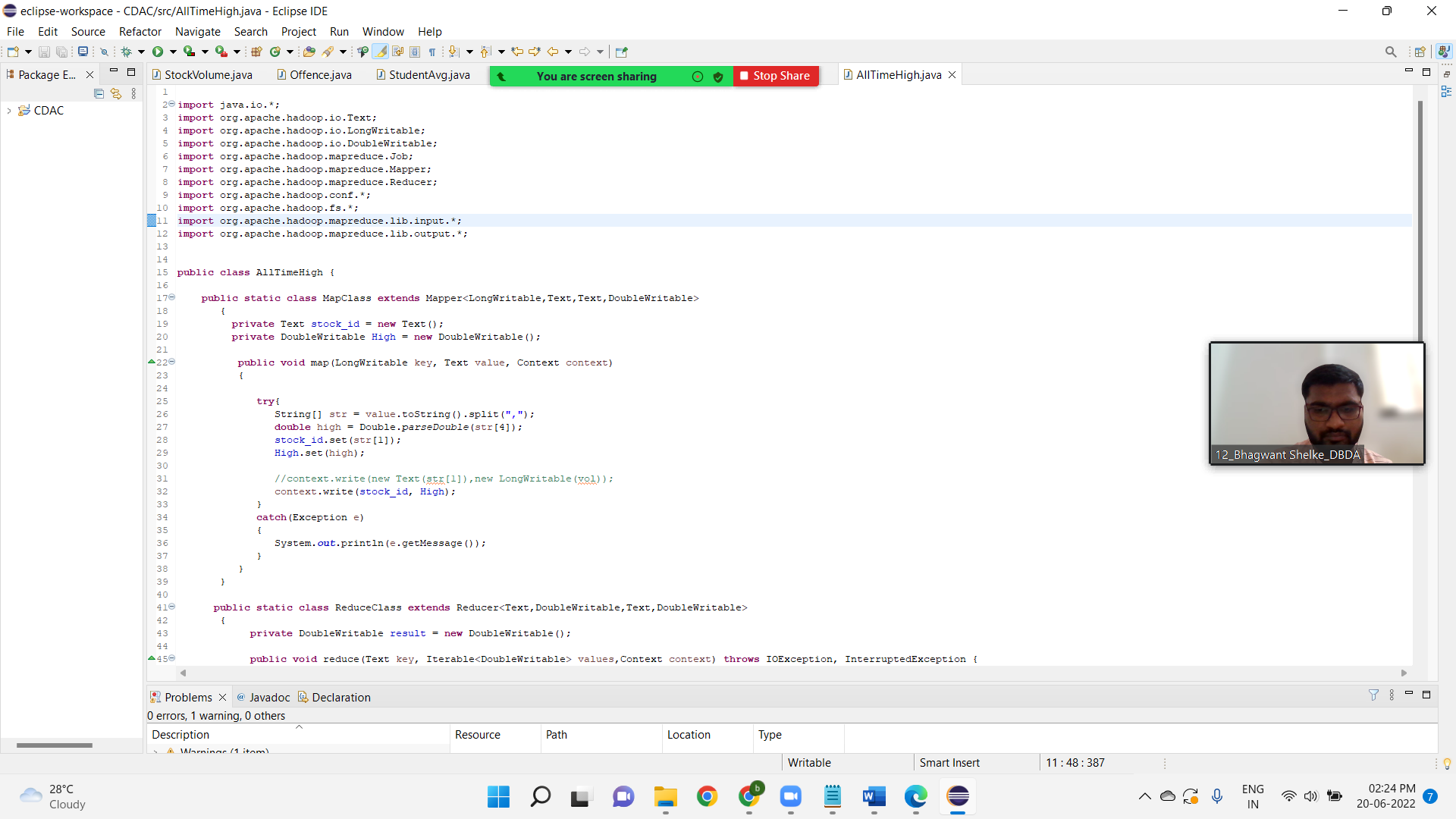
MARKS : 40 MARKS

**NAME – BHAGWANT SHELKE**

**ID – 20340325012**

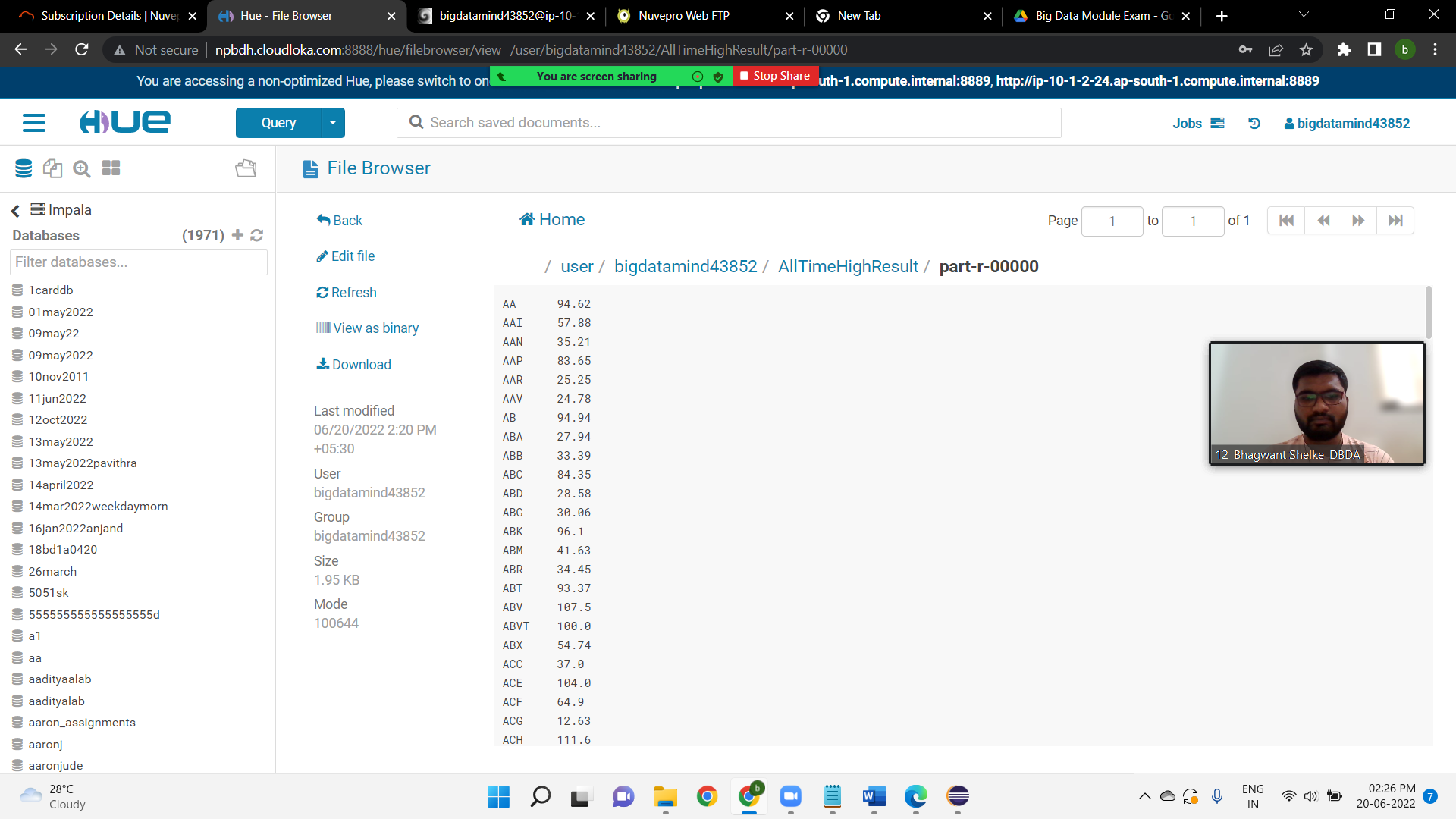
**Q1. MapReduce**

**Find all time High price for each stock**



hadoop jar myjar.jar AllTimeHigh cdac/NYSE.csv AllTimeHighResult

**OUTPUT**



**Hive**

create table customer1(custno INT, firstname STRING, lastname STRING, age INT, profession STRING)

row format delimited

fields terminated by ','

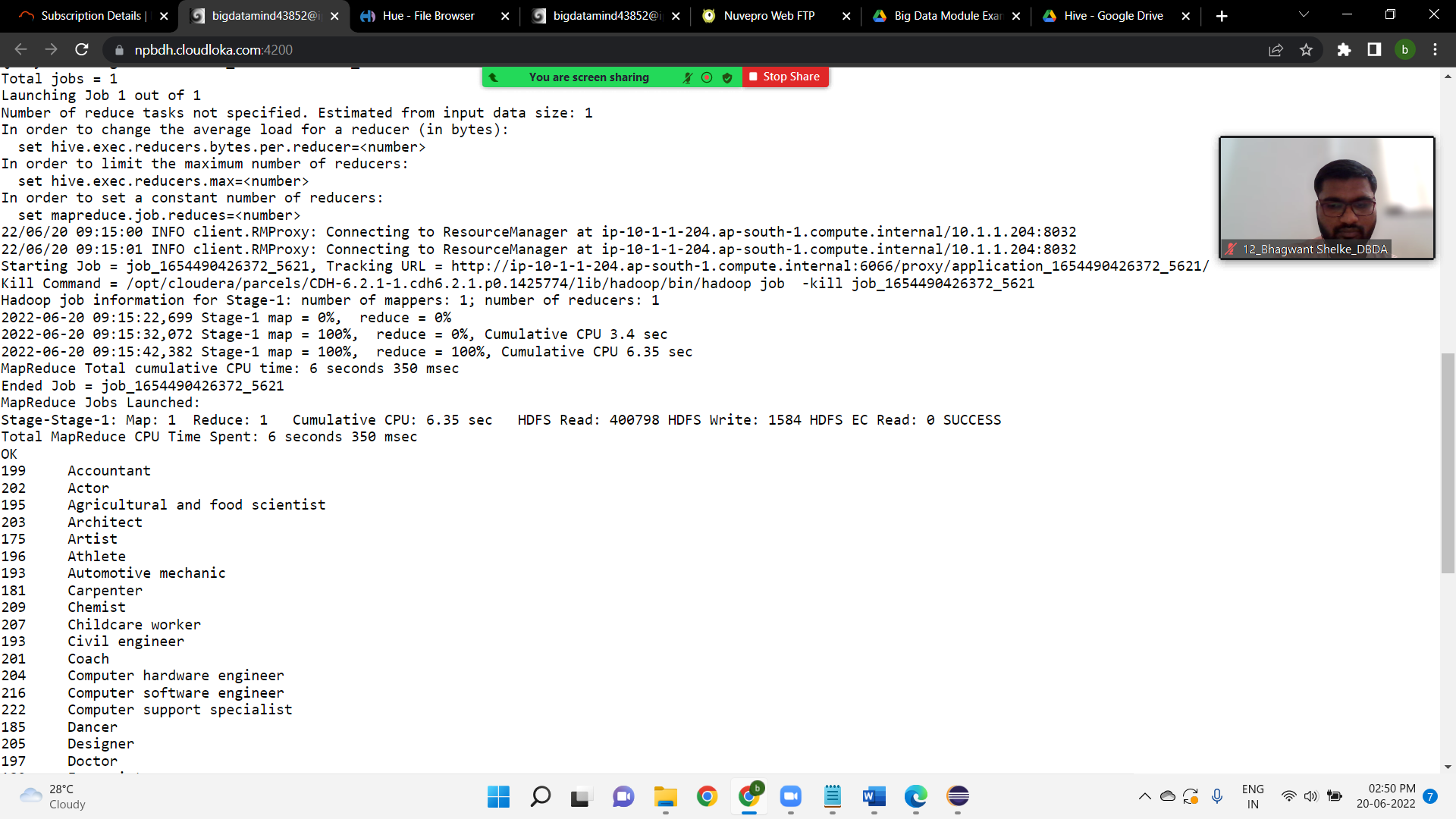
stored as textfile;

LOAD DATA LOCAL INPATH 'custs.txt' OVERWRITE INTO TABLE customer1;

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

select count(custno) as count, profession from customer1 group by profession;

**OUTPUT**



create table txnrecords1(txnno INT, txndate STRING, custno INT, amount DOUBLE,

category STRING, product STRING, city STRING, state STRING, spendby STRING)

row format delimited

fields terminated by ','

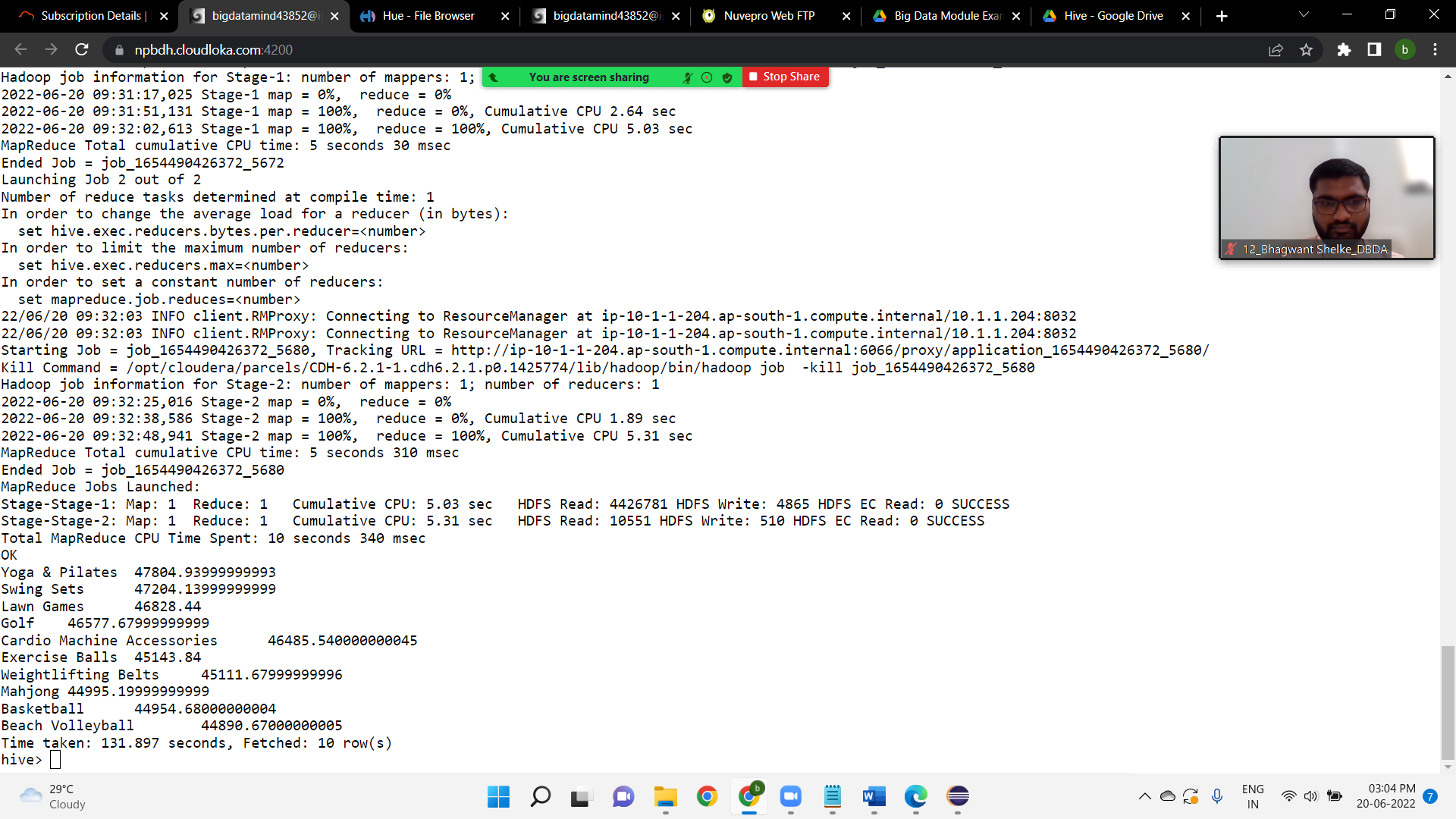
stored as textfile;

LOAD DATA LOCAL INPATH 'txns1.txt' OVERWRITE INTO TABLE txnrecords1;

**2) Write a program to find the top 10 products sales wise**

select product , sum(amount) as total from txnrecords1 group by product order by total desc limit 10;

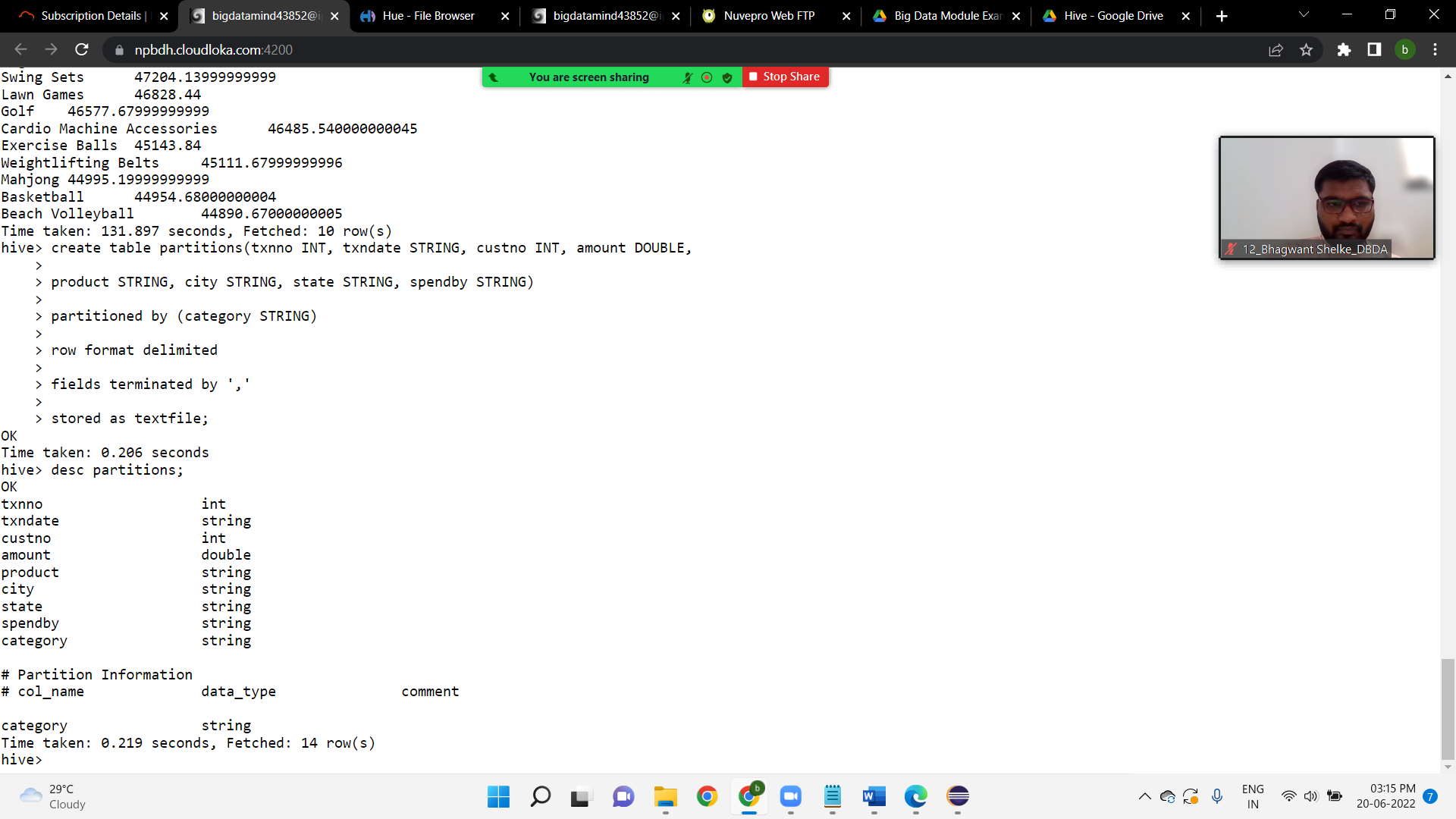
**OUTPUT**



**3) Write a program to create partitioned table on category**

create table partitions(txnno INT, txndate STRING, custno INT, amount DOUBLE, product STRING, city STRING, state STRING, spendby STRING) partitioned by (category STRING) row format delimited fields terminated by ',' stored as textfile;

**OUTPUT**



**PySpark**

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

airlineRDD =sc.textFile("/user/bigdatamind43852/airlines.csv")

>>> airlineRDD1 = airlineRDD.map(lambda a.encode("ascii","ignore"))

>>> header=airlineRDD1.first()

>>> airlineRDD2 = airlineRDD1.filter(lambda a : a != header)

>>> arrayRDD = airlineRDD2.map(lambda a: a.split(","))

>>>key\_value = arrayRDD.map(lambda a: (a[0], int(a[3])))

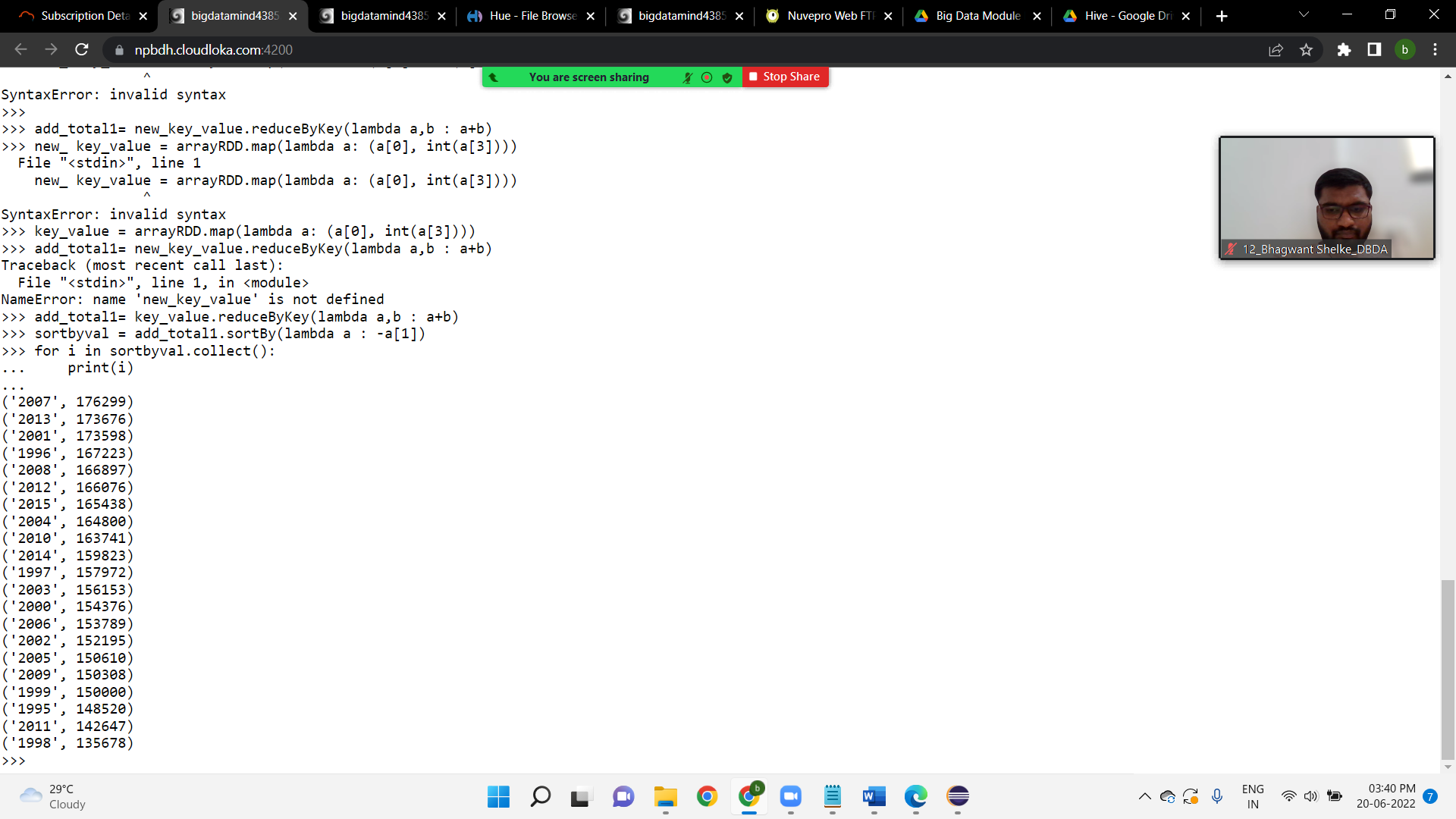
>>> add\_total1= key\_value.reduceByKey(lambda a,b : a+b)

>>> sortbyval = add\_total1.sortBy(lambda a : -a[1])

>>> for i in sortbyval.collect():

... print(i)

...



**2. Identifying the highest revenue generation for which year**

>>> key\_value1 = arrayRDD.map(lambda a: (a[0],

float(a[2])\*int(a[3])))

>>> for line in key\_value1.collect():

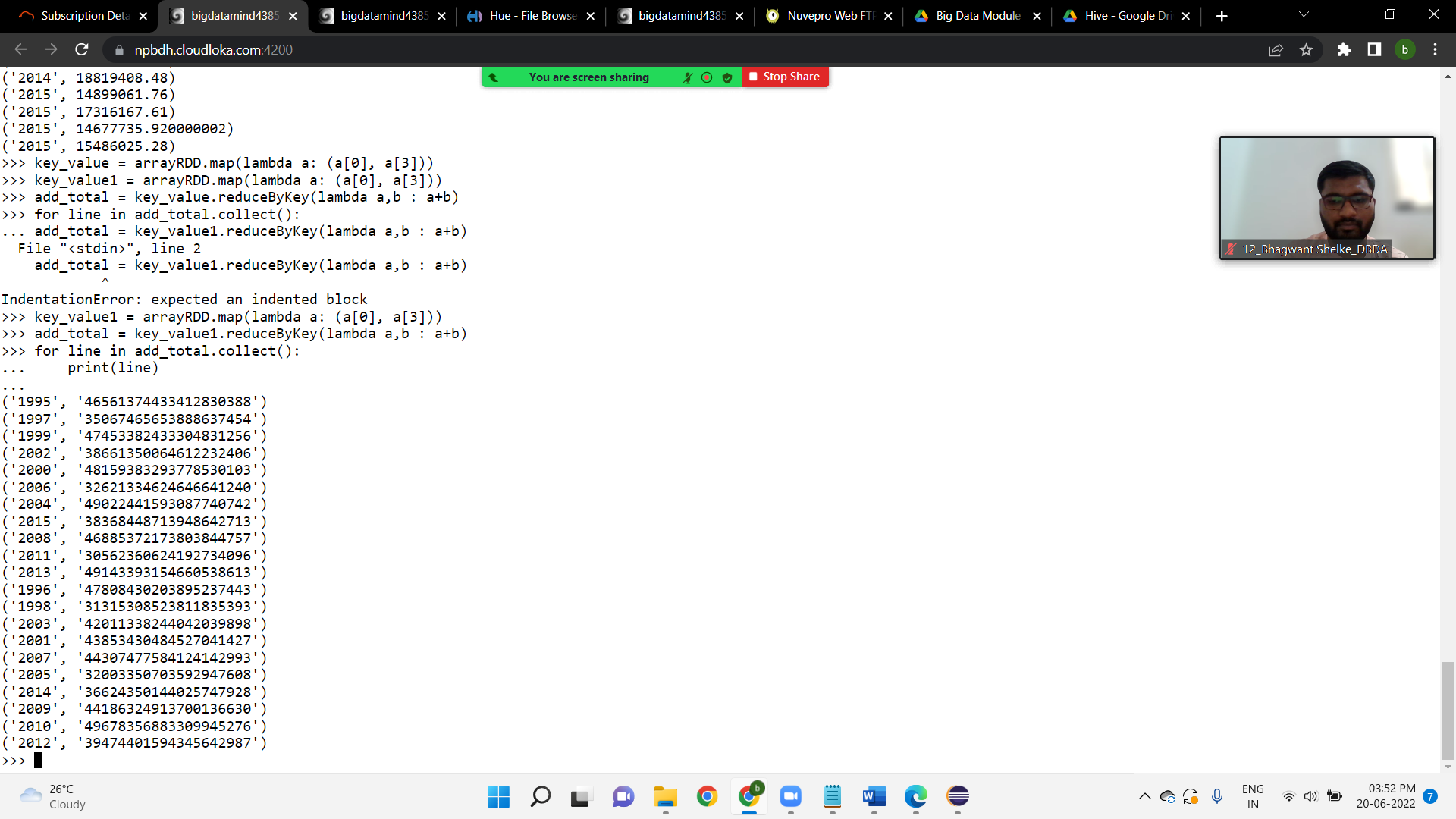
... print(line)

>>> key\_value1 = arrayRDD.map(lambda a: (a[0], a[3]))

>>> add\_total = key\_value1.reduceByKey(lambda a,b : a+b)

>>> for line in add\_total.collect():

... print(line)



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

>>> key\_value1 = arrayRDD.map(lambda a: (a[0],a[1], a[3]))

>>> for i in key\_value1.collect().take(5):

... print(i)

­­­­­­­­­­­­­­

...

