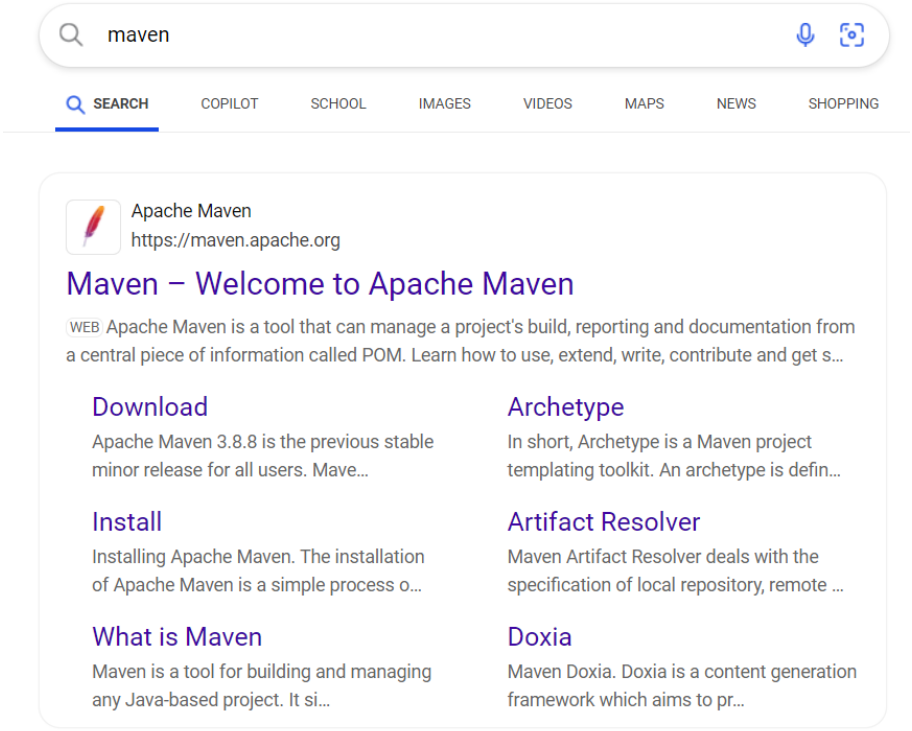
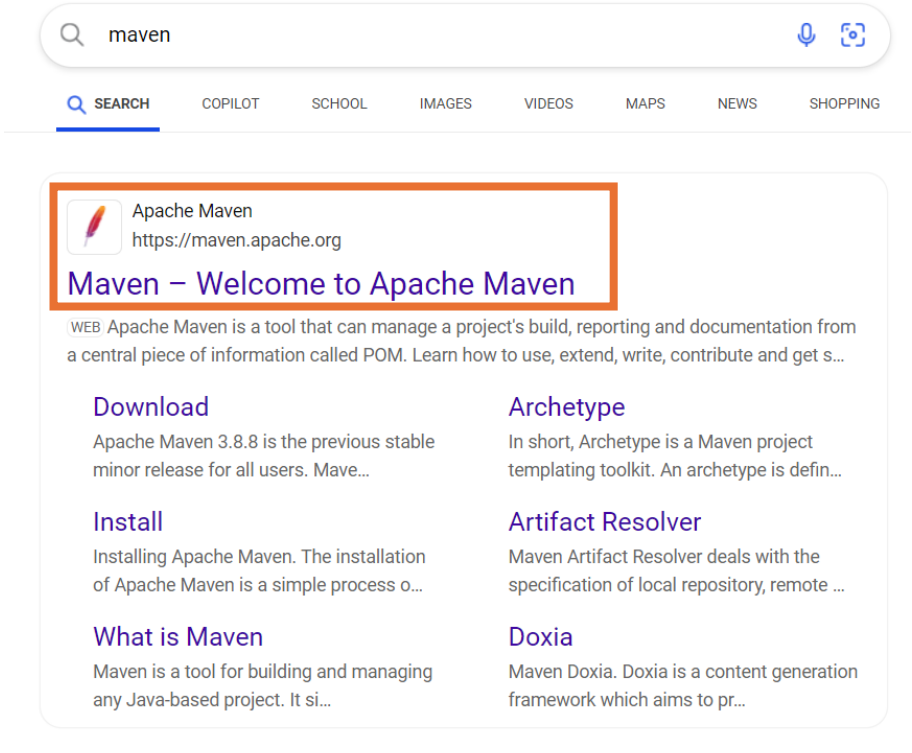
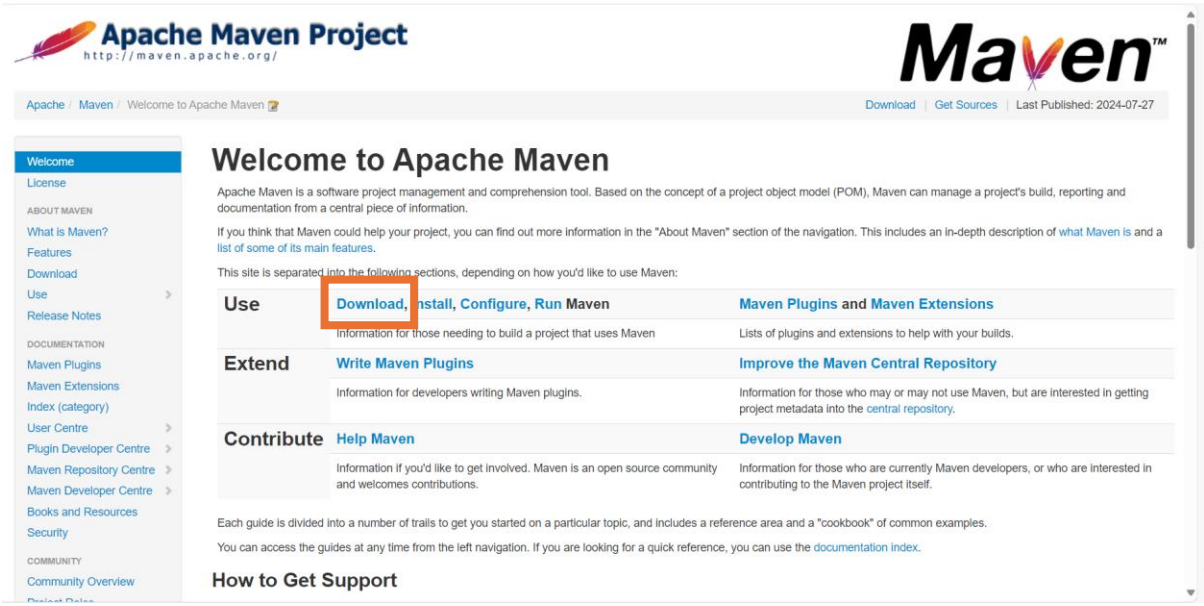
Step 1 : Type in google Maven



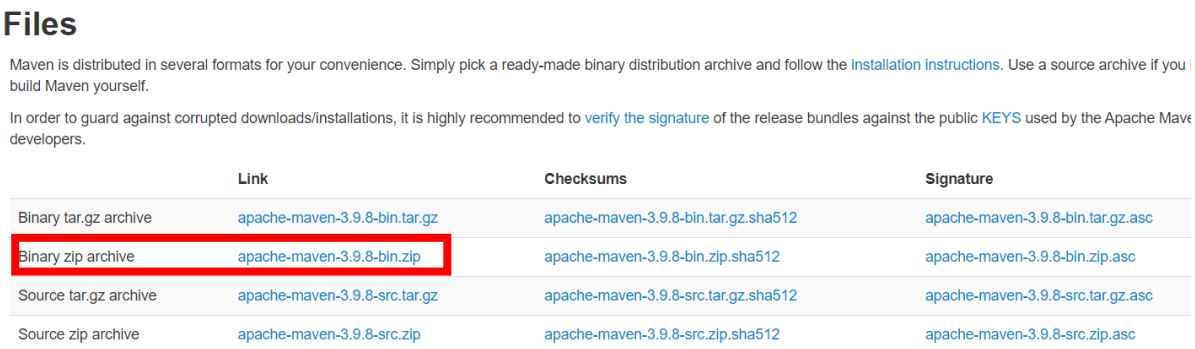
Step 2: Click maven website



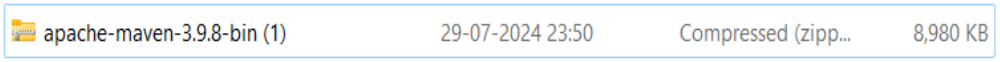
Step 3: Click on the Download button on official page



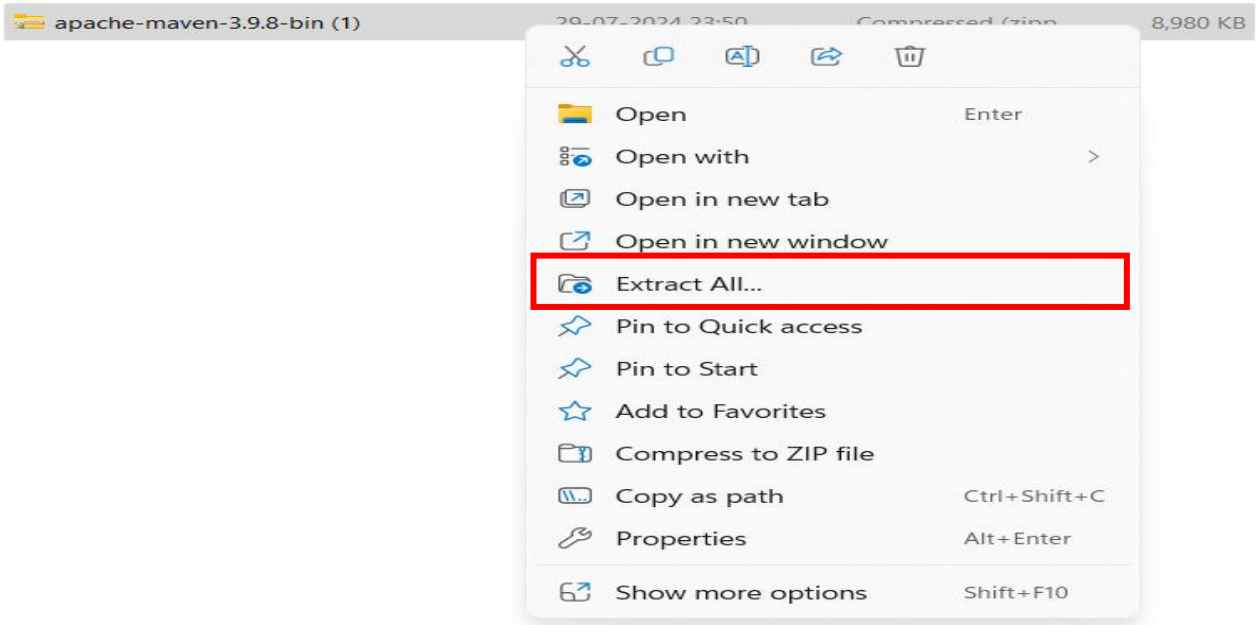
Step 4: Download the binary zip file version



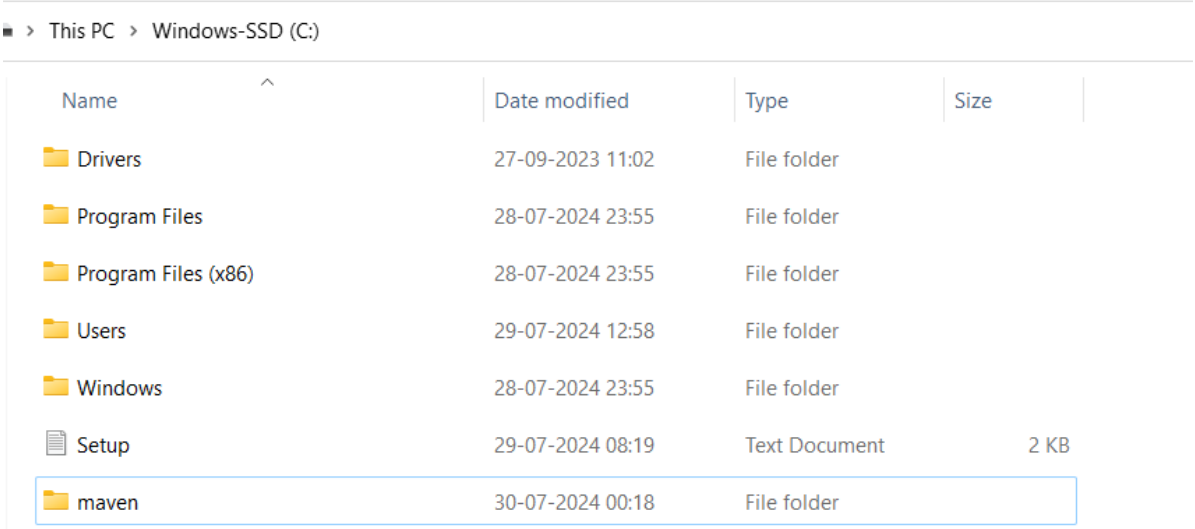
Step 5: Go to the folder where it has downloaded



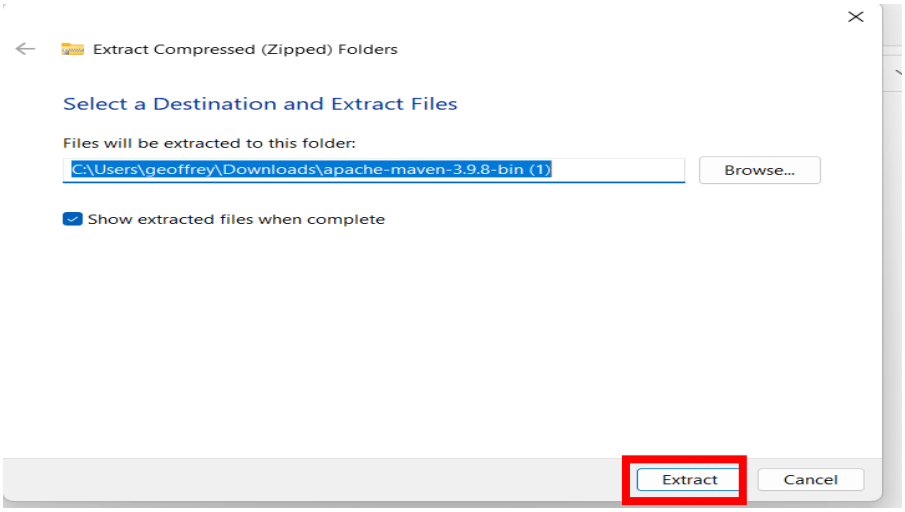
Step 6: Right click the downloaded file and select extract all



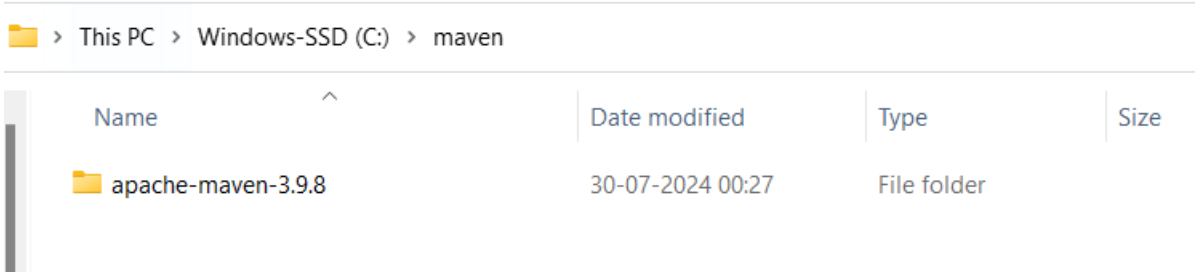
Step 7: Create new folder in C Folder named maven



Step 8: Extract all the files in the created maven file

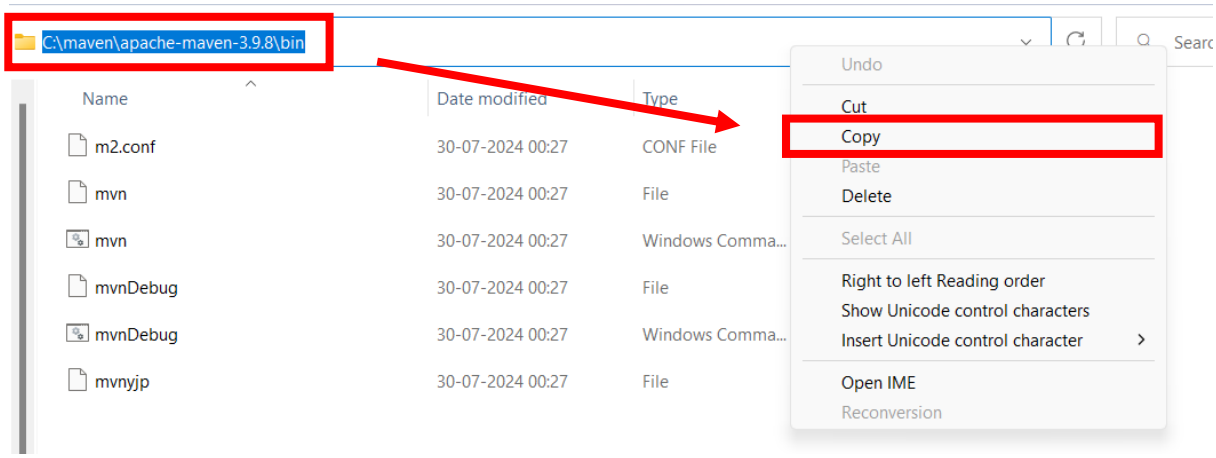


Step 9: Go to the Extracted Folder C:\maven

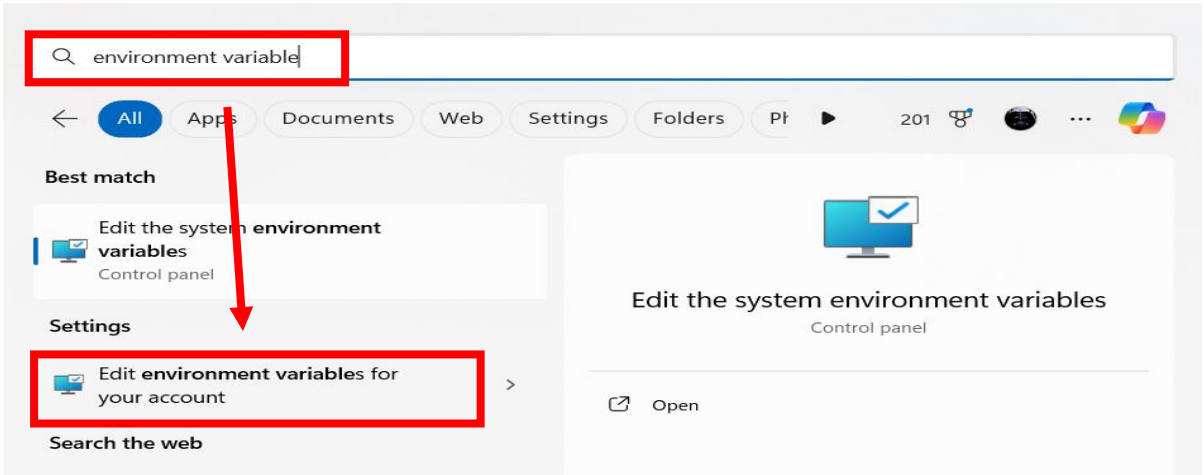


Step 10: Click apache-maven-3.9.8 and Go inside bin folder

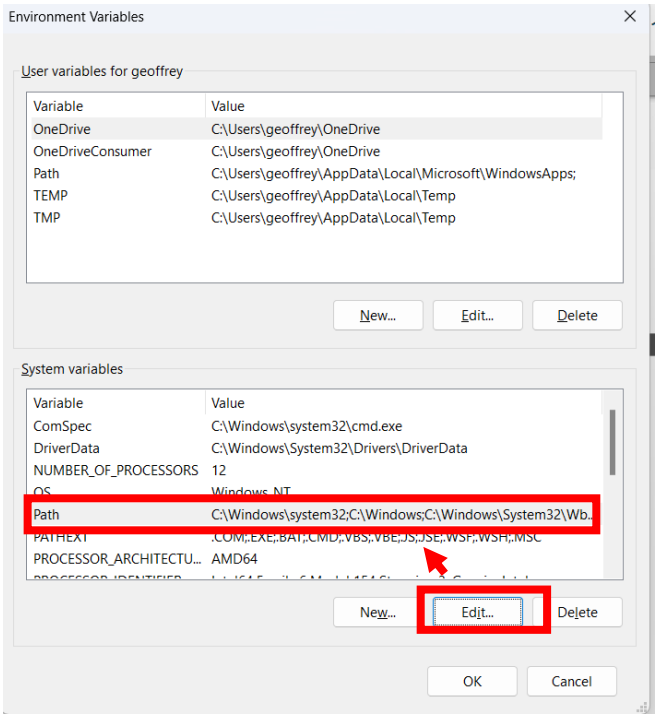
Step 11 Copy the bin path by clicking menu bar and right click then select copy



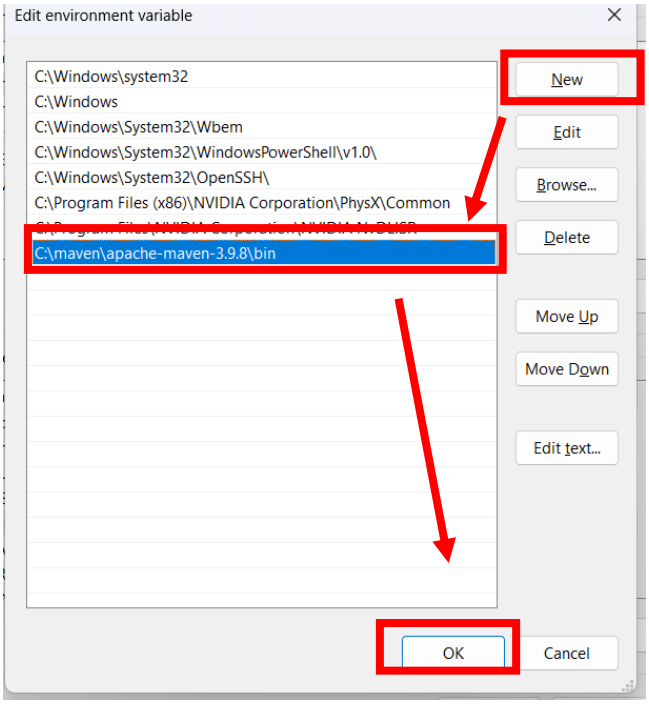
Step 12 Type in search environment variables and select environment variables



Step : 13 Double Click of the path in System variables (or) Select Path and click edit

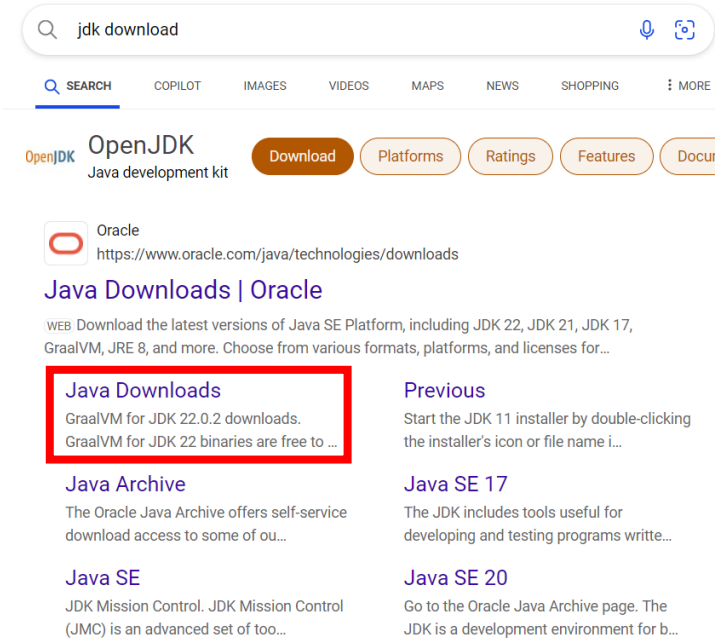


Step : 14 Click new and paste the copied path of step 11 under system variables path and close everything

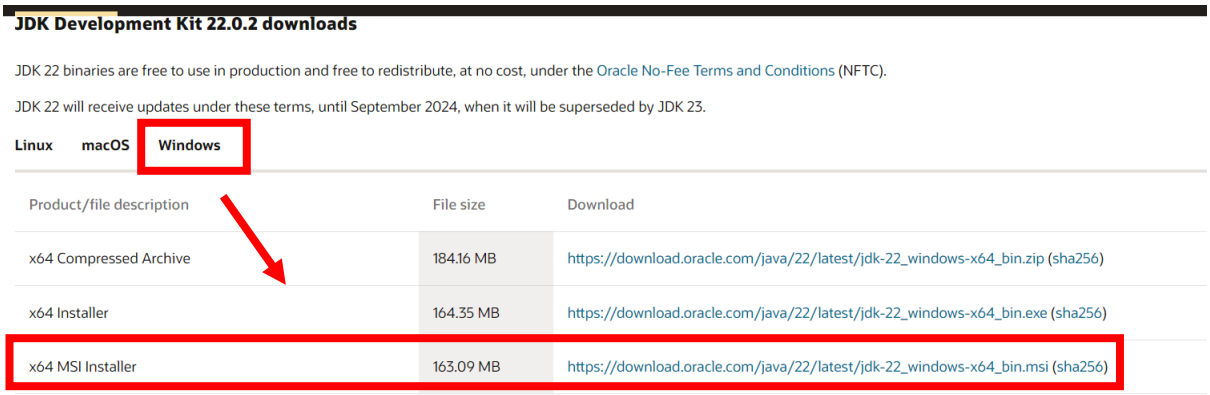


Downloading the java from website

Step 16: Go to google type jdk download and select the link

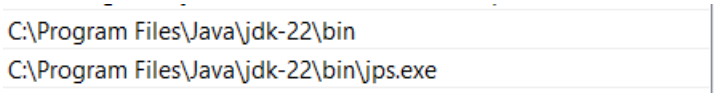


Step 17: Download 22.0.2 jdk by Switching to the windows tab on this page and download the x64 MSI installer



Step 18 : Open the Downloaded jdk kit and choose the directory and click Next and install it .

Step19: Go to Environment Variable and add Path of the java and JPS.exe file



Creating the Maven Project

Step 20: Open command prompt and type 'mvn --version' for verification of maven

```
C:\Users\geoffrey>mvn --version
Apache Maven 3.9.8 (36645f6c9b5079805ea5009217e36f2cffd34256)
Maven home: C:\maven\apache-maven-3.9.8
Java version: 22.0.2, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-22
Default locale: en_IN, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```

Step 21: Change the directory to the respective folder.

```
C:\Users\geoffrey>cd /

C:\>
```

Step 22: Create the maven project

```
C:\>mvn archetype:generate -DgroupId=com.example.SocketWordCount -DartifactId=SocketWordCount -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

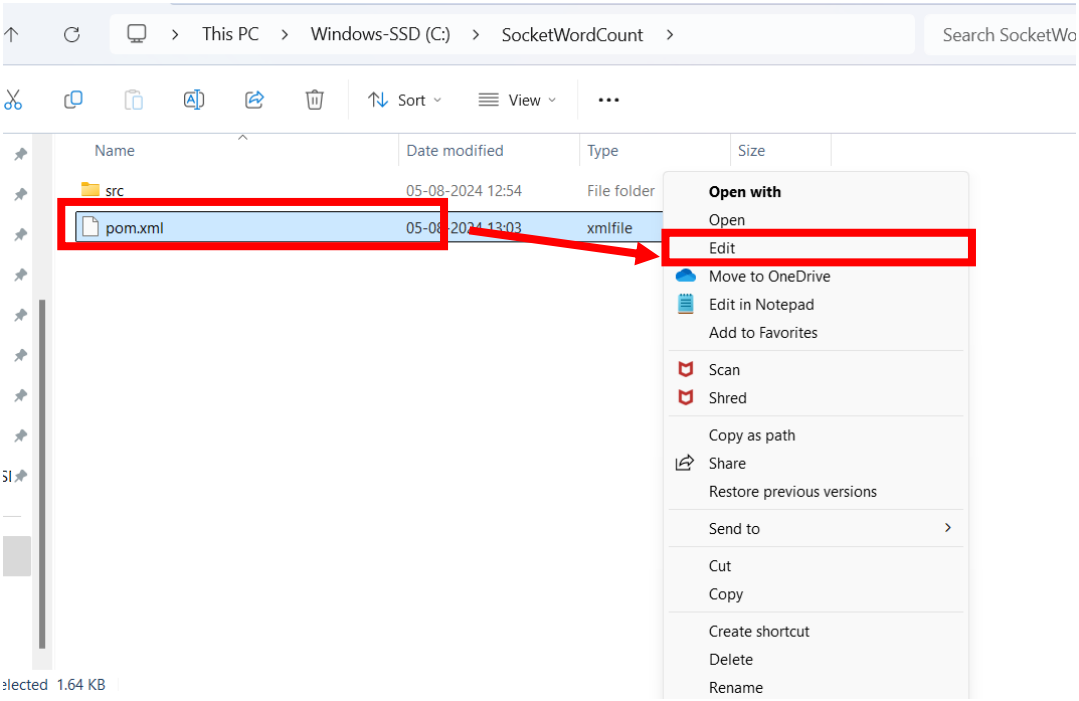
mvn archetype:generate -DgroupId=com.example.SocketWordCount -DartifactId=SocketWordCount -DarchetypeArtifactId=maven-archetype-quickstart -DInteractiveMode=False

```
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< org.apache.maven:standalone-pom >-------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] --------------------------------[ pom ]---------------------------------
[INFO]
[INFO] >>> archetype:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< archetype:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO]
[INFO] --- archetype:3.2.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.0
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: basedir, Value: C:\
[INFO] Parameter: package, Value: com.example.SocketWordCount
[INFO] Parameter: groupId, Value: com.example.SocketWordCount
[INFO] Parameter: artifactId, Value: SocketWordCount
[INFO] Parameter: packageName, Value: com.example.SocketWordCount
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\SocketWordCount
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.924 s
[INFO] Finished at: 2024-08-05T12:54:04+05:30
[INFO] ------------------------------------------------------------------------
```

Step 23: Change the directory to the maven project directory which has been created

```
C:\>cd SocketWordCount
```

Go to the C:/SocketWordCount/pom.xml and Right click and select edit




```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.example.SocketWordCount</groupId>
    <artifactId>SocketWordCount</artifactId>
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>SocketWordCount</name>
    <url>http://maven.apache.org</url>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
```

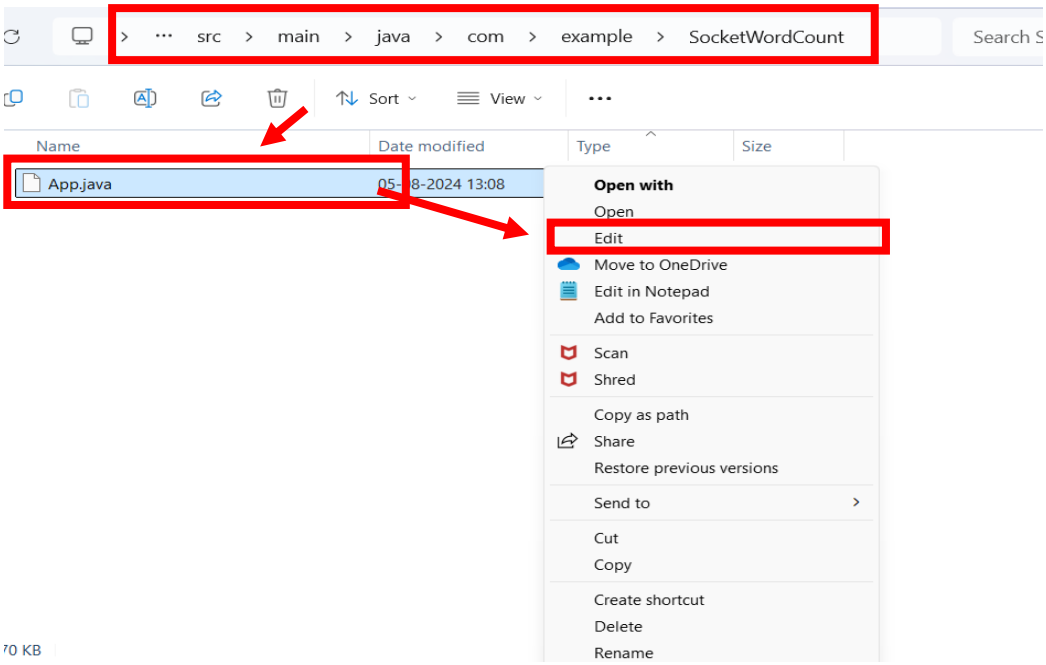
```
        <dependency>
            <groupId>org.apache.flink</groupId>
            <artifactId>flink-java</artifactId>
            <version>1.18.0</version>
        </dependency>
        <dependency>
            <groupId>org.apache.flink</groupId>
            <artifactId>flink-streaming-java</artifactId>
            <version>1.19.1</version> <!-- Update to match your
Flink version -->
            <scope>provided</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.1</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-jar-plugin</artifactId>
                <version>3.2.0</version>
                <configuration>
                    <archive>
                        <manifest>
                            <addClasspath>true</addClasspath>

<mainClass>com.example.SocketWordCount.SocketWordCount</mainClass>
                        </manifest>
                    </archive>
                </configuration>
            </plugin>
        </plugins>
    </build>
```

</project>

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd https://mvnrepository.com/artifact/org.apache.flink/flink-streaming-java">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example.SocketWordCount</groupId>
  <artifactId>SocketWordCount</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>SocketWordCount</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-java</artifactId>
        <version>1.18.0</version>
    </dependency>

   <dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-streaming-java</artifactId>
    <version>1.17.1</version>
    <scope>provided</scope>
   </dependency>

  </dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-jar-plugin</artifactId>
            <version>3.2.0</version>
            <configuration>
                <archive>
                    <manifest>
                        <addClasspath>true</addClasspath>
                        <mainClass>com.example.SocketWordCount.SocketWordCount</mainClass>
                    </manifest>
                </archive>
            </configuration>
        </plugin>
    </plugins>
</build>

</project>
```

Step 25: Now edit the App.java file

Go to the C:\SocketWordCount\src\main\java\com\example\SocketWordCount\ App.java **and Right click and select edit**




```java
package com.example.SocketWordCount;

import org.apache.flink.api.common.functions.FlatMapFunction;
import org.apache.flink.api.java.tuple.Tuple2;
import org.apache.flink.streaming.api.datastream.DataStream;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.util.Collector;

public class SocketWordCount {

    public static void main(String[] args) throws Exception {
        // Set up the execution environment
        final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();

        // Connect to the socket server on localhost:9999
        DataStream<String> text = env.socketTextStream("localhost", 9999);

        // Parse the data, group by word, and aggregate the counts
        DataStream<Tuple2<String, Integer>> counts = text
            .flatMap(new Tokenizer())
            .keyBy(value -> value.f0)
            .sum(1);

        // Print the result to the console
        counts.print();

        // Execute the program
        env.execute("Socket WordCount Example");
    }

    // User-defined function to split lines into words
    public static final class Tokenizer implements FlatMapFunction<String, Tuple2<String, Integer>> {
        @Override
        public void flatMap(String value, Collector<Tuple2<String, Integer>> out) {
            // Normalize and split the line into words
            String[] tokens = value.toLowerCase().split("\\W+");

            // Emit the words with a count of 1
            for (String token : tokens) {
                if (token.length() > 0) {
                    out.collect(new Tuple2<>(token, 1));
                }
            }
        }
    }
}
```

Step 26: Open the java file and to code it , Delete everything in the file and Copy all and paste in the document and save the file

```
package com.example.SocketWordCount;
import org.apache.flink.api.common.functions.FlatMapFunction;
import org.apache.flink.api.java.tuple.Tuple2;
import org.apache.flink.streaming.api.datastream.DataStream;
import
org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.util.Collector;
public class SocketWordCount {

    public static void main(String[] args) throws Exception {
        // Set up the execution environment
        final StreamExecutionEnvironment env =
StreamExecutionEnvironment.getExecutionEnvironment();

        // Connect to the socket server on localhost:9999
        DataStream<String> text = env.socketTextStream("localhost", 9999);

        // Parse the data, group by word, and aggregate the counts
        DataStream<Tuple2<String, Integer>> counts = text
            .flatMap(new Tokenizer())
            .keyBy(value -> value.f0)
            .sum(1);

        // Print the result to the console
        counts.print();

        // Execute the program
        env.execute("Socket WordCount Example");
    }

    // User-defined function to split lines into words
    public static final class Tokenizer implements FlatMapFunction<String,
Tuple2<String, Integer>> {
        @Override
        public void flatMap(String value, Collector<Tuple2<String,
Integer>> out) {
            // Normalize and split the line into words
            String[] tokens = value.toLowerCase().split("\\W+");

            // Emit the words with a count of 1
            for (String token : tokens) {
                if (token.length() > 0) {
                    out.collect(new Tuple2<>(token, 1));
                }
            }
        }
    }
}
```

Step 27: Rename the file to `SocketWordCount`.java

Note: Class name and file name of java program should be same.

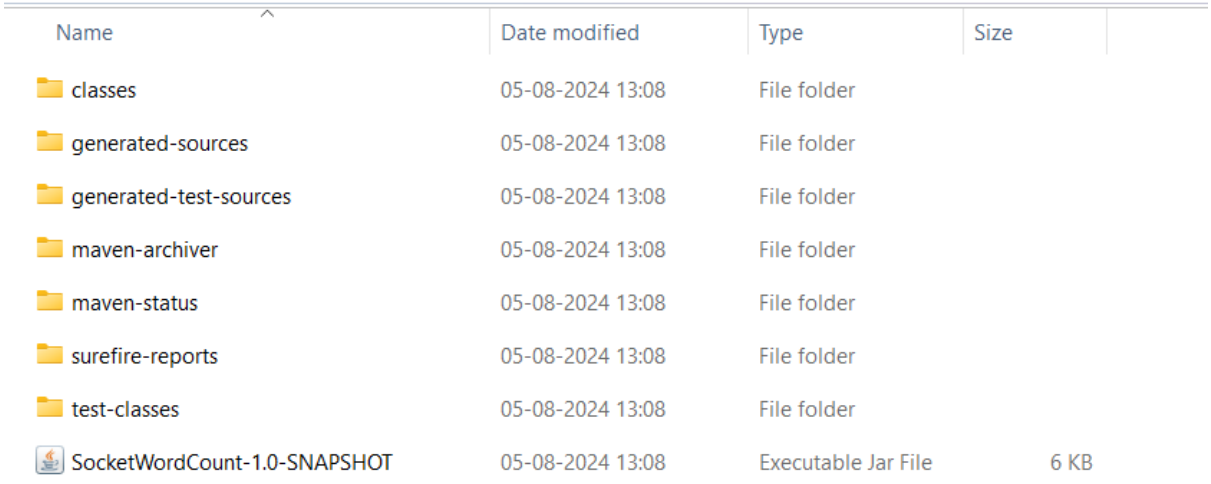
Step 28: Open the command prompt and type cd SocketWordCount **(Your created folder name using maven)**

Step 29: Now type mvn clean install (or) mvn clean package (or) mvn clean compile**(jar file will be created)**

```
C:\SocketWordCount>mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------< com.example.SocketWordCount:SocketWordCount >-------------
[INFO] Building SocketWordCount 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ SocketWordCount ---
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ SocketWordCount ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\SocketWordCount\src\main\resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ SocketWordCount ---
[INFO] Recompiling the module because of changed source code.
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 1 source file with javac [debug target 1.8] to target\classes
[WARNING] bootstrap class path is not set in conjunction with -source 8
  not setting the bootstrap class path may lead to class files that cannot run on JDK 8
    --release 8 is recommended instead of -source 8 -target 1.8 because it sets the bootstrap class path automatically
[WARNING] source value 8 is obsolete and will be removed in a future release
[WARNING] target value 8 is obsolete and will be removed in a future release
[WARNING] To suppress warnings about obsolete options, use -Xlint:-options.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ SocketWordCount ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\SocketWordCount\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ SocketWordCount ---
[INFO] Recompiling the module because of changed dependency.
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 1 source file with javac [debug target 1.8] to target\test-classes
[WARNING] bootstrap class path is not set in conjunction with -source 8
```

```
[INFO] Recompiling the module because of changed dependency.
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 1 source file with javac [debug target 1.8] to target\test-classes
[WARNING] bootstrap class path is not set in conjunction with -source 8
 not setting the bootstrap class path may lead to class files that cannot run on JDK 8
   --release 8 is recommended instead of -source 8 -target 1.8 because it sets the bootstrap class path automatically
[WARNING] source value 8 is obsolete and will be removed in a future release
[WARNING] target value 8 is obsolete and will be removed in a future release
[WARNING] To suppress warnings about obsolete options, use -Xlint:-options.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ SocketWordCount ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit.JUnit3Provider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.example.SocketWordCount.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.017 s -- in com.example.SocketWordCount.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.2.0:jar (default-jar) @ SocketWordCount ---
[INFO] Building jar: C:\SocketWordCount\target\SocketWordCount-1.0-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.240 s
[INFO] Finished at: 2024-08-05T13:00:45+05:30
[INFO] ------------------------------------------------------------------------
```

Step 30 : Now go into jar generated folder (Location directory shown in Building jar **)**



Name	Date modified	Type	Size
classes	05-08-2024 13:08	File folder	
generated-sources	05-08-2024 13:08	File folder	
generated-test-sources	05-08-2024 13:08	File folder	
maven-archiver	05-08-2024 13:08	File folder	
maven-status	05-08-2024 13:08	File folder	
surefire-reports	05-08-2024 13:08	File folder	
test-classes	05-08-2024 13:08	File folder	
SocketWordCount-1.0-SNAPSHOT	05-08-2024 13:08	Executable Jar File	6 KB

Right Click SocketWordCount-1.0-SNAPSHOT **and select copy and paste the jar file to the** C: **drive**