SPARK framework introduction

Go to the Bigdata User login

Pwd - hadoop

Now search for Oracle virtual box

Double Click Hadoop

Go to login,

Login - Ubuntu

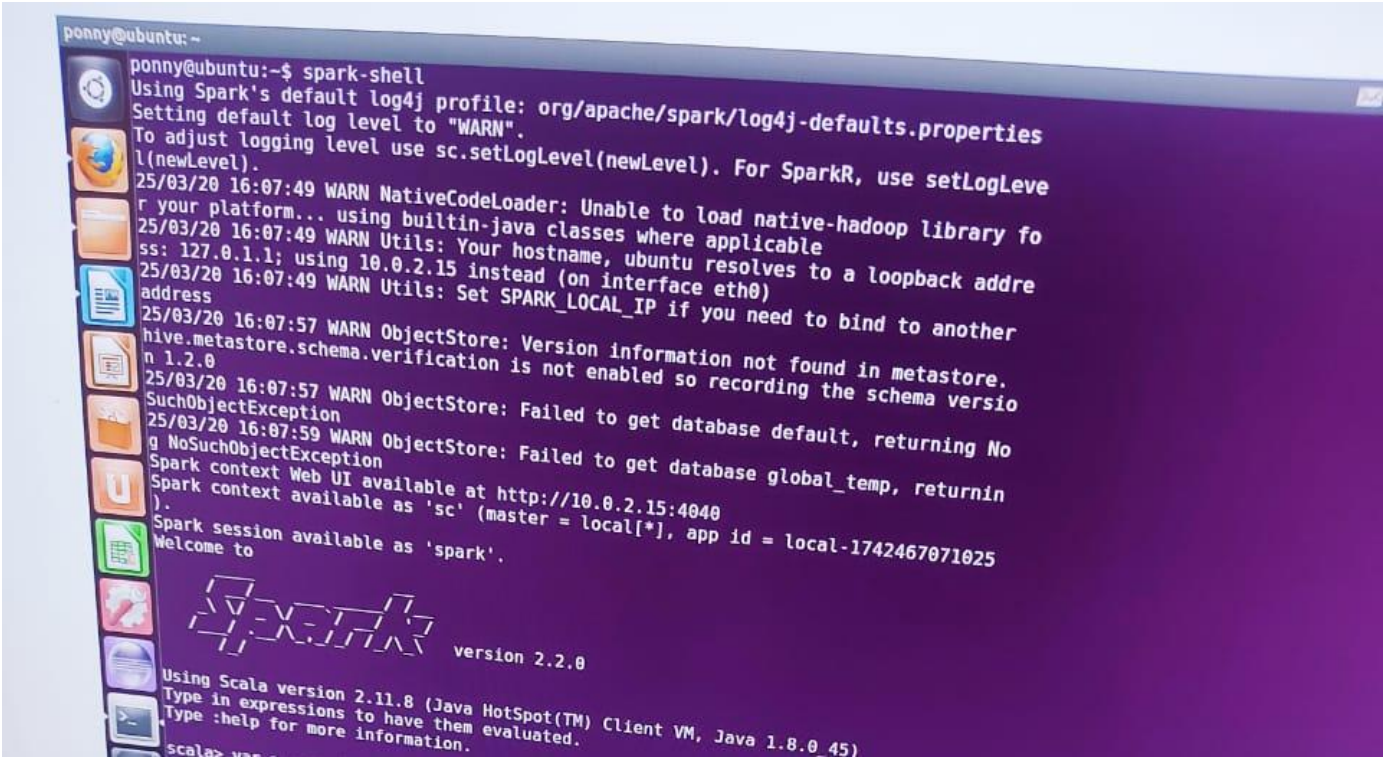
Passwd - vitcc123

Experiment 2: User Defined Function using Scala in spark framework

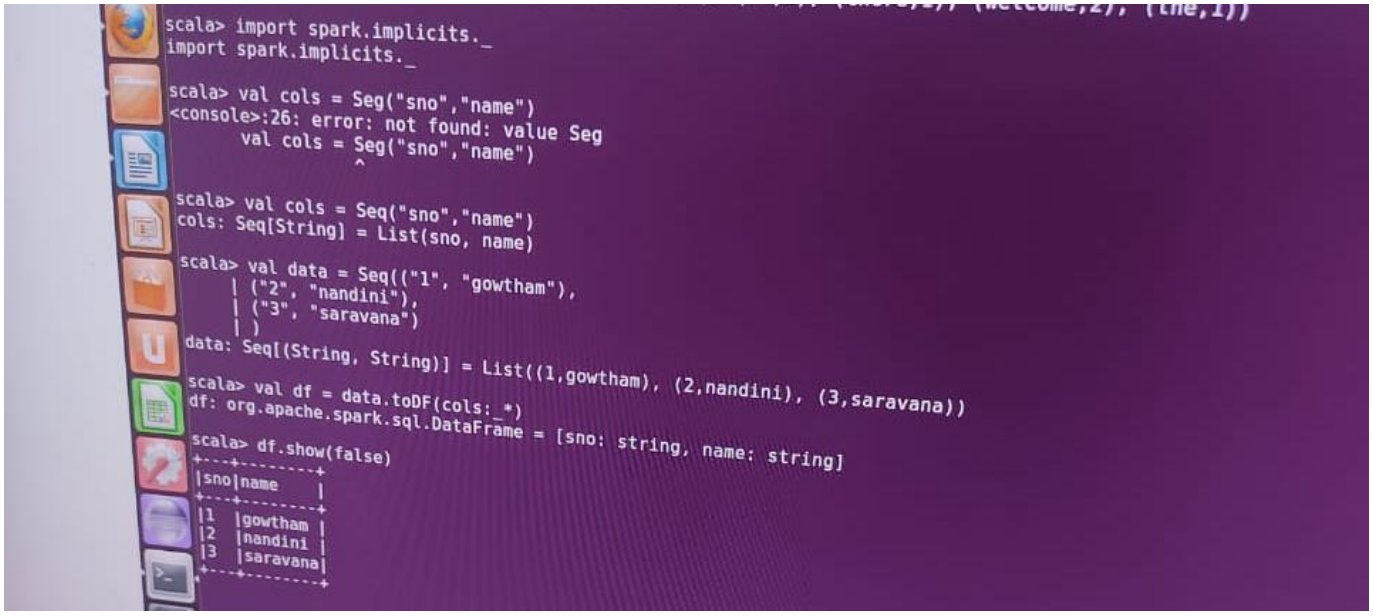
Open terminal

Type the command: spark-shell

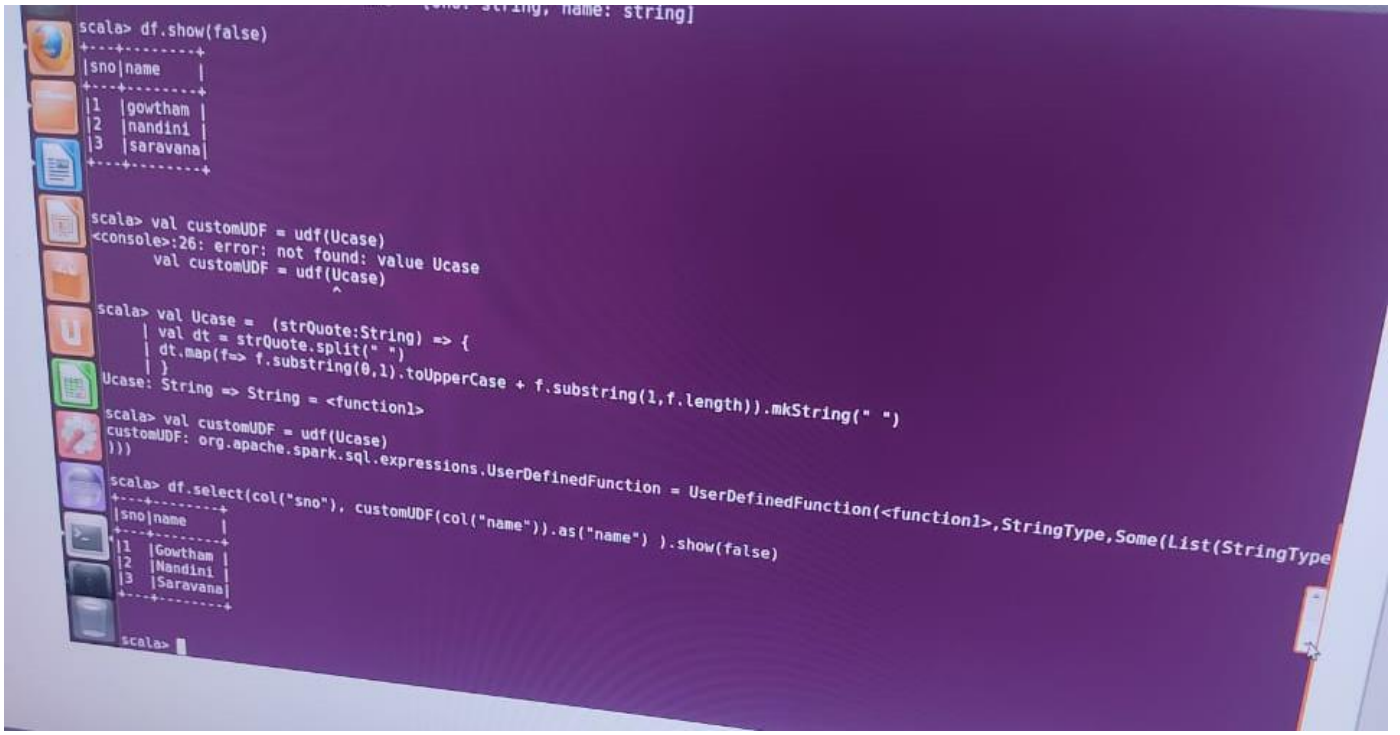
1. **Follow the commands given in the screenshot --**



2.


scala> import spark.implicits._
import spark.implicits._
scala> val cols = Seq("sno","name")
<console>:26: error: not found: value Seg
val cols = Seg("sno","name")
^
scala> val cols = Seq("sno","name")
cols: Seq[String] = List(sno, name)
scala> val data = Seq(("1", "gowtham"),
| ("2", "nandini"),
| ("3", "saravana")
|)
data: Seq[(String, String)] = List((1,gowtham), (2,nandini), (3,saravana))
scala> val df = data.toDF(cols:_*)
df: org.apache.spark.sql.DataFrame = [sno: string, name: string]
scala> df.show(false)
+---+--------+
|sno|name |
+---+--------+
1	gowtham
2	nandini
3	saravana
+---+--------+

3.


scala> df.show(false)
+---+--------+
|sno|name |
+---+--------+
1	gowtham
2	nandini
3	saravana
+---+--------+
scala> val customUDF = udf(Ucase)
<console>:26: error: not found: value Ucase
val customUDF = udf(Ucase)
^
scala> val Ucase = (strQuote:String) => {
| val dt = strQuote.split(" ")
| dt.map(f=> f.substring(0,1).toUpperCase + f.substring(1,f.length)).mkString(" ")
| }
Ucase: String => String = <function1>
scala> val customUDF = udf(Ucase)
customUDF: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,Some(List(StringType
)))
scala> df.select(col("sno"), customUDF(col("name")).as("name")).show(false)
+---+--------+
|sno|name |
+---+--------+
1	Gowtham
2	Nandini
3	Saravana
+---+--------+
scala>