

**Tribhuvan University
Institute of Science and Technology
Patan Multiple Campus**



**LAB REPORT ON
CLOUD COMPUTING (BIT408)**

A partial fulfilment of the requirements for Bachelors in Information Technology(BIT)

Submitted To:
Sachita Nand Mishra
Lecturer, Patan Multiple Campus

Submitted By:
Suprabha Aryal
BIT 7th Semester
Exam Roll No. : 604/078
Class Roll No. :01/078

List of Lab Works

S.N.	Title	Date	Signature
1.	Installation and Testing of tomcat Server on Java IDE		
2.	SOAP Web Service in Java (JAX-WS)		
3.	Consuming SOAP Web services in java		
4.	Implement Windows Hyper V virtualization		
5.	Implement Virtualization using VirtualBox		
6.	Simulate a cloud scenario using CloudSim		
7.	Installation and testing of Hadoop single node cluster on windows		
8.	Map reduce word count program using java		
9.	Develop and application for Google App Engine		

Lab-1 Installation and testing of Apache Tomcat in Java IDE

Step 1: Download Tomcat from <https://tomcat.apache.org/download-10.cgi>:

The screenshot shows the Apache Tomcat download page for version 10.1.44. The left sidebar contains links for Tomcat versions 11.0, 10.1, 9.0, Upgrading, Connectors, Native 2, Native 1.3, Wiki, Migration Guide, Presentations, and Specifications. Under 'Problems?', there are links for Security Reports, Find help, FAQ, Mailing Lists, Bug Database, and IRC. Under 'Get Involved', there are links for Overview, Source code, Buildbot, and Tools. Under 'Media', there is a link for Twitter. The main content area is titled '10.1.44' and includes a note: 'Please see the [README](#) file for packaging information. It explains what every distribution contains.' Below this is a section titled 'Binary Distributions' with a bulleted list of download links for Core, Full documentation, Deployer, and Embedded components in both zip and tar.gz formats. Another section titled 'Source Code Distributions' lists tar.gz and zip links.

Step: Extract the downloaded tomcat in your preferred folder:

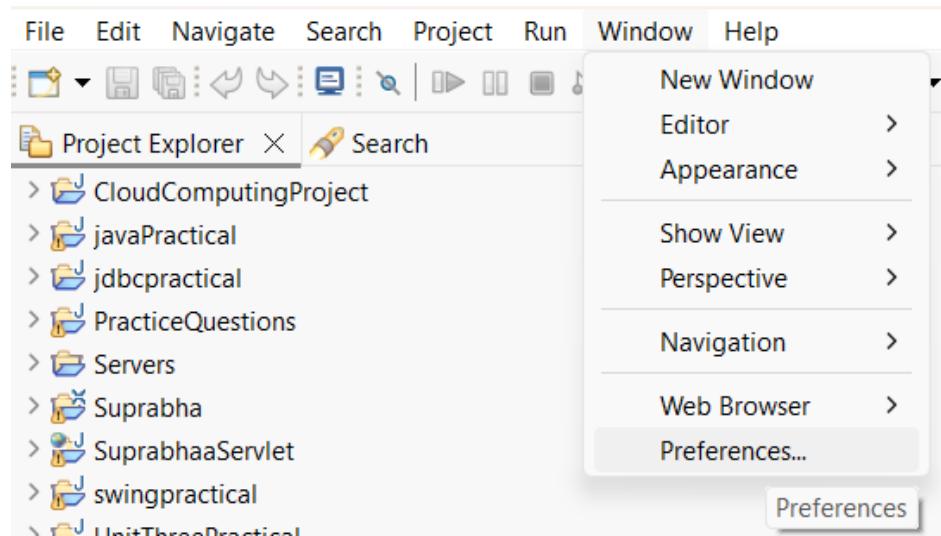
The screenshot shows a file explorer window with the following directory structure:

```
Downloads > SuprabhaFiles > TomcatInstallation > apache-tomcat-10.1.44
```

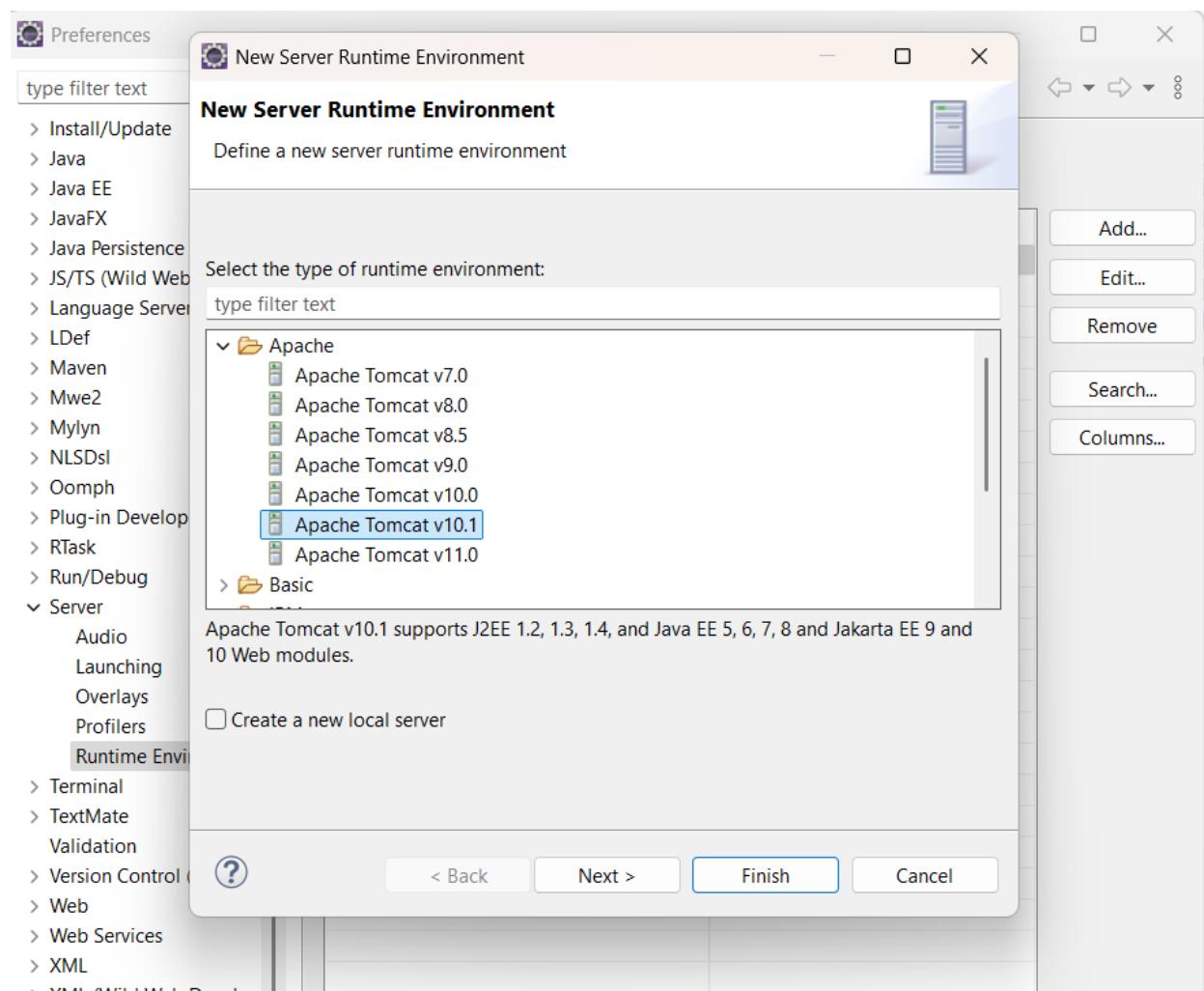
The contents of the 'apache-tomcat-10.1.44' folder are:

Name	Date modified	Type	Size
BUILDING	8/26/2025 9:25 AM	Text Document	25 KB
CONTRIBUTING	8/26/2025 9:25 AM	Markdown Source ...	7 KB
LICENSE	8/26/2025 9:25 AM	File	61 KB
NOTICE	8/26/2025 9:25 AM	File	3 KB
README	8/26/2025 9:25 AM	Markdown Source ...	4 KB
RELEASE-NOTES	8/26/2025 9:25 AM	File	7 KB
RUNNING	8/26/2025 9:25 AM	Text Document	17 KB
webapps	8/26/2025 9:25 AM	File folder	
lib	8/26/2025 9:25 AM	File folder	
temp	8/26/2025 9:25 AM	File folder	
bin	8/26/2025 9:25 AM	File folder	
conf	8/26/2025 9:25 AM	File folder	
logs	8/4/2025 1:14 PM	File folder	
work	8/4/2025 1:14 PM	File folder	

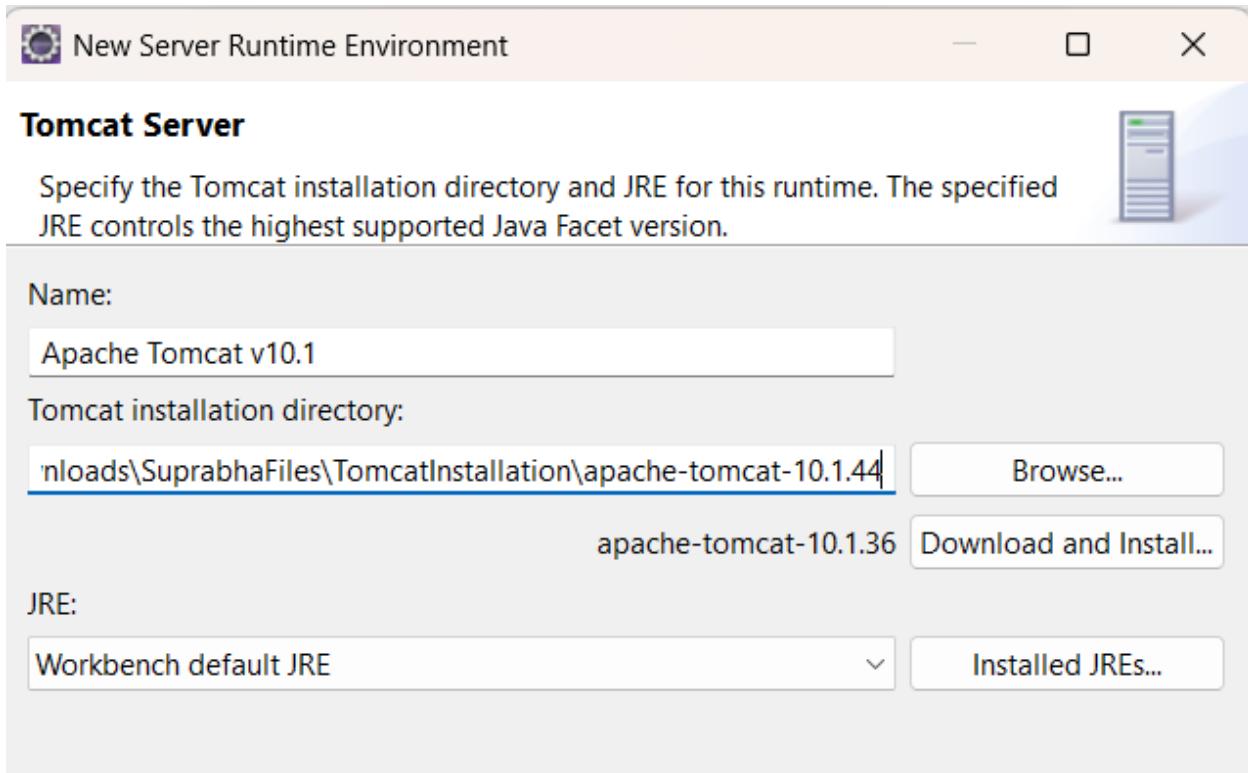
Step 3: Add Tomcat to the IDE. Here, I'm using Eclipse . First, Open the Window> Preferences:



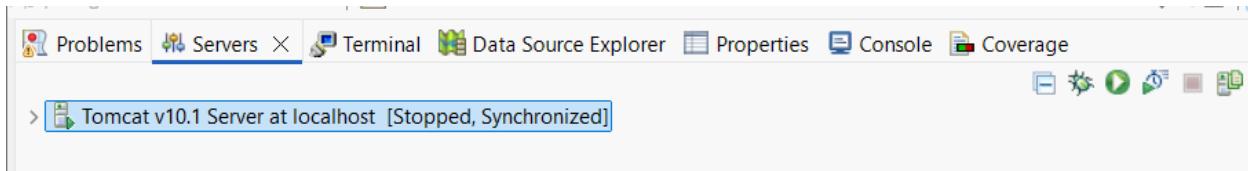
Step 4: After going to Preferences, Select the type of runtime environment:



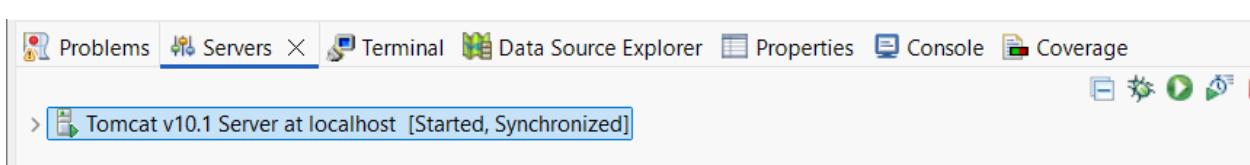
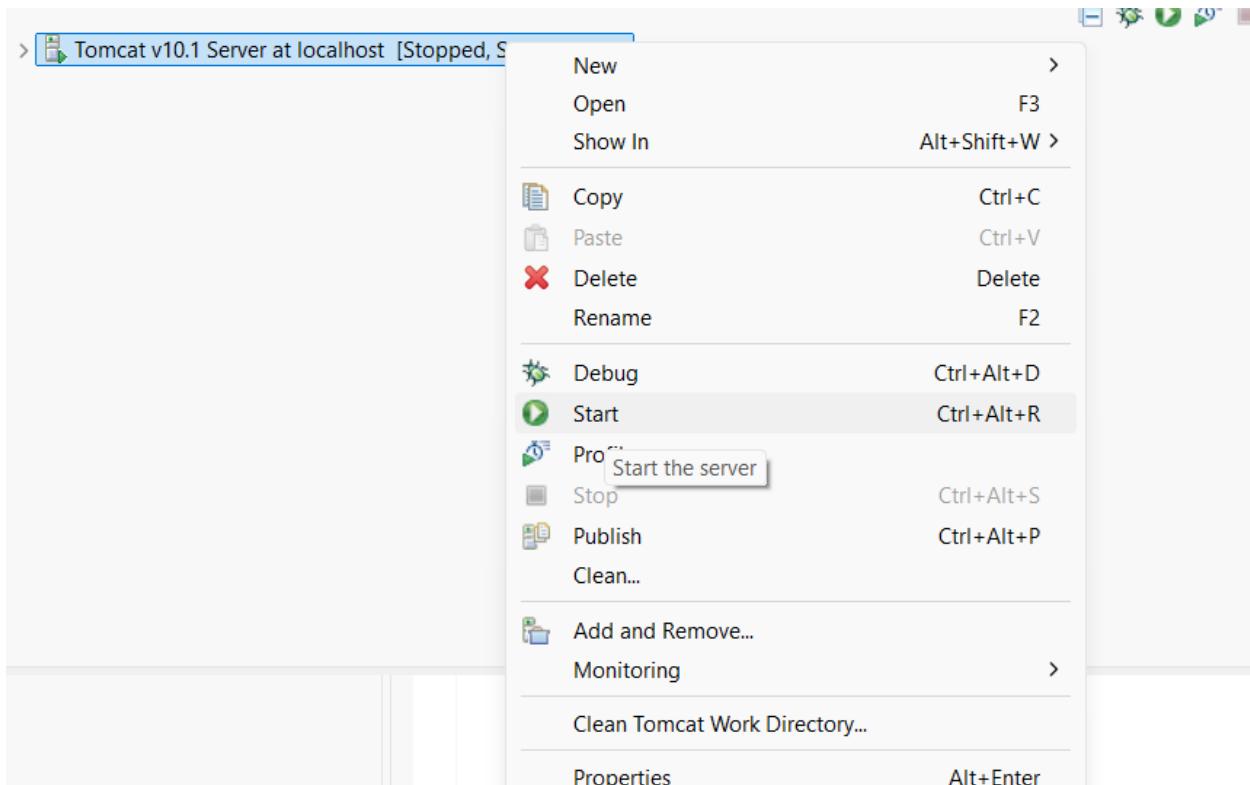
Step 5: And then click next and browse to the directory where you have extracted the apache tomcat and click finish.



The server is stopped,



Step 6: Start the Server:



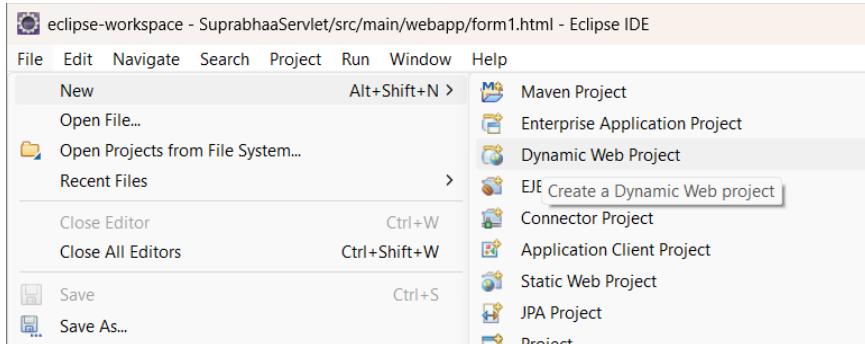
The screenshot shows the Eclipse IDE interface with the Console view active. The log output from the Tomcat server is displayed, indicating a successful startup. The log message is as follows:

```
Tomcat v10.1 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-24\bin\javaw.exe (Aug 26, 2025, 9:50:58 AM elapsed: 0:00:57)
INFO: At least one JAR was scanned for TDUs yet contained no TDUs. Enable debug logging for this log
Aug 26, 2025 9:51:03 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
Aug 26, 2025 9:51:03 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in [2449] milliseconds
```

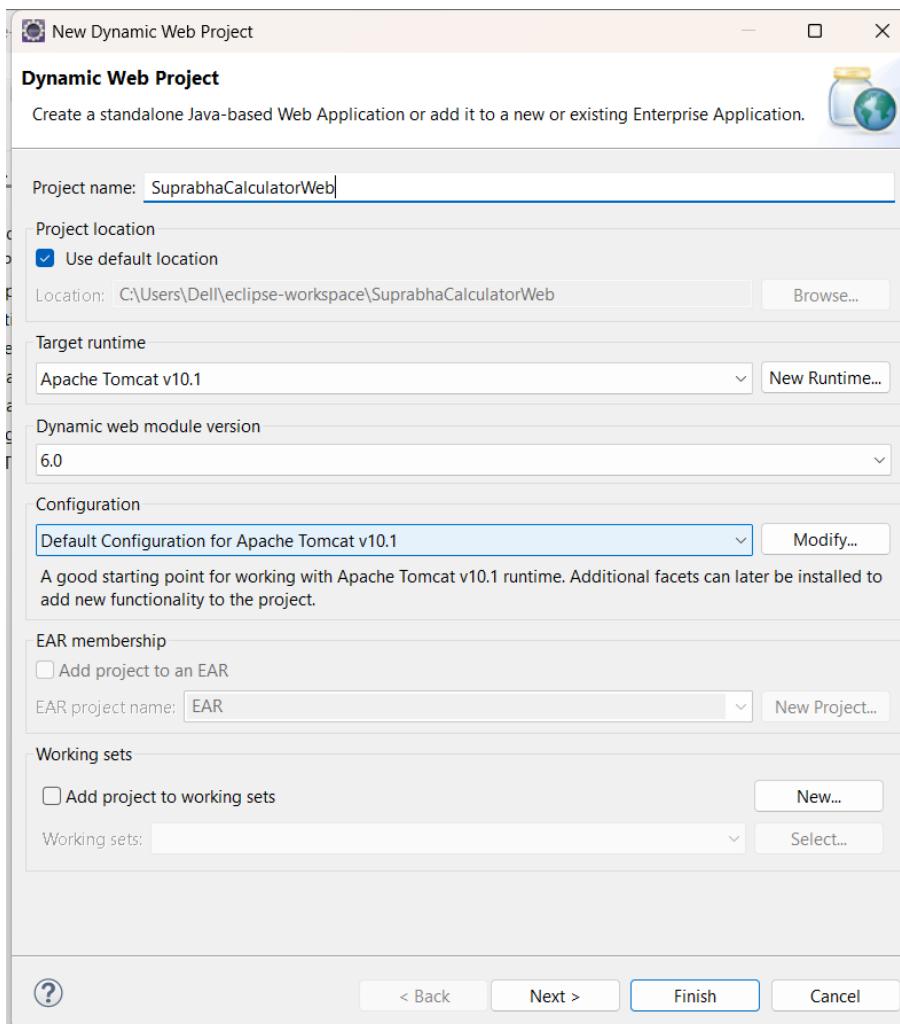
Lab-2 SOAP Web Service in Java (JAX-WS)

Develop a simple SOAP based Web Service in Java using JAX-WS, called as "CalculatorService".

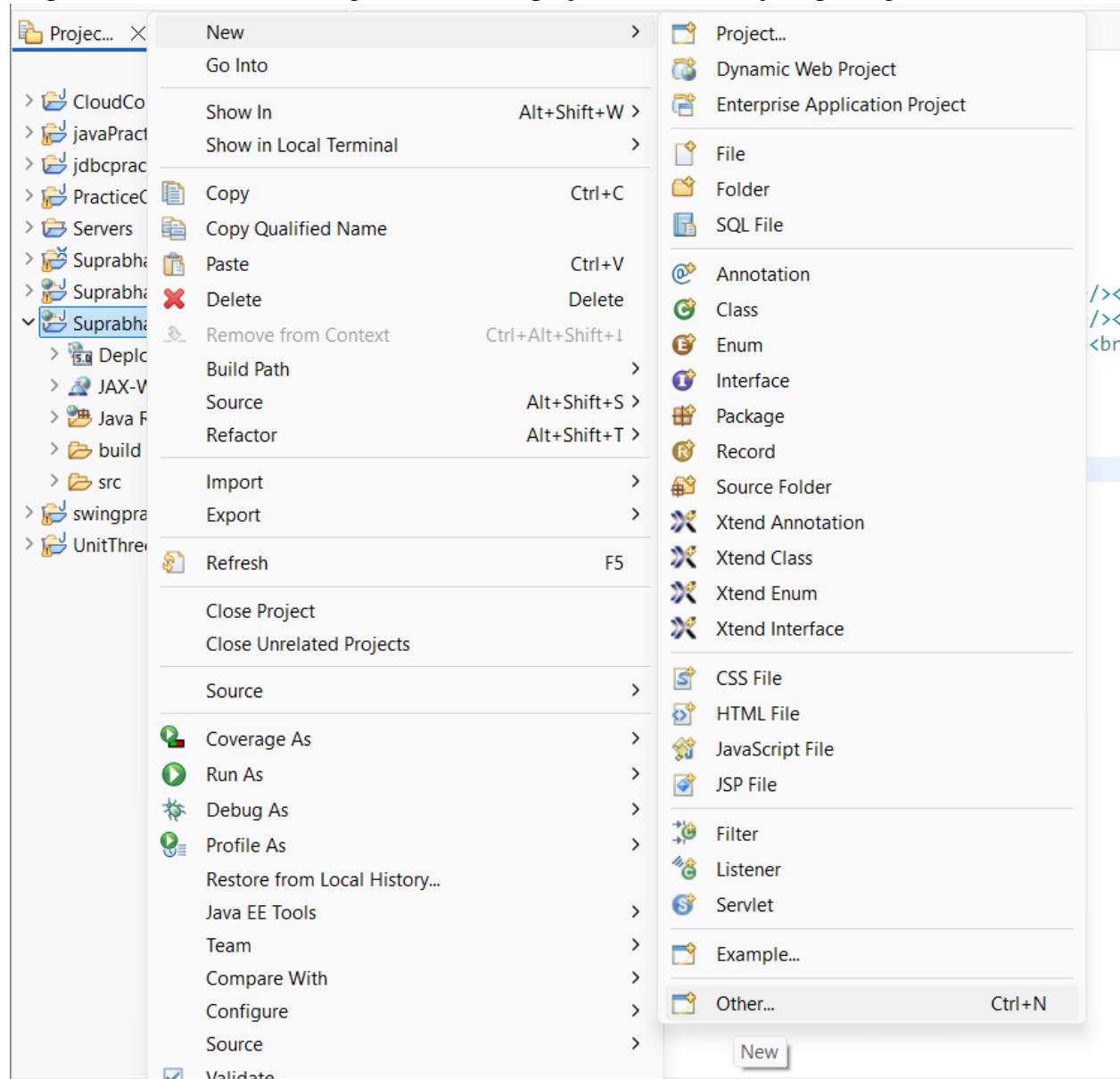
Step 1: Create a new Java Web application project. Go to, File > New > Dynamic Web Project:



Enter the project name and then click finish:

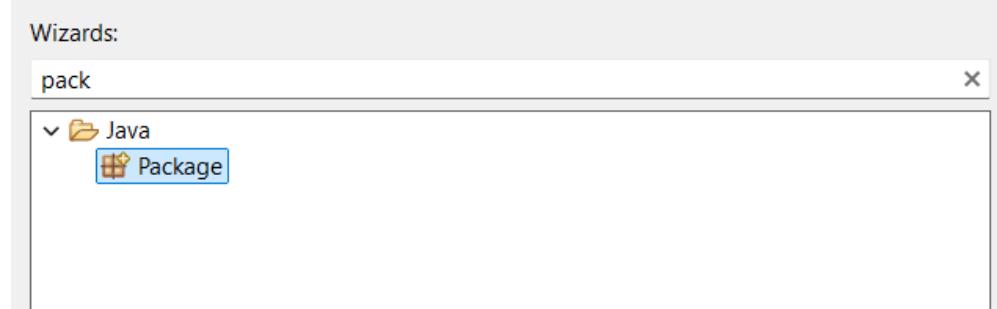


Step 2: Add a web service , right click on the project and create a java package:



Select a wizard

Create a Java package



Java Package

Create a new Java package.



Creates folders corresponding to packages.

Source folder:

SuprabhaCalculatorWeb/src/main/java

[Browse...](#)

Name:

com.demo.calc

Create package-info.java

Generate comments (configure templates and default value [here](#))

Create a CalculatorService.java class with
package com.demo.calc;

```
import javax.jws.WebMethod;
import javax.jws.WebService;

@WebService
public class CalculatorService {

    @WebMethod
    public int add(int a, int b) {
        return a + b;
    }

    @WebMethod
    public int subtract(int a, int b) {
        return a - b;
    }

    @WebMethod
    public int multiply(int a, int b) {
        return a * b;
    }

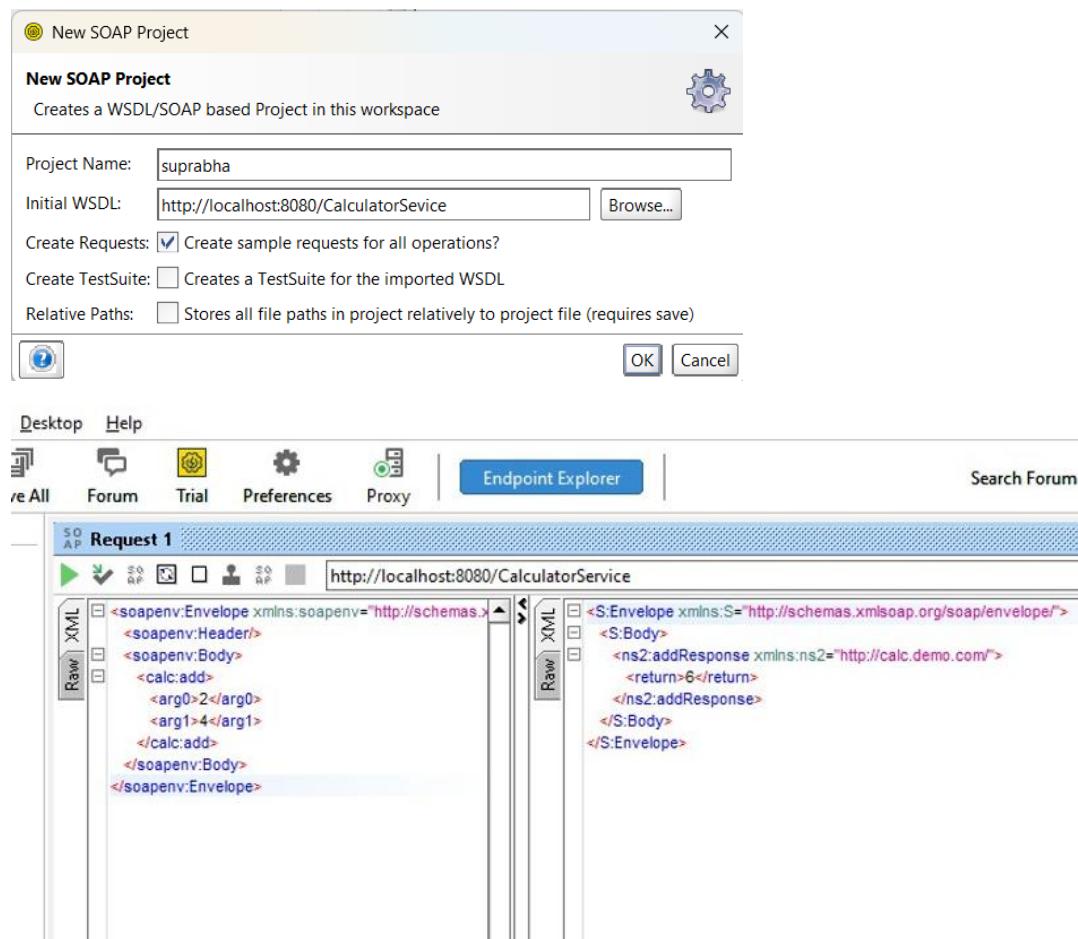
    @WebMethod
    public double divide(int a, int b) {
        if (b == 0) throw new IllegalArgumentException("Division by zero!");
        return (double) a / b;
    }
}
```

Create CaclulatorPublisher.java class with
package com.demo.calc;

```
import javax.xml.ws.Endpoint;
```

```
public class CalculatorPublisher {  
    public static void main(String[] args) {  
        String url = "http://localhost:8080/CalculatorService";  
        Endpoint.publish(url, new CalculatorService());  
        System.out.println("Service published at: " + url + "?wsdl");  
    }  
}  
  
_  
|  
|   src/main/java  
|   |   com.demo.calc  
|   |       CalculatorPublisher.java  
|   |       CalculatorService.java
```

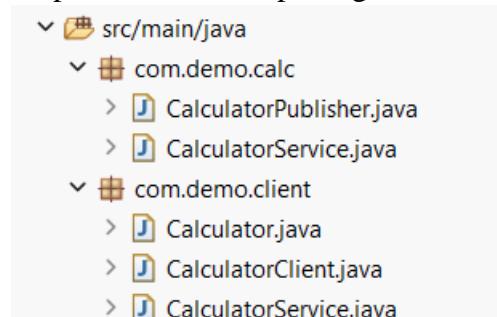
Using SOAP UI to test this file:



Lab-3 Consuming Soap Web Services in java

Client to make user of web services add method

Step 1: create a new package named client and create following interfaces and classes:



With the code as follows

```
package com.demo.client;
public class CalculatorClient {
    public static void main(String[] args) {
        CalculatorService service = new CalculatorService(null);
        Calculator port = service.getCalculatorPort();

        int result = port.add(7, 5);
        System.out.println("Result from CalculatorService: " + result);
    }
}
```

Create Calculator interface with:

```
package com.demo.client;

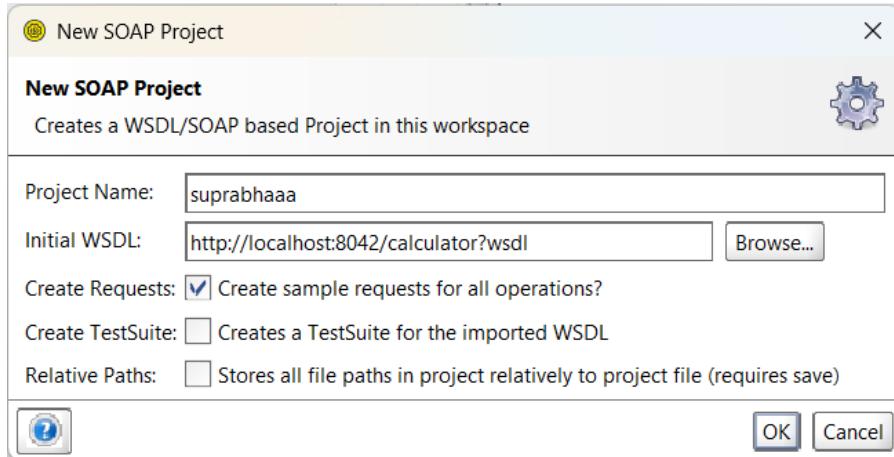
import javax.jws.WebMethod;
import javax.jws.WebService;
import javax.jws.soap.SOAPBinding;
@WebService(targetNamespace = "http://calc.demo.com/")
@SOAPBinding(style = SOAPBinding.Style.DOCUMENT, use =
SOAPBinding.Use.LITERAL) public interface Calculator {
    @WebMethod
    int add(int a, int b);
```

```
}
```

And, finally create a calculatorservice class extending service as such: package com.demo.client;

```
import javax.xml.namespace.QName; import  
javax.xml.ws.Service; import java.net.URL; public class  
CalculatorService extends Service {  
  
    private static final QName SERVICE_NAME = new  
    QName("http://calc.demo.com/",  
  
        "CalculatorService");  
    public CalculatorService(URL wsdlDocumentLocation) {  
        super(wsdlDocumentLocation, SERVICE_NAME);  
    }  
    public Calculator getCalculatorPort() {  
        return super.getPort(new QName("http://calc.demo.com/", "CalculatorServicePort"),  
  
Calculator.class);  
  
    }  
}
```

Compile the client and test it using SOAP UI



Desktop Help

All Forum Trial Preferences Proxy Endpoint Explorer Search Forum

Request 1

http://localhost:8042/CalculatorService

Raw XML

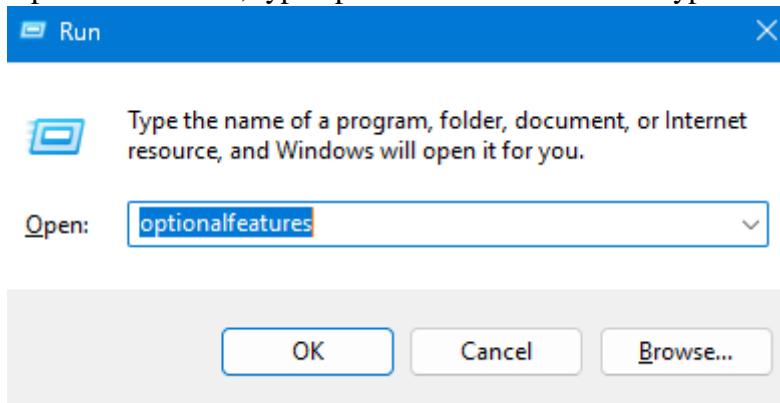
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Header/>
 <soapenv:Body>
 <calc:add>
 <arg0>10</arg0>
 <arg1>20</arg1>
 </calc:add>
 </soapenv:Body>
</soapenv:Envelope>

Raw XML

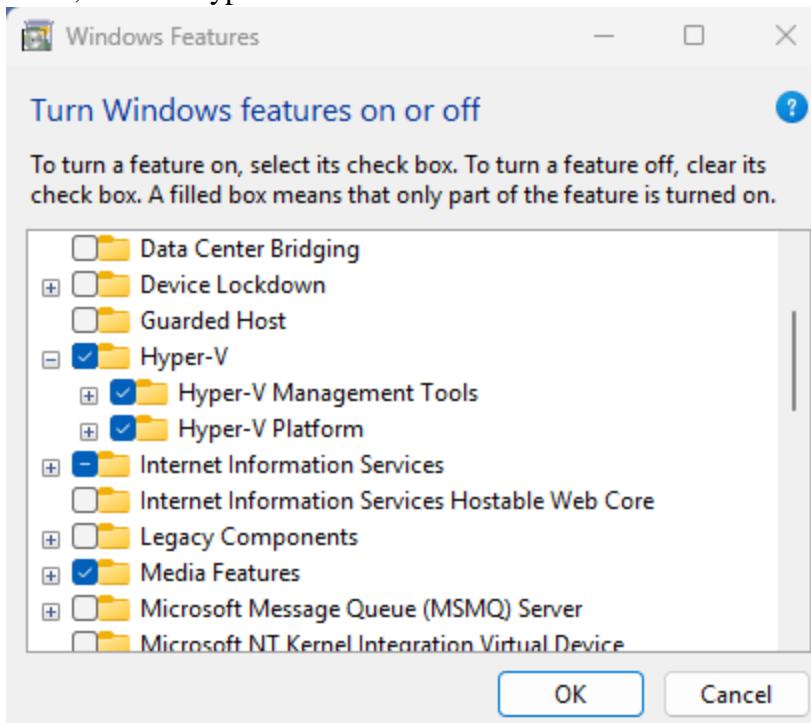
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
 <S:Body>
 <ns2:addResponse xmlns:ns2="http://calc.demo.com/">
 <return>30</return>
 </ns2:addResponse>
 </S:Body>
</S:Envelope>

Lab-4 Implement Windows Hyper V virtualization

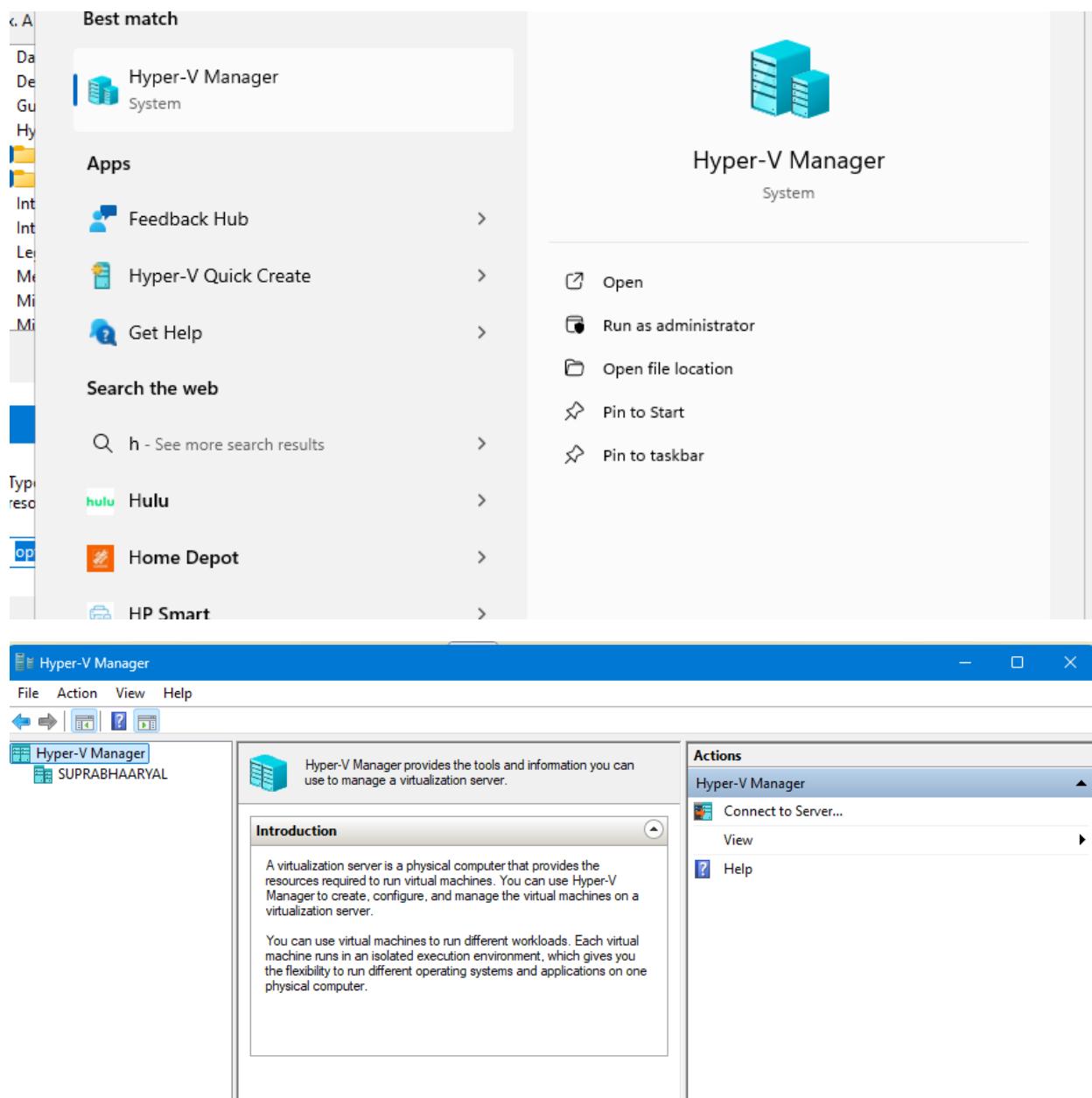
Open windows+R, type optional features to enable Hyper-V:



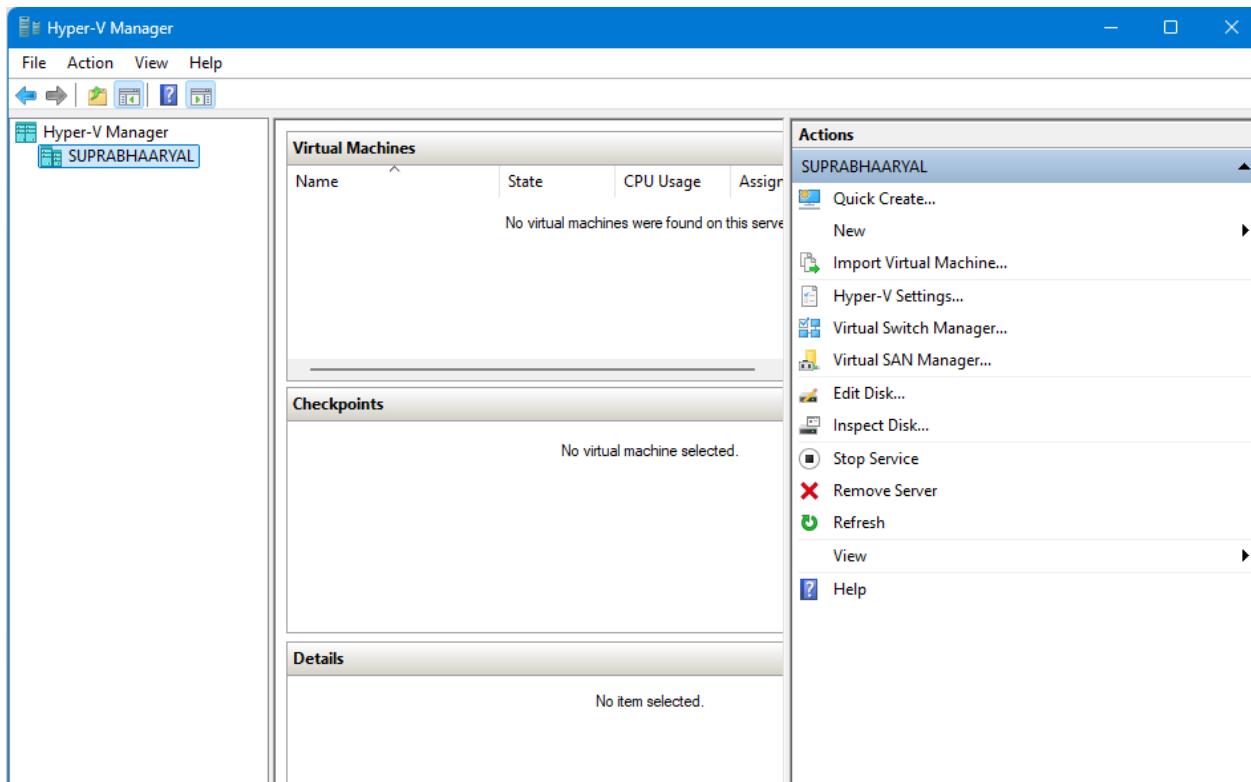
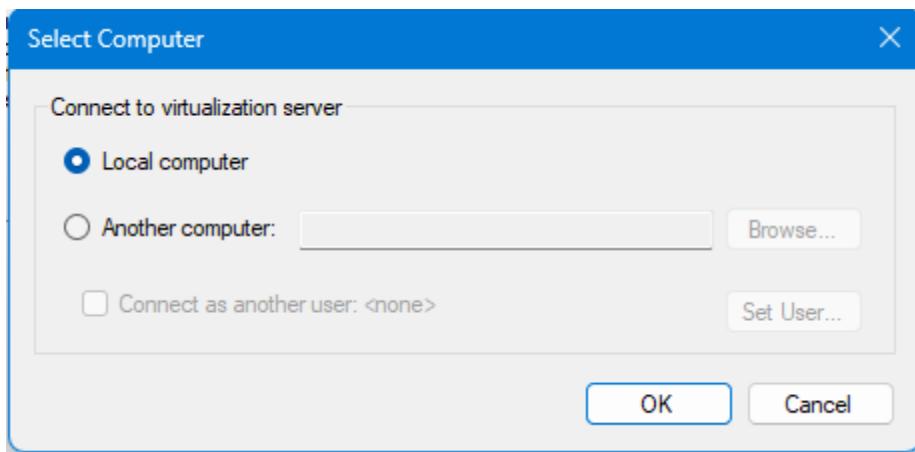
Then, Enable Hyper-V:



Launch Hyper-V Manager:



Connect to virtualization server as a Local computer:



Create a new Virtual Machine:

The screenshot shows the Windows Start Menu with the "Hyper-V Manager" pinned icon. The "Actions" menu is open under the "Hyper-V Manager" pinned icon. The "New" option is highlighted, which has opened the "New Virtual Machine Wizard". The wizard window title is "New Virtual Machine Wizard".

Before You Begin

This wizard helps you create a virtual machine. You can use virtual machines in place of physical computers for a variety of uses. You can use this wizard to configure the virtual machine now, and you can change the configuration later using Hyper-V Manager.

To create a virtual machine, do one of the following:

- Click Finish to create a virtual machine that is configured with default values.
- Click Next to create a virtual machine with a custom configuration.

Do not show this page again

< Previous Next > Finish Cancel

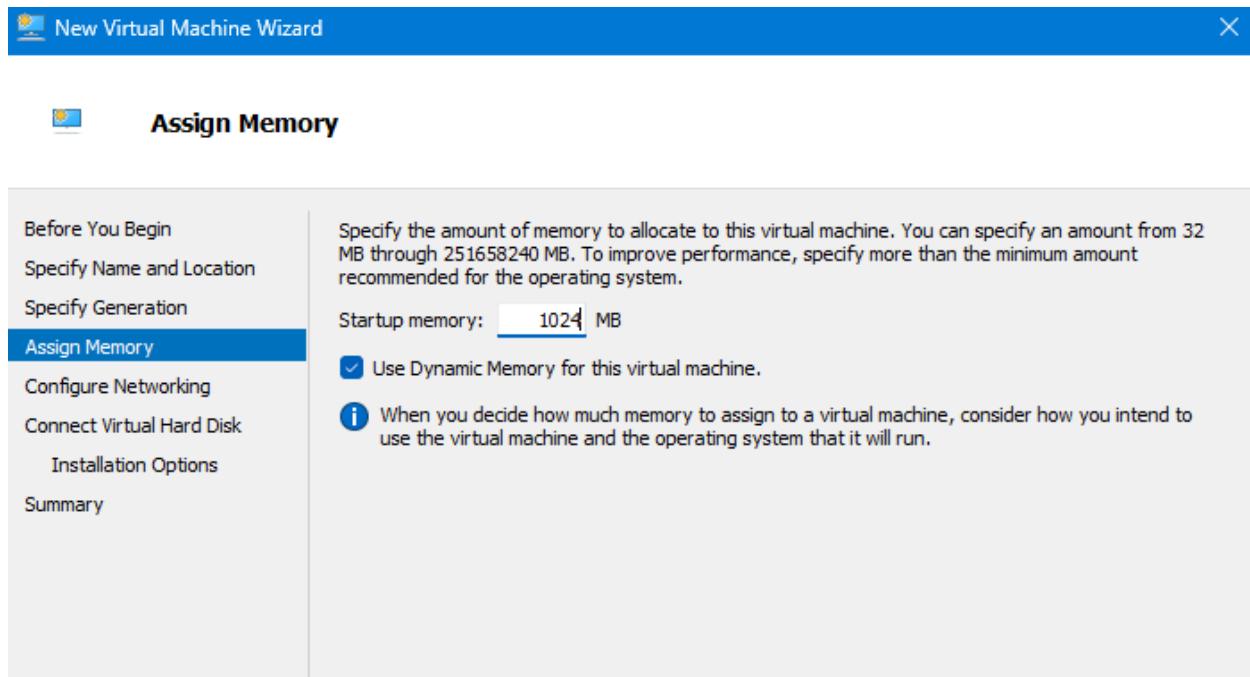
Specify Name and Location:

The screenshot shows the 'Specify Name and Location' step of the New Virtual Machine Wizard. On the left, a navigation pane lists steps: Before You Begin, Specify Name and Location (which is selected and highlighted in blue), Specify Generation, Assign Memory, Configure Networking, Connect Virtual Hard Disk, Installation Options, and Summary. The main pane contains instructions to choose a name and location for the virtual machine. It notes that the name is displayed in Hyper-V Manager and recommends using a name that helps identify the virtual machine. The 'Name' field is filled with 'SuprabhaHyper-V'. Below it, there's a checkbox for 'Store the virtual machine in a different location' which is unchecked. The 'Location' field shows 'C:\ProgramData\Microsoft\Windows\Hyper-V\' with a 'Browse...' button. A warning message states: 'If you plan to take checkpoints of this virtual machine, select a location that has enough free space. Checkpoints include virtual machine data and may require a large amount of space.' At the bottom are buttons for '< Previous', 'Next >', 'Finish', and 'Cancel'.

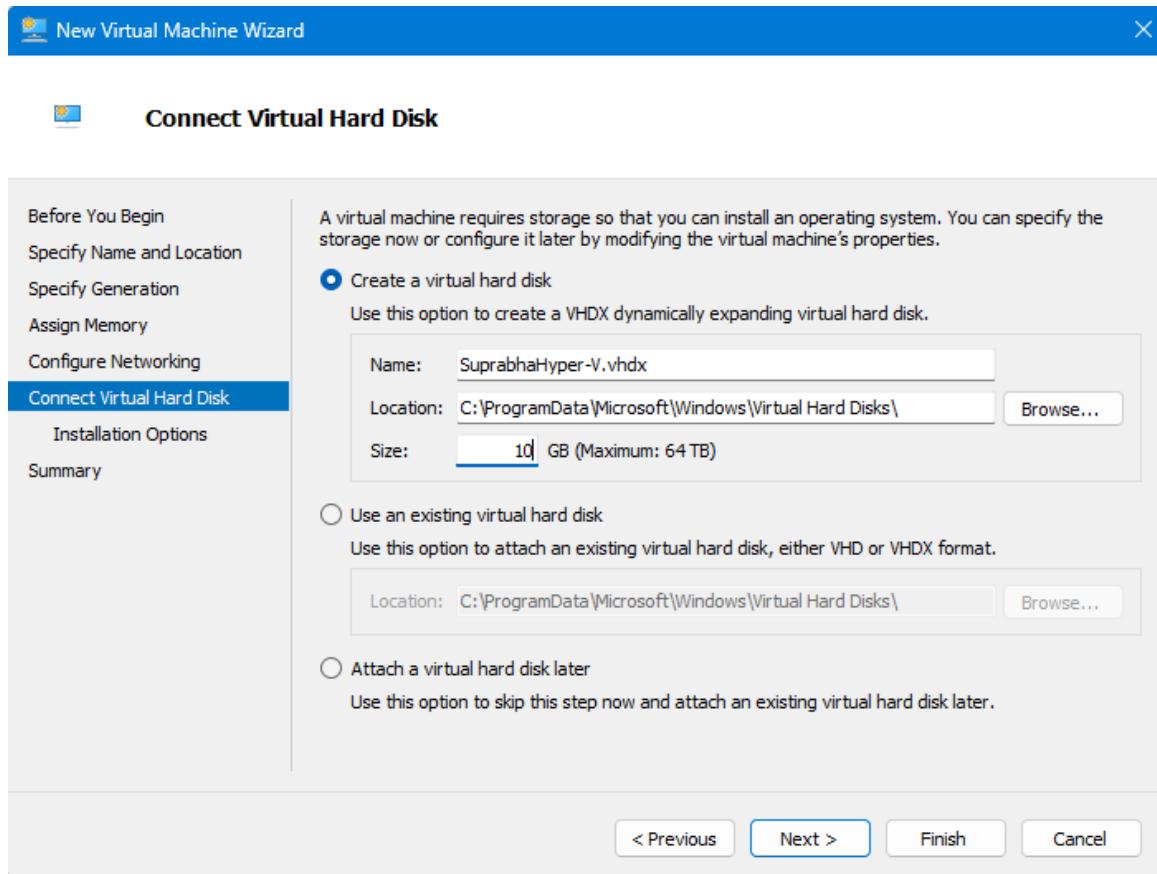
Specify Generation:

The screenshot shows the 'Specify Generation' step of the New Virtual Machine Wizard. The navigation pane on the left shows: Before You Begin, Specify Name and Location (selected), Specify Generation (highlighted in blue), Assign Memory, Configure Networking, Connect Virtual Hard Disk, Installation Options, and Summary. The main pane asks to choose the generation of the virtual machine. It offers two options: 'Generation 1' (radio button is empty) and 'Generation 2' (radio button is checked). A descriptive text explains that Generation 1 supports 32-bit and 64-bit guest operating systems and provides virtual hardware available in all previous versions of Hyper-V. A warning message for Generation 2 states: 'Once a virtual machine has been created, you cannot change its generation.' At the bottom are buttons for '< Previous', 'Next >', 'Finish', and 'Cancel'.

Assign memory :



Setup harddisk:





Installation Options

You can install an operating system now if you have access to the setup media, or you can install it later.

Install an operating system later

Install an operating system from a bootable CD/DVD-ROM

Media

Physical CD/DVD drive:

Image file (.iso):

Install an operating system from a bootable floppy disk

Media

Virtual floppy disk (.vfd):



You have successfully completed the New Virtual Machine Wizard. You are about to create the following virtual machine.

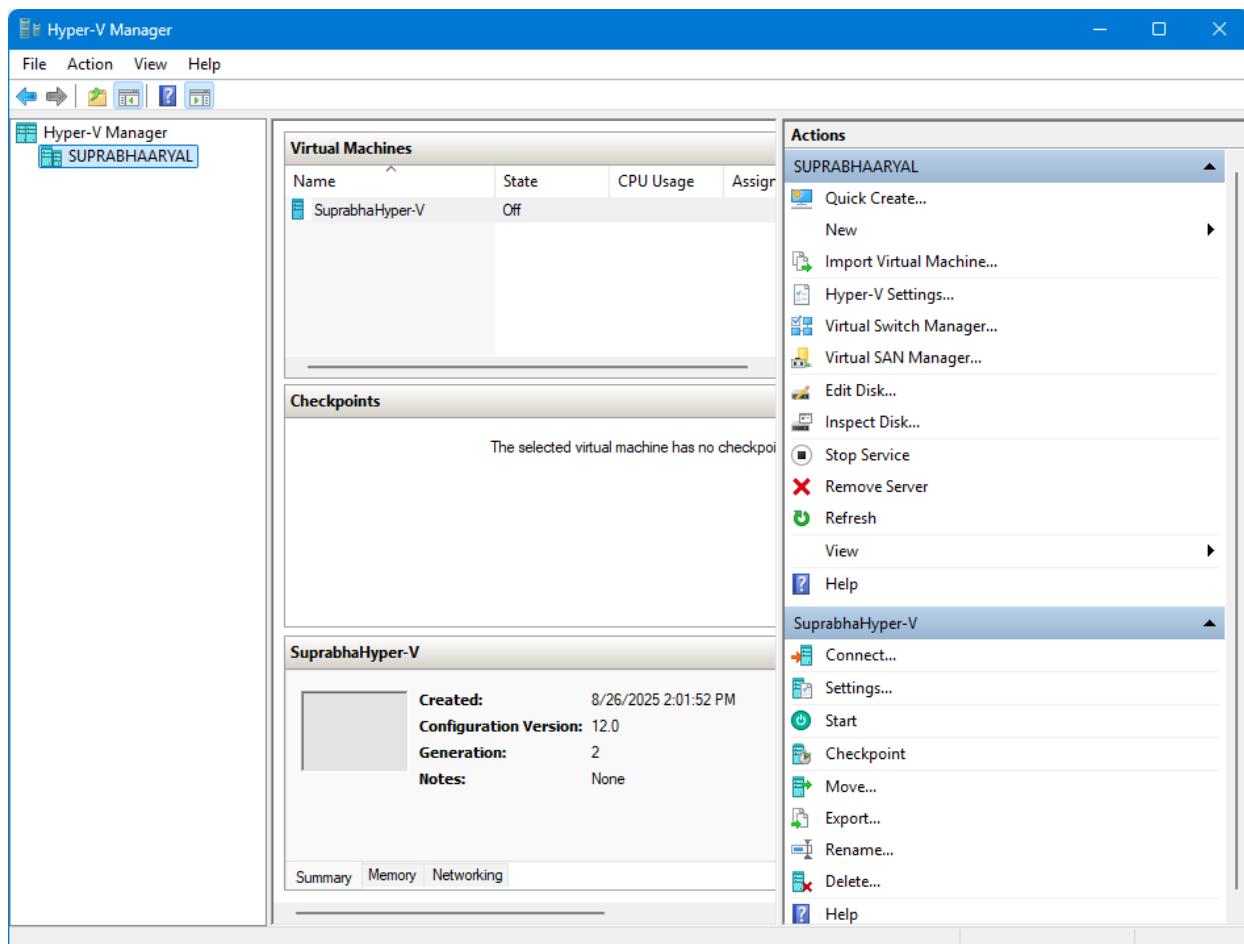
Description:

Name:	SuprabhaHyper-V
Generation:	Generation 2
Memory:	1024 MB
Network:	Not Connected
Hard Disk:	C:\ProgramData\Microsoft\Windows\Virtual Hard Disks\SuprabhaHyper-V.vhdx (\
Operating System:	Will be installed at a later time

To create the virtual machine and close the wizard, click Finish.

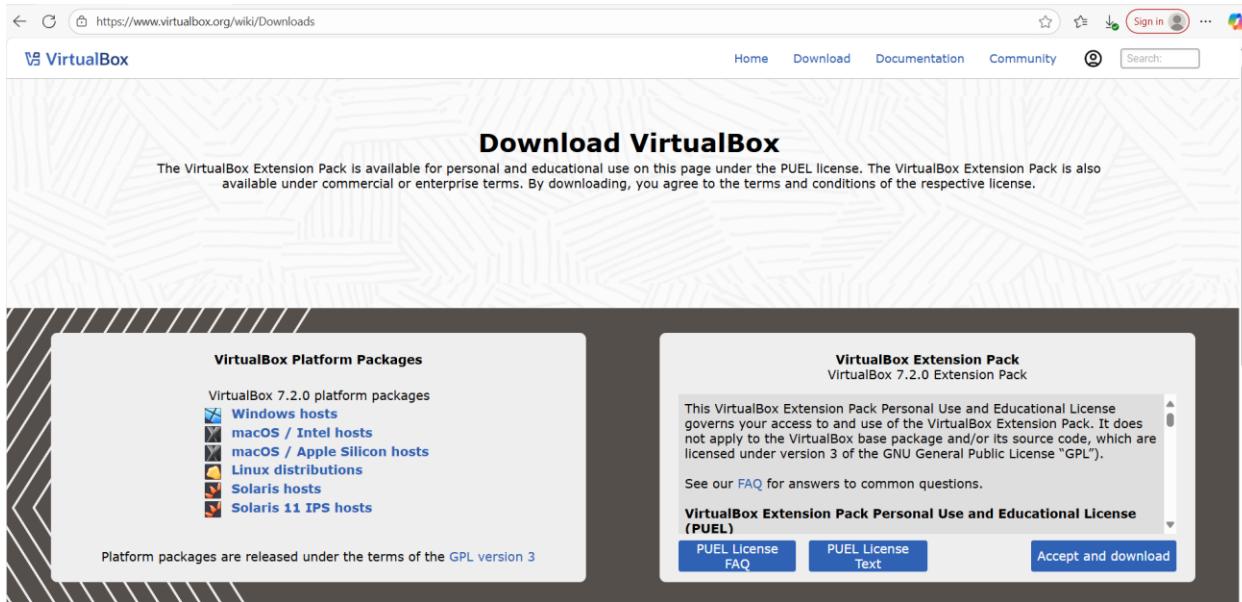
< Previous Next > **Finish** Cancel

The Setup is completed , Now you can connect to the Virtual Machine:

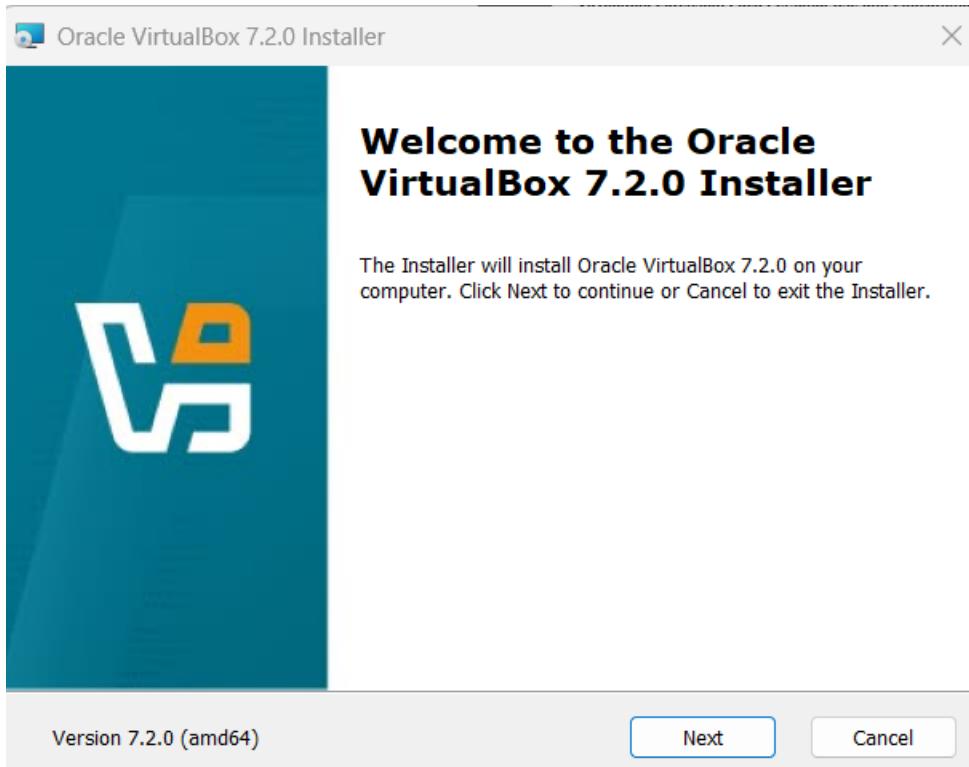


Lab-5 Implement Virtualization using VirtualBox

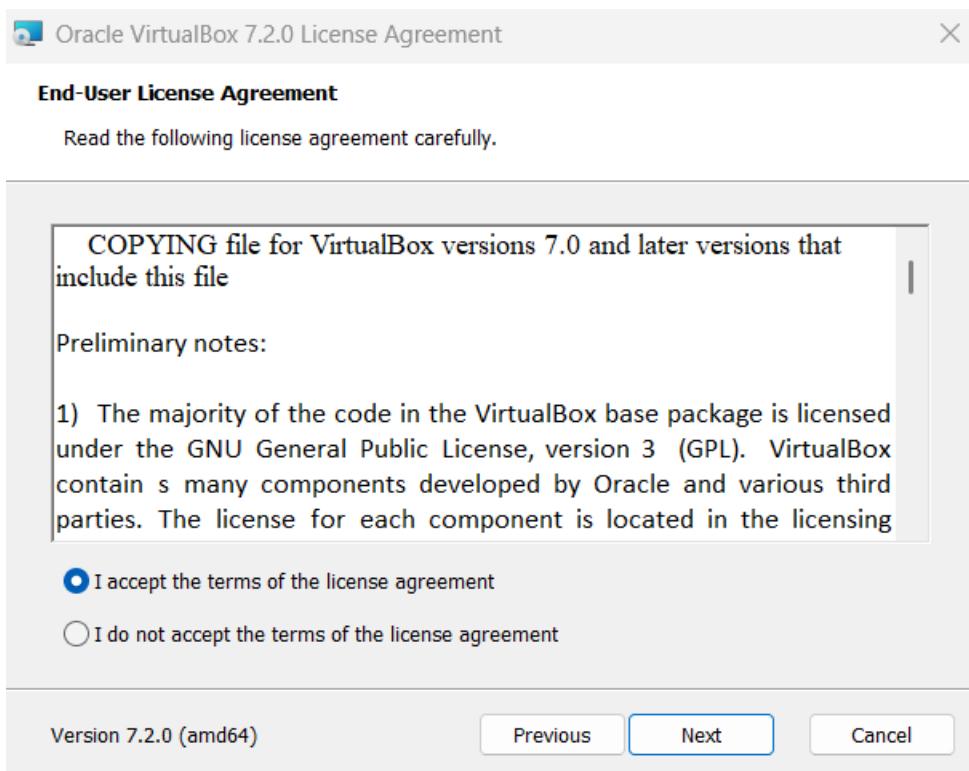
Download Virtualbox exe from <https://www.virtualbox.org/wiki/Downloads> :



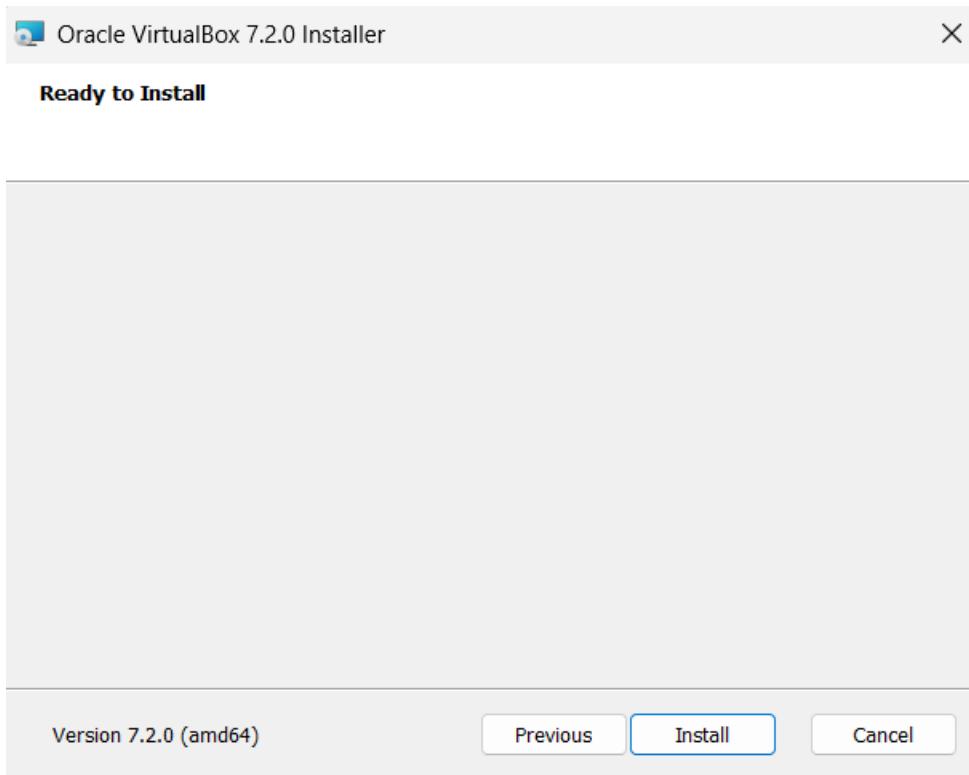
Follow the installation process:

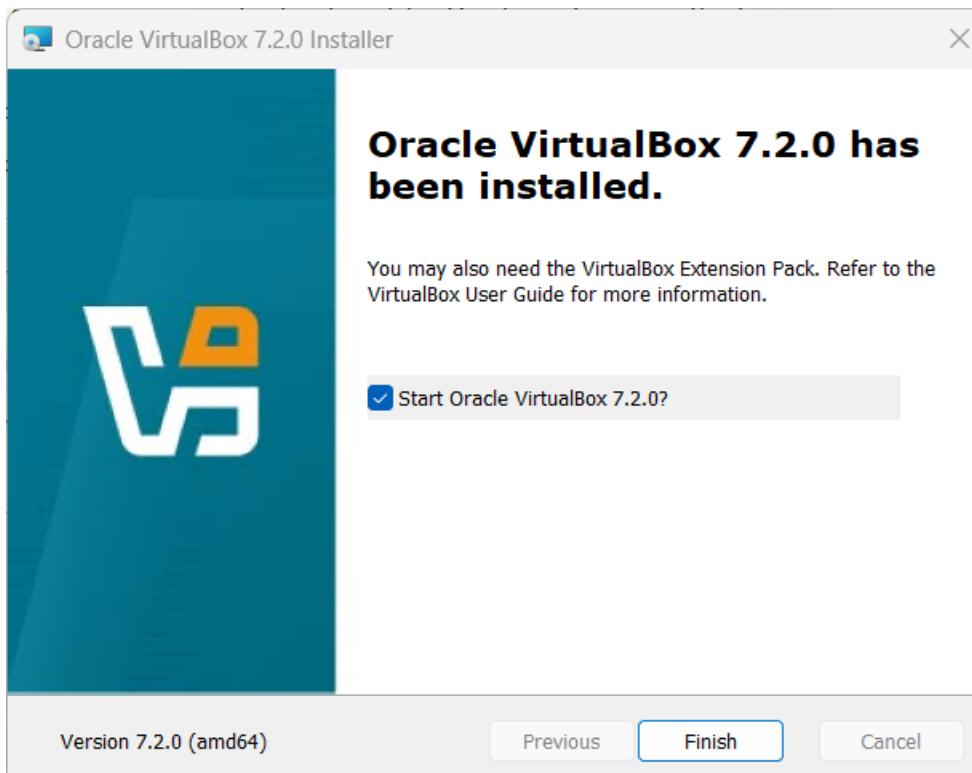


Accept the terms:

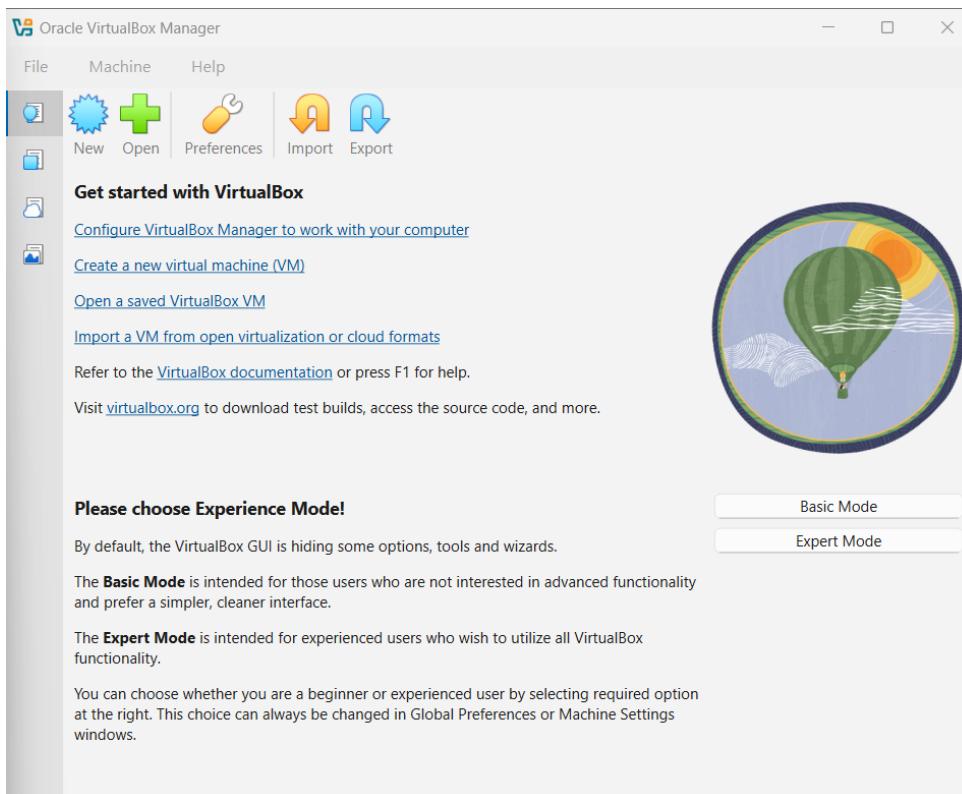


After following the necessary steps, install the application:



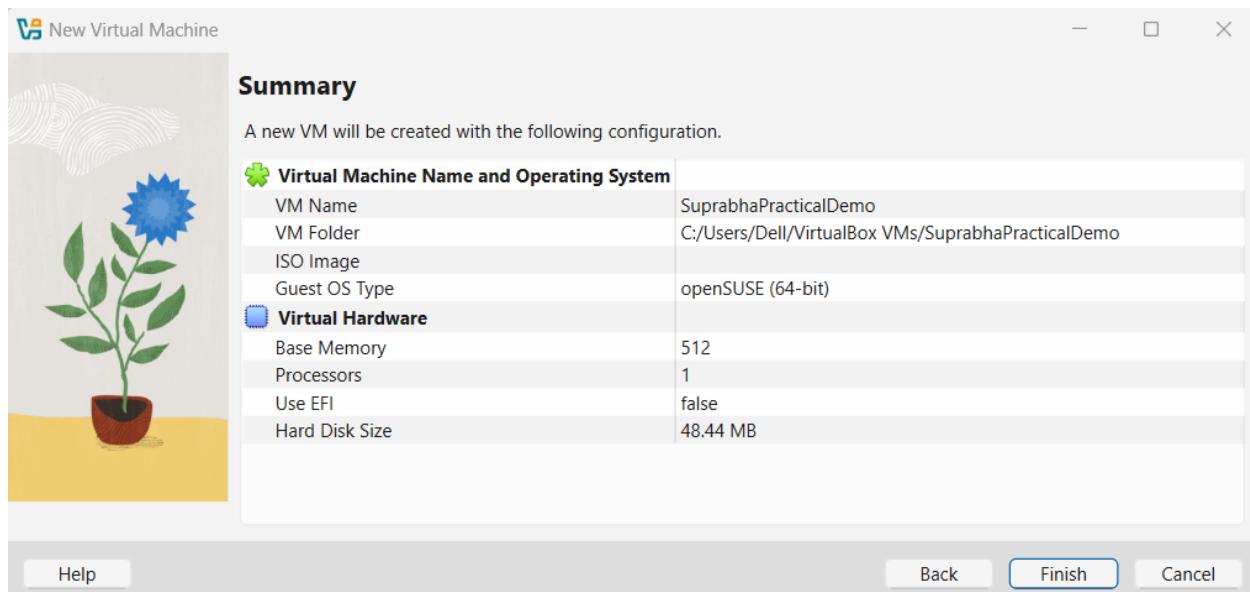


Launch the Application:

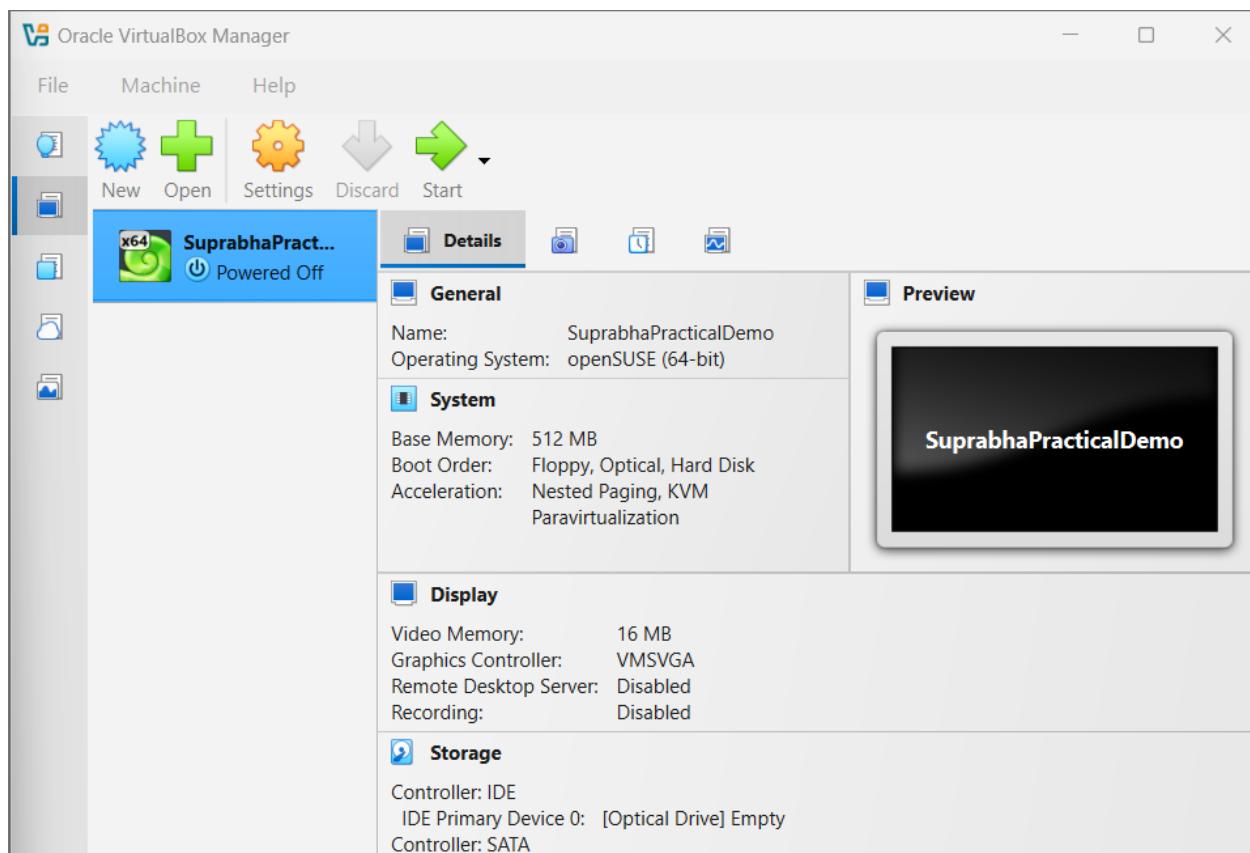


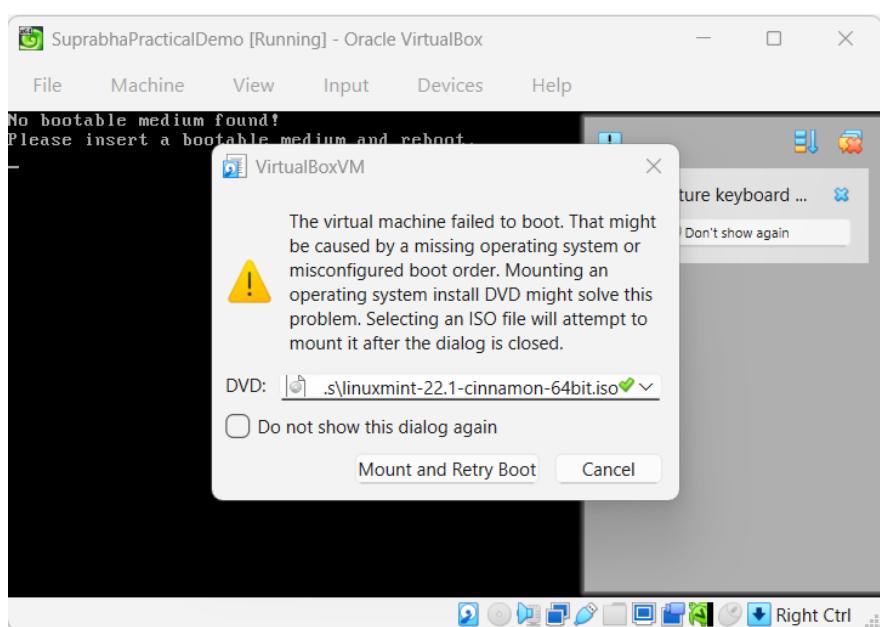
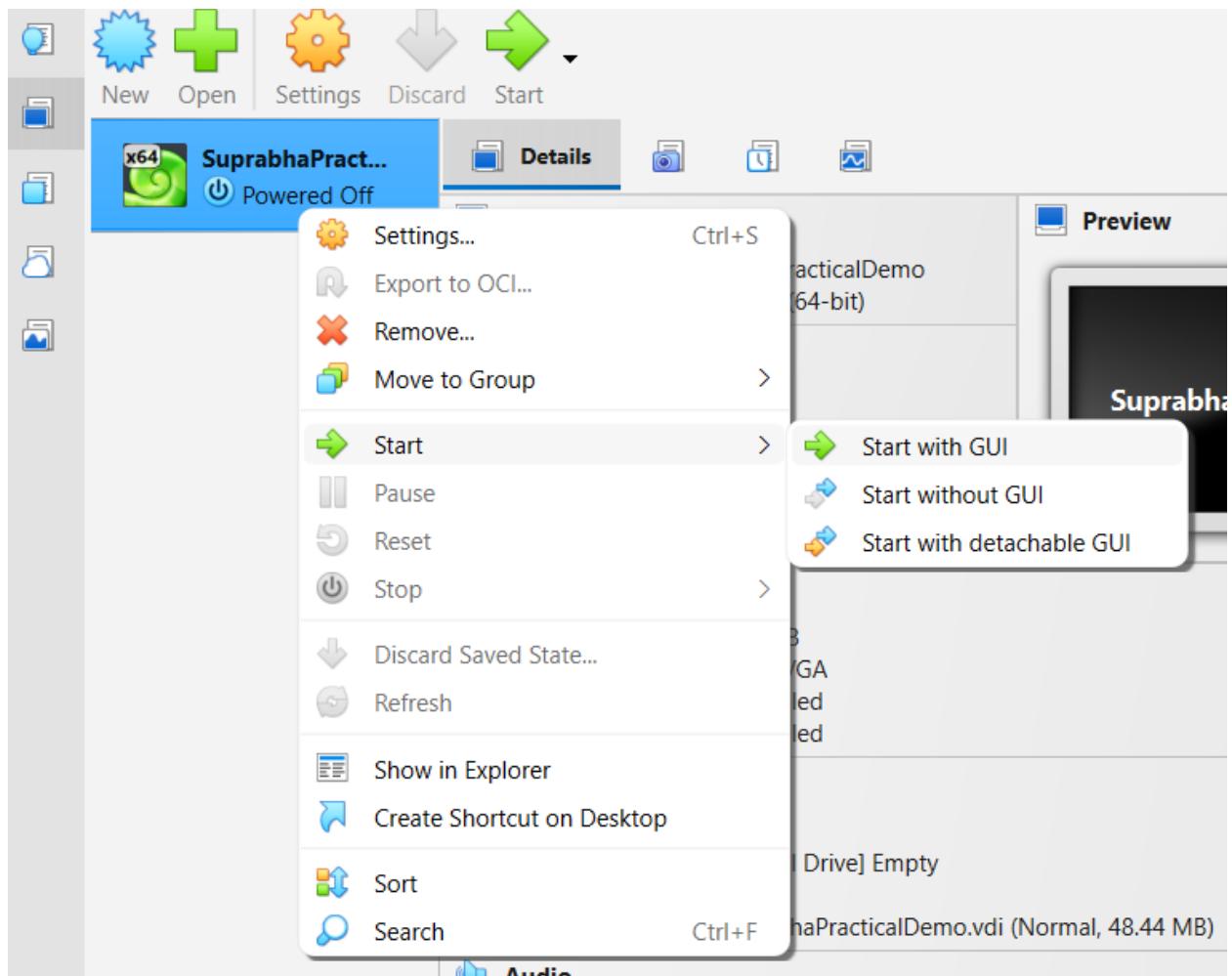
And then create a new virtual machine:

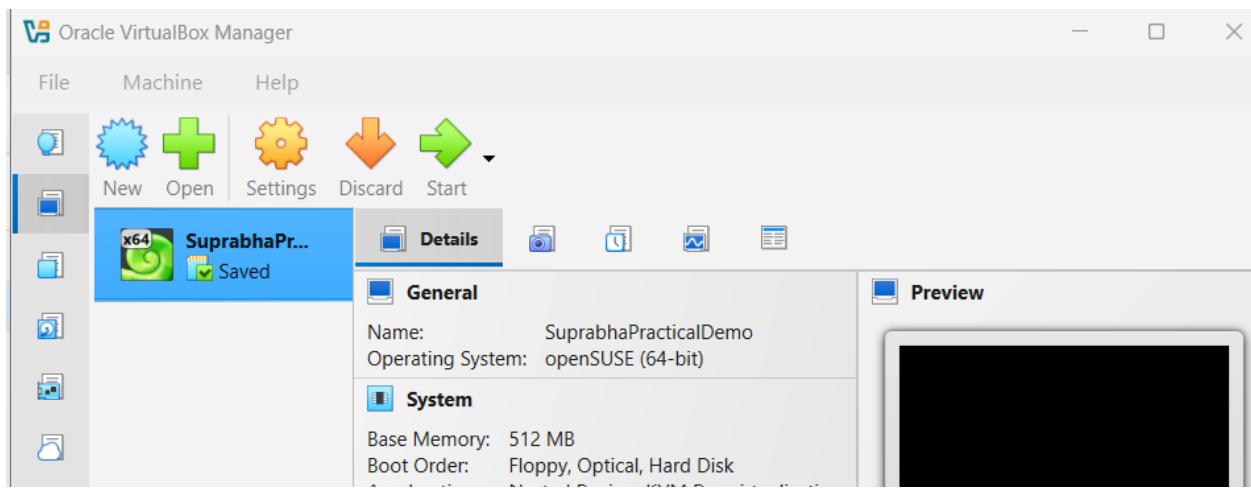
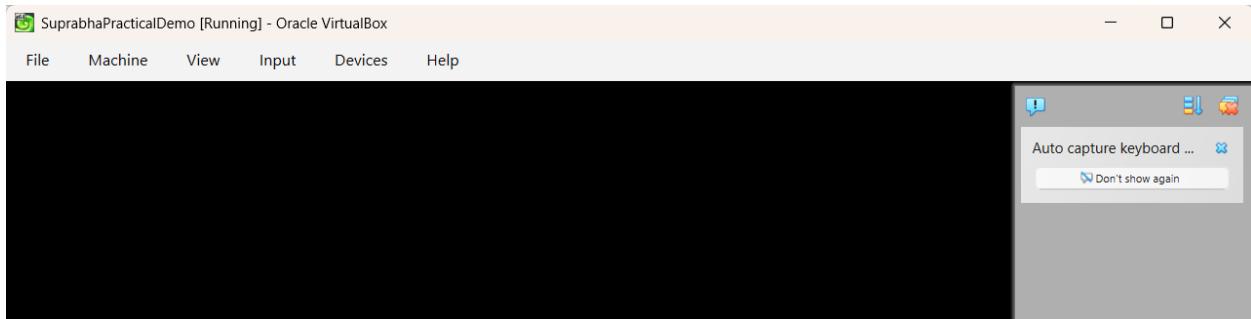
The screenshot shows the Oracle VirtualBox Manager interface. At the top, there's a menu bar with 'File', 'Machine' (which is selected), and 'Help'. Below the menu is a toolbar with icons for 'New...', 'Open...', 'Import', and 'Export'. A central window titled 'Get started with VirtualBox' contains the 'Virtual machine name and operating system' configuration screen. This screen includes fields for 'VM Name' (SuprabhaPracticalDemo), 'VM Folder' (C:\Users\...), 'ISO Image' (not selected), 'OS Edition' (Linux), 'OS Distribution' (openSUSE), and 'OS Version' (openSUSE (64-bit)). There's also a checkbox for 'Proceed with Unattended Installation' and a note that 'No ISO image is selected, the guest OS will need to be installed manually.' At the bottom of this window are 'Help', 'Back', 'Next', and 'Cancel' buttons. The second window shown is the 'Specify virtual hardware' screen, which allows configuring memory (512 MB), number of CPUs (1), and disk size (48.44 MB). It also has an 'Use EFI' checkbox. The bottom of this window has 'Help', 'Back', 'Next', and 'Cancel' buttons.



Start it up:

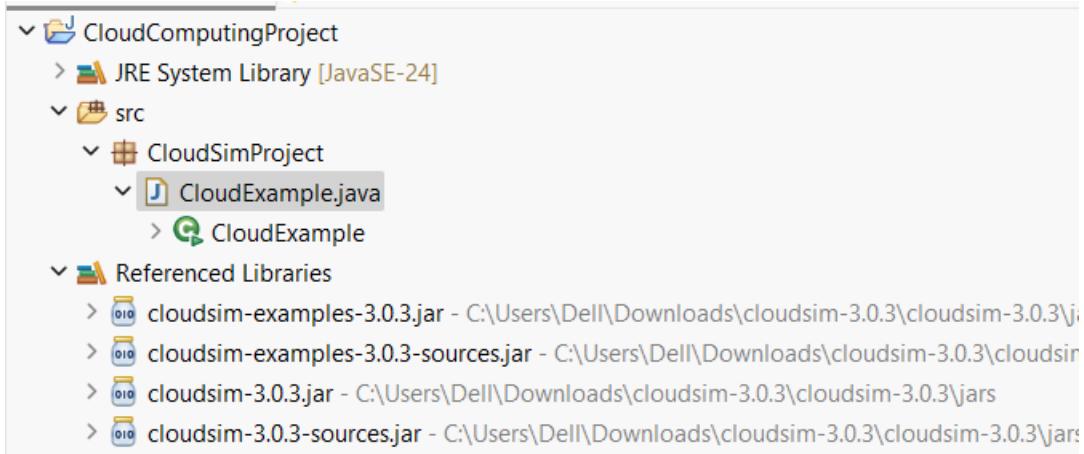






Lab-6 Simulate a cloud scenario using CloudSim

Import cloudsim and the necessary dependencies required ;



And execute the following code in **CloudExample.java**:

package CloudSimProject;

```
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.List;

import org.cloudbus.cloudsim.*;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.provisioners.*;

public class CloudExample {

    private static List<Cloudlet> cloudletList;
    private static List<Vm> vmlist;

    public static void main(String[] args) {
        Log.printLine("Starting Suprabha's CloudSimProject...");
        try {
            // ----- 1. INIT -----
            int numUsers = 1;
            Calendar calendar = Calendar.getInstance();
            boolean traceFlag = false;
            CloudSim.init(numUsers, calendar, traceFlag);

            // ----- 2. DATACENTERS -----
            createDatacenter("Datacenter_0");
        }
    }
}
```

```

createDatacenter("Datacenter_1");

// ----- 3. BROKER -----
DatacenterBroker broker = createBroker();
int brokerId = broker.getId();

// ----- 4. VMs -----
vmlist = new ArrayList<>();
int mips = 250;
long size = 10_000; // image size (MB)
int ram = 512; // VM memory (MB)
long bw = 1_000; // bandwidth
int pes = 1; // number of vCPUs
String vmm = "Xen";

vmlist.add(new Vm(0, brokerId, mips, pes, ram, bw, size, vmm,
                  new CloudletSchedulerTimeShared()));
vmlist.add(new Vm(1, brokerId, mips, pes, ram, bw, size, vmm,
                  new CloudletSchedulerTimeShared()));
broker.submitVmList(vmlist);

// ----- 5. CLOUDLETS -----
cloudletList = new ArrayList<>();
long length = 40_000;
long fileSize = 300;
long outputSize = 300;
UtilizationModel util = new UtilizationModelFull();

Cloudlet c1 = new Cloudlet(0, length, pes, fileSize, outputSize,
                         util, util, util);
c1.setUserId(brokerId);
Cloudlet c2 = new Cloudlet(1, length, pes, fileSize, outputSize,
                         util, util, util);
c2.setUserId(brokerId);

cloudletList.add(c1);
cloudletList.add(c2);
broker.submitCloudletList(cloudletList);

// pin each cloudlet to a different VM
broker.bindCloudletToVm(c1.getCloudletId(), vmlist.get(0).getId());
broker.bindCloudletToVm(c2.getCloudletId(), vmlist.get(1).getId());

// ----- 6. RUN SIM -----
CloudSim.startSimulation();
List<Cloudlet> newList = broker.getCloudletReceivedList();

```

```

CloudSim.stopSimulation();

printCloudletList(newList);
Log.printLine("Suprabha's CloudSimProject finished!");

} catch (Exception e) {
    e.printStackTrace();
    Log.printLine("The simulation has been terminated due to an unexpected error.");
}

/*
----- helper methods -----
*/

private static Datacenter createDatacenter(String name) {
    List<Host> hostList = new ArrayList<>();
    List<Pe> peList = new ArrayList<>();

    int mips = 1000;
    peList.add(new Pe(0, new PeProvisionerSimple(mips)));

    int hostId = 0;
    int ram = 2048; // MB
    long storage = 1_000_000; // MB
    int bw = 10_000;

    hostList.add(new Host(
        hostId,
        new RamProvisionerSimple(ram),
        new BwProvisionerSimple(bw),
        storage,
        peList,
        new VmSchedulerSpaceShared(peList)
    ));

    String arch = "x86";
    String os = "Linux";
    String vmm = "Xen";
    double timeZone = 10.0;
    double cost = 3.0;
    double costPerMem = 0.05;
    double costPerStorage = 0.001;
    double costPerBw = 0.0;

    DatacenterCharacteristics characteristics = new DatacenterCharacteristics(
        arch, os, vmm, hostList, timeZone,
        cost, costPerMem, costPerStorage, costPerBw);
}

```

```

try {
    return new Datacenter(name, characteristics,
        new VmAllocationPolicySimple(hostList),
        new LinkedList<Storage>(), 0);
} catch (Exception e) {
    e.printStackTrace();
    return null;
}
}

private static DatacenterBroker createBroker() {
    try {
        return new DatacenterBroker("Broker");
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

/* ----- the bit that was broken ----- */
private static void printCloudletList(List<Cloudlet> list) {
    DecimalFormat dft = new DecimalFormat("###.##");

    Log.println("\n===== OUTPUT =====");
    Log.println(String.format(
        "%-12s%-10s%-15s%-10s%-10s%-15s",
        "Cloudlet ID", "STATUS", "Datacenter ID", "VM ID",
        "Time", "Start Time", "Finish Time"));

    for (Cloudlet cloudlet : list) {
        if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS) {
            Log.println(String.format(
                "%-12d%-10s%-15d%-10d%-10s%-15s",
                cloudlet.getCloudletId(), "SUCCESS",
                cloudlet.getResourceId(), cloudlet.getVmId(),
                dft.format(cloudlet.getActualCPUTime()),
                dft.format(cloudlet.getExecStartTime()),
                dft.format(cloudlet.getFinishTime()))); // **only TWO ) before ;**
        }
    }
}

```

```
Problems Terminal Data Source Explorer Properties Console X Coverage TCP/IP Monitor
<terminated> CloudExample (2) [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (Aug 26, 2025, 12:58:38 PM – 12:58:39 PM e
Starting Suprabha's CloudSimProject...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #1 to Host #0 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Creation of VM #1 failed in Datacenter #2
0.1: Broker: Trying to Create VM #1 in Datacenter_1
0.2: Broker: VM #1 has been created in Datacenter #3, Host #0
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
160.2: Broker: Cloudlet 0 received
160.2: Broker: Cloudlet 1 received
160.2: Broker: All Cloudlets executed. Finishing...
160.2: Broker: Destroying VM #0
160.2: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID STATUS Datacenter ID VM ID Time Start Time Finish Time
0 SUCCESS 2 0 160 0.2 160.2
1 SUCCESS 3 1 160 0.2 160.2
Suprabha's CloudSimProject finished!
```

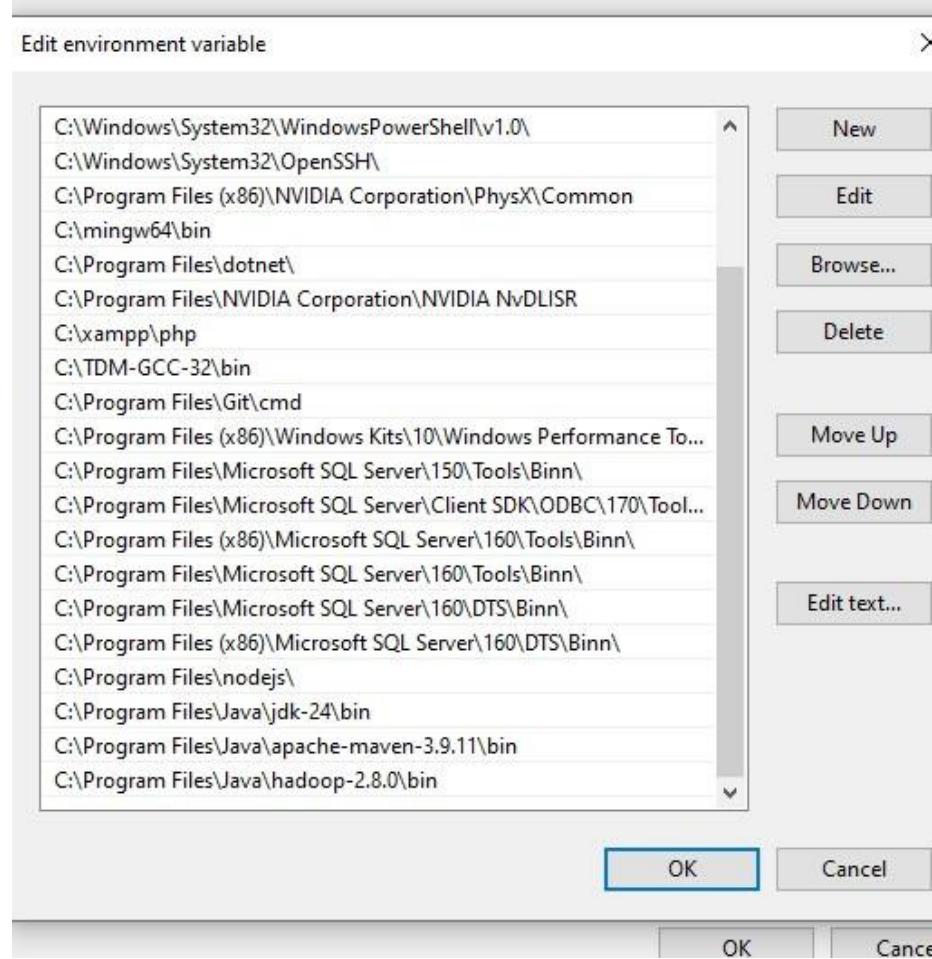
Lab-7 Installation and testing of Hadoop single node cluster on windows

Download Hadoop from <http://archive.apache.org/dist/hadoop/core//hadoop-2.8.0/hadoop-2.8.0.tar.gz> and then, install and setup JDK and Hadoop set up system variables for both:

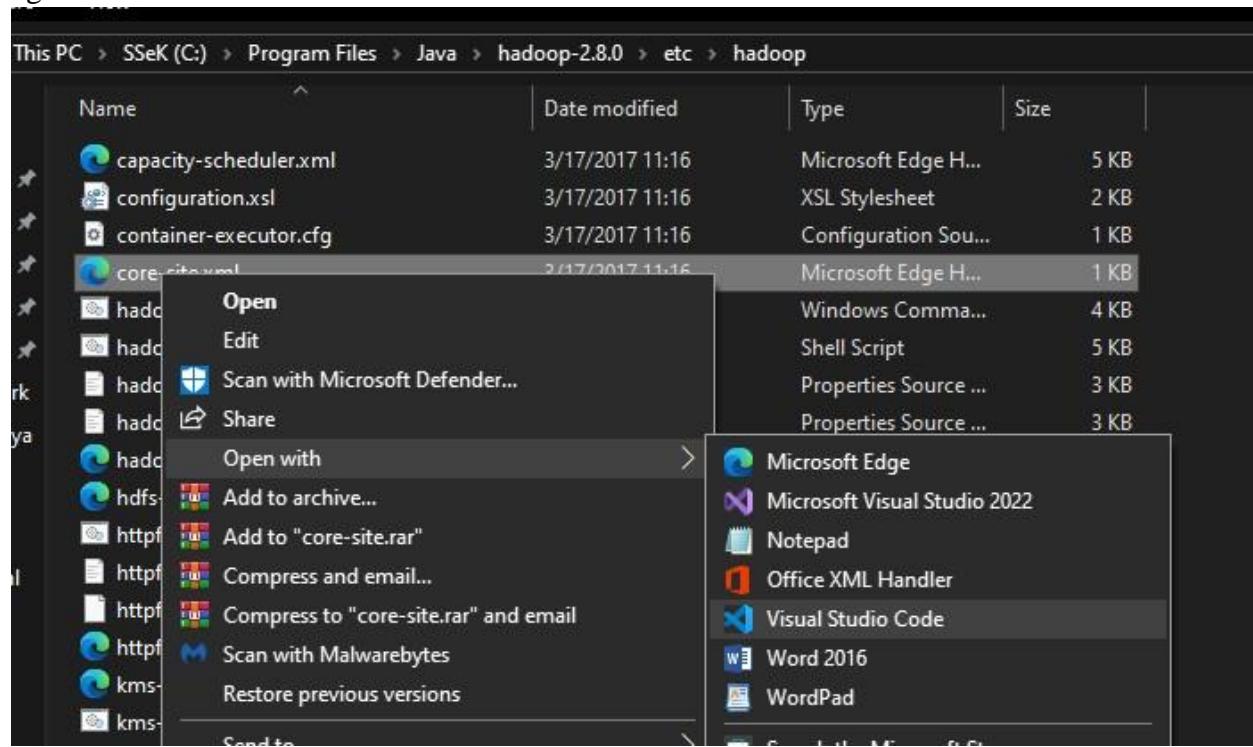
System variables	
Variable	Value
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
HADOOP_HOME	C:\Program Files\Java\hadoop-2.8.0
JAVA_HOME	C:\Program Files\Java\jdk-24\
NUMBER_OF_PROCESSORS	12
OS	Windows NT

New... Edit...

Then go to the system environment variables and add this to the path:



Go to Hadoop folder hadoop-2.8.0\etc\hadoop, edit core-site.xml with a rich text editor, eg Visual Studio Code



Modify

```
<configuration>
```

```
</configuration>
```

To

```
<configuration>
```

```
<property>
```

```
<name>fs.defaultFS</name>
```

```
<value>hdfs://localhost:9000</value>
```

```
</property>
```

```
</configuration>
```

Find mapred-site.xml.template and modify it to mapred-site.xml and add :

```
<property>
```

```
<name>mapreduce.framework.name</name>
```

```
<value>yarn</value>
```

```
</property>
```

To the configuration tags

Create the following highlighted folders

	Name	Date modified	Type	Size
	bin	7/19/2025 00:04	File folder	
	etc	7/19/2025 00:04	File folder	
	include	7/19/2025 00:04	File folder	
	lib	7/19/2025 00:04	File folder	
	libexec	7/19/2025 00:04	File folder	
	sbin	7/19/2025 00:04	File folder	
	share	7/19/2025 00:08	File folder	
Work				
shya	datanode	7/19/2025 00:20	File folder	
	namenode	7/19/2025 00:20	File folder	
	LICENSE.txt	3/17/2017 11:16	Text Document	97 KB
	NOTICE.txt	3/17/2017 11:16	Text Document	16 KB
	README.txt	3/17/2017 11:16	Text Document	2 KB

Edit Hadoop-2.8.0\etc\hadoop\hdfs-site.xml with

```
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>C:\hadoop-2.8.0\data\namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>C:\hadoop-2.8.0\data\datanode</value>
</property>
```

And Hadoop-2.8.0\etc\hadoop\yarn-site.xml with

```
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```

Edit the file Hadoop-2.8.0\etc/hadoop\hadoop-env.cmd and write @rem in front of “set JAVA_HOME=%JAVA_HOME%”. Write set JAVA_HOME={JDK directory} at the next row. This is C:\Program Files\Java\jdk-24 for me

Download Hadoop Configuration.zip from
<https://github.com/MuhammadBilalYar/HADOOP-INSTALLATION-ON-WINDOW10/blob/master/Hadoop%20Configuration.zip>

And replace the bin from Hadoop-2.8.0 with bin from configuration

Run cmd at this directory

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.6093]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\Java\hadoop-2.8.0\sbin>
```

Runn the following commands

```
C:\Java\hadoop-2.8.0\sbin>hdfs namenode -format
C:\Java\hadoop-2.8.0\sbin>start-dfs.cmd
C:\Java\hadoop-2.8.0\sbin>start-yarn.cmd
  starting yarn daemons
C:\Java\hadoop-2.8.0\sbin>
```

In browser go to localhost:8088



The image shows a screenshot of a Hadoop cluster metrics interface. At the top left is a yellow dog icon next to the word "hadoop". To the right, the text "All Applications" is partially visible. On the left, there's a sidebar with a tree icon and the text "Cluster Metrics". Below this are links for "About", "Nodes", "Node Labels", "Applications", and application states: "NEW", "NEW SAVING", "SUBMITTED", "ACCEPTED", and "RUNNING". The main area contains three tables: "Cluster Metrics", "Cluster Nodes Metrics", and "Scheduler Metrics".

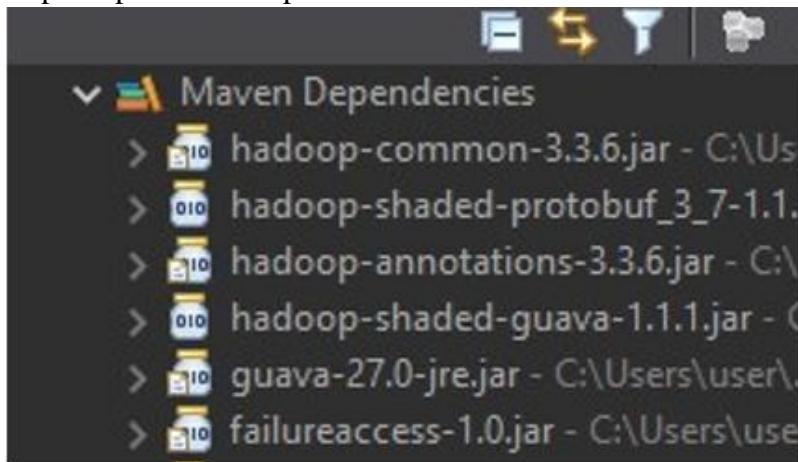
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running
0	0	0	0	0

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
0	0	0

Scheduler Metrics
[Table structure shown]

Lab-8 Mapreduce wordcount program using java

Import apache Hadoop 3.3.6 :



Create a word count reducer class with following code

package mapred;

```
import java.io.IOException;
import
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;

import com.ctc.wstx.util.WordSet;

public class mapred {
    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 class SumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

@Override

protected void reduce(Text key, Iterable<IntWritable> values, Context
context) throws IOException, InterruptedException {

int sum = 0;

for (IntWritable val : values) {

    sum += val.get();

}

context.write(key, new IntWritable(sum));

}

}

public static void main(String[] args) throws Exception {

Configuration conf = new Configuration();
Job job = Job.getInstance(conf, "word
count");
job.setJarByClass(WordSet.class);
job.setMapperClass(TokenizerMapper.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);
}

```

```
}
```

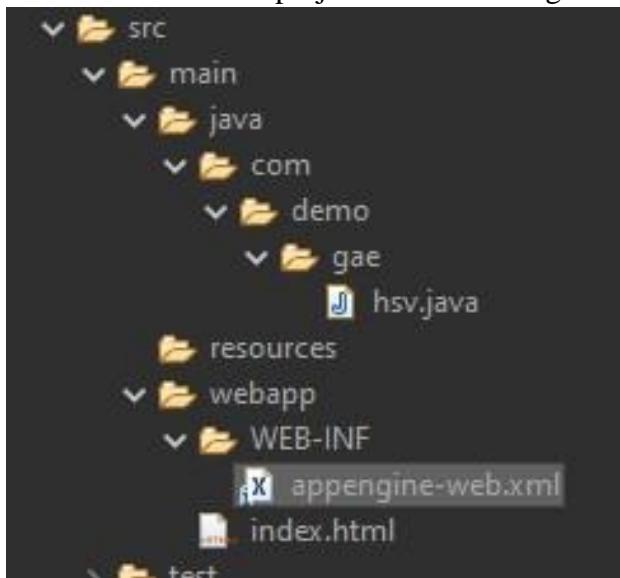
```
}
```



A screenshot of the Eclipse IDE interface, specifically focusing on the Terminal view. The terminal window shows the output of a Java application named 'mapred'. The output text is:
<terminated> mapred [Java Application]
Usage: wordcount 445 224

Lab-9 Develop and application for Google App Engine

Create a new maven project with following structure



With each containing the following code Hsv.java

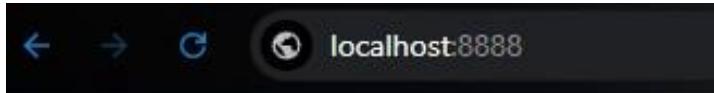
```
package com.demo.gae;  
  
import java.io.IOException;  
  
import jakarta.servlet.http.*;  
  
@SuppressWarnings("serial")  
public class hsv extends  
HttpServlet {  
  
    @Override public void doGet(HttpServletRequest req,  
    HttpServletResponse resp) throws IOException {  
        resp.setContentType("text/plain");  
        resp.getWriter().println("Hello, world from Java 17!");  
    }  
}
```

```
Appengine-web.xml  
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">  
    <runtime>java17</runtime>  
    <threadsafe>true</threadsafe>  
</appengine-web-app>
```

```
Index.html
<!DOCTYPE html>

<html>
  <head>
    <title>Hello GAE</title>
  </head>
  <body>
    <h1>Hello from Java 17 on App Engine!</h1>
    <p><a href="/hello">Go to Servlet</a></p>
  </body>
</html>
```

Run the application by
mvn appengine:run
gcloud app deploy



Hello App Engine!

Available Servlets:
[GoogleAppEngine](#)

