PROBLEM STATEMENT: Design and develop a distributed application to find frequency of words from sample text data. Use sample text data and process it using MapReduce

Step 1: Create a Java Project in Eclipse

1. **Create a New Java Project**:
    - Open Eclipse and go to **File** → **New** → **Java Project**.
    - Name the project (e.g., `WordCount`).
    - Click **Finish**.
2. **Create a Package**:
    - In the **Project Explorer**, right-click on `src` → **New** → **Package**.
    - Name the package (e.g., `WordCount`).
3. **Create the Classes**:
    - Right-click on the package you created → **New** → **Class**.
    - Create the following classes:
        - **WCMapper**
        - **WCReducer**
        - **WCDriver** (Driver class)

Step 2: Add the Code

Add the following code in their respective classes:

//WCMapper.java

```
// Importing libraries

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;
```

```
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;

public class WCMapper extends MapReduceBase implements Mapper<LongWritable,
                                                                Text,
Text, IntWritable> {

        // Map function
        public void map(LongWritable key, Text value, OutputCollector<Text,
                                IntWritable> output, Reporter rep) throws IOException
        {

                String line = value.toString();

                // Splitting the line on spaces
                for (String word : line.split(" "))
                {
                        // Clean the word: remove punctuation/special characters
                        word = word.replaceAll("[^a-zA-Z]", ""); // <-- Added line

                        if (word.length() > 0)
                        {
                                output.collect(new Text(word.toLowerCase()), new IntWritable(1));
// <-- Made it lowercase
                        }
                }
```

```
        }
}
```

//WCReducer.java

```
// Importing libraries
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class WCReducer extends MapReduceBase implements Reducer<Text,
                                                        IntWritable, Text, IntWritable>
{

        // Reduce function
        public void reduce(Text key, Iterator<IntWritable> value,
                                OutputCollector<Text, IntWritable> output,
                                                        Reporter rep) throws IOException
        {

                int count = 0;

                // Counting the frequency of each words
```

```
            while (value.hasNext())
            {
                IntWritable i = value.next();
                count += i.get();
            }

            output.collect(key, new IntWritable(count));
        }
}
```

//WCDriver.java

```
// Importing libraries
import java.io.IOException;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {
```

```
public int run(String args[]) throws IOException
{
        if (args.length < 2)
        {
                System.out.println("Please give valid inputs");
                return -1;
        }

        JobConf conf = new JobConf(WCDriver.class);
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(WCMapper.class);
        conf.setReducerClass(WCReducer.class);
        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);
        JobClient.runJob(conf);
        return 0;
}

// Main Method
public static void main(String args[]) throws Exception
{
        int exitCode = ToolRunner.run(new WCDriver(), args);
        System.out.println(exitCode);
```

```
    }
}
```

Step 3: Build the Project

1. **Build the Project**:
    - Click on **Project** → **Build Project** in Eclipse.
    - Make sure the project compiles without any errors.
2. **Check Build Path**:
    - Go to **Project Explorer** → Right-click on your project → **Build Path** → **Configure Build Path**.
    - Ensure that all the Hadoop JAR files you added are present in the **Libraries** section.
        i. hadoop-common (e.g., `hadoop-common.jar`)
        ii. Hadoop-mapreduce-client-core (e.g., `hadoop-mapreduce-client-core-2.x.x.jar`)

Step 4: Create the JAR File

1. **Export to JAR**:
    - Go to **File** → **Export**.
    - Choose **Java** → **Runnable JAR file**.
    - Choose **Launch configuration** as `WordCount`.
    - Select the destination path and name your JAR file (e.g., `Wcount.jar`).
    - Click **Finish**.

Step 5: Prepare the Input Files

1. **Create the Log File (`logfile.txt`)**:
    - The `WCFile.txt` should be placed in the HDFS directory. Here's the content of the file:

      Hello Hello, Sampada Sampada. This is Word. I am from Cloudera.

Upload the Input File to HDFS:

- Use the Hadoop shell to upload the input file to HDFS:

hdfs dfs -put /path/to/WCfile.txt /user/cloudera/WCfile.txt

Step 6: Configure the Run Configuration

1. **Set up Run Configuration**:
    - In Eclipse, go to **Run** → **Run Configurations**.
    - Select **Java Application** and click **New**.
    - In the **Main** tab, select the **Project** (your Hadoop project) and the **Main Class** (`W`).

Step 8: Run the Job

1. **Run the Hadoop Job**:
   [cloudera@quickstart ~]$ had oop jar /home/cloudera/Wcount.jar WordCount.WordCount /user/cloudera/WCFile.txt /user/cloudera/dir51
2. hdfs dfs -ls /user/cloudera/dir51

```
                Bytes Written=30
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/dir51

Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2025-04-26 23:53 /user/cloudera/dir51/_SUCCESS
-rw-r--r--   1 cloudera cloudera         30 2025-04-26 23:53 /user/cloudera/dir51/part-r-00000
[cloudera@quickstart ~]$
```

3. Hadoop fs -cat /user/cloudera/dir51/part-00000