Part 1

ClusterID: student137

Password: DistributedCMU1$

Task 7 screenshot

A screenshot of a computer

Description automatically generated

**Part 2**

Code:  
package org.myorg;  
  
import org.apache.spark.SparkConf;  
import org.apache.spark.api.java.JavaRDD;  
import org.apache.spark.api.java.JavaSparkContext;  
import org.apache.spark.api.java.function.Function;  
  
import java.io.BufferedReader;  
import java.io.InputStreamReader;  
import java.util.Arrays;  
import java.util.List;  
  
*/\*\*  
 \* ShakespeareAnalytics.java  
 \* Author: Rheann Sequeira  
 \* @author: rsequeir  
 \*/*public class ShakespeareAnalytics {  
  
 public static void main(String[] args) {  
 *// Check if input file is provided* if (args.length == 0) {  
 System.*out*.println("NO INPUT FILE");  
 System.*exit*(1);  
 }  
  
 *// Create a SparkConf object with local master and app name* SparkConf conf = new SparkConf().setMaster("local").setAppName("ShakespeareAnalytics");  
 JavaSparkContext sc = new JavaSparkContext(conf);  
  
 *// Load the input file into a JavaRDD* JavaRDD<String> lines = sc.textFile(args[0]);  
  
 *// Count the number of lines in the input file* long noOfLines = lines.count();  
  
 *// Task 1: Count the number of words* Function<String, Boolean> filter = k -> (!k.isEmpty());  
 JavaRDD<String> words = lines.flatMap(line -> Arrays.*asList*(line.split("[^a-zA-Z]+")));  
 long noOfWords = words.filter(word -> !word.isEmpty()).count();  
  
 *// Task 2: Count the number of distinct words* long noOfDistinctWords = words.filter(filter).distinct().count();  
  
 *// Task 3: Count the number of symbols* JavaRDD<String> symbols = lines.flatMap(line -> Arrays.*asList*(line.split("")));  
 long noOfSymbols = symbols.count();  
  
 *// Task 4: Count the number of distinct symbols* long noOfDistinctSymbols = symbols.distinct().count();  
  
 *// Task 5: Count the number of distinct letters* JavaRDD<String> letters = symbols.filter(s -> s.matches("[a-zA-Z]"));  
 long noOfDistinctLetters = letters.distinct().count();  
  
 *// Display the results* System.*out*.println("Number of lines: " + noOfLines);  
 System.*out*.println("Number of words: " + noOfWords);  
 System.*out*.println("Number of distinct words in 'All's Well That Ends Well': " + noOfDistinctWords);  
 System.*out*.println("Number of symbols: " + noOfSymbols);  
 System.*out*.println("Number of distinct symbols: " + noOfDistinctSymbols);  
 System.*out*.println("Number of distinct letters: " + noOfDistinctLetters);  
  
 *// Task 6: Search for a word in the input file* try (BufferedReader reader = new BufferedReader(new InputStreamReader(System.*in*))) {  
 System.*out*.print("Enter search word (This search is case-sensitive): ");  
 String input = reader.readLine();  
  
 JavaRDD<String> linesFiltered = lines.filter(line -> line.contains(input));  
  
 List<String> result = linesFiltered.collect();  
 if (result.isEmpty()) {  
 System.*out*.println("No lines found containing the word \"" + input + "\".");  
 } else {  
 System.*out*.println("Lines containing the word \"" + input + "\":");  
 result.forEach(System.*out*::println);  
 }  
 } catch (Exception e) {  
 System.*out*.println("Error processing input: " + e.getMessage());  
 e.printStackTrace();  
 }  
  
 *// Stop the Spark context* sc.stop();  
 }  
}

Output:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated