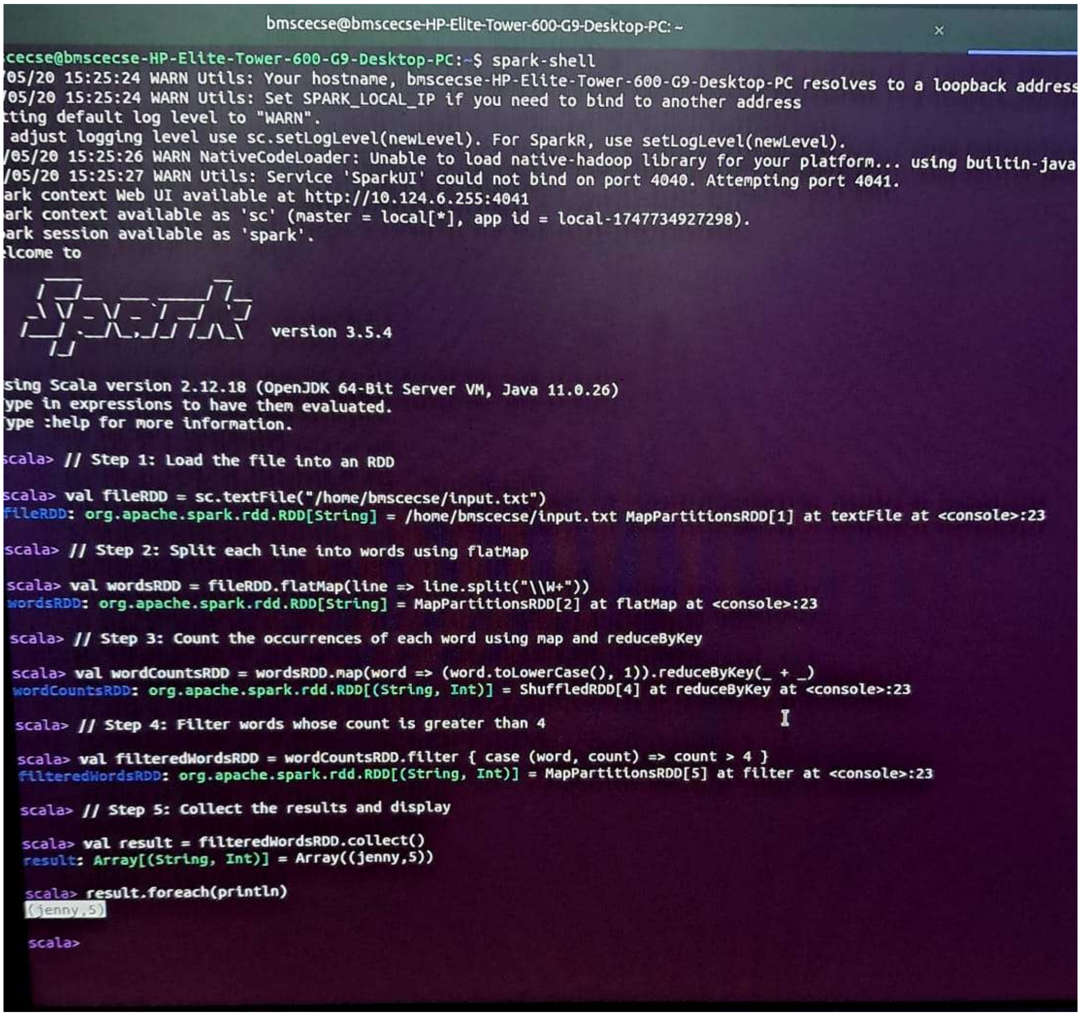
LAB – 9

Using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark.

```
//Create input.txt file

nano input.txt

//enter the contents and save the file

// Step 1: Load the file into an RDD

val fileRDD = sc.textFile("input.txt")

// Step 2: Split each line into words using flatMap
val wordsRDD = fileRDD.flatMap(line => line.split("\\W+"))

// Step 3: Count the occurrences of each word using map and reduceByKey
val wordCountsRDD = wordsRDD.map(word => (word.toLowerCase(), 1)).reduceByKey(_ + _)

// Step 4: Filter words whose count is greater than 4
val filteredWordsRDD = wordCountsRDD.filter { case (word, count) => count > 4 }

// Step 5: Collect the results and display
val result = filteredWordsRDD.collect()
result.foreach(println)
```



```
cecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ spark-shell
/05/20 15:25:24 WARN Utils: Your hostname, bmscecse-HP-Elite-Tower-600-G9-Desktop-PC resolves to a loopback address
/05/20 15:25:24 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
tting default log level to "WARN".
 adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
/05/20 15:25:26 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
/05/20 15:25:27 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
ark context Web UI available at http://10.124.6.255:4041
ark context available as 'sc' (master = local[*], app id = local-1747734927298).
ark session available as 'spark'.
lcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.4
      /_/

sing Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.26)
ype in expressions to have them evaluated.
ype :help for more information.

scala> // Step 1: Load the file into an RDD

scala> val fileRDD = sc.textFile("/home/bmscecse/input.txt")
fileRDD: org.apache.spark.rdd.RDD[String] = /home/bmscecse/input.txt MapPartitionsRDD[1] at textFile at <console>:23

scala> // Step 2: Split each line into words using flatMap

scala> val wordsRDD = fileRDD.flatMap(line => line.split("\\W+"))
wordsRDD: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at <console>:23

scala> // Step 3: Count the occurrences of each word using map and reduceByKey

scala> val wordCountsRDD = wordsRDD.map(word => (word.toLowerCase(), 1)).reduceByKey(_ + _)
wordCountsRDD: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:23

scala> // Step 4: Filter words whose count is greater than 4

scala> val filteredWordsRDD = wordCountsRDD.filter { case (word, count) => count > 4 }
filteredWordsRDD: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[5] at filter at <console>:23

scala> // Step 5: Collect the results and display

scala> val result = filteredWordsRDD.collect()
result: Array[(String, Int)] = Array((jenny,5))

scala> result.foreach(println)
(jenny,5)

scala>
```