PROGRAM NO:4

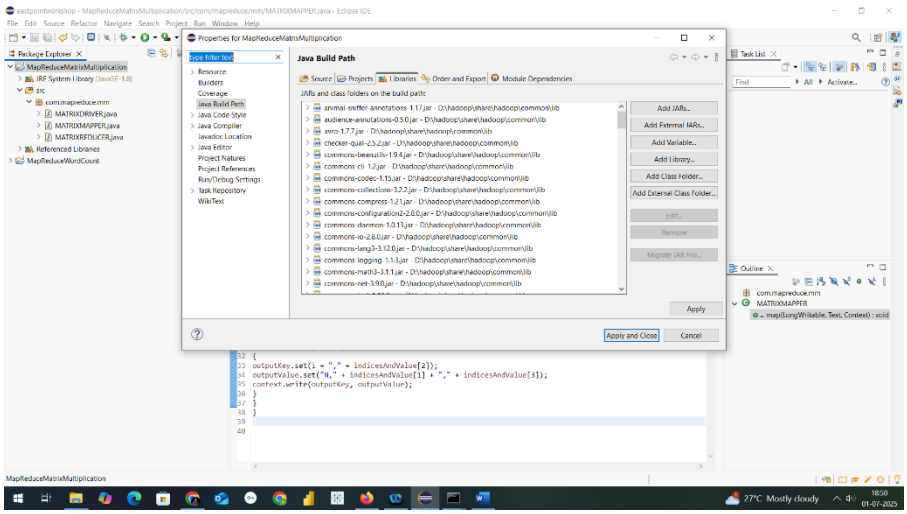
Problem Statement : Develop a MapReduce program to find the tags associated with each movie by analyzing movie lens data

Step1:goto Search button run cmd prompt as administrator
Step2: Initiate all required component of hadoop by command
start-all.cmd
Step3: Create a weather.csv file where we have folder running hadoop java program in my case I have created in Desktop folder
D:\hadoopprograms\movietag
U can create anywhere as u wish with content looks as in below

userId	movieId	tag	timestamp
15	318	brad pitt	1215184630
15	318	edward norton	1215184590
15	318	fight club	1215184270
16	319	mrVIjay	1215184279

Step3: So open Eclipse

Create a java project with name MoviesTagsWithTitles
Select execution environment as JavaSE-1.8
And click on next&finish
Step4: Right Click on Projectname--- Click on New---create a package with name com.mapreduce.mm and click on Finish
Step5: Add required libraries to support hadoop by navigating as in below
Right Click on Project-------Right Click on BuildPath------- Configure Buildpath Then goto Libraries as in screen below



Step6: Add necessary External jar file from D:\hadoop\share\hadoop
Like clients,common,hdfs,mapreduce & yarn to support packages for hadoop

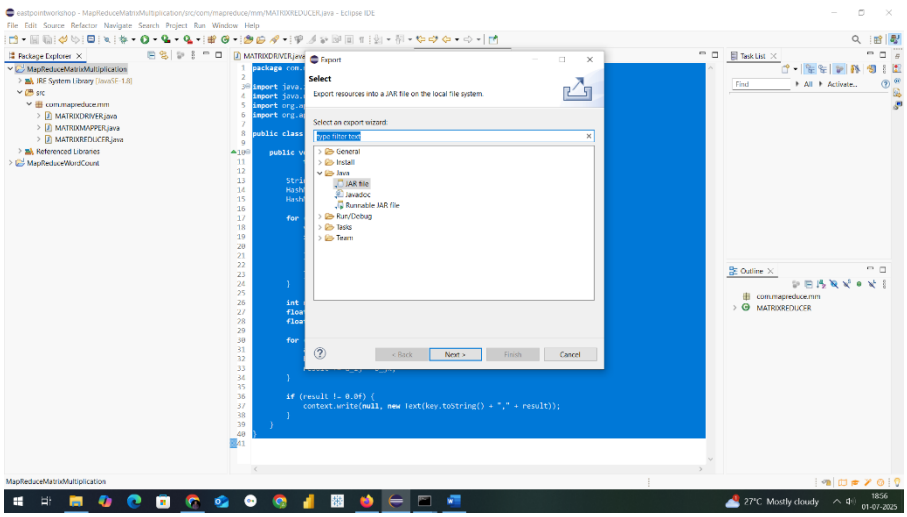
Step7: Create a class within package com.mapreduce.tag with name MoviesTagsWithTitles and paste this code

```
package com.mapreduce.tg
import java.io.IOException;
import java.util.HashSet;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MovieTagsWithTitles {

   public static class TagsMapper extends Mapper<LongWritable, Text, Text,
Text> {
      public void map(LongWritable key, Text value, Context context)
            throws IOException, InterruptedException {
         String[] fields = value.toString().split(",", 4); // userId, movieId, tag,
timestamp
         if (fields.length >= 3 && !fields[0].equals("userId")) { // skip header
            String movieId = fields[1].trim();
            String tag = fields[2].trim();
            context.write(new Text(movieId), new Text(tag));
```

```
            }
        }
    }

    public static class TagsReducer extends Reducer<Text, Text, Text, Text> {
        public void reduce(Text movieId, Iterable<Text> tags, Context context)
                throws IOException, InterruptedException {
            HashSet<String> uniqueTags = new HashSet<>();
            for (Text tag : tags) {
                uniqueTags.add(tag.toString());
            }
            context.write(movieId, new Text(String.join(", ", uniqueTags)));
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        conf.set("mapreduce.framework.name", "local");
        conf.set("fs.defaultFS", "file:///");
        conf.set("n", "3");

        Job job = Job.getInstance(conf, "Movie Tags By Movie ID");

        job.setJarByClass(MovieTagsWithTitles.class);
        job.setMapperClass(TagsMapper.class);
        job.setReducerClass(TagsReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        // Input/output from command-line args
        FileInputFormat.addInputPath(job, new Path(args[0]));    // path to tags.csv
        FileOutputFormat.setOutputPath(job, new Path(args[1]));  // output folder

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Step8: Procedure to create jar file from eclipse project

Now Right click on Project and click on export and select jar as in screen



THE OUTPUT WILL BE CREATED THIS TIME IN LOCALFOLDER WHERE MENTIONED
----→ D:\hadoopprograms\movieoutput
In part-r-00000 text file

NOW TO CHECK THE RESULT OF A PROGRAM