**Q1. MapReduce :-**

**Ans:-**

**import java.io.\*;**

**import java.io.IOException;**

**import java.util.StringTokenizer;**

**import org.apache.hadoop.conf.Configuration;**

**import org.apache.hadoop.fs.Path;**

**import org.apache.hadoop.io.IntWritable;**

**import org.apache.hadoop.io.LongWritable;**

**import org.apache.hadoop.io.Text;**

**import org.apache.hadoop.mapreduce.Job;**

**import org.apache.hadoop.mapreduce.Mapper;**

**import org.apache.hadoop.mapreduce.Reducer;**

**import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;**

**import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;**

**public class Stockclass {**

**public static class MapClass extends Mapper<LongWritable, Text, Text, LongWritable>**

**{**

**// private Text stock\_id = new Text();**

**// private IntWritable High = new IntWritable();**

**public void Map (LongWritable key, Text value, Context context )**

**{**

**try {**

**String[] str = value.toString().split(",");**

**long vol = Long.*parseLong*(str[7]);**

**context.write(new Text(str[1]),new LongWritable(vol));**

**}**

**catch(Exception e)**

**{**

**System.*out*.println(e.getMessage());**

**}**

**}**

**}**

**public static class ReduceClass extends Reducer<Text,LongWritable,Text,LongWritable>**

**{**

**private LongWritable result = new LongWritable();**

**public void reduce(Text key, Iterable<LongWritable>values,Context context)**

**throws IOException, InterruptedException{**

**long sum = 0;**

**for(LongWritable val : values)**

**{**

**sum+= val.get();**

**}**

**result.set(sum);**

**context.write(key,result);**

**}**

**}**

**public static void main(String[] args) throws Exception {**

**Configuration conf = new Configuration();**

**Job job = Job.*getInstance*(conf, "Counting volume ");**

**job.setJarByClass(Stockclass.class);**

**job.setMapperClass(MapClass.class);**

**job.setReducerClass(ReduceClass.class);**

**job.setNumReduceTasks(1);**

**job.setMapOutputKeyClass(Text.class);**

**// job.setMapOutputValueClass();**

**job.setOutputKeyClass(Text.class);**

**job.setOutputValueClass(LongWritable.class);**

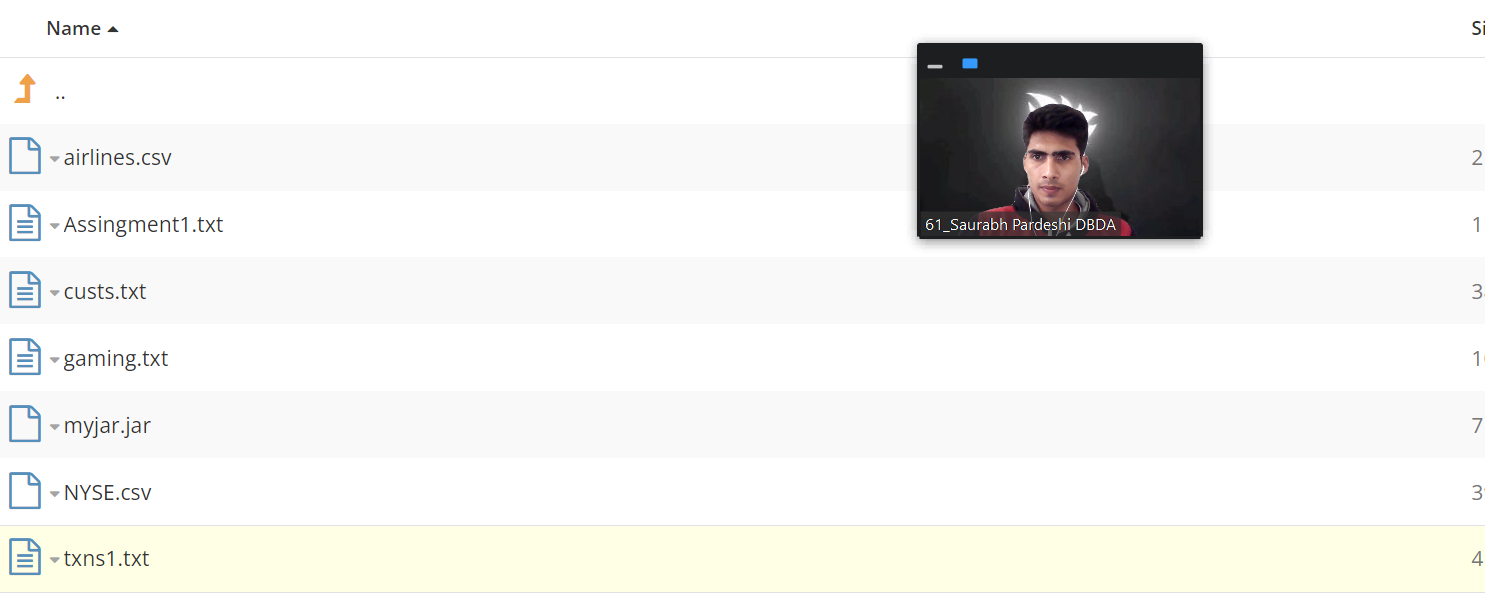
**FileInputFormat.*addInputPath*(job, new Path(args[0]));**

**FileOutputFormat.*setOutputPath*(job, new Path(args[1]));**

**System.*exit*(job.waitForCompletion(true) ? 0 : 1);**

**}**

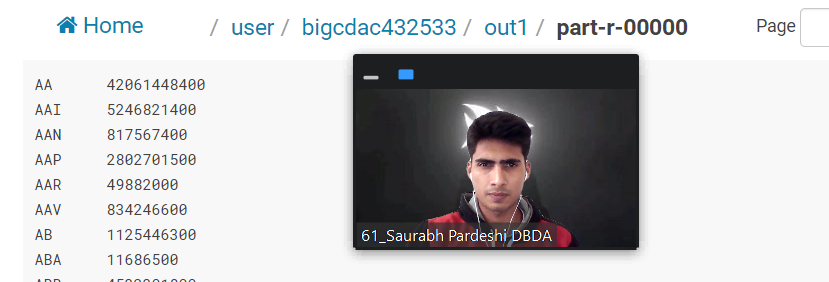
**}**

****

**On Webshell:-**

**hadoop jar myjar.jar Stockclass /user/bigcdac432533/NYSE.csv**

**/user/bigcdac432533/out1**

****

**HIVE:-**

**Question 2 ➖**

**ANS:-**

[bigcdac432533@ip-10-1-1-204 ~]$ hadoop fs -put custs.txt

[bigcdac432533@ip-10-1-1-204 ~]$ hive

hive> create database Saurabh;

hive> use saurabh;

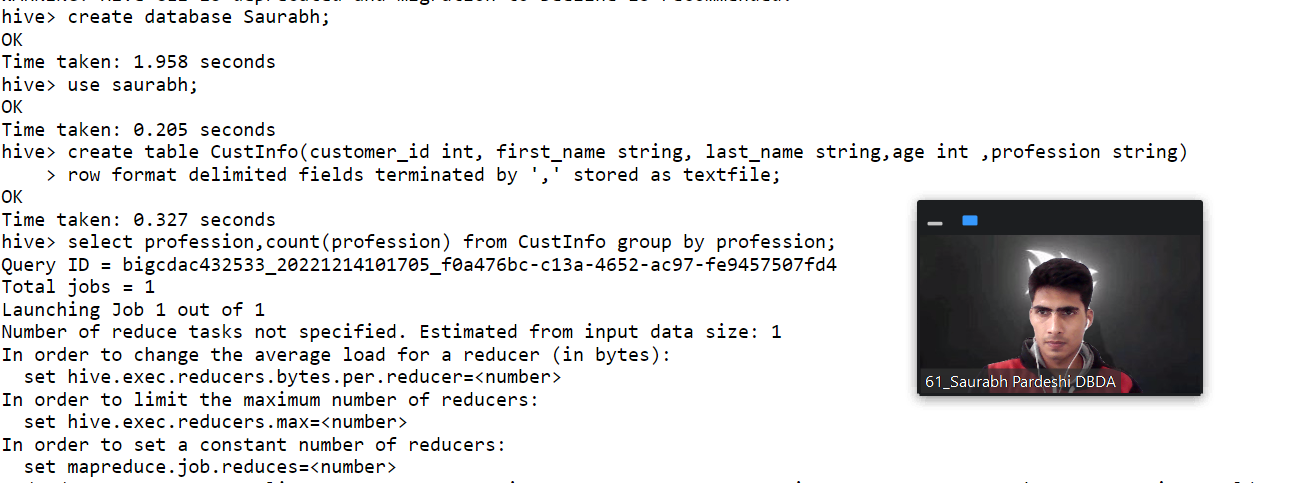
hive> create table CustInfo(customer\_id int, first\_name string, last\_name string,age int ,profession string)

> row format delimited fields terminated by ',' stored as textfile;

hive> load data local inpath 'custs.txt'

> overwrite into table CustInfo;

hive> select profession,count(profession) from CustInfo group by profession limit 10;



hive> create database Raghuvanshi;

hive> use Raghuvanshi;

hive> create table sales(txns\_id int,txns\_date string,customer\_id int,amt double,cat string,product string,city string, state string,spendby string)

> row format delimited fields terminated by ',' stored as textfile;

hive> LOAD DATA LOCAl INPATH 'txns1.txt' into table sales;

hive> select product,sum(amt) as total\_amt from sales group by product order by total\_amt desc limit 10;

