|  |  |
| --- | --- |
| **EXP.NO: 5** | **IMPLEMENT SERVER LOGIN ANALYSIS USING SPARK** |
| **DATE:** |

# AIM :

To perform the server login analysis using spark.

# PROCEDURE :

**STEP 1:** Open the spark terminal.

**STEP 2:** Restart all the daemons.

**STEP 3:** Configure the Spark.

**STEP 4:** Load the data.

**STEP 5:** Parse the data.

**STEP 6:** Analyze the data to find failed login attempts.

**STEP 7:** Analyze the data to find login counts per user.

# PROGRAM:

import org.apache.spark.sql.SparkSession import org.apache.spark.sql.functions.\_

object ServerLoginAnalysis {

def main(args: Array[String]): Unit = {

// Initialize Spark session

val spark = SparkSession.builder

.appName("Server Login Analysis")

.getOrCreate()

// Load the data

val filePath = "path/to/your/logins.csv"

val loginsDF = spark.read.option("header", "true").csv(filePath)

// Show the data loginsDF.show()

// Analysis: Count of failed login attempts

val failedLoginsDF = loginsDF.filter(col("login\_status") === "failed") val failedLoginsCount = failedLoginsDF.count()

println(s"Total Failed Login Attempts: $failedLoginsCount")

// Analysis: Count of logins per user

val loginCountsDF = loginsDF.groupBy("user\_id").agg(count("login\_status").alias("login\_count")) loginCountsDF.show()

// Analysis: Count of successful and failed logins per user

val loginStatusCountsDF = loginsDF.groupBy("user\_id").agg( count(when(col("login\_status") === "success", true)).alias("success\_count"), count(when(col("login\_status") === "failed", true)).alias("failed\_count")

)

loginStatusCountsDF.show()

// Stop the Spark session spark.stop()

}

}

# OUTPUT :

|  |  |  |
| --- | --- | --- |
| **PREPARATION** | **30** |  |
| **LAB PERFORMANCE** | **30** |  |
| **REPORT** | **40** |  |
| **TOTAL** | **100** |  |
| **INITIAL OF FACULTY** | |  |

**RESULT:**

Thus, the Server login analysis using spark is executed successfully.