Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri, Nagpur - 441 110
NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

Department of Artificial Intelligence & Data Science

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Session 2025-2026

Vision: Dream of where you want.	**Mission:** Means to achieve Vision

Program Educational Objectives of the program (PEO): (broad statements that describe the professional and career accomplishments)

PEO1	**Preparation**	**P: Preparation**	**Pep-CL abbreviation pronounce as Pep-si-lL easy to recall**
PEO2	**Core Competence**	**E: Environment (Learning Environment)**	
PEO3	**Breadth**	**P: Professionalism**	
PEO4	**Professionalism**	**C: Core Competence**	
PEO5	**Learning Environment**	**L: Breadth (Learning in diverse areas)**	

Program Outcomes (PO): (statements that describe what a student should be able to do and know by the end of a program)

Keywords of POs:
Engineering knowledge, Problem analysis, Design/development of solutions, Conduct Investigations of Complex Problems, Engineering Tool Usage, The Engineer and The World, Ethics, Individual and Collaborative Team work, Communication, Project Management and Finance, Life-Long Learning

PSO Keywords: Cutting edge technologies, Research

"I am an engineer, and I know how to apply engineering knowledge to investigate, analyse and design solutions to complex problems using tools for entire world following all ethics in a collaborative way with proper management skills throughout my life." *to contribute to the development of cutting-edge technologies and Research*.

Integrity: I will adhere to the Laboratory Code of Conduct and ethics in its entirety.

Name and Signature of Student and Date
(Signature and Date in Handwritten)

Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri, Nagpur - 441 110
NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

Department of Artificial Intelligence & Data Science

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Session	2025-26 (ODD)	**Course Name**	BDH Lab
Semester	7 AIDS	**Course Code**	22ADS703
Roll No	21	**Name of Student**	**Sanskruti.Paunikar**

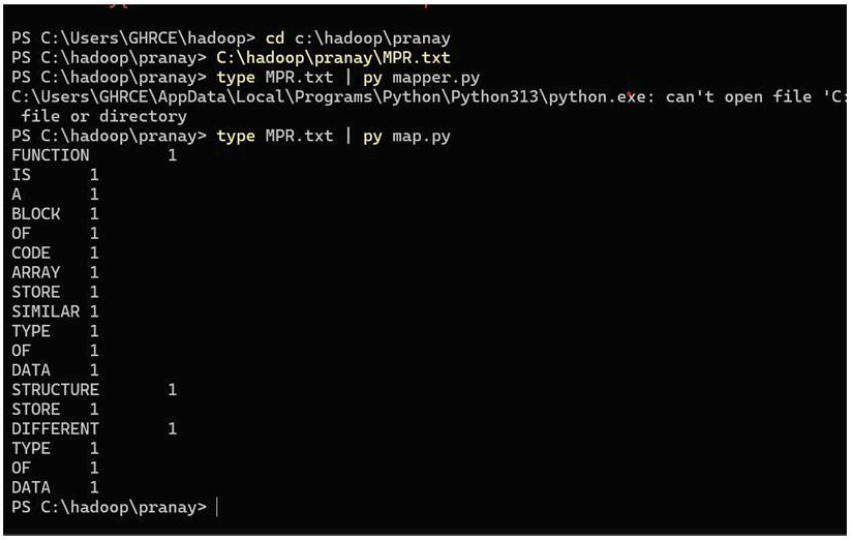
Practical Number	**3**
Course Outcome	**CO1:-** 1. Understand big data analytics and its business applications. **CO2:-** Analyze the HADOOP and Map Reduce technologies associated with big data analytics. **CO3:-** Apply Big Data analytics Using Pig and Hive.
Aim	Write a MapReduce Program to Calculate Frequency of Words from Datasets.
Problem Definition	
Theory (100 words)	MapReduce is a distributed programming model in Hadoop used for processing large datasets across clusters. It consists of two main functions: **Mapper** and **Reducer**. The Mapper reads input data and breaks it into key-value pairs, usually emitting each word as a key and 1 as its value. The Reducer then aggregates these pairs by summing the counts of identical words, producing the total frequency of each word in the dataset. This parallel processing approach improves efficiency, scalability, and fault tolerance, making MapReduce ideal for big data analytics and word count operations in distributed systems..

Procedure and Execution
 	
(100 Words)
	Steps of Implementation:-
	Prepare Input:
	Upload text file to HDFS:
	`hdfs dfs -put input.txt /user/hadoop/input`
	
	Mapper:
	Emits `(word, 1)` for each word.
	
	Reducer:
	Sums values for each word → `(word, total_count)`.
	
	Run Job:
	`hadoop jar wordcount.jar WordCount /user/hadoop/input /user/hadoop/output`
	
	View Output:
	`hdfs dfs -cat /user/hadoop/output/part-r-00000`

```
Code:
import java.io.IOException;
port java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {
public static class TokenizerMapper extends Mapper<Object,
Text, Text, IntWritable> {
private final static IntWritable one = new IntWritable(1);
private Text word = new Text();
public void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
StringTokenizer itr = new StringTokenizer(value.toString());
while (itr.hasMoreTokens()) {
word.set(itr.nextToken());
context.write(word, one);
}
}
}
public static class IntSumReducer extends Reducer<Text,
IntWritable, Text, IntWritable> {
private IntWritable result = new IntWritable();
public void reduce(Text key, Iterable<IntWritable> values,
Context context)
throws IOException, InterruptedException {
int sum = 0;
for (IntWritable val : values) {
sum += val.get();
}
result.set(sum);
context.write(key, result);
}
}
public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();
```

	Job job = Job.getInstance(conf, "word count");
	job.setJarByClass(WordCount.class);
	job.setMapperClass(TokenizerMapper.class);
	job.setCombinerClass(IntSumReducer.class);
	job.setReducerClass(IntSumReducer.class);
	job.setOutputKeyClass(Text.class);
	job.setOutputValueClass(IntWritable.class);
	FileInputFormat.addInputPath(job, new Path(args[0]));
	FileOutputFormat.setOutputPath(job, new Path(args[1]));
	System.exit(job.waitForCompletion(true) ? 0 : 1);
	}
	}

	Output: 
Output Analysis	The output lists each unique word with its frequency count. The Mapper divides the text and assigns counts, while the Reducer aggregates them. This demonstrates the efficiency of Hadoop's distributed computation. The output confirms correct functionality—accurate word counts indicate successful data partitioning, parallel execution, and combination of intermediate results, showcasing MapReduce's scalability and reliability for large-scale text processing tasks.
Link of student Github profile where lab assignment has been uploaded	https://github.com/sanskruti-1234/BDH.git
Conclusion	The WordCount MapReduce program effectively demonstrates parallel data processing in Hadoop. It distributes data across nodes, computes word frequencies efficiently, and consolidates

	results through reducers. This experiment validates the power of Hadoop MapReduce for handling large-scale text analytics tasks in a distributed environment.
Plag Report (Similarity index < 12%)	
Date	30/10/2025