Big Data Analytics Journal

Submitted By

Name: Vatsal Parekh
Class: MSC CS (SEM-III)
Roll No: 31031523022
MSc CS – Part II

Department Of Computer Science
Somaiya Vidyavihar University
SK Somaiya College

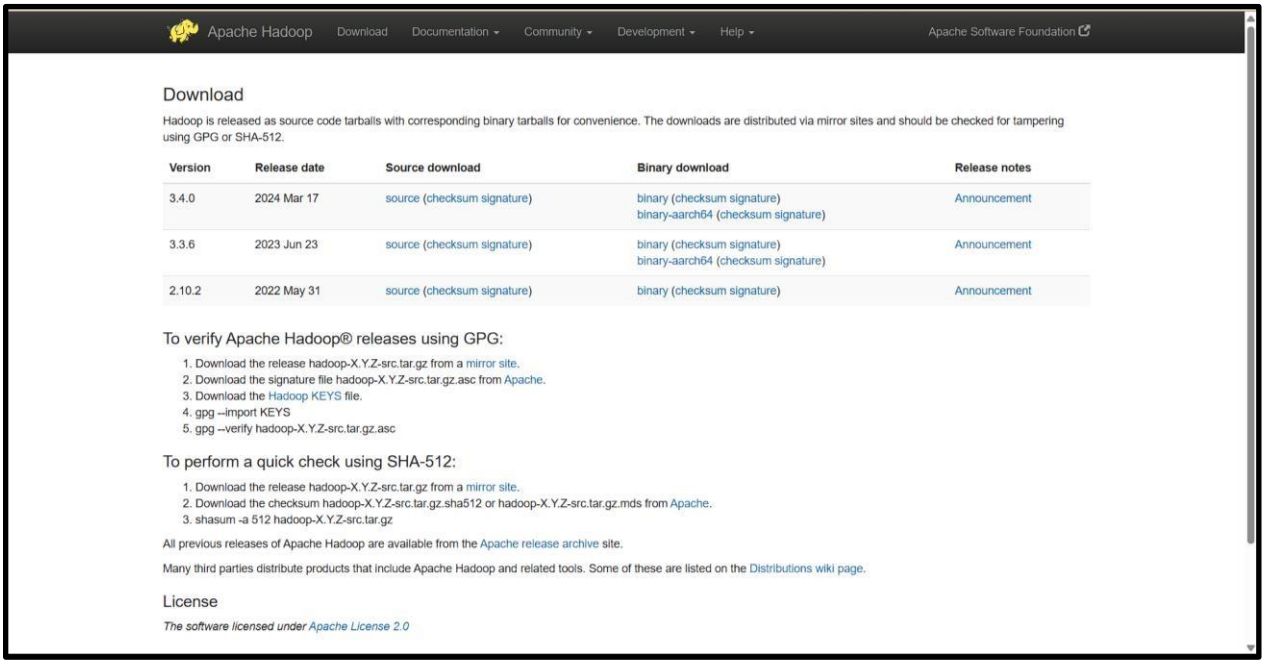
Name: Vatsal Parekh
Roll No: 31031523022

Index:

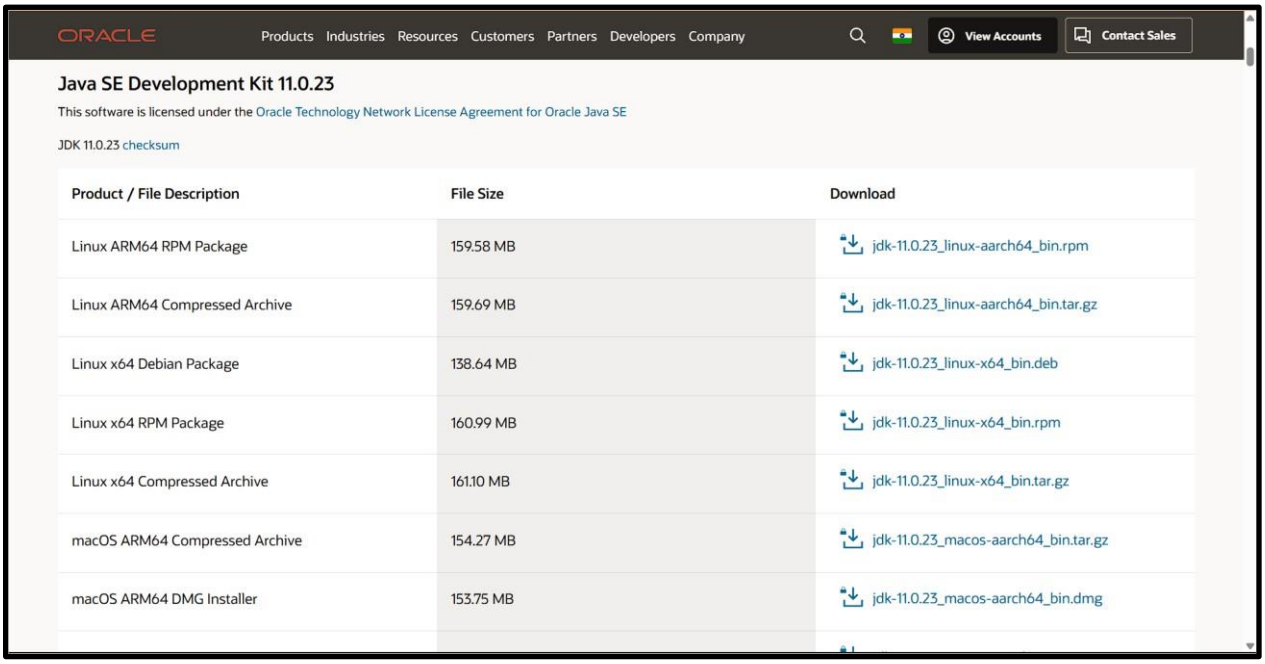
Name: Vatsal Parekh
Roll No: 31031523022

Practical 1: Installation of Hadoop on Windows.

1. Download the latest Hadoop Binary version from here.

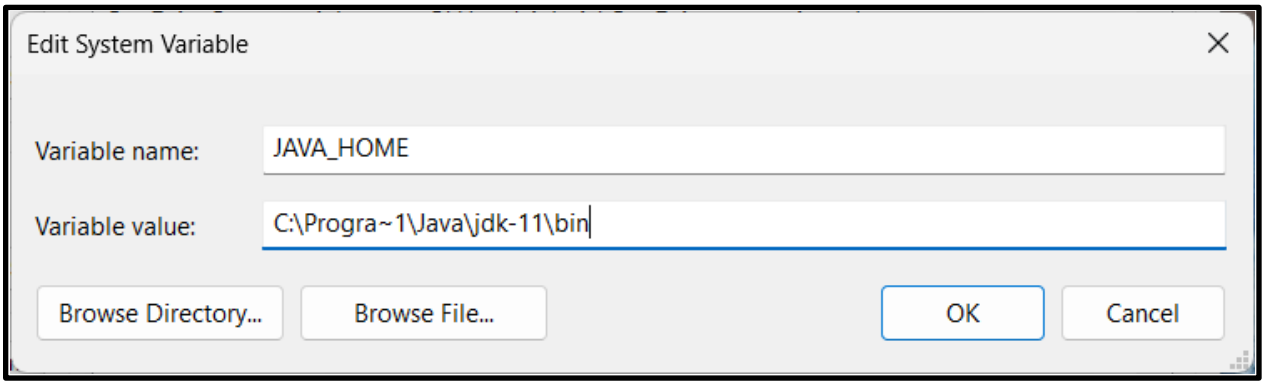


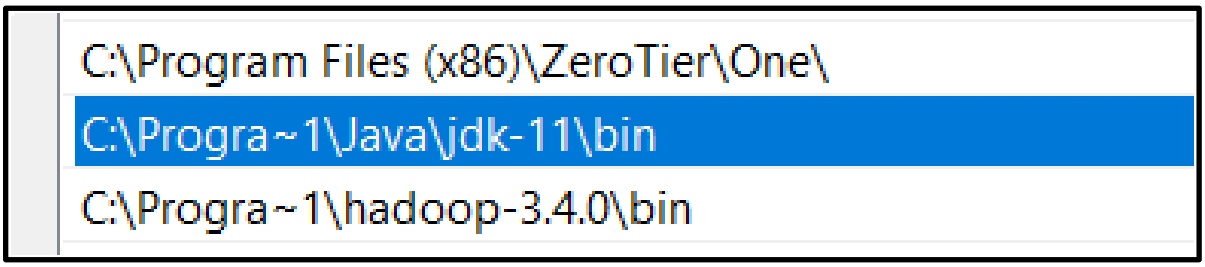
2. Download Java version 11.0.23 from here.



3. Run the installer and follow the steps to install Java.

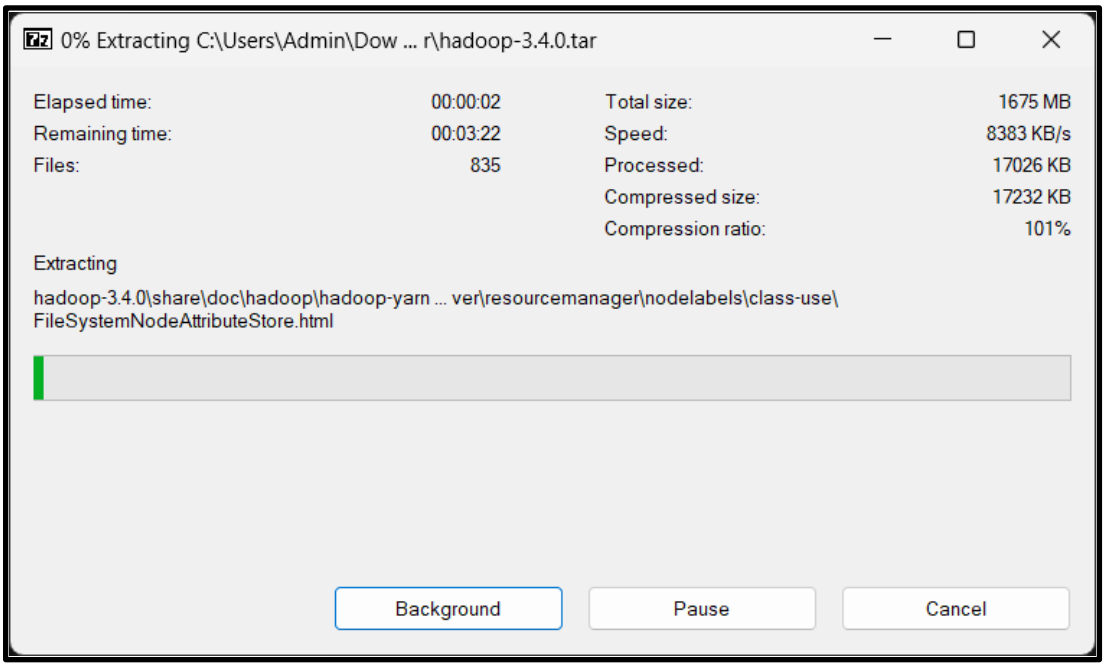
4: Make sure to set the Java Environment Variables by creating a new variable and also adding it to the path variable.





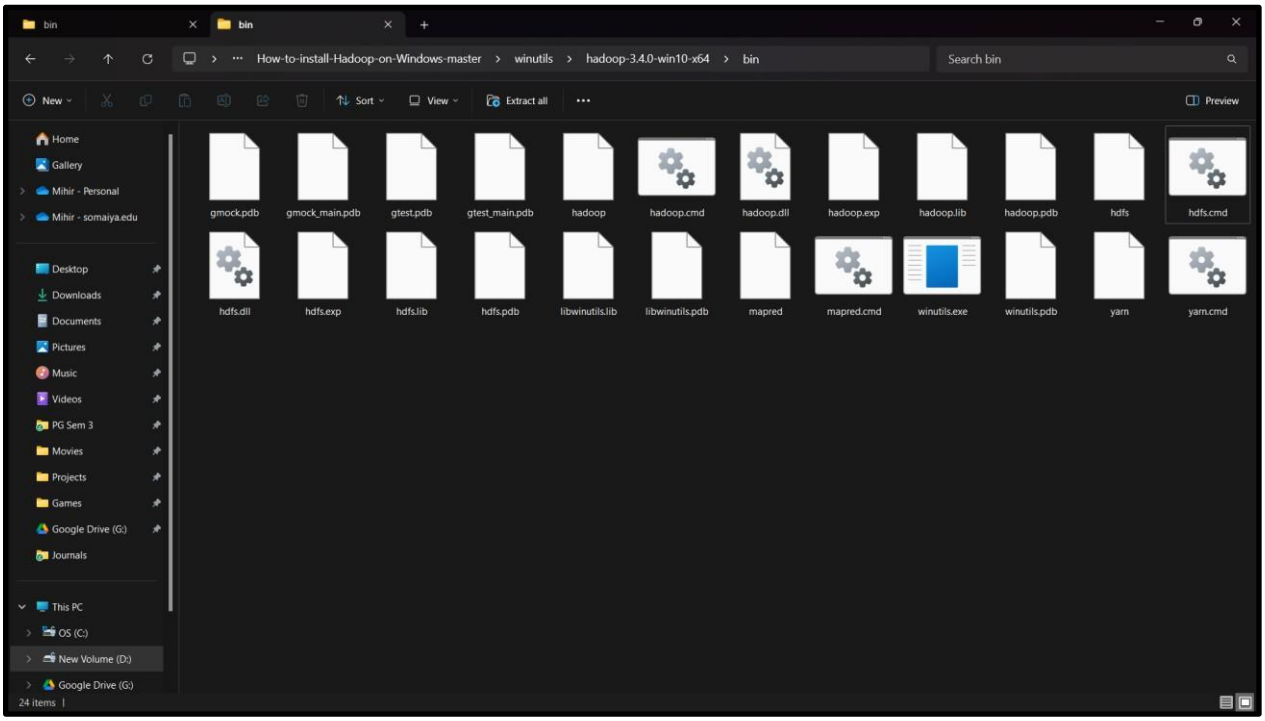
Name: Vatsal Parekh
Roll No: 31031523022

5. Extract the previously downloaded Hadoop archive and copy the contents to your desired location.

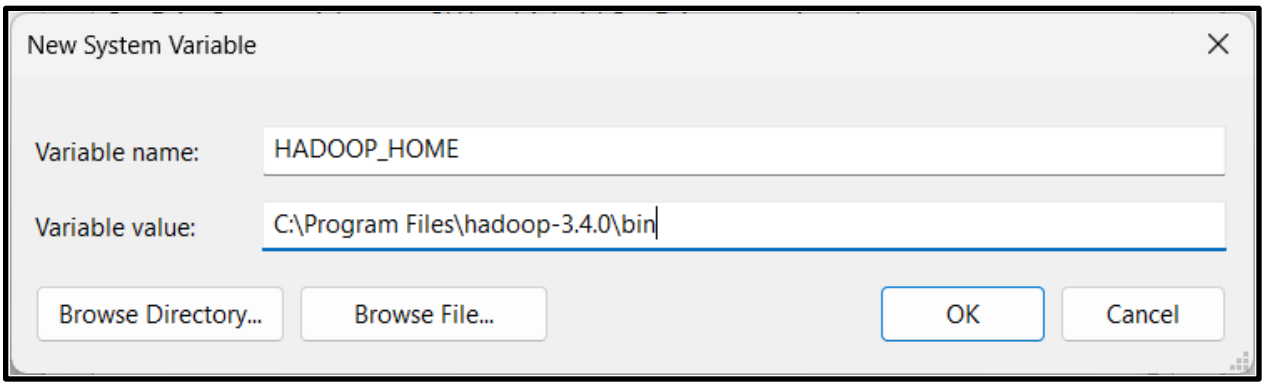


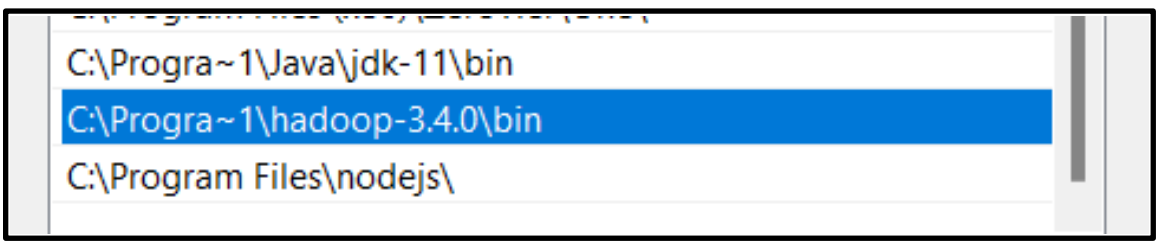
Name: Vatsal Parekh
Roll No: 31031523022

6. Download and Install Hadoop native IO binary from here. To install copy all the contents of the archive to the bin folder of your Hadoop.

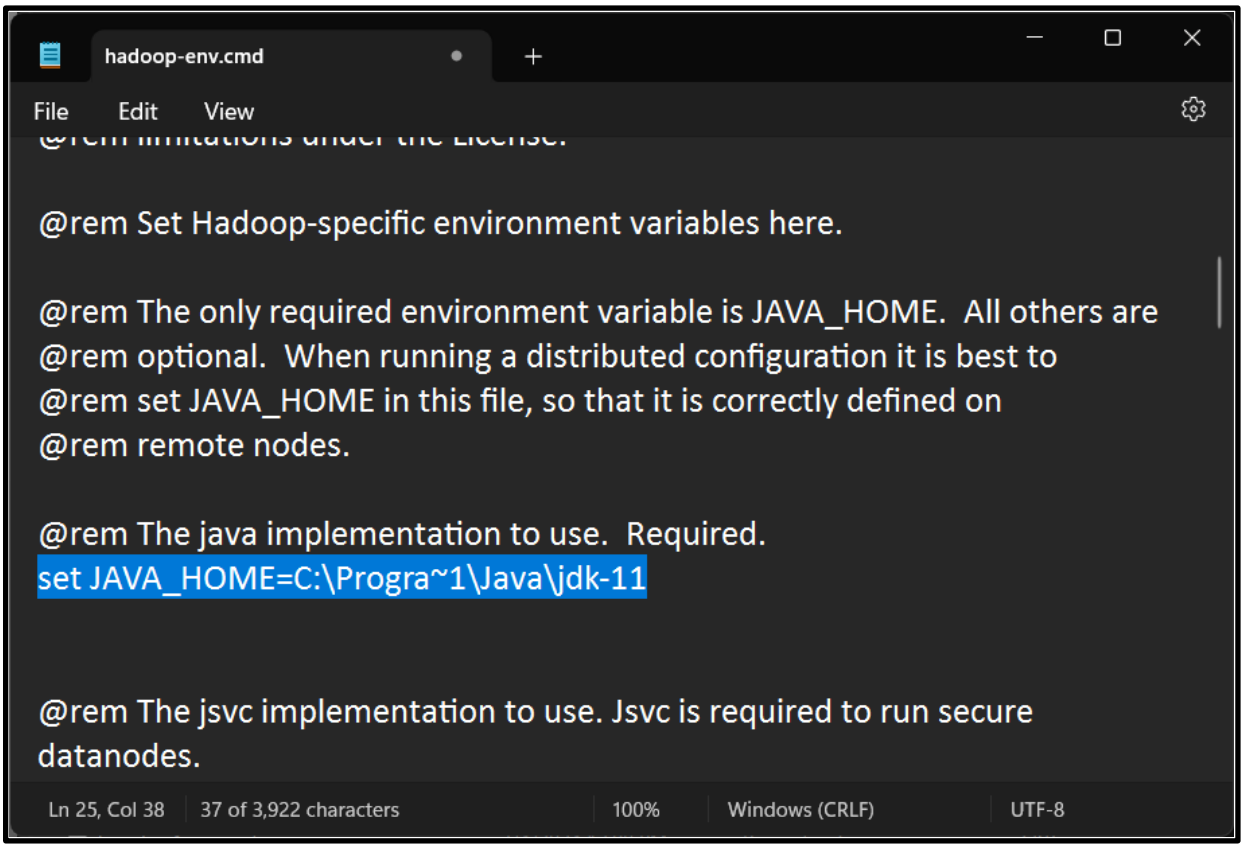


7. Set up the Hadoop Environment Variables.

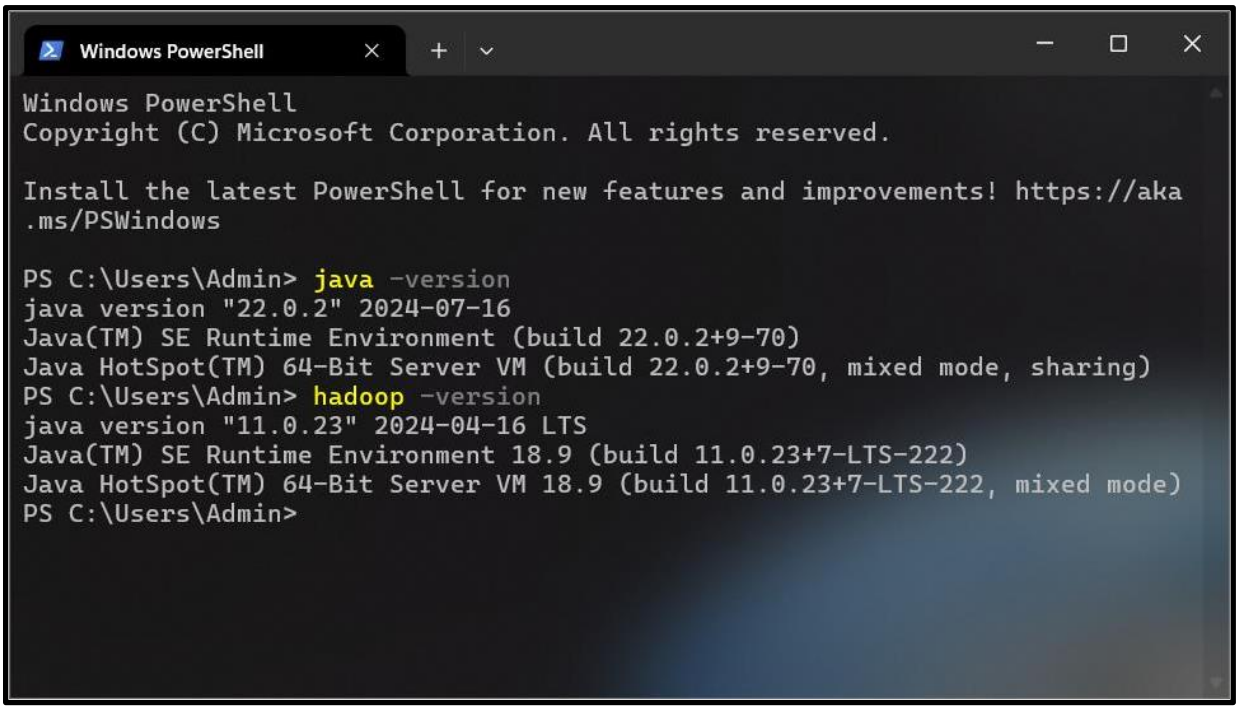




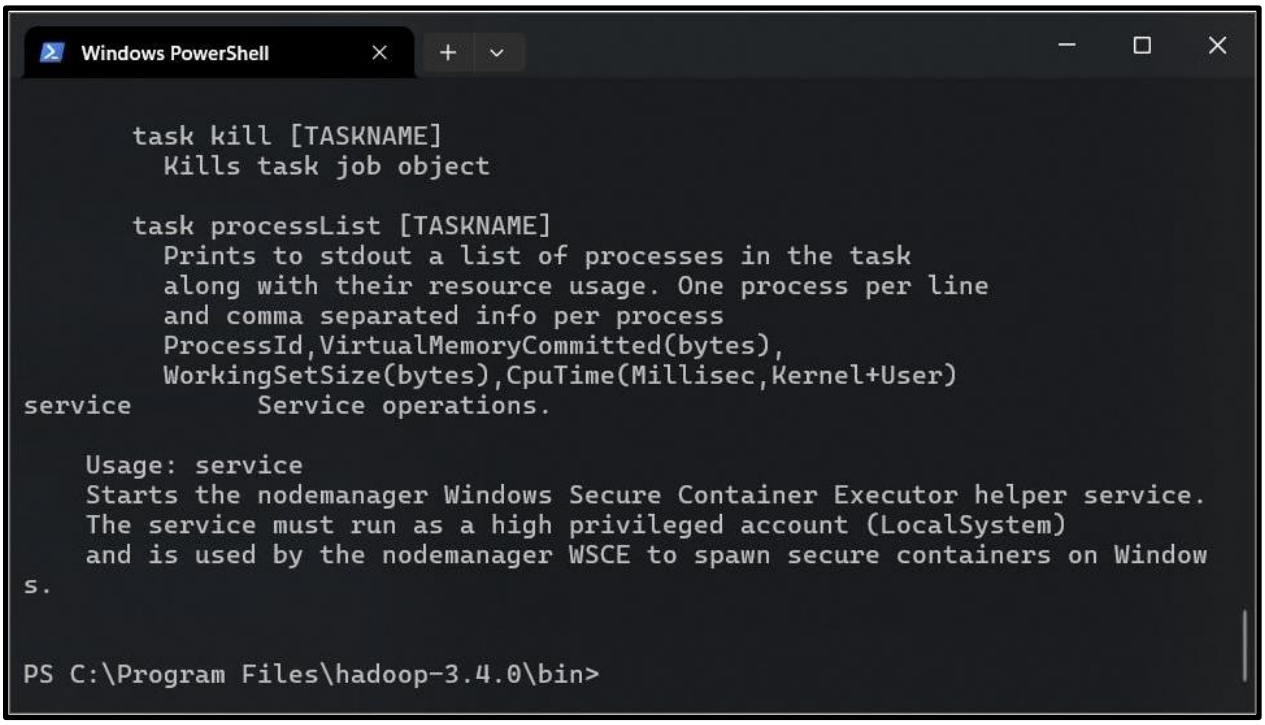
8. Set the JAVA_HOME in the hadoop_env.cmd file located inside the etc folder.



9. Check the versions of both Java and Hadoop.

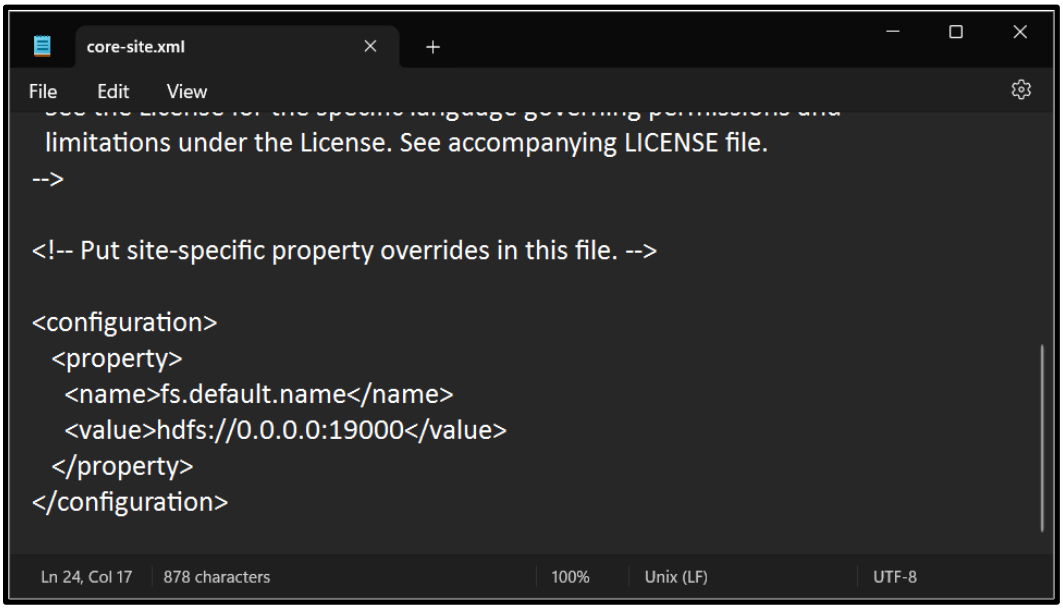


10. To verify all the above steps are completed successfully, open the bin folder in terminal and run the command winutils.exe



11. Hadoop Configurations. All the files will be located in the etc folder of your Hadoop installation.

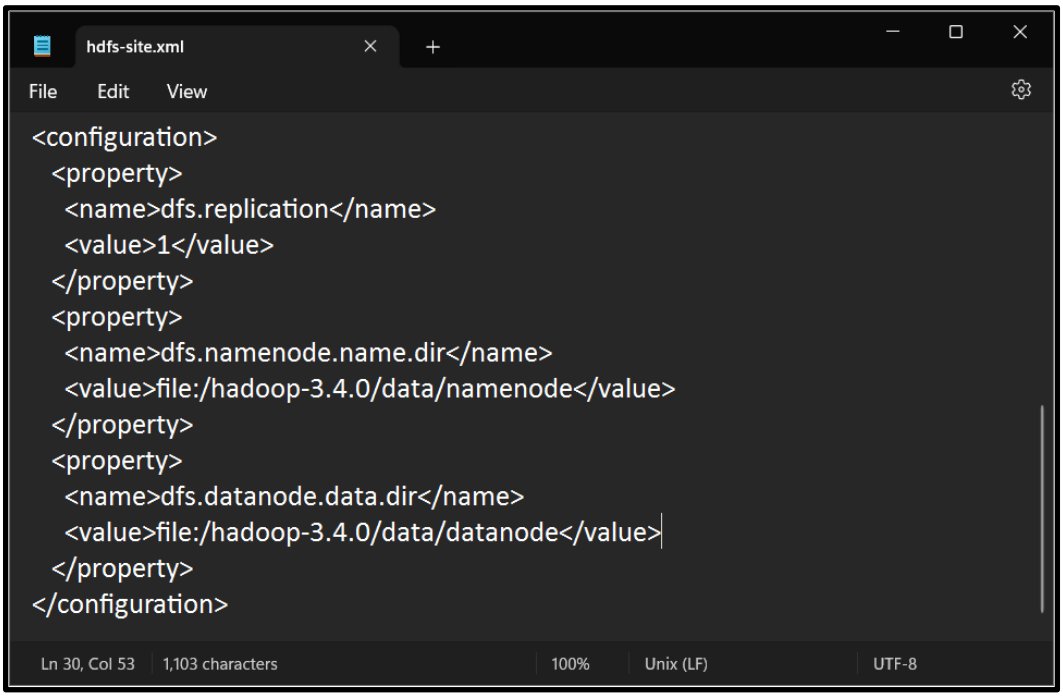
11a. Configure core-site (core-site.xml)



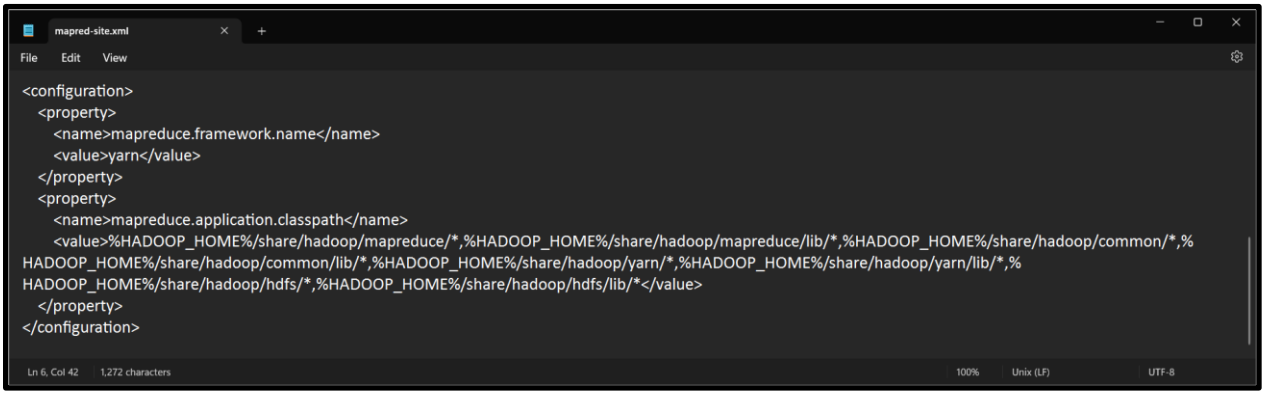
Name: Vatsal Parekh
Roll No: 31031523022

11b. Configure HDFS (hdfs-site.xml)

Before configuring HDFS, make a folder called as data in your Hadoop installation. Make two subfolders named as namenode and datanode.

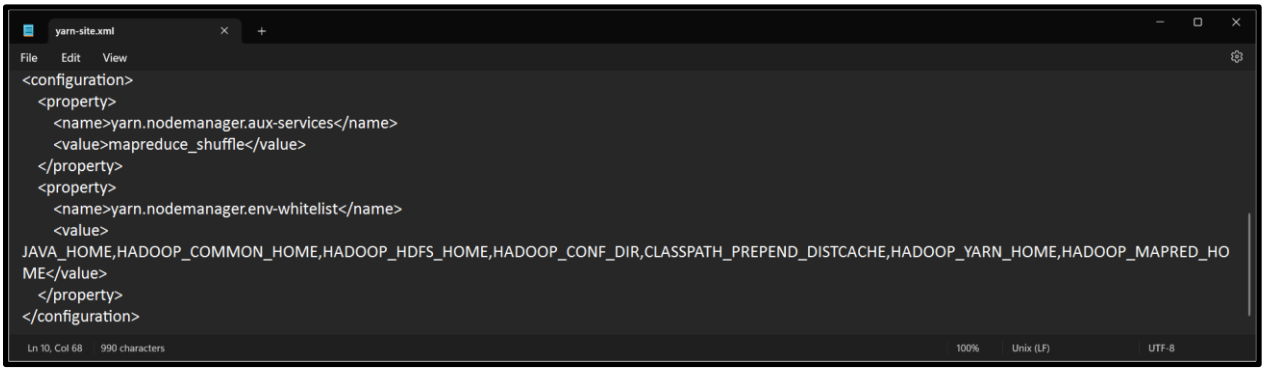


11c. Configure MapReduce (mapred-site.xml)

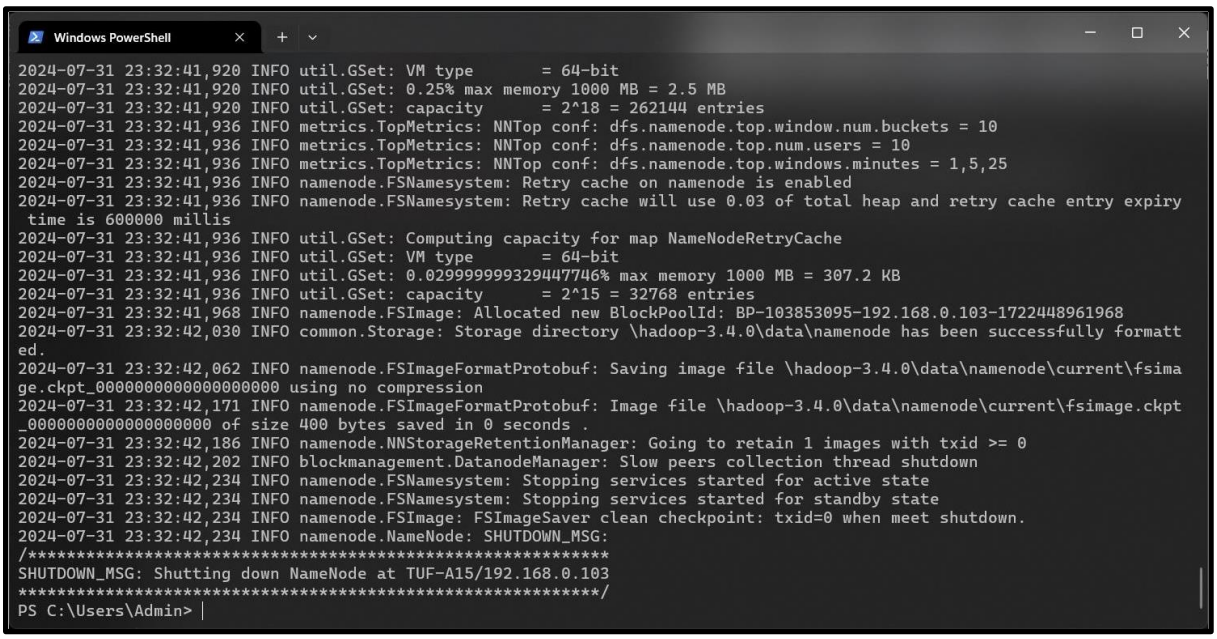


Name: Vatsal Parekh
Roll No: 31031523022

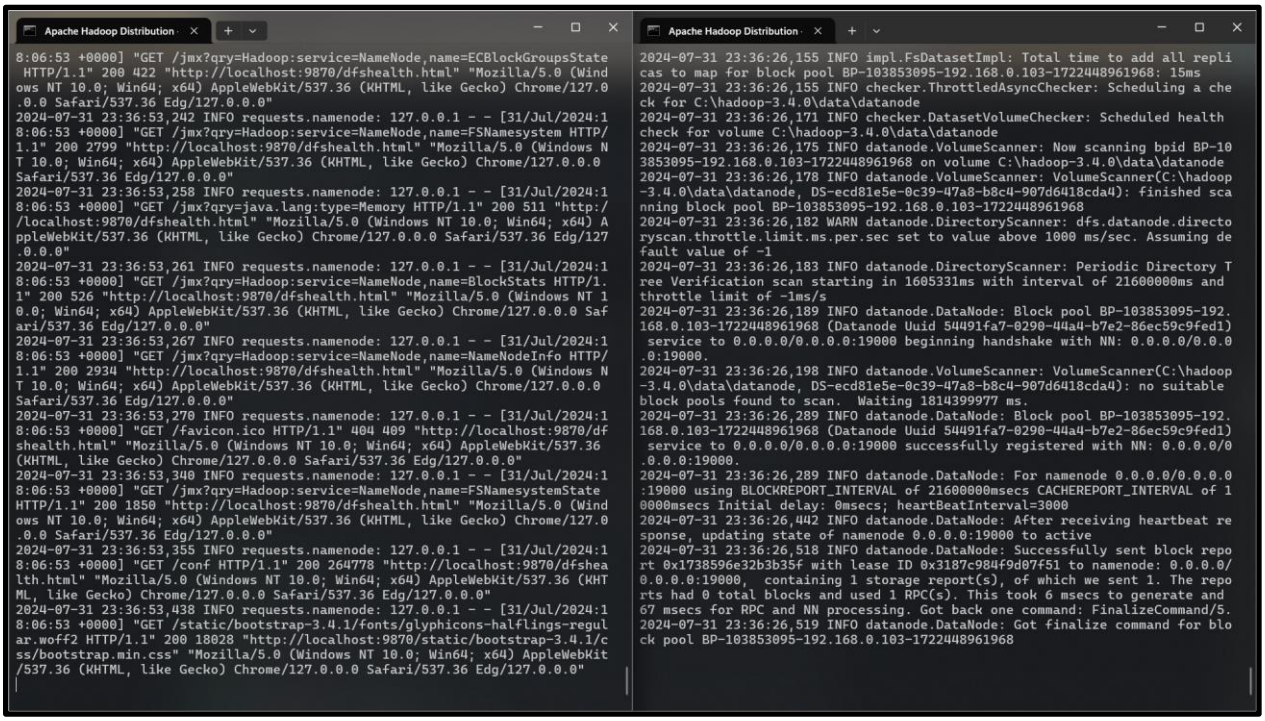
11d. Configure YARN (yarn-site.xml)



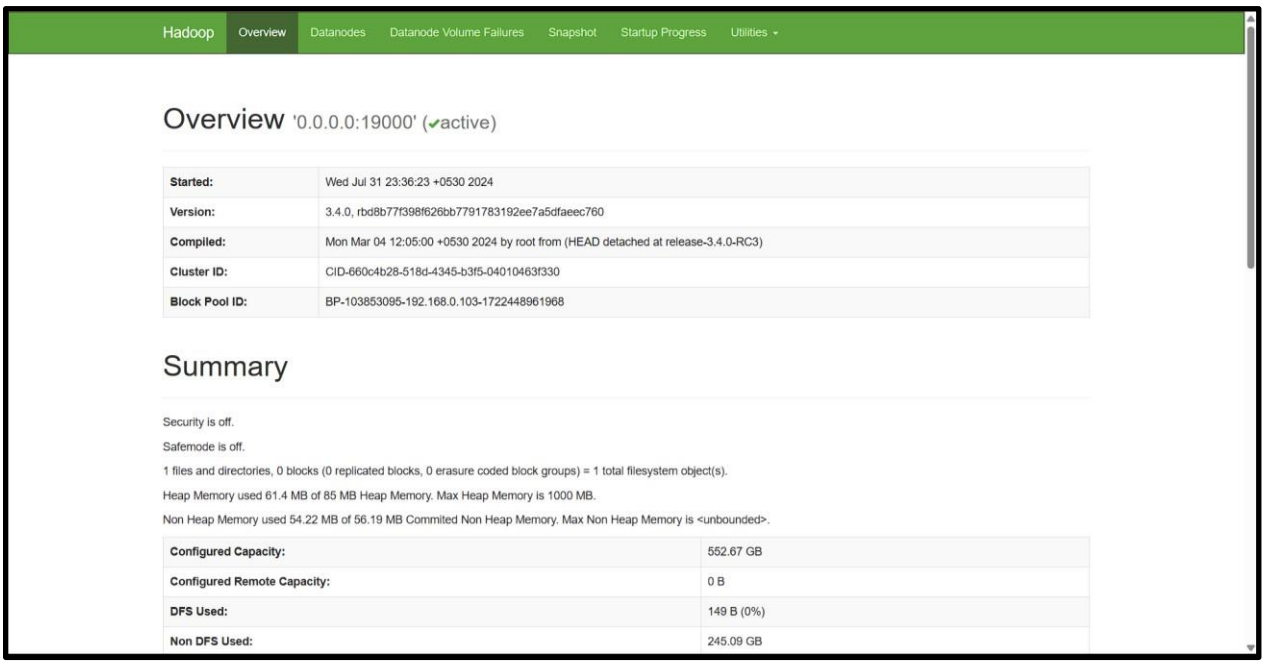
12. Initialise HDFS using the following command hdfs namenode -format



13. Start HDFS Daemons by going to the sbin folder of Hadoop installation and running the command start-dfs.cmd. Processes will start running.

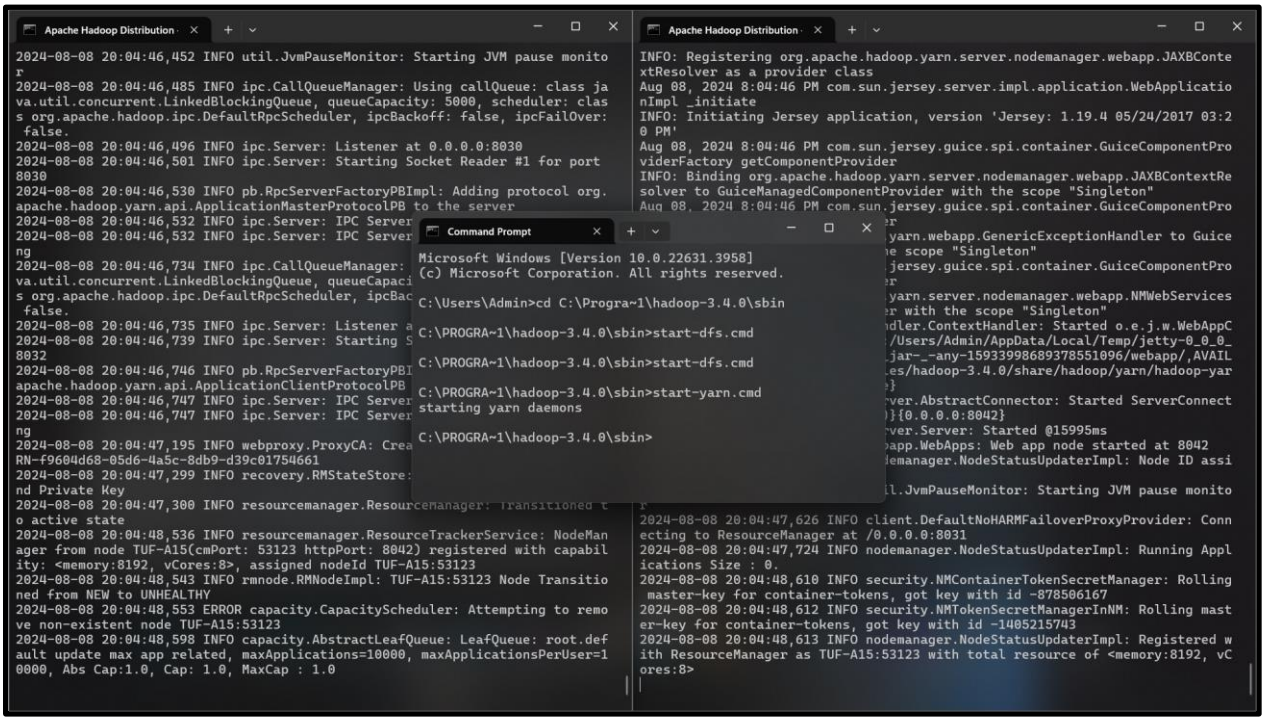


14. Verify HDFS web portal UI through this link.

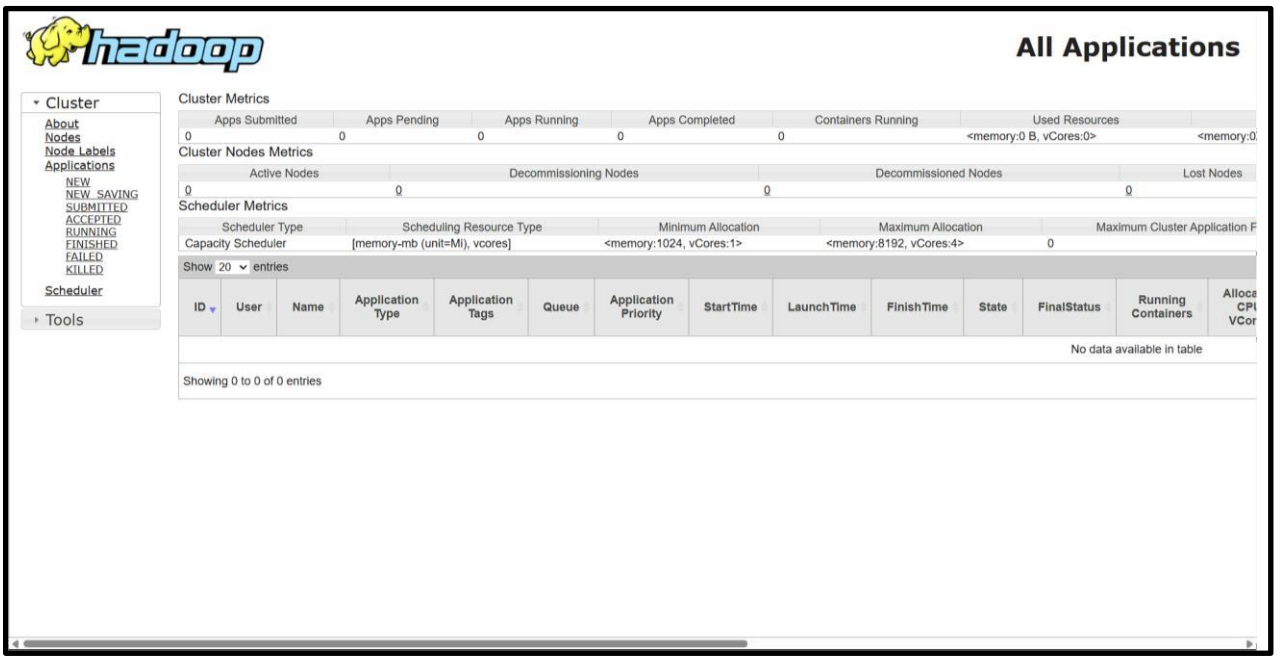


15. Start YARN Daemons by using the following command start-yarn.cmd.

Name: Vatsal Parekh
Roll No: 31031523022



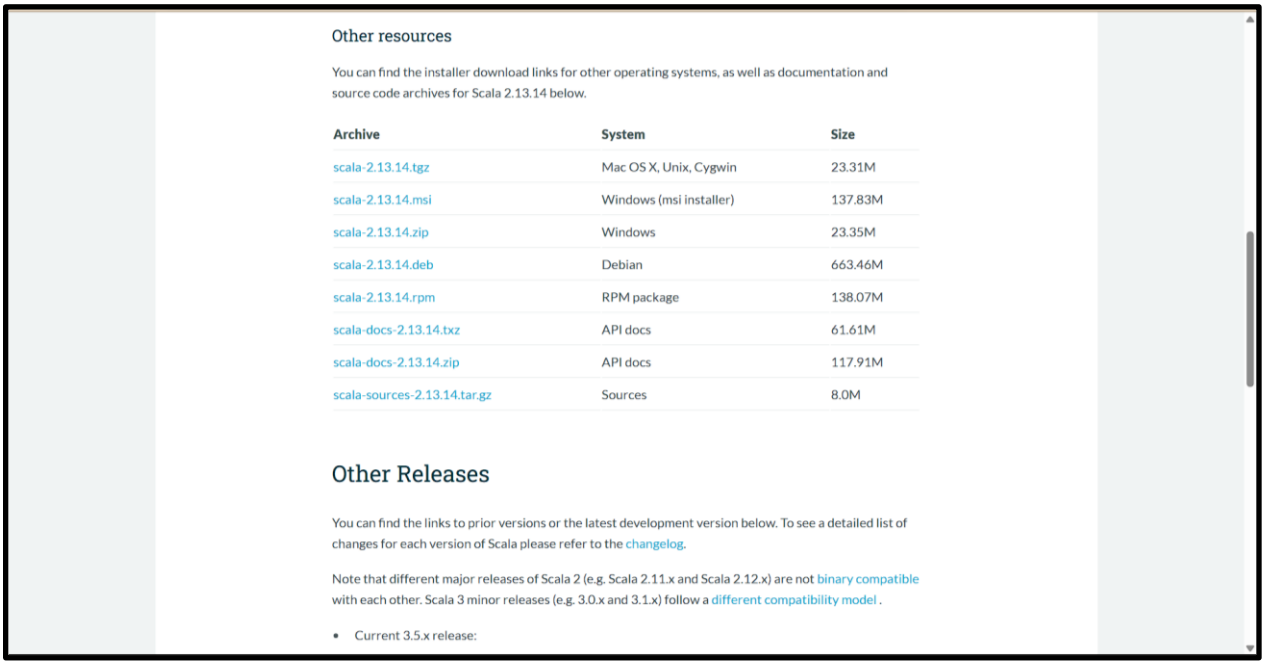
16. Verify YARN resource manager through this link.



Name: Vatsal Parekh
Roll No: 31031523022

Practical 2: Installation of Scala and Apache Spark

1. Go to this link and download Scala msi installer.



Other resources

You can find the installer download links for other operating systems, as well as documentation and source code archives for Scala 2.13.14 below.

Archive	System	Size
scala-2.13.14.tgz	Mac OS X, Unix, Cygwin	23.31M
scala-2.13.14.msi	Windows (msi installer)	137.83M
scala-2.13.14.zip	Windows	23.35M
scala-2.13.14.deb	Debian	663.46M
scala-2.13.14.rpm	RPM package	138.07M
scala-docs-2.13.14.txz	API docs	61.61M
scala-docs-2.13.14.zip	API docs	117.91M
scala-sources-2.13.14.tar.gz	Sources	8.0M

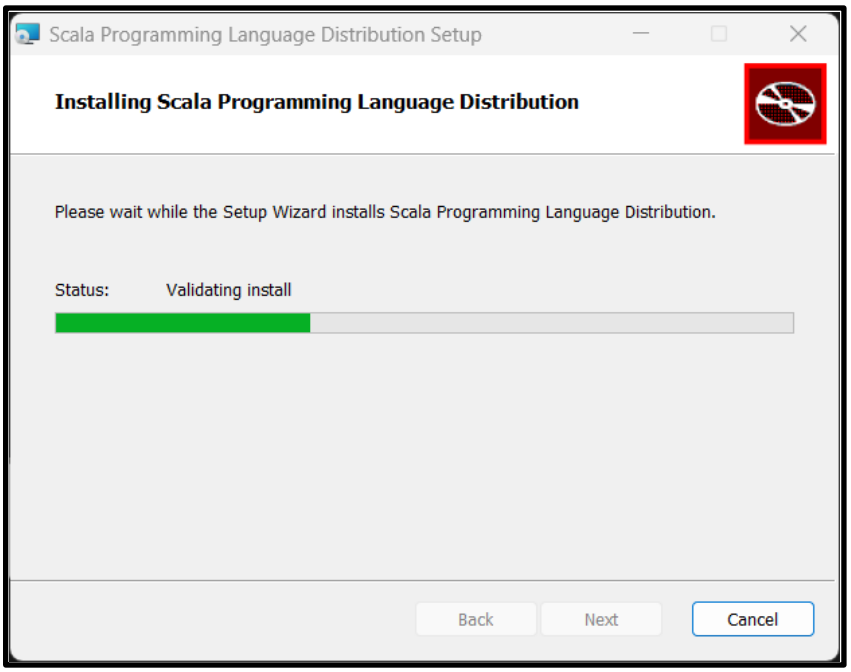
Other Releases

You can find the links to prior versions or the latest development version below. To see a detailed list of changes for each version of Scala please refer to the changelog.

Note that different major releases of Scala 2 (e.g. Scala 2.11.x and Scala 2.12.x) are not binary compatible with each other. Scala 3 minor releases (e.g. 3.0.x and 3.1.x) follow a different compatibility model .

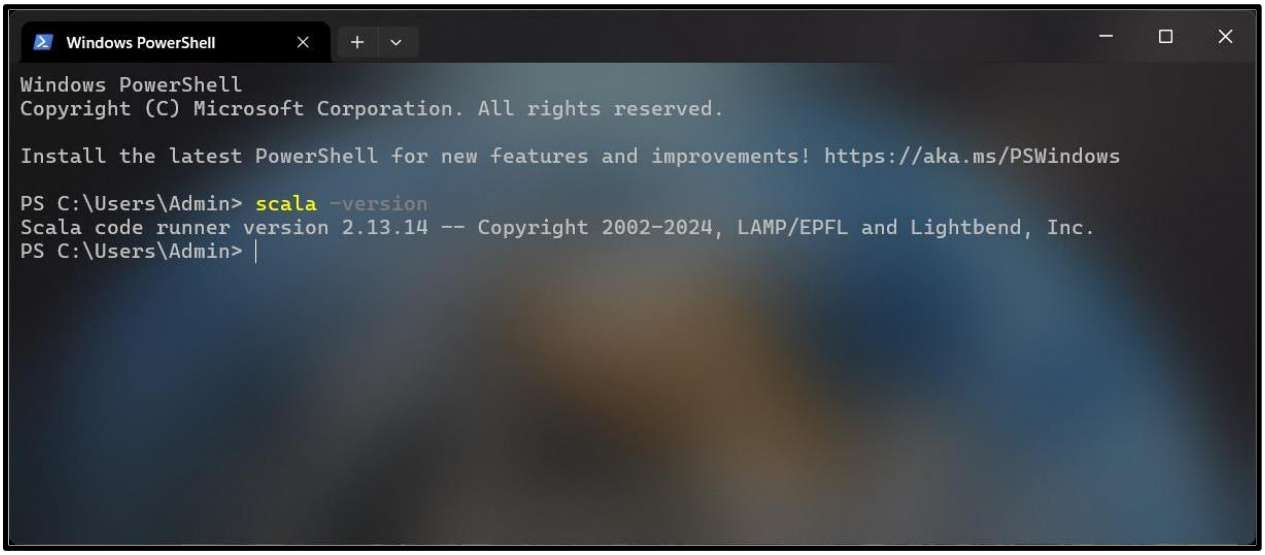
- Current 3.5.x release:

2. Run the installer.

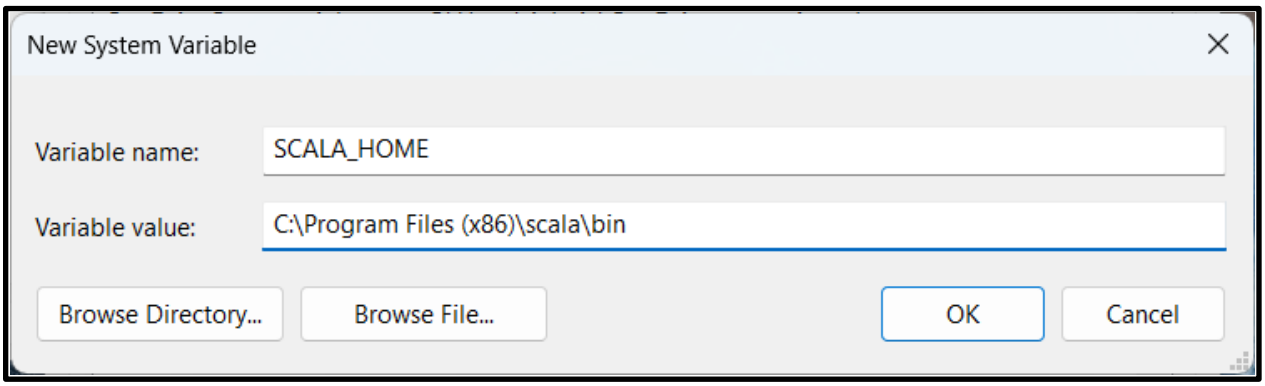


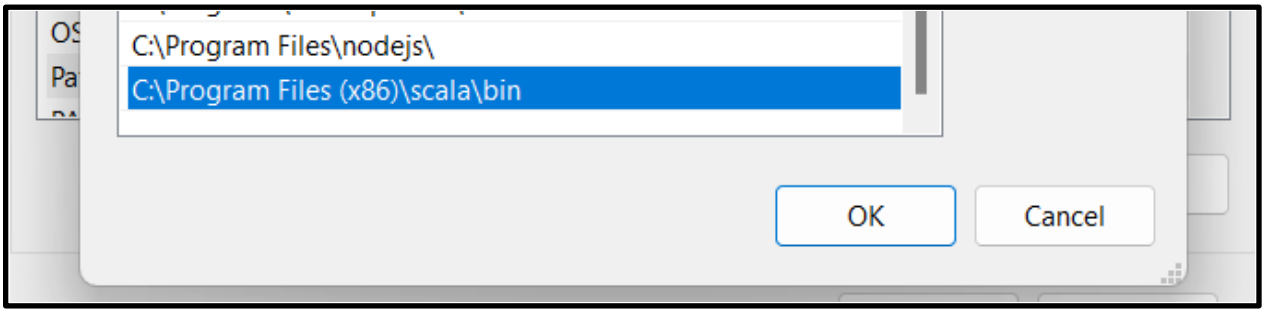
Name: Vatsal Parekh
Roll No: 31031523022

3. Using the command scala -version check if scala is installed.



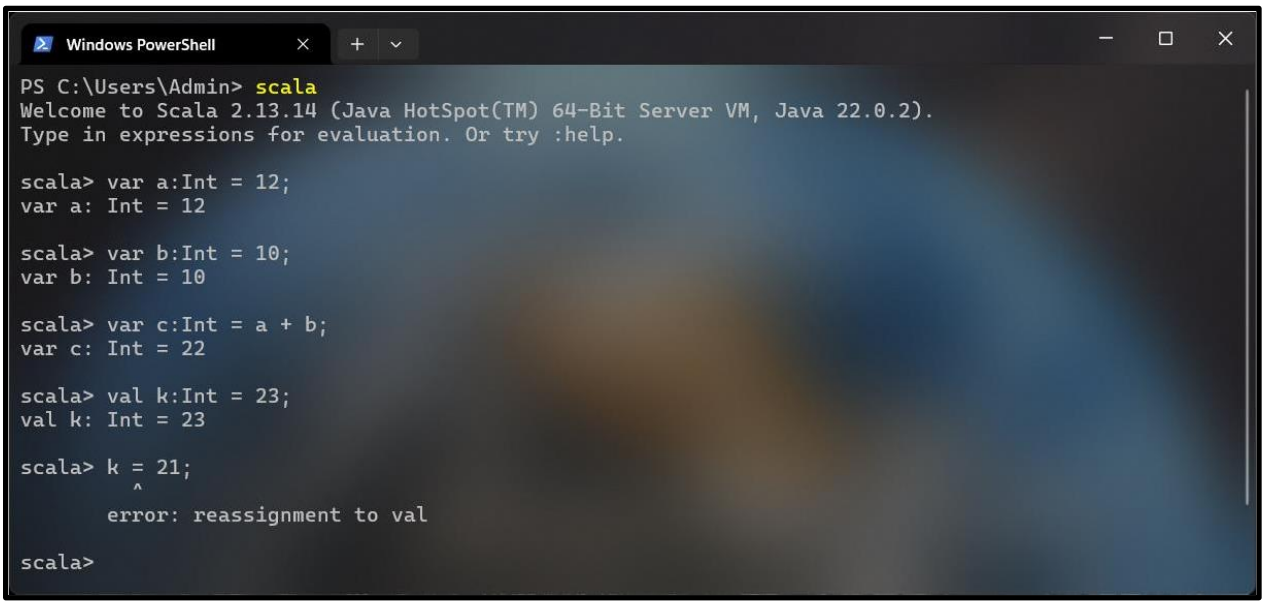
4. Configure the System Environment Variable.



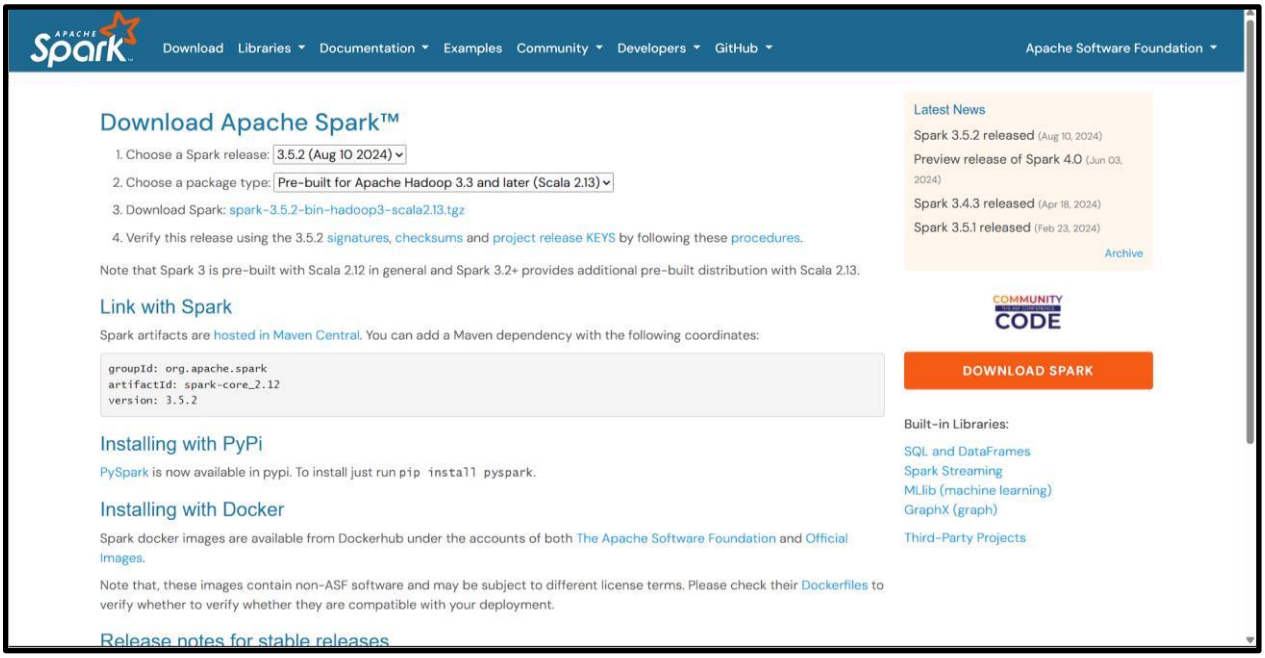


Name: Vatsal Parekh
Roll No: 31031523022

5. For mutable variables we use keyword ‘var’, whereas for immutable variables we use the keyword ‘val’. Error will be thrown if values are reassigned to an immutable variable.

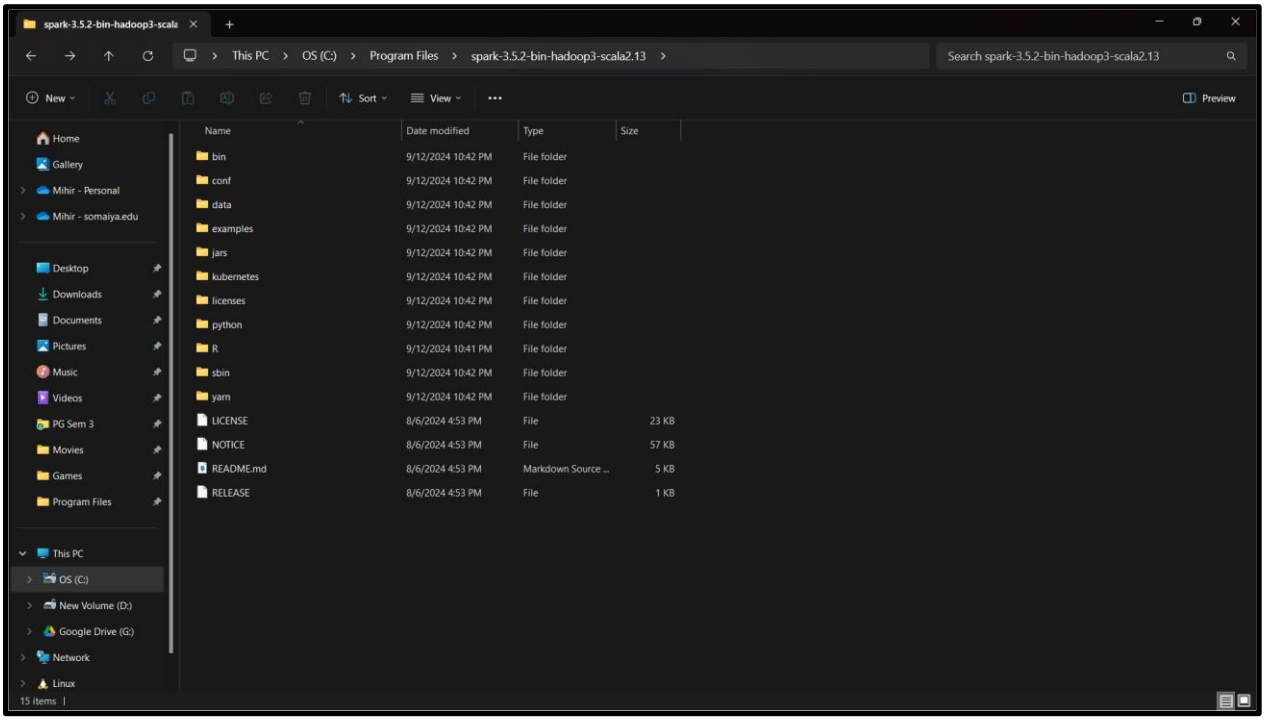


6. Download Apache Spark from here.

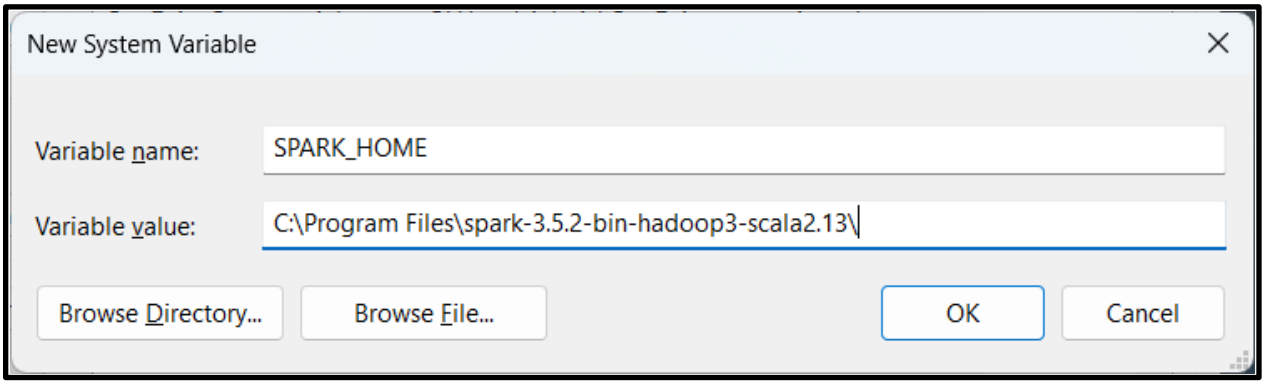


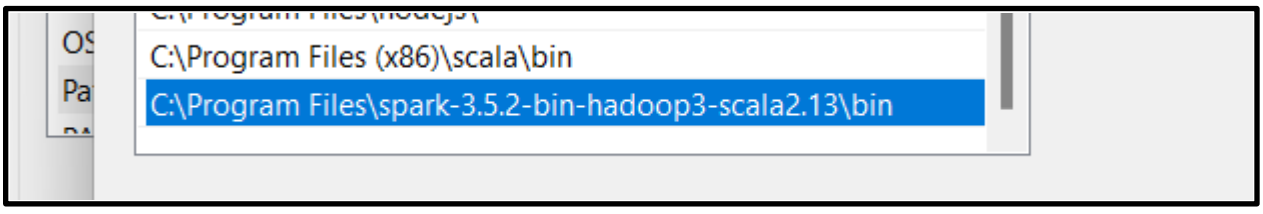
Name: Vatsal Parekh
Roll No: 31031523022

7. Extract the zip and paste it in Program Files.



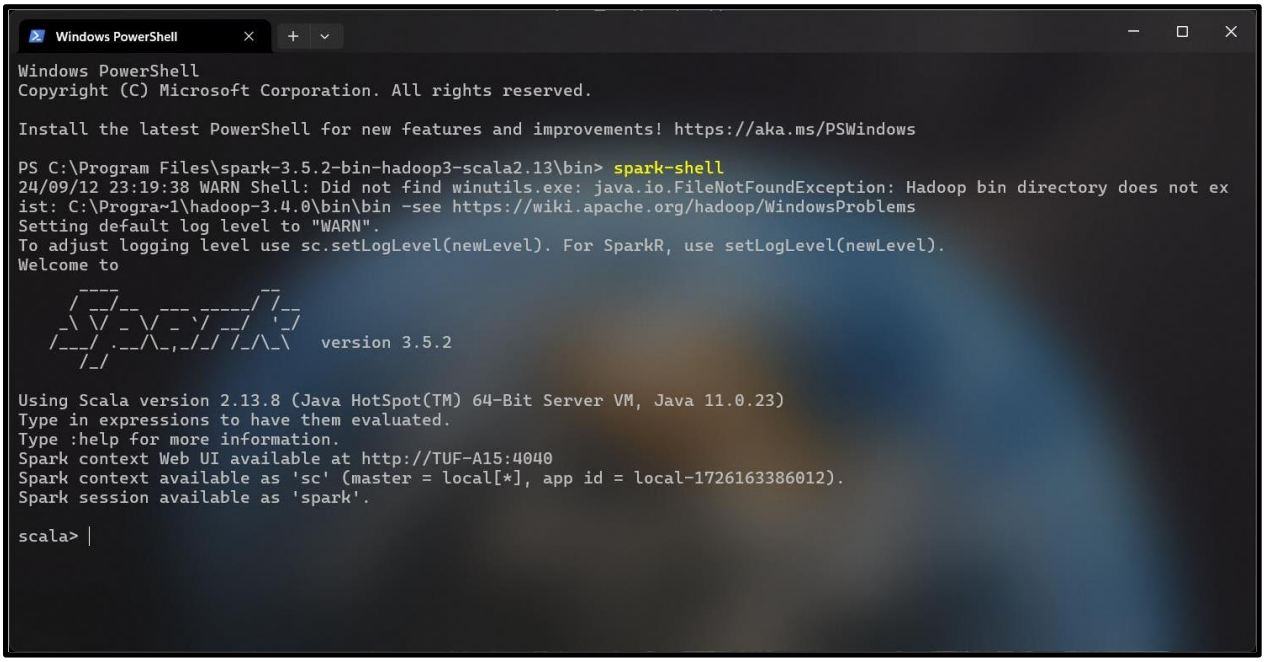
8. Configure the user and system environment variables.





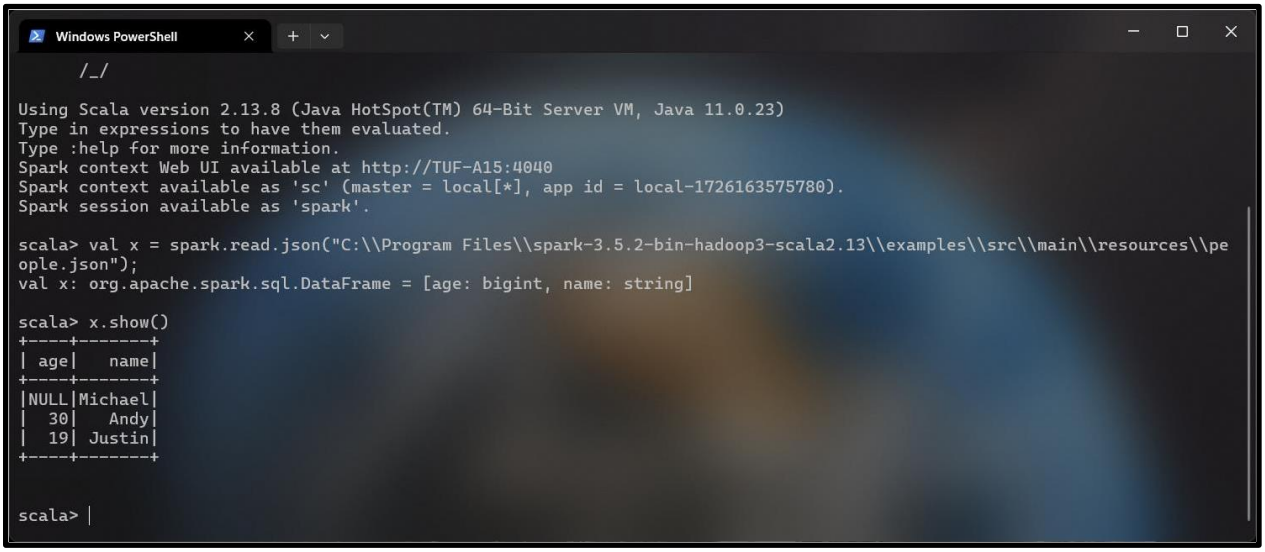
Name: Vatsal Parekh
Roll No: 31031523022

9. Run spark-shell in the bin folder of spark.



10. Run the following commands

```
val x = spark.read.json("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.json");
x.show()
```



x.printSchema()

```
scala> x.printSchema()
root
 |-- age: long (nullable = true)
 |-- name: string (nullable = true)
```

x.select($"name",$"age").show()

```
scala> x.select($"name",$"age").show()
+-------+----+
|   name| age|
+-------+----+
|Michael|NULL|
|   Andy|  30|
| Justin|  19|
+-------+----+
```

x.filter($"age">20).show()

```
scala> x.filter($"age">20).show()
+---+----+
|age|name|
+---+----+
| 30|Andy|
+---+----+
```

x.select($"age"+1).show()

```
scala> x.select($"age"+1).show()
+----------+
|(age + 1)|
+----------+
|      NULL|
|        31|
|        20|
+----------+
```

x.createOrReplaceTempView("people")

val sqlDF = spark.sql("Select * from people")

```
scala> x.createOrReplaceTempView("people")

scala> val sqlDF = spark.sql("Select * from people")
val sqlDF: org.apache.spark.sql.DataFrame = [age: bigint, name: string]
```

sqlDF.show()

```
scala> sqlDF.show()
+----+-------+
| age|   name|
+----+-------+
|NULL|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+
```

df.createGlobalTempView("people")

```
scala> df.createGlobalTempView("people")
24/09/19 10:40:27 WARN HiveConf: HiveConf of name hive.stats.jdbc.timeout does not exist
24/09/19 10:40:27 WARN HiveConf: HiveConf of name hive.stats.retries.wait does not exist
24/09/19 10:40:29 WARN ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not enabled so recording the schema version 2.3.0
24/09/19 10:40:29 WARN ObjectStore: setMetaStoreSchemaVersion called but recording version is disabled: version = 2.3.0, comment = Set by MetaStore UNKNOWN@172.23.1.71
24/09/19 10:40:29 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
```

spark.sql("SELECT * FROM global_temp.people").show()

```
scala> spark.sql("SELECT * FROM global_temp.people").show()
+----+-------+
| age|   name|
+----+-------+
|NULL|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+
```

spark.newSession().sql("SELECT * FROM global_temp.people").show()

```
scala> spark.newSession().sql("SELECT * FROM global_temp.people").show()
+----+-------+
| age|   name|
+----+-------+
|NULL|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+
```

case class Person(name: String, age: Long)

```
scala> case class Person(name: String, age: Long)
class Person
```

val caseClassDS = Seq(Person("Andy", 32)).toDS()

```
scala> val caseClassDS = Seq(Person("Andy", 32)).toDS()
val caseClassDS: org.apache.spark.sql.Dataset[Person] = [name: string, age: bigint]
```

caseClassDS.show()

```
scala> caseClassDS.show()
+----+---+
|name|age|
+----+---+
|Andy| 32|
+----+---+
```

val primitiveDS = Seq(1, 2, 3).toDS()

```
scala> val primitiveDS = Seq(1, 2, 3).toDS()
val primitiveDS: org.apache.spark.sql.Dataset[Int] = [value: int]
```

primitiveDS.map(_ + 1).collect()

```
scala> primitiveDS.map(_ + 1).collect()
val res8: Array[Int] = Array(2, 3, 4)
```

val path = "C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.json"

```
scala> val path = "C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.json"
val path: String = C:\Program Files\spark-3.5.2-bin-hadoop3-scala2.13\examples\src\main\resources\people.json
```

val peopleDS = spark.read.json(path).as[Person]

```
scala> val peopleDS = spark.read.json(path).as[Person]
val peopleDS: org.apache.spark.sql.Dataset[Person] = [age: bigint, name: string]
```

peopleDS.show()

```
scala> peopleDS.show()
+----+-------+
| age|   name|
+----+-------+
|NULL|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+
```

import spark.implicits._

```
scala> import spark.implicits._
import spark.implicits._
```

val peopleDF = spark.sparkContext.textFile("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.txt").map(_.split(",")).map(attributes => Person(attributes(0), attributes(1).trim.toInt)).toDF()

```
scala> val peopleDF = spark.sparkContext.textFile("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.txt").map(_.split(",")).map(attributes => Person(attributes(0), attributes(1).trim.toInt)).toDF()
val peopleDF: org.apache.spark.sql.DataFrame = [name: string, age: bigint]
```

peopleDF.createOrReplaceTempView("people")

val teenagersDF = spark.sql("SELECT name, age FROM people WHERE age BETWEEN 13 AND 19")

```
scala> peopleDF.createOrReplaceTempView("people")

scala> val teenagersDF = spark.sql("SELECT name, age FROM people WHERE age BETWEEN 13 AND 19")
val teenagersDF: org.apache.spark.sql.DataFrame = [name: string, age: bigint]
```

teenagersDF.map(teenager => "Name: " + teenager(0)).show()

```
scala> teenagersDF.map(teenager => "Name: " + teenager(0)).show()
+------------+
|       value|
+------------+
|Name: Justin|
+------------+
```

teenagersDF.map(teenager => "Name: " + teenager.getAs[String]("name")).show()

```
scala> teenagersDF.map(teenager => "Name: " + teenager.getAs[String]("name")).show()
+------------+
|       value|
+------------+
|Name: Justin|
+------------+
```

implicit val mapEncoder = org.apache.spark.sql.Encoders.kryo[Map[String, Any]]

```
scala> implicit val mapEncoder = org.apache.spark.sql.Encoders.kryo[Map[String, Any]]
val mapEncoder: org.apache.spark.sql.Encoder[Map[String,Any]] = class[value[0]: binary]
```

teenagersDF.map(teenager => teenager.getValuesMap[Any](List("name", "age"))).collect()

```
scala> teenagersDF.map(teenager => teenager.getValuesMap[Any](List("name", "age"))).collect()
val res29: Array[Map[String,Any]] = Array(Map(name -> Justin, age -> 19))
```

11. Perform the same operations on people.csv.

val a = spark.read.option("header", "true").csv("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.csv");

a.show()

```
scala> val a = spark.read.option("header", "true").csv("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\
src\\main\\resources\\people.csv");
val a: org.apache.spark.sql.DataFrame = [name;age;job: string]

scala> a.show()
+-----------------+
|      name;age;job|
+-----------------+
|Jorge;30;Developer|
|  Bob;32;Developer|
+-----------------+
```

a.printSchema()

```
scala> a.printSchema()
root
 |-- name;age;job: string (nullable = true)
```

12. Performing operations on custom data.

val mydata = spark.read.format("csv").option("inferschema", "true").option("header", "true").load("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\banking.csv")

```
scala> val mydata = spark.read.format("csv").option("inferschema", "true").option("header", "true").load("C:\\Program Fi
les\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\banking.csv")
val mydata: org.apache.spark.sql.DataFrame = [age: int, job: string ... 19 more fields]
```

mydata.show()

Name: Vatsal Parekh
Roll No: 31031523022

```
scala> mydata.show()
+---+-----------+-------+--------------------+-------+-------+----+---------+-----+-----------+--------+--------+-----+--------+----------+-----------+--------------+--------------+---------+-----------+---+
|age|        job|marital|           education|default|housing|loan|  contact|month|day_of_week|duration|campaign|pdays|previous|  poutcome|emp_var_rate|cons_price_idx|cons_conf_idx|euribor3m|nr_employed|  y|
+---+-----------+-------+--------------------+-------+-------+----+---------+-----+-----------+--------+--------+-----+--------+----------+-----------+--------------+--------------+---------+-----------+---+
| 44| blue-collar|married|            basic.4y|unknown|    yes|  no| cellular|  aug|        thu|     210|       1|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.963|     5228.1|  0|
| 53| technician|married|             unknown|     no|     no|  no| cellular|  nov|        fri|     138|       1|  999|       0|nonexistent|       -0.1|         93.2|         -42.0|    4.021|     5195.8|  0|
| 28| management| single|university.degree|     no|    yes|  no| cellular|  jun|        thu|     339|       3|    6|       2|   success|       -1.7|        94.055|         -39.8|    0.729|     4991.6|  1|
| 39|   services|married|        high.school|     no|     no|  no| cellular|  apr|        fri|     185|       2|  999|       0|nonexistent|       -1.8|        93.075|         -47.1|    1.405|     5099.1|  0|
| 55|    retired|married|            basic.4y|     no|    yes|  no| cellular|  aug|        fri|     137|       1|    3|       1|   success|       -2.9|        92.201|         -31.4|    0.869|     5076.2|  1|
| 30| management|divorced|            basic.4y|     no|    yes|  no| cellular|  jul|        tue|      68|       8|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.961|     5228.1|  0|
| 37| blue-collar|married|            basic.4y|     no|    yes|  no| cellular|  may|        thu|     204|       1|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.327|     5099.1|  0|
| 39| blue-collar|divorced|           basic.9y|     no|    yes|  no| cellular|  may|        fri|     191|       1|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.313|     5099.1|  0|
| 36|      admin.|married|university.degree|     no|     no|  no| cellular|  jun|        mon|     174|       1|    3|       1|   success|       -2.9|        92.963|         -40.8|    1.266|     5076.2|  1|
| 27| blue-collar| single|            basic.4y|     no|    yes|  no| cellular|  apr|        thu|     191|       2|  999|       1|   failure|       -1.8|        93.075|         -47.1|    1.41|     5099.1|  0|
| 34|  housemaid| single|university.degree|     no|     no|  no|telephone|  may|        fri|      62|       2|  999|       0|nonexistent|        1.1|        93.994|         -36.4|    4.864|     5191.0|  0|
| 41| management|married|university.degree|     no|    yes|  no| cellular|  aug|        thu|     789|       1|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.964|     5228.1|  0|
| 55| management|married|university.degree|     no|     no|  no| cellular|  aug|        mon|     372|       3|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.965|     5228.1|  1|
| 33|   services|divorced|        high.school|     no|    yes|  no| cellular|  may|        tue|      75|       5|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.291|     5099.1|  0|
| 26|      admin.|married|        high.school|     no|     no| yes|telephone|  jun|        mon|    1021|       1|  999|       0|nonexistent|        1.4|        94.465|         -41.8|    4.96|     5228.1|  0|
| 52|   services|married|        high.school|unknown|    yes|  no| cellular|  jul|        thu|     117|       2|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.962|     5228.1|  0|
| 35|   services|married|        high.school|     no|     no|  no| cellular|  apr|        thu|    1034|       2|  999|       0|nonexistent|       -1.8|        93.075|         -47.1|    1.365|     5099.1|  1|
| 27|      admin.| single|university.degree|     no|     no|  no|telephone|  oct|        tue|     540|       1|  999|       0|nonexistent|       -0.1|        93.798|         -40.4|    4.86|     5195.8|  1|
| 28| blue-collar|married|           basic.9y|unknown|     no|  no|telephone|  may|        thu|     140|       1|  999|       0|nonexistent|        1.1|        93.994|         -36.4|    4.86|     5191.0|  0|
| 26|  unemployed| single|            basic.9y|     no|    yes| yes| cellular|  jul|        mon|     104|       4|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.96|     5228.1|  0|
| 41|  unemployed|married|           basic.9y|unknown|    yes|  no|telephone|  apr|        fri|     246|       1|  999|       1|   failure|       -1.8|        93.075|         -47.1|    1.405|     5099.1|  0|
| 35| blue-collar| single|             unknown|     no|     no| yes|telephone|  jun|        fri|    1114|       1|  999|       0|nonexistent|        1.4|        94.465|         -41.8|    4.967|     5228.1|  0|
| 40|      admin.|married|university.degree|unknown|    yes|  no|telephone|  jul|        wed|     340|       1|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.963|     5228.1|  0|
| 32|  technician| single|professional.course|     no|     no|  no| cellular|  jul|        thu|      35|       1|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.968|     5228.1|  0|
| 41| blue-collar|married|        high.school|     no|    yes| yes| cellular|  jul|        thu|     241|       3|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.962|     5228.1|  0|
| 30|entrepreneur| single|university.degree|     no|    yes|  no| cellular|  may|        tue|     160|       2|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.344|     5099.1|  0|
| 49|  technician|divorced|             unknown|     no|    yes| yes| cellular|  oct|        thu|      81|       1|  999|       0|nonexistent|       -3.4|        92.431|         -26.9|    0.754|     5017.5|  0|
| 37|      admin.|married|        high.school|     no|    yes|  no| cellular|  apr|        thu|     226|       1|    2|       1|   success|       -1.8|        93.075|         -47.1|    1.365|     5099.1|  0|
| 35| blue-collar|married|            basic.6y|unknown|    yes|  no| cellular|  may|        fri|     746|       2|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.313|     5099.1|  0|
| 38| blue-collar| single|            basic.6y|unknown|     no|  no|telephone|  jul|        tue|      41|       5|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.961|     5228.1|  0|
| 47|   services|divorced|        high.school|     no|     no|  no|telephone|  jun|        tue|     115|       1|  999|       0|nonexistent|        1.4|        94.465|         -41.8|    4.961|     5228.1|  0|
| 46|      admin.|married|university.degree|     no|     no|  no| cellular|  may|        thu|     227|       2|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.327|     5099.1|  0|
| 27|  technician|married|professional.course|     no|    yes|  no| cellular|  may|        mon|     275|       1|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.299|     5099.1|  0|
| 29|  technician|married|        high.school|     no|    yes|  no| cellular|  jun|        fri|     365|       3|  999|       0|nonexistent|       -2.9|        92.963|         -40.8|    1.268|     5076.2|  1|
| 32|   services|divorced|            basic.9y|unknown|     no|  no| cellular|  jul|        wed|      68|       2|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.963|     5228.1|  0|
| 36| blue-collar|married|            basic.6y|unknown|     no|  no| cellular|  may|        wed|      54|       1|  999|       2|   failure|       -1.8|        92.893|         -46.2|    1.334|     5099.1|  0|
| 29| blue-collar|married|            basic.4y|     no|    yes| yes| cellular|  jul|        mon|      93|       1|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.96|     5228.1|  0|
| 47|  technician|married|        high.school|unknown|     no|  no| cellular|  apr|        mon|     267|       2|  999|       1|   failure|       -1.8|        93.075|         -47.1|    1.405|     5099.1|  0|
| 44| blue-collar|married|            basic.4y|     no|    yes|  no|telephone|  jun|        mon|      17|      25|  999|       0|nonexistent|        1.4|        94.465|         -41.8|    4.96|     5228.1|  0|
| 50|  management|married|university.degree|     no|    yes| yes| cellular|  aug|        thu|     121|      11|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.962|     5228.1|  0|
| 36| blue-collar|married|            basic.9y|     no|     no|  no|telephone|  may|        thu|     294|       2|  999|       0|nonexistent|        1.1|        93.994|         -36.4|    4.86|     5191.0|  0|
| 42| blue-collar|married|            basic.4y|     no|    yes|  no|telephone|  may|        wed|      24|       1|  999|       0|nonexistent|        1.1|        93.994|         -36.4|    4.857|     5191.0|  0|
| 44| blue-collar|married|            basic.4y|     no|     no|  no| cellular|  jul|        tue|     345|       5|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.961|     5228.1|  0|
| 72|    retired|divorced|            basic.6y|     no|    yes|  no| cellular|  nov|        wed|     244|       2|  999|       0|nonexistent|       -3.4|        92.649|         -30.1|    0.715|     5017.5|  1|
| 48| blue-collar|married|           basic.9y|unknown|     no|  no| cellular|  aug|        tue|     213|       3|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.966|     5228.1|  0|
| 36|  management|married|        high.school|unknown|    yes|  no| cellular|  nov|        thu|     371|       1|  999|       0|nonexistent|       -0.1|         93.2|         -42.0|    4.076|     5195.8|  0|
| 35|  housemaid|married|            basic.4y|     no|    yes|  no| cellular|  aug|        mon|      87|       1|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.965|     5228.1|  0|
| 43|entrepreneur|married|            basic.6y|     no|    yes|  no| cellular|  may|        mon|     195|       3|  999|       1|   failure|       -1.8|        92.893|         -46.2|    1.354|     5099.1|  0|
| 56|    retired|married|            basic.4y|unknown|    yes|  no| cellular|  aug|        wed|     102|       1|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.967|     5228.1|  0|
| 42| blue-collar|married|           basic.9y|unknown|    yes|  no|telephone|  jun|        fri|      93|       2|  999|       0|nonexistent|        1.4|        94.465|         -41.8|    4.959|     5228.1|  0|
+---+-----------+-------+--------------------+-------+-------+----+---------+-----+-----------+--------+--------+-----+--------+----------+-----------+--------------+--------------+---------+-----------+---+
only showing top 20 rows
```

mydata.show(50)

```
scala> mydata.show(50)
+---+-----------+-------+--------------------+-------+-------+----+---------+-----+-----------+--------+--------+-----+--------+----------+-----------+--------------+--------------+---------+-----------+---+
|age|        job|marital|           education|default|housing|loan|  contact|month|day_of_week|duration|campaign|pdays|previous|  poutcome|emp_var_rate|cons_price_idx|cons_conf_idx|euribor3m|nr_employed|  y|
+---+-----------+-------+--------------------+-------+-------+----+---------+-----+-----------+--------+--------+-----+--------+----------+-----------+--------------+--------------+---------+-----------+---+
| 44| blue-collar|married|            basic.4y|unknown|    yes|  no| cellular|  aug|        thu|     210|       1|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.963|     5228.1|  0|
| 53| technician|married|             unknown|     no|     no|  no| cellular|  nov|        fri|     138|       1|  999|       0|nonexistent|       -0.1|         93.2|         -42.0|    4.021|     5195.8|  0|
| 28| management| single|university.degree|     no|    yes|  no| cellular|  jun|        thu|     339|       3|    6|       2|   success|       -1.7|        94.055|         -39.8|    0.729|     4991.6|  1|
| 39|   services|married|        high.school|     no|     no|  no| cellular|  apr|        fri|     185|       2|  999|       0|nonexistent|       -1.8|        93.075|         -47.1|    1.405|     5099.1|  0|
| 55|    retired|married|            basic.4y|     no|    yes|  no| cellular|  aug|        fri|     137|       1|    3|       1|   success|       -2.9|        92.201|         -31.4|    0.869|     5076.2|  1|
| 30| management|divorced|            basic.4y|     no|    yes|  no| cellular|  jul|        tue|      68|       8|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.961|     5228.1|  0|
| 37| blue-collar|married|            basic.4y|     no|    yes|  no| cellular|  may|        thu|     204|       1|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.327|     5099.1|  0|
| 39| blue-collar|divorced|           basic.9y|     no|    yes|  no| cellular|  may|        fri|     191|       1|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.313|     5099.1|  0|
| 36|      admin.|married|university.degree|     no|     no|  no| cellular|  jun|        mon|     174|       1|    3|       1|   success|       -2.9|        92.963|         -40.8|    1.266|     5076.2|  1|
| 27| blue-collar| single|            basic.4y|     no|    yes|  no| cellular|  apr|        thu|     191|       2|  999|       1|   failure|       -1.8|        93.075|         -47.1|    1.41|     5099.1|  0|
| 34|  housemaid| single|university.degree|     no|     no|  no|telephone|  may|        fri|      62|       2|  999|       0|nonexistent|        1.1|        93.994|         -36.4|    4.864|     5191.0|  0|
| 41| management|married|university.degree|     no|    yes|  no| cellular|  aug|        thu|     789|       1|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.964|     5228.1|  0|
| 55| management|married|university.degree|     no|     no|  no| cellular|  aug|        mon|     372|       3|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.965|     5228.1|  1|
| 33|   services|divorced|        high.school|     no|    yes|  no| cellular|  may|        tue|      75|       5|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.291|     5099.1|  0|
| 26|      admin.|married|        high.school|     no|     no| yes|telephone|  jun|        mon|    1021|       1|  999|       0|nonexistent|        1.4|        94.465|         -41.8|    4.96|     5228.1|  0|
| 52|   services|married|        high.school|unknown|    yes|  no| cellular|  jul|        thu|     117|       2|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.962|     5228.1|  0|
| 35|   services|married|        high.school|     no|     no|  no| cellular|  apr|        thu|    1034|       2|  999|       0|nonexistent|       -1.8|        93.075|         -47.1|    1.365|     5099.1|  1|
| 27|      admin.| single|university.degree|     no|     no|  no|telephone|  oct|        tue|     540|       1|  999|       0|nonexistent|       -0.1|        93.798|         -40.4|    4.86|     5195.8|  1|
| 28| blue-collar|married|           basic.9y|unknown|     no|  no|telephone|  may|        thu|     140|       1|  999|       0|nonexistent|        1.1|        93.994|         -36.4|    4.86|     5191.0|  0|
| 26|  unemployed| single|            basic.9y|     no|    yes| yes| cellular|  jul|        mon|     104|       4|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.96|     5228.1|  0|
| 41|  unemployed|married|           basic.9y|unknown|    yes|  no|telephone|  apr|        fri|     246|       1|  999|       1|   failure|       -1.8|        93.075|         -47.1|    1.405|     5099.1|  0|
| 35| blue-collar| single|             unknown|     no|     no| yes|telephone|  jun|        fri|    1114|       1|  999|       0|nonexistent|        1.4|        94.465|         -41.8|    4.967|     5228.1|  0|
| 40|      admin.|married|university.degree|unknown|    yes|  no|telephone|  jul|        wed|     340|       1|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.963|     5228.1|  0|
| 32|  technician| single|professional.course|     no|     no|  no| cellular|  jul|        thu|      35|       1|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.968|     5228.1|  0|
| 41| blue-collar|married|        high.school|     no|    yes| yes| cellular|  jul|        thu|     241|       3|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.962|     5228.1|  0|
| 30|entrepreneur| single|university.degree|     no|    yes|  no| cellular|  may|        tue|     160|       2|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.344|     5099.1|  0|
| 49|  technician|divorced|             unknown|     no|    yes| yes| cellular|  oct|        thu|      81|       1|  999|       0|nonexistent|       -3.4|        92.431|         -26.9|    0.754|     5017.5|  0|
| 37|      admin.|married|        high.school|     no|    yes|  no| cellular|  apr|        thu|     226|       1|    2|       1|   success|       -1.8|        93.075|         -47.1|    1.365|     5099.1|  0|
| 35| blue-collar|married|            basic.6y|unknown|    yes|  no| cellular|  may|        fri|     746|       2|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.313|     5099.1|  0|
| 38| blue-collar| single|            basic.6y|unknown|     no|  no|telephone|  jul|        tue|      41|       5|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.961|     5228.1|  0|
| 47|   services|divorced|        high.school|     no|     no|  no|telephone|  jun|        tue|     115|       1|  999|       0|nonexistent|        1.4|        94.465|         -41.8|    4.961|     5228.1|  0|
| 46|      admin.|married|university.degree|     no|     no|  no| cellular|  may|        thu|     227|       2|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.327|     5099.1|  0|
| 27|  technician|married|professional.course|     no|    yes|  no| cellular|  may|        mon|     275|       1|  999|       0|nonexistent|       -1.8|        92.893|         -46.2|    1.299|     5099.1|  0|
| 29|  technician|married|        high.school|     no|    yes|  no| cellular|  jun|        fri|     365|       3|  999|       0|nonexistent|       -2.9|        92.963|         -40.8|    1.268|     5076.2|  1|
| 32|   services|divorced|            basic.9y|unknown|     no|  no| cellular|  jul|        wed|      68|       2|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.963|     5228.1|  0|
| 36| blue-collar|married|            basic.6y|unknown|     no|  no| cellular|  may|        wed|      54|       1|  999|       2|   failure|       -1.8|        92.893|         -46.2|    1.334|     5099.1|  0|
| 29| blue-collar|married|            basic.4y|     no|    yes| yes| cellular|  jul|        mon|      93|       1|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.96|     5228.1|  0|
| 47|  technician|married|        high.school|unknown|     no|  no| cellular|  apr|        mon|     267|       2|  999|       1|   failure|       -1.8|        93.075|         -47.1|    1.405|     5099.1|  0|
| 44| blue-collar|married|            basic.4y|     no|    yes|  no|telephone|  jun|        mon|      17|      25|  999|       0|nonexistent|        1.4|        94.465|         -41.8|    4.96|     5228.1|  0|
| 50|  management|married|university.degree|     no|    yes| yes| cellular|  aug|        thu|     121|      11|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.962|     5228.1|  0|
| 36| blue-collar|married|            basic.9y|     no|     no|  no|telephone|  may|        thu|     294|       2|  999|       0|nonexistent|        1.1|        93.994|         -36.4|    4.86|     5191.0|  0|
| 42| blue-collar|married|            basic.4y|     no|    yes|  no|telephone|  may|        wed|      24|       1|  999|       0|nonexistent|        1.1|        93.994|         -36.4|    4.857|     5191.0|  0|
| 44| blue-collar|married|            basic.4y|     no|     no|  no| cellular|  jul|        tue|     345|       5|  999|       0|nonexistent|        1.4|        93.918|         -42.7|    4.961|     5228.1|  0|
| 72|    retired|divorced|            basic.6y|     no|    yes|  no| cellular|  nov|        wed|     244|       2|  999|       0|nonexistent|       -3.4|        92.649|         -30.1|    0.715|     5017.5|  1|
| 48| blue-collar|married|           basic.9y|unknown|     no|  no| cellular|  aug|        tue|     213|       3|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.966|     5228.1|  0|
| 36|  management|married|        high.school|unknown|    yes|  no| cellular|  nov|        thu|     371|       1|  999|       0|nonexistent|       -0.1|         93.2|         -42.0|    4.076|     5195.8|  0|
| 35|  housemaid|married|            basic.4y|     no|    yes|  no| cellular|  aug|        mon|      87|       1|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.965|     5228.1|  0|
| 43|entrepreneur|married|            basic.6y|     no|    yes|  no| cellular|  may|        mon|     195|       3|  999|       1|   failure|       -1.8|        92.893|         -46.2|    1.354|     5099.1|  0|
| 56|    retired|married|            basic.4y|unknown|    yes|  no| cellular|  aug|        wed|     102|       1|  999|       0|nonexistent|        1.4|        93.444|         -36.1|    4.967|     5228.1|  0|
| 42| blue-collar|married|           basic.9y|unknown|    yes|  no|telephone|  jun|        fri|      93|       2|  999|       0|nonexistent|        1.4|        94.465|         -41.8|    4.959|     5228.1|  0|
+---+-----------+-------+--------------------+-------+-------+----+---------+-----+-----------+--------+--------+-----+--------+----------+-----------+--------------+--------------+---------+-----------+---+
only showing top 50 rows
```

mydata.select($"age", $"y").show()

```
scala> mydata.select($"age", $"y").show()
+---+---+
|age|  y|
+---+---+
| 44|  0|
| 53|  0|
| 28|  1|
| 39|  0|
| 55|  1|
| 30|  0|
| 37|  0|
| 39|  0|
| 36|  1|
| 27|  0|
| 34|  0|
| 41|  0|
| 55|  1|
| 33|  0|
| 26|  0|
| 52|  0|
| 35|  1|
| 27|  1|
| 28|  0|
| 26|  0|
+---+---+
only showing top 20 rows
```

mydata.count()

```
scala> mydata.count()
val res39: Long = 41188
```

mydata.count.toDouble

```
scala> mydata.count.toDouble
warning: 1 deprecation (since 2.13.3); for details, enable `:setting -deprecation` or `:replay -deprecation`
val res40: Double = 41188.0
```

Practical 3: Spark GraphX

Name: Vatsal Parekh
Roll No: 31031523022

```
import org.apache.spark._
```

```
import org.apache.spark.rdd.RDD
```

```
import org.apache.spark.graphx._
```

```
scala> import org.apache.spark._
import org.apache.spark._

scala> import org.apache.spark.rdd.RDD
import org.apache.spark.rdd.RDD

scala> import org.apache.spark.graphx._
import org.apache.spark.graphx._
```

```
val vertices = Array((1L,("A")),(2L,("B")),(3L,("C")))
```

```
scala> val vertices = Array((1L,("A")),(2L,("B")),(3L,("C")))
val vertices: Array[(Long, String)] = Array((1,A), (2,B), (3,C))
```

```
val vRDD = sc.parallelize(vertices)
```

```
scala> val vRDD = sc.parallelize(vertices)
warning: 1 deprecation (since 2.13.0); for details, enable `:setting -deprecation` or `:replay -deprec
ation`
val vRDD: org.apache.spark.rdd.RDD[(Long, String)] = ParallelCollectionRDD[0] at parallelize at <conso
le>:1
```

```
vRDD.take(1)
```

```
vRDD.take(2)
```

```
scala> vRDD.take(1)
val res0: Array[(Long, String)] = Array((1,A))

scala> vRDD.take(2)
val res1: Array[(Long, String)] = Array((1,A), (2,B))
```

val edges = Array(Edge(1L,2L,1800),Edge(2L,3L,800),Edge(3L,1L,1400))

```
scala> val edges = Array(Edge(1L,2L,1800),Edge(2L,3L,800),Edge(3L,1L,1400))
val edges: Array[org.apache.spark.graphx.Edge[Int]] = Array(Edge(1,2,1800), Edge(2,3,800), Edge(3,1,14
00))
```

val eRDD = sc.parallelize(edges)

```
scala> val eRDD = sc.parallelize(edges)
warning: 1 deprecation (since 2.13.0); for details, enable `:setting -deprecation` or `:replay -deprec
ation`
val eRDD: org.apache.spark.rdd.RDD[org.apache.spark.graphx.Edge[Int]] = ParallelCollectionRDD[1] at pa
rallelize at <console>:1
```

eRDD.take(2)

```
scala> eRDD.take(2)
val res2: Array[org.apache.spark.graphx.Edge[Int]] = Array(Edge(1,2,1800), Edge(2,3,800))
```

val nowhere = "nowhere"

```
scala> val nowhere = "nowhere"
val nowhere: String = nowhere
```

val graph = Graph(vRDD,eRDD,nowhere)

```
scala> val graph = Graph(vRDD,eRDD,nowhere)
val graph: org.apache.spark.graphx.Graph[String,Int] = org.apache.spark.graphx.impl.GraphImpl@2e008502
```

Name: Vatsal Parekh
Roll No: 31031523022

#To check number of Airports

val numairports = graph.numVertices

```
scala> val numairports = graph.numVertices
val numairports: Long = 3
```

#To check routes

val numairports = graph.numEdges

```
scala> val numairports = graph.numEdges
val numairports: Long = 3
```

#Route having distance > 1000

(graph.edges.filter{case Edge(src,dst,prop)=>prop>1000}.collect.foreach(println))

```
scala> (graph.edges.filter{case Edge(src,dst,prop)=>prop>1000}.collect.foreach(println))
warning: 1 deprecation (since 2.13.3); for details, enable `:setting -deprecation` or `:replay -deprec
ation`
Edge(1,2,1800)
Edge(3,1,1400)
```

#Triplet Information

graph.triplets.take(3).foreach(println)

```
scala> graph.triplets.take(3).foreach(println)
((1,A),(2,B),1800)
((2,B),(3,C),800)
((3,C),(1,A),1400)
```

#Indegree

val i = graph.inDegrees

i.collect()

```
scala> val i = graph.inDegrees
val i: org.apache.spark.graphx.VertexRDD[Int] = VertexRDDImpl[25] at RDD at VertexRDD.scala:57

scala> i.collect()
val res5: Array[(org.apache.spark.graphx.VertexId, Int)] = Array((1,1), (2,1), (3,1))
```

#Outdegrees

val o = graph.outDegrees

o.collect()

```
scala> val o = graph.outDegrees
val o: org.apache.spark.graphx.VertexRDD[Int] = VertexRDDImpl[29] at RDD at VertexRDD.scala:57

scala> o.collect()
val res6: Array[(org.apache.spark.graphx.VertexId, Int)] = Array((1,1), (2,1), (3,1))
```

#Total Degree

val t = graph.degrees

t.collect()

```
scala> val t = graph.degrees
val t: org.apache.spark.graphx.VertexRDD[Int] = VertexRDDImpl[33] at RDD at VertexRDD.scala:57

scala> t.collect()
val res8: Array[(org.apache.spark.graphx.VertexId, Int)] = Array((1,2), (2,2), (3,2))
```

Name: Vatsal Parekh
Roll No: 31031523022

Practical 4: Working with PySpark

```
!pip install pyspark
```

```
import pyspark

import pandas as pd

pd.read_csv('data.csv')
```

	Name	Age	Experience	Salary
0	Willetta	27.0	52.0	62860.0
1	Merrie	45.0	34.0	57591.0
2	Joleen	43.0	42.0	61343.0
3	Nananne	25.0	33.0	75478.0
4	Jennica	56.0	10.0	71299.0
5	Lizzie	55.0	27.0	62050.0
6	Kaja	28.0	18.0	52610.0
7	Aubrie	55.0	18.0	77702.0
8	Ginnie	71.0	26.0	88700.0
9	Correy	47.0	25.0	83288.0
10	Rochette	24.0	12.0	56123.0
11	Hayley	55.0	51.0	72758.0
12	Mathilda	42.0	44.0	62825.0
13	Veda	61.0	31.0	68511.0
14	Jessy	59.0	41.0	85482.0
15	John	30.0	5.0	50000.0
16	Mike	25.0	NaN	NaN
17	NaN	40.0	NaN	NaN
18	Sarah	NaN	3.0	40000.0
19	Sarah	NaN	NaN	NaN

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName('Practise').getOrCreate()

# inferSchema determines the datatype of each column.

df = spark.read.csv('data.csv', header=True, inferSchema=True)

df.show()
```

```
+--------+----+----------+------+
|    Name| Age|Experience|Salary|
+--------+----+----------+------+
|Willetta|  27|        52| 62860|
|  Merrie|  45|        34| 57591|
|  Joleen|  43|        42| 61343|
| Nananne|  25|        33| 75478|
| Jennica|  56|        10| 71299|
|  Lizzie|  55|        27| 62050|
|    Kaja|  28|        18| 52610|
|  Aubrie|  55|        18| 77702|
|  Ginnie|  71|        26| 88700|
|  Correy|  47|        25| 83288|
|Rochette|  24|        12| 56123|
|  Hayley|  55|        51| 72758|
|Mathilda|  42|        44| 62825|
|    Veda|  61|        31| 68511|
|   Jessy|  59|        41| 85482|
|    John|  30|         5| 50000|
|    Mike|  25|      NULL|  NULL|
|    NULL|  40|      NULL|  NULL|
|   Sarah|NULL|         3| 40000|
|   Sarah|NULL|      NULL|  NULL|
+--------+----+----------+------+
```

```
# Type of the data

print(type(df))
```

```
<class 'pyspark.sql.dataframe.DataFrame'>
```

```
# Check Schema

df.printSchema()
```

```
root
 |-- Name: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Experience: integer (nullable = true)
 |-- Salary: integer (nullable = true)
```

```
# Get Column Names

df.columns
```

```
['Name', 'Age', 'Experience', 'Salary']
```

```
# First 3 Data Values

df.head(3)
```

```
[Row(Name='Willetta', Age=27, Experience=52, Salary=62860),
 Row(Name='Merrie', Age=45, Experience=34, Salary=57591),
 Row(Name='Joleen', Age=43, Experience=42, Salary=61343)]
```

```
# Selecting Specific Columns

df.select(['Name', 'Age']).show()
```

```
+--------+---+
|    Name|Age|
+--------+---+
|Willetta| 27|
|  Merrie| 45|
|  Joleen| 43|
| Nananne| 25|
| Jennica| 56|
|  Lizzie| 55|
|    Kaja| 28|
|  Aubrie| 55|
|  Ginnie| 71|
|  Correy| 47|
|Rochette| 24|
|  Hayley| 55|
|Mathilda| 42|
|    Veda| 61|
|   Jessy| 59|
|    John| 30|
|    Mike| 25|
|    NULL| 40|
|   Sarah|NULL|
|   Sarah|NULL|
+--------+---+
```

```
# Check Datatypes

df.dtypes
```

```
[('Name', 'string'), ('Age', 'int'), ('Experience', 'int'), ('Salary', 'int')]
```

```
# Describe dataset

df.describe().show()
```

```
+-------+--------+-----------------+------------------+-----------------+
|summary|    Name|              Age|        Experience|           Salary|
+-------+--------+-----------------+------------------+-----------------+
|  count|      19|               18|                17|               17|
|   mean|    NULL|43.77777777777778|27.764705882352942|66389.41176470589|
| stddev|    NULL|14.63901135988258| 15.29513571272214|13310.58934673266|
|    min|  Aubrie|               24|                 3|            40000|
|    max|Willetta|               71|                52|            88700|
+-------+--------+-----------------+------------------+-----------------+
```

Name: Vatsal Parekh
Roll No: 31031523022

```
# Adding columns in data frame

df = df.withColumn('Experience After 2 Years', df['Experience'] + 2)

df.show()
```

```
+--------+----+----------+------+----------------------+
|    Name| Age|Experience|Salary|Experience After 2 Years|
+--------+----+----------+------+----------------------+
|Willetta|  27|        52| 62860|                    54|
|  Merrie|  45|        34| 57591|                    36|
|  Joleen|  43|        42| 61343|                    44|
| Nananne|  25|        33| 75478|                    35|
| Jennica|  56|        10| 71299|                    12|
|  Lizzie|  55|        27| 62050|                    29|
|    Kaja|  28|        18| 52610|                    20|
|  Aubrie|  55|        18| 77702|                    20|
|  Ginnie|  71|        26| 88700|                    28|
|  Correy|  47|        25| 83288|                    27|
|Rochette|  24|        12| 56123|                    14|
|  Hayley|  55|        51| 72758|                    53|
|Mathilda|  42|        44| 62825|                    46|
|    Veda|  61|        31| 68511|                    33|
|   Jessy|  59|        41| 85482|                    43|
|    John|  30|         5| 50000|                     7|
|    Mike|  25|      NULL|  NULL|                  NULL|
|    NULL|  40|      NULL|  NULL|                  NULL|
|   Sarah|NULL|         3| 40000|                     5|
|   Sarah|NULL|      NULL|  NULL|                  NULL|
+--------+----+----------+------+----------------------+
```

```
# Drop the column

df = df.drop('Experience After 2 Years')

df.show()
```

```
+--------+----+----------+------+
|    Name| Age|Experience|Salary|
+--------+----+----------+------+
|Willetta|  27|        52| 62860|
|  Merrie|  45|        34| 57591|
|  Joleen|  43|        42| 61343|
| Nananne|  25|        33| 75478|
| Jennica|  56|        10| 71299|
|  Lizzie|  55|        27| 62050|
|    Kaja|  28|        18| 52610|
|  Aubrie|  55|        18| 77702|
|  Ginnie|  71|        26| 88700|
|  Correy|  47|        25| 83288|
|Rochette|  24|        12| 56123|
|  Hayley|  55|        51| 72758|
|Mathilda|  42|        44| 62825|
|    Veda|  61|        31| 68511|
|   Jessy|  59|        41| 85482|
|    John|  30|         5| 50000|
|    Mike|  25|      NULL|  NULL|
|    NULL|  40|      NULL|  NULL|
|   Sarah|NULL|         3| 40000|
|   Sarah|NULL|      NULL|  NULL|
+--------+----+----------+------+
```

```
# Rename the columns
```

```
df.withColumnRenamed('Name', 'New Name').show()
```

```
+--------+----+----------+------+
|New Name| Age|Experience|Salary|
+--------+----+----------+------+
|Willetta|  27|        52| 62860|
|  Merrie|  45|        34| 57591|
|  Joleen|  43|        42| 61343|
| Nananne|  25|        33| 75478|
| Jennica|  56|        10| 71299|
|  Lizzie|  55|        27| 62050|
|    Kaja|  28|        18| 52610|
|  Aubrie|  55|        18| 77702|
|  Ginnie|  71|        26| 88700|
|  Correy|  47|        25| 83288|
|Rochette|  24|        12| 56123|
|  Hayley|  55|        51| 72758|
|Mathilda|  42|        44| 62825|
|    Veda|  61|        31| 68511|
|   Jessy|  59|        41| 85482|
|    John|  30|         5| 50000|
|    Mike|  25|      NULL|  NULL|
|    NULL|  40|      NULL|  NULL|
|   Sarah|NULL|         3| 40000|
|   Sarah|NULL|      NULL|  NULL|
+--------+----+----------+------+
```

```
df.na.drop().show()
```

```
+--------+---+----------+------+
|    Name|Age|Experience|Salary|
+--------+---+----------+------+
|Willetta| 27|        52| 62860|
|  Merrie| 45|        34| 57591|
|  Joleen| 43|        42| 61343|
| Nananne| 25|        33| 75478|
| Jennica| 56|        10| 71299|
|  Lizzie| 55|        27| 62050|
|    Kaja| 28|        18| 52610|
|  Aubrie| 55|        18| 77702|
|  Ginnie| 71|        26| 88700|
|  Correy| 47|        25| 83288|
|Rochette| 24|        12| 56123|
|  Hayley| 55|        51| 72758|
|Mathilda| 42|        44| 62825|
|    Veda| 61|        31| 68511|
|   Jessy| 59|        41| 85482|
|    John| 30|         5| 50000|
+--------+---+----------+------+
```

```
df.na.drop(how="all").show()
```

Name: Vatsal Parekh
Roll No: 31031523022

```
+--------+----+----------+------+
|    Name| Age|Experience|Salary|
+--------+----+----------+------+
|Willetta|  27|        52| 62860|
|  Merrie|  45|        34| 57591|
|  Joleen|  43|        42| 61343|
| Nananne|  25|        33| 75478|
| Jennica|  56|        10| 71299|
|  Lizzie|  55|        27| 62050|
|    Kaja|  28|        18| 52610|
|  Aubrie|  55|        18| 77702|
|  Ginnie|  71|        26| 88700|
|  Correy|  47|        25| 83288|
|Rochette|  24|        12| 56123|
|  Hayley|  55|        51| 72758|
|Mathilda|  42|        44| 62825|
|    Veda|  61|        31| 68511|
|   Jessy|  59|        41| 85482|
|    John|  30|         5| 50000|
|    Mike|  25|      NULL|  NULL|
|    NULL|  40|      NULL|  NULL|
|   Sarah|NULL|         3| 40000|
|   Sarah|NULL|      NULL|  NULL|
+--------+----+----------+------+
```

```
df.na.drop(how="any").show()
```

```
+--------+---+----------+------+
|    Name|Age|Experience|Salary|
+--------+---+----------+------+
|Willetta| 27|        52| 62860|
|  Merrie| 45|        34| 57591|
|  Joleen| 43|        42| 61343|
| Nananne| 25|        33| 75478|
| Jennica| 56|        10| 71299|
|  Lizzie| 55|        27| 62050|
|    Kaja| 28|        18| 52610|
|  Aubrie| 55|        18| 77702|
|  Ginnie| 71|        26| 88700|
|  Correy| 47|        25| 83288|
|Rochette| 24|        12| 56123|
|  Hayley| 55|        51| 72758|
|Mathilda| 42|        44| 62825|
|    Veda| 61|        31| 68511|
|   Jessy| 59|        41| 85482|
|    John| 30|         5| 50000|
+--------+---+----------+------+
```

```
df.na.drop(how="any", thresh=2).show()
```

Name: Vatsal Parekh
Roll No: 31031523022

```
+--------+----+----------+------+
|    Name| Age|Experience|Salary|
+--------+----+----------+------+
|Willetta|  27|        52| 62860|
|  Merrie|  45|        34| 57591|
|  Joleen|  43|        42| 61343|
| Nananne|  25|        33| 75478|
| Jennica|  56|        10| 71299|
|  Lizzie|  55|        27| 62050|
|    Kaja|  28|        18| 52610|
|  Aubrie|  55|        18| 77702|
|  Ginnie|  71|        26| 88700|
|  Correy|  47|        25| 83288|
|Rochette|  24|        12| 56123|
|  Hayley|  55|        51| 72758|
|Mathilda|  42|        44| 62825|
|    Veda|  61|        31| 68511|
|   Jessy|  59|        41| 85482|
|    John|  30|         5| 50000|
|    Mike|  25|      NULL|  NULL|
|   Sarah|NULL|         3| 40000|
+--------+----+----------+------+
```

```
df.na.drop(how="any", thresh=3).show()
```

```
+--------+----+----------+------+
|    Name| Age|Experience|Salary|
+--------+----+----------+------+
|Willetta|  27|        52| 62860|
|  Merrie|  45|        34| 57591|
|  Joleen|  43|        42| 61343|
| Nananne|  25|        33| 75478|
| Jennica|  56|        10| 71299|
|  Lizzie|  55|        27| 62050|
|    Kaja|  28|        18| 52610|
|  Aubrie|  55|        18| 77702|
|  Ginnie|  71|        26| 88700|
|  Correy|  47|        25| 83288|
|Rochette|  24|        12| 56123|
|  Hayley|  55|        51| 72758|
|Mathilda|  42|        44| 62825|
|    Veda|  61|        31| 68511|
|   Jessy|  59|        41| 85482|
|    John|  30|         5| 50000|
|   Sarah|NULL|         3| 40000|
+--------+----+----------+------+
```

```
df.na.drop(how="any", subset=['Experience']).show()
```

Name: Vatsal Parekh
Roll No: 31031523022

```
+--------+----+----------+------+
|    Name| Age|Experience|Salary|
+--------+----+----------+------+
|Willetta|  27|        52| 62860|
|  Merrie|  45|        34| 57591|
|  Joleen|  43|        42| 61343|
| Nananne|  25|        33| 75478|
| Jennica|  56|        10| 71299|
|  Lizzie|  55|        27| 62050|
|    Kaja|  28|        18| 52610|
|  Aubrie|  55|        18| 77702|
|  Ginnie|  71|        26| 88700|
|  Correy|  47|        25| 83288|
|Rochette|  24|        12| 56123|
|  Hayley|  55|        51| 72758|
|Mathilda|  42|        44| 62825|
|    Veda|  61|        31| 68511|
|   Jessy|  59|        41| 85482|
|    John|  30|         5| 50000|
|   Sarah|NULL|         3| 40000|
+--------+----+----------+------+
```

```
df.na.fill({'Name' : 'Missing', 'Salary': 0, 'Experience' : 0, 'Age' : 0}).show()
```

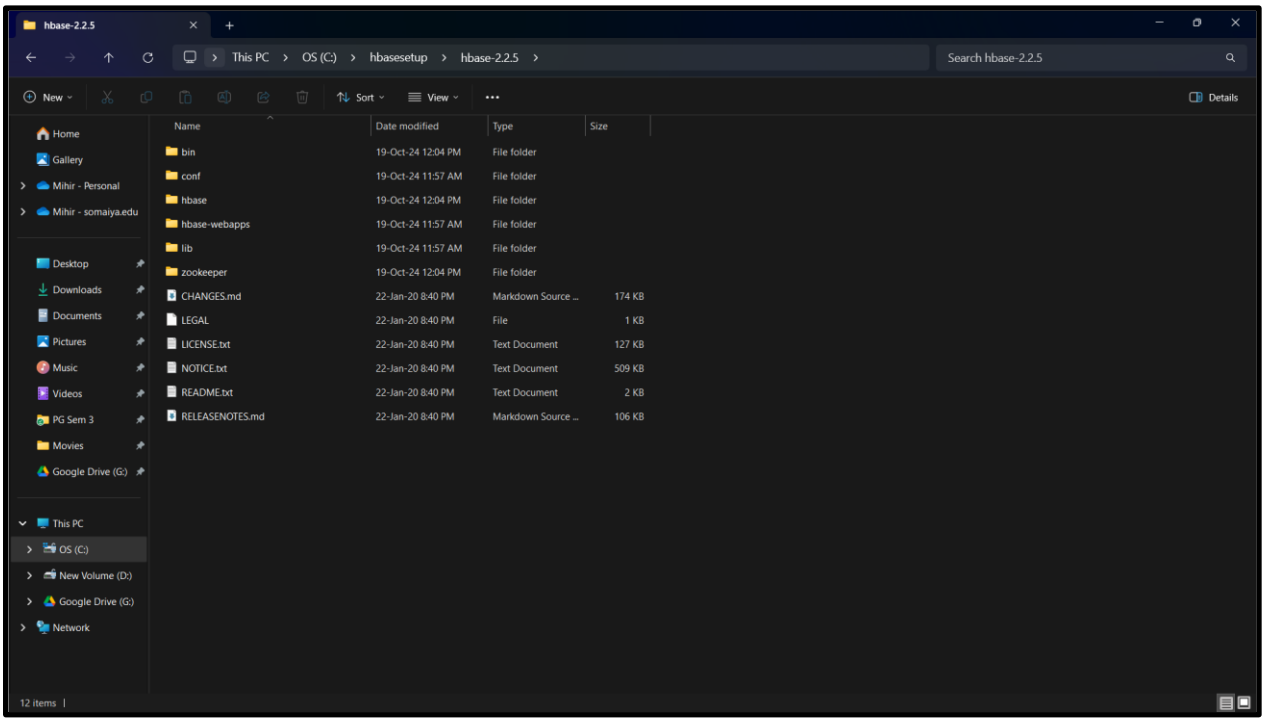
```
+--------+---+----------+------+
|    Name|Age|Experience|Salary|
+--------+---+----------+------+
|Willetta| 27|        52| 62860|
|  Merrie| 45|        34| 57591|
|  Joleen| 43|        42| 61343|
| Nananne| 25|        33| 75478|
| Jennica| 56|        10| 71299|
|  Lizzie| 55|        27| 62050|
|    Kaja| 28|        18| 52610|
|  Aubrie| 55|        18| 77702|
|  Ginnie| 71|        26| 88700|
|  Correy| 47|        25| 83288|
|Rochette| 24|        12| 56123|
|  Hayley| 55|        51| 72758|
|Mathilda| 42|        44| 62825|
|    Veda| 61|        31| 68511|
|   Jessy| 59|        41| 85482|
|    John| 30|         5| 50000|
|    Mike| 25|         0|     0|
| Missing| 40|         0|     0|
|   Sarah|  0|         3| 40000|
|   Sarah|  0|         0|     0|
+--------+---+----------+------+
```

Name: Vatsal Parekh
Roll No: 31031523022

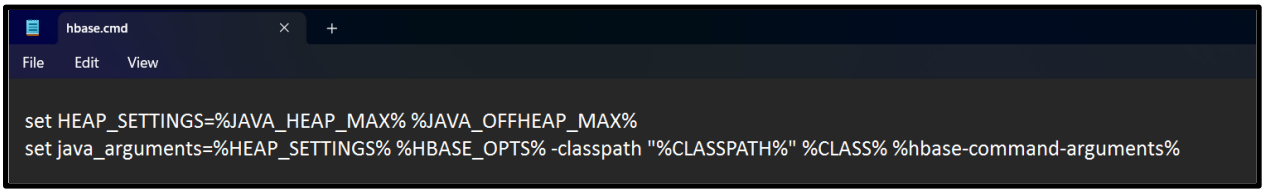
Practical 5: Installation of HBase

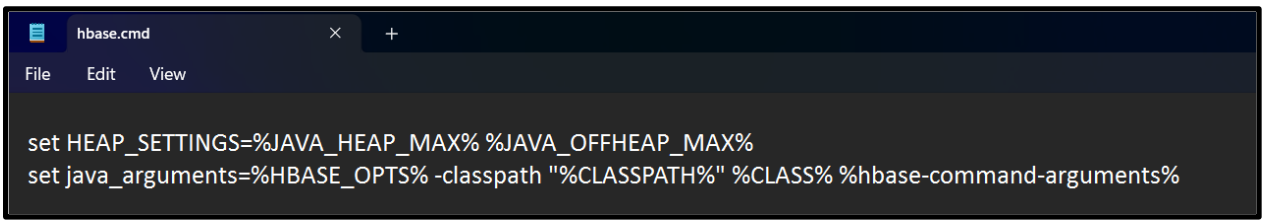
1. Download HBase bin file from here.

2. Create a new folder in C drive named hbasesetup. Extract the files and paste it in the hbasesetup. Create 2 new folders hbase and zookeeper inside.



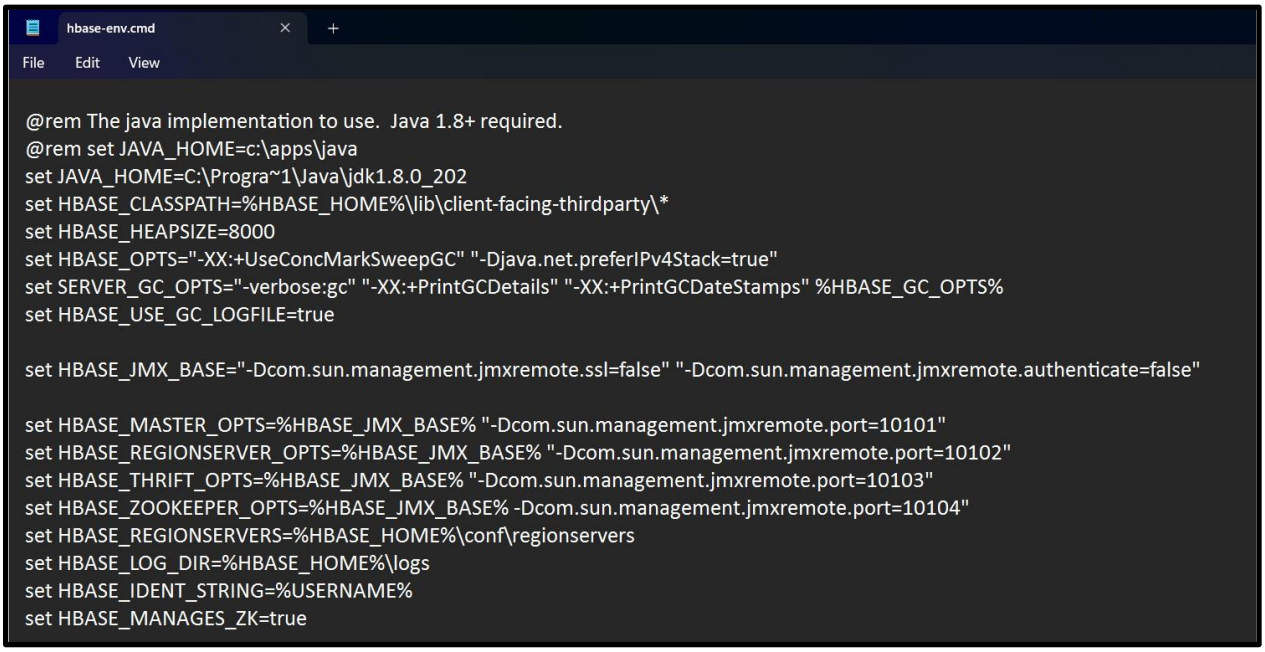
3. Open the bin folder. Search for the file hbase.cmd and edit it in notepad. Search for java_arguments and remove %HEAP_SETTINGS%.





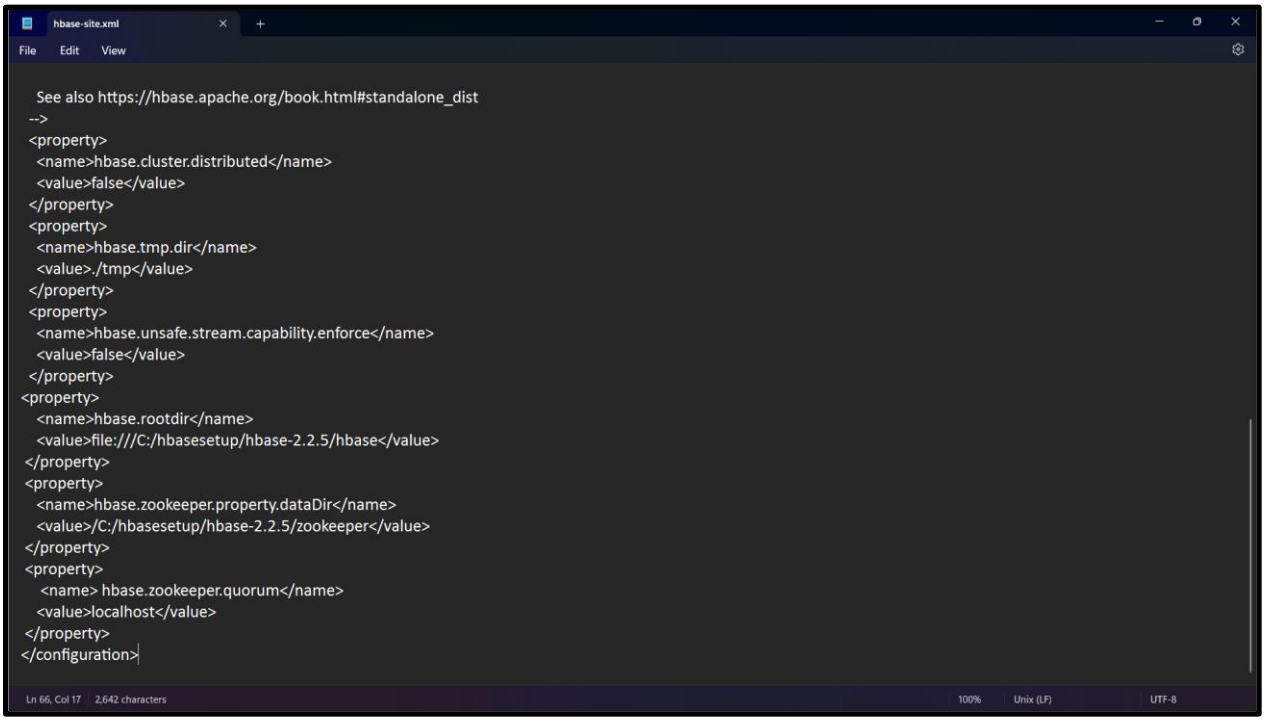
4. Open the conf folder and edit the file hbase-env.cmd in notepad. Add the following lines inside.

```
set JAVA_HOME=C:\Progra~1\Java\jdk1.8.0_202

set HBASE_CLASSPATH=%HBASE_HOME%\lib\client-facing-thirdparty\*

set HBASE_HEAPSIZE=8000

set HBASE_OPTS="-XX:+UseConcMarkSweepGC" "-Djava.net.preferIPv4Stack=true"

set SERVER_GC_OPTS="-verbose:gc" "-XX:+PrintGCDetails" "-XX:+PrintGCDateStamps"
%HBASE_GC_OPTS%

set HBASE_USE_GC_LOGFILE=true

set HBASE_JMX_BASE="-Dcom.sun.management.jmxremote.ssl=false" "-
Dcom.sun.management.jmxremote.authenticate=false"

set HBASE_MASTER_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10101"

set HBASE_REGIONSERVER_OPTS=%HBASE_JMX_BASE% "-
Dcom.sun.management.jmxremote.port=10102"

set HBASE_THRIFT_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10103"

set HBASE_ZOOKEEPER_OPTS=%HBASE_JMX_BASE% -Dcom.sun.management.jmxremote.port=10104"

set HBASE_REGIONSERVERS=%HBASE_HOME%\conf\regionservers

set HBASE_LOG_DIR=%HBASE_HOME%\logs

set HBASE_IDENT_STRING=%USERNAME%

set HBASE_MANAGES_ZK=true
```

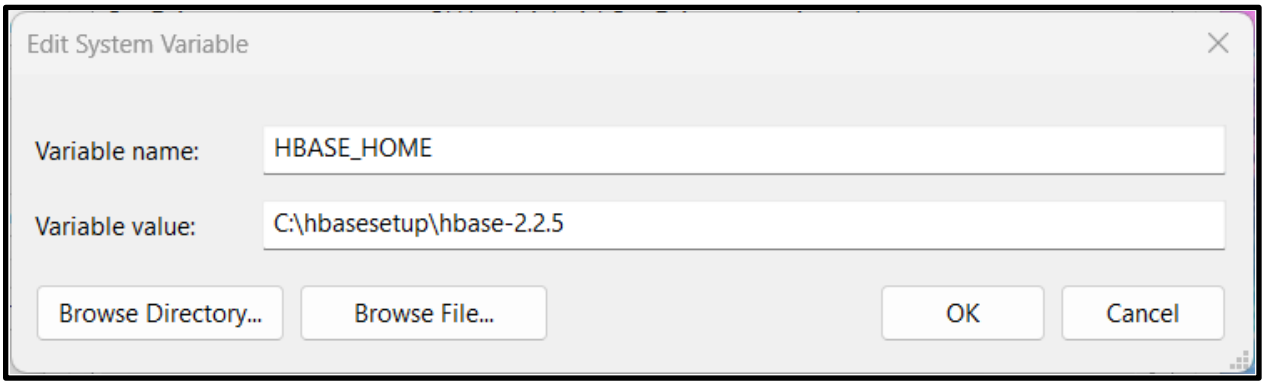


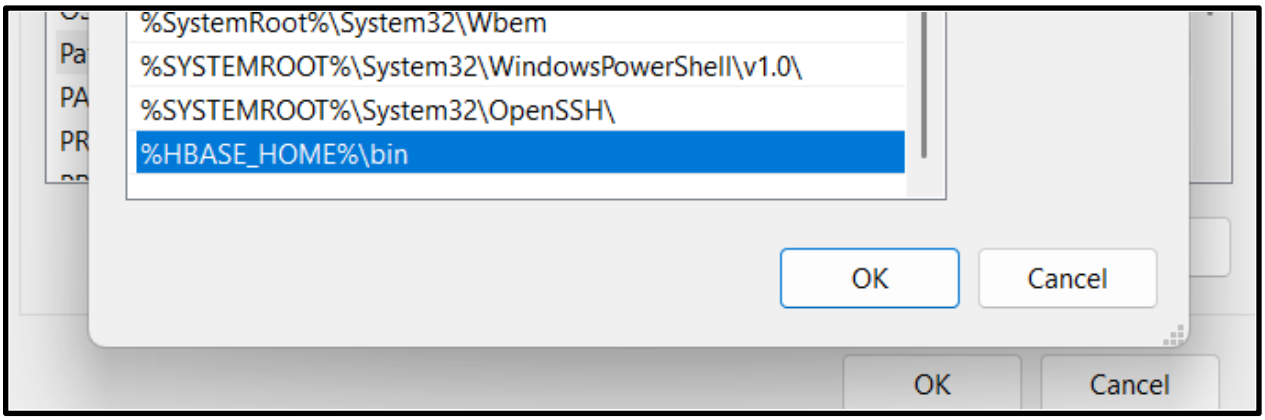
5. Open the conf folder and edit the file hbase-site.xml in notepad. Add the following lines after the last property tag.

```
<property>
    <name>hbase.rootdir</name>
    <value>file:///C:/hbasesetup/hbase-2.2.5/hbase</value>
 </property>
 <property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>/C:/hbasesetup/hbase-2.2.5/zookeeper</value>
 </property>
 <property>
     <name> hbase.zookeeper.quorum</name>
    <value>localhost</value>
 </property>
```



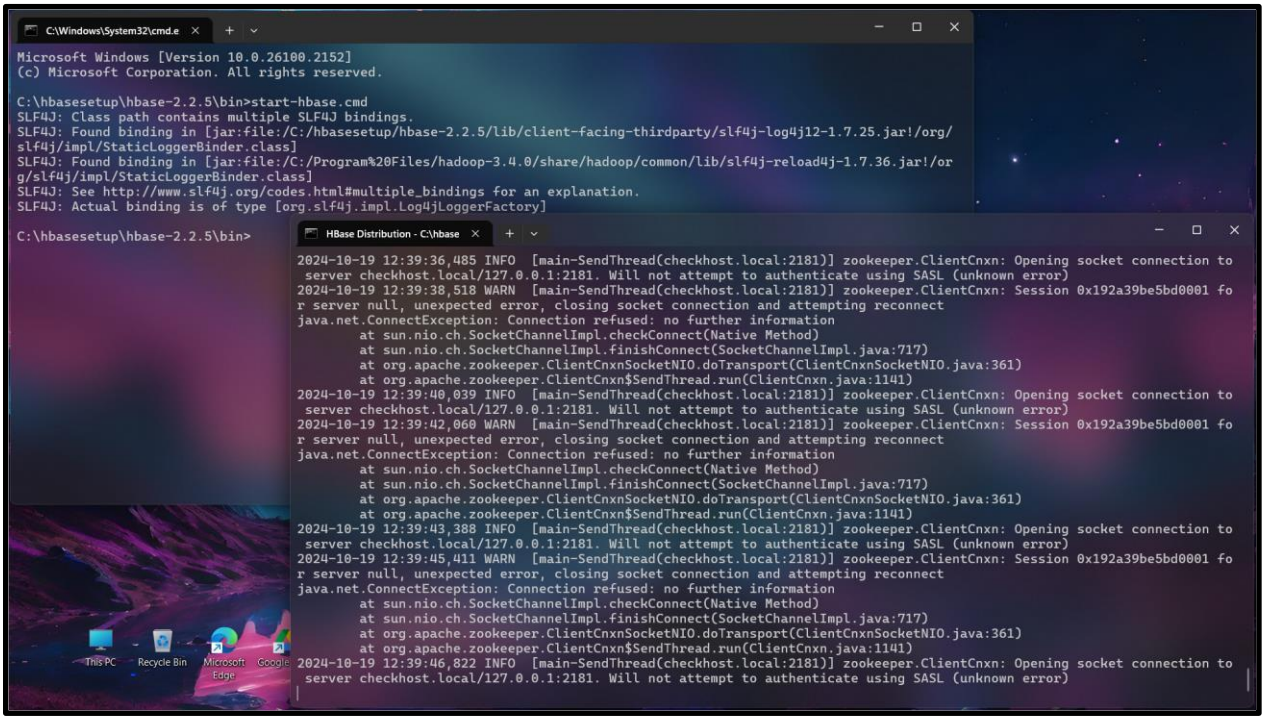
6. Set the HBase environment variables and it too path as well.



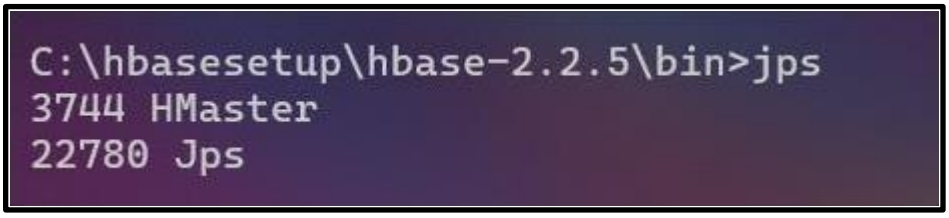


Name: Vatsal Parekh
Roll No: 31031523022

7. Open command prompt and navigate to the hbase bin folder. Run the command start-hbase.cmd to start hbase



8. Using the command jps you can check that our HMaster is running.



9. Start the HBase Shell now with the command hbase shell. Initial startup may take some time.

10. Your HBase Shell has been started. Ignore the warnings received while starting the shell.