Practical Journal

Of

ADVANCED CLOUD COMPUTING

Submitted by

25FMSCIT020-Kshitij Singh

for the award of the Degree of

MASTERS OF SCIENCE (INFORMATION TECHNOLOGY)

PART-1

DEPARTMENT OF INFORMATION TECHNOLOGY

KISHINCHAND CHELLARAM COLLEGE

(Affiliated to University of HSNC)

MUMBAI, 400020

2025-26

KISHINCHAND CHELLARAM COLLEGE

CHURCHGATE, MUMBAI – 400 020.

DEPARTMENT OF INFORMATION TECHNOLOGY

M.SC.I.T PART- I

CERTIFICATE

This is to certify that the Practical conducted by, Mr. **KSHITIJ SINGH** for M.Sc. (IT) Part-I Semester- I, Seat No: **25FMSCIT020** at Kishinchand Chellaram College in partial fulfilment for the MASTERS OF SCIENCE (INFORMATION TECHNOLOGY). Degree Examination for semester I has been periodically examined and signed, and the course of term work has been satisfactorily carried out for the year 2025 - 2026. This Practical journal had not been submitted for any other examination and does not form part of any other course undergone by the candidate.

Signature	**Signature**	**Signature**
Lecturer-In-Charge	**External Examiner**	**Course Coordination**
Guided By	**Examined By**	**Certified BY**

College Stamp

SUBJECT CODE- BIT511B

ADVANCED CLOUD COMPUTING

Practical no.	Index	Remarks
1	Write a program for implementing Client Server communication model using TCP.	
2	A client server based program using UDP to find if the number entered is even or odd number.	
3	Write a program to show the object communication using RMI to display current date and time.	
4	A multicast socket example.	
5	To configure and host multiple websites on a single Apache server inside a Virtual Machine using Virtual Hosts, and access them from a host machine by mapping domain names to the VM's IP address.	
6	SSH Key Login.	
7	Network & Compliance Hardening for a Linux Server.	
8	Show the implementation of web services.	

Practical no-1

Aim: Write a program for implementing Client Server communication model using TCP.

Code:

1. TcpServerPrime.java

```
import java.net.*;
import java.io.*;

class TcpServerPrime
{
	public static void main(String args[])
	{
		try
		{
			ServerSocket ss=new ServerSocket(8001);
			System.out.println("Server Started. ..... ");
			Socket s=ss.accept();

			DataInputStream in=new
			DataInputStream(s.getInputStream());
			int x=in.readInt();

			DataOutputStream otc=new
			DataOutputStream(s.getOutputStream());
			int y=x/2;

			if(x==1 || x==2 || x==3)
			{
				otc.writeUTF(x + "is Prime");
				System.exit(0);
			}
			for(int i=2;i<=y;i++)
			{
				if(x%i !=0)
				{
					otc.writeUTF(x + "is Prime");
				}
				else
				{
					otc.writeUTF(x + "is not Prime");
```

```
                        }
                    }
                }
                catch(Exception e)
                {
                    System.out.println(e.toString());
                }
        }
}
```

2. TcpClient.java

```
import java.net.*;
import java.io.*;

class tcpClientPrime
{
        public static void main(String args[])
        {
                try
                {
                        Socket cs=new Socket("LocalHost",8001);
                        BufferedReader infu=new BufferedReader(new
                        InputStreamReader(System.in));
                        System.out.println("Enter a number: ");

                        int a=Integer.parseInt(infu.readLine());
                        DataOutputStream out=new
                        DataOutputStream(cs.getOutputStream());
                        out.writeInt(a);

                        DataInputStream in=new
DataInputStream(cs.getInputStream());
                        System.out.println(in.readUTF());
                        cs.close();
                }
                catch(Exception e)
                {
                        System.out.println(e.toString());
                }
        }
}
```

Output:

Server:

```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Kshitij Singh\Desktop\java>javac TcpServerPrime.java

C:\Users\Kshitij Singh\Desktop\java>java TcpServerPrime
Server Started.........

C:\Users\Kshitij Singh\Desktop\java>
```

Client:

```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Kshitij Singh\Desktop\java>javac TcpClientPrime.java

C:\Users\Kshitij Singh\Desktop\java>java TcpClientPrime
Enter a number:
10
10is not Prime

C:\Users\Kshitij Singh\Desktop\java>
```

Practical no-2

Aim: A client server based program using UDP to find if the number entered is even or odd number.

Code:

1. udpServerEO.java

```
import java.io.*;
import java.net.*;
public class udpServerEO
{
	public static void main(String args[])
	{
		try
		{
			DatagramSocket ds = new DatagramSocket(2000);
			byte b[] = new byte[1024];
			DatagramPacket dp = new DatagramPacket(b,b.length);
			ds.receive(dp);
			String str = new
			String(dp.getData(),0,dp.getLength());
			System.out.println(str);
			int a= Integer.parseInt(str);
			String s= new String();
			if (a%2 == 0)
				s = "Number is even";
			else
				s = "Number is odd";
			byte b1[] = new byte[1024];
			b1 = s.getBytes();
			DatagramPacket dp1 = new

	DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000);
			ds.send(dp1);
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
	}
}
```

2. udpClientEO.java

```
import java.io.*;
import java.net.*;
public class udpClientEO
{
	public static void main(String args[])
	{
		try
		{
			DatagramSocket ds = new DatagramSocket(1000);
			BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
			System.out.println("Enter a number : ");
			String num = br.readLine();
			byte b[] = new byte[1024];
			b=num.getBytes();
			DatagramPacket dp = new
DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);
			ds.send(dp);
			byte b1[] = new byte[1024];
			DatagramPacket dp1 = new DatagramPacket(b1,b1.length);
			ds.receive(dp1);
			String str = new
			String(dp1.getData(),0,dp1.getLength());
			System.out.println(str);
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
	}
}
```

Output:

Server:

```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Kshitij Singh\Desktop\java>javac udpServerEO.java

C:\Users\Kshitij Singh\Desktop\java>java udpServerEO
36

C:\Users\Kshitij Singh\Desktop\java>
```

Client:

```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Kshitij Singh\Desktop\java>javac udpClientEO.java

C:\Users\Kshitij Singh\Desktop\java>java udpClientEO
Enter a number :
36
Number is even

C:\Users\Kshitij Singh\Desktop\java>
```

Practical no-3

Aim: Write a program to show the object communication using RMI to display current date and time.

Code:

1. **InterDate.java**

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface InterDate extends Remote {
   String display() throws RemoteException;
}
```

2. **ServerDate.java**

```
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.util.Date;

public class ServerDate extends UnicastRemoteObject implements InterDate {

   public ServerDate() throws RemoteException {
      super();
   }

   public String display() throws RemoteException {
      Date d = new Date();
      return d.toString();
   }

   public static void main(String[] args) throws Exception {
      ServerDate s1 = new ServerDate();
      Naming.rebind("DS", s1);
      System.out.println("Object registered...");
   }
}
```

3. ClientDate.java

```
import java.rmi.*;
import java.io.*;
public class ClientDate
{
	public static void main(String args[])
	throws Exception
	{
		String s1;
		InterDate h1 = (InterDate)Naming.lookup("DS");
		s1 = h1.display();
		System.out.println(s1);
	}
}
```

Output:

Server:

```
C:\Users\Kshitij Singh\Desktop\java>javac ServerDate.java

C:\Users\Kshitij Singh\Desktop\java>java ServerDate
Object registered...
```

Client:

```
C:\Users\Kshitij Singh\Desktop\java>javac ClientDate.java

C:\Users\Kshitij Singh\Desktop\java>java ClientDate
Sat Nov 29 20:13:40 IST 2025

C:\Users\Kshitij Singh\Desktop\java>
```

Practivcal no-4

Aim: A multicast socket example.

Code:

1.BroadcastServer.java

```
import java.net.*;
import java.io.*;
import java.util.*;

public class BroadcastServer
{
	public static final int PORT=1234;
	public static void main(String args[])
	throws Exception {
		MulticastSocket  socket;
		DatagramPacket packet;
		InetAddress address;

		address=InetAddress.getByName("239.1.2.3");
		socket=new MulticastSocket();

		socket.joinGroup(address);

		byte[] data=null;
		for(;;)

		{
			Thread.sleep(10000);
			System.out.println("Sending");
			String str=("This is neha calling...");
			data=str.getBytes();

			packet=new
DatagramPacket(data,str.length(),address,PORT);

			socket.send(packet);
		}
	}
}
```

2.BroadcastClient.java

```
import java.net.*;
import java.io.*;

public class BroadcastClient
{
	public static final int PORT=1234;
	public static void main(String args[])
	throws Exception {

			MulticastSocket  socket;
			DatagramPacket packet;
			InetAddress address;

			address=InetAddress.getByName("239.1.2.3");
			socket=new MulticastSocket(PORT);

			socket.joinGroup(address);
			byte[] data=new byte[100];

			packet=new DatagramPacket(data,data.length);
			for(;;)
			{

				socket.receive(packet);

				String str=new String(packet.getData());
				System.out.println("Message received from" +
packet.getAddress() +"Message is:"+str);
			}
	}
}
```

Output:

Server:

```
C:\Users\Kshitij Singh\Desktop\java>javac BroadcastServer.java
Note: BroadcastServer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\Kshitij Singh\Desktop\java>java BroadcastServer
Sending
Sending
Sending
Sending
Sending
Sending
Sending
```

Client:

```
C:\Users\Kshitij Singh\Desktop\java>javac BroadcastClient.java
Note: BroadcastClient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\Kshitij Singh\Desktop\java>java BroadcastClient
Message received from/192.168.56.1Message is:This is Kshitij calling........
Message received from/192.168.56.1Message is:This is Kshitij calling........
Message received from/192.168.56.1Message is:This is Kshitij calling........
Message received from/192.168.56.1Message is:This is Kshitij calling........
Message received from/192.168.56.1Message is:This is Kshitij calling........
Message received from/192.168.56.1Message is:This is Kshitij calling........
Message received from/192.168.56.1Message is:This is Kshitij calling........
```

Practical no-5

Aim: To configure and host multiple websites on a single Apache server inside a Virtual Machine using Virtual Hosts, and access them from a host machine by mapping domain names to the VM's IP address.

Apache supports Virtual Hosting, which allows a single web server to serve multiple domain names using different configuration files. Each website is stored in a different directory and mapped using a unique domain like example.com, test.com. This simulates shared hosting on production servers.

Step 1: Install Apache (If Not Installed)

sudo apt update

sudo apt install apache2 –y

```
ubuntuu@ubuntuu:~$ sudo apt update
[sudo] password for ubuntuu:
Hit:1 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 126 kB in 3s (46.0 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
102 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntuu@ubuntuu:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.8).
0 upgraded, 0 newly installed, 0 to remove and 102 not upgraded.
ubuntuu@ubuntuu:~$
```

Start and enable Apache:

sudo systemctl start apache2

sudo systemctl enable apache2

```
[sudo] password for ubuntuu:
Hit:1 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 126 kB in 3s (46.0 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
102 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntuu@ubuntuu:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.8).
0 upgraded, 0 newly installed, 0 to remove and 102 not upgraded.
ubuntuu@ubuntuu:~$ sudo systemctl start apache2
ubuntuu@ubuntuu:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
ubuntuu@ubuntuu:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-11-29 22:49:54 IST; 7min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1335 (apache2)
      Tasks: 55 (limit: 9205)
     Memory: 7.8M (peak: 8.3M)
        CPU: 449ms
     CGroup: /system.slice/apache2.service
             ├─1335 /usr/sbin/apache2 -k start
             ├─1338 /usr/sbin/apache2 -k start
             └─1339 /usr/sbin/apache2 -k start
```

Step 2: Create Website Folders and HTML Pages

sudo mkdir -p /var/www/example.com
sudo mkdir -p /var/www/test.com

Add simple HTML files:

echo "<h1>Hello from example.com</h1>" | sudo tee /var/www/example.com/index.html
echo "<h1>Hello from test.com</h1>" | sudo tee /var/www/test.com/index.html

Step 3: Set Permissions

sudo chown -R www-data:www-data /var/www/example.com
sudo chown -R www-data:www-data /var/www/test.com
sudo chmod -R 755 /var/www

chown = **change ownership**
-R = apply changes **recursively** (to all files and folders inside)
www-data:www-data = set the **owner and group** to Apache's default user (www-data)
/var/www/example.com = the folder you are changing

```
ubuntuu@ubuntuu:~$ sudo mkdir -p /var/www/example.com
ubuntuu@ubuntuu:~$ sudo mkdir -p /var/www/test.com
ubuntuu@ubuntuu:~$ echo "<h1> Hello from example.com</h1>'|sudo tee /var/www/example.com/index.html
>
> <h1>hello from example.com</h1>
> ^C
ubuntuu@ubuntuu:~$ echo "<h1> Hello from test.com</h1>'|sudo tee /var/www/test.com/index.html

<h1>hello from test.com</h1>
> ^C
ubuntuu@ubuntuu:~$ sudo shown -R www-data:www-data /var/www/example.com
sudo: shown: command not found
ubuntuu@ubuntuu:~$ sudo chown -R www-data:www-data /var/www/example.com
ubuntuu@ubuntuu:~$ sudo chown -R www-data:www-data /var/www/test.com
ubuntuu@ubuntuu:~$ sudo chown -R 755 /var/www
ubuntuu@ubuntuu:~$
```

Step 4: Create Apache Virtual Host Config Files (ctrl + O SAVE ,Enter,Ctrl +X/Z)

1. example.com Virtual Host

sudo nano /etc/apache2/sites-available/example.com.conf

```
<VirtualHost *:80>
    ServerName example.com
    DocumentRoot /var/www/example.com
    <Directory /var/www/example.com>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/example.com-error.log
    CustomLog ${APACHE_LOG_DIR}/example.com-access.log combined
</VirtualHost>
```

2. test.com Virtual Host

sudo nano /etc/apache2/sites-available/test.com.conf

```
<VirtualHost *:80>
    ServerName test.com
    DocumentRoot /var/www/test.com
    <Directory /var/www/test.com>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/test.com-error.log
    CustomLog ${APACHE_LOG_DIR}/test.com-access.log combined
</VirtualHost>
```

- **Step 5: Enable the Sites and Reload Apache**

sudo a2ensite example.com.conf
sudo a2ensite test.com.conf
sudo systemctl reload apache2
Check config: sudo apache2ctl configtest
Output : Syntax OK

```
ubuntuu@ubuntuu:~$ sudo shown -R www-data:www-data /var/www/example.com
sudo: shown: command not found
ubuntuu@ubuntuu:~$ sudo chown -R www-data:www-data /var/www/example.com
ubuntuu@ubuntuu:~$ sudo chown -R www-data:www-data /var/www/test.com
ubuntuu@ubuntuu:~$ sudo chown -R 755 /var/www
ubuntuu@ubuntuu:~$ sudo nano /etc/apache2/sites-available/example.com.conf
ubuntuu@ubuntuu:~$ sudo nano /etc/apache2/sites-available/test.com.conf
ubuntuu@ubuntuu:~$ sudo a2ensite example.com.conf
Enabling site example.com.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntuu@ubuntuu:~$ sudo a2ensite test.com.conf
Enabling site test.com.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntuu@ubuntuu:~$ sudo systemctl reload apache2
ubuntuu@ubuntuu:~$
```

```
ubuntuu@ubuntuu:~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using fd17:625c:f037:2:a00:27ff:fecd:310c. Set the 'ServerName' directive globally to suppress
this message
Syntax OK
ubuntuu@ubuntuu:~$
```

🔹 Step 6: Get Your VM's IP Address

ip a

```
ubuntuu@ubuntuu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:cd:31:0c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 82835sec preferred_lft 82835sec
    inet6 fd17:625c:f037:2:65dc:b92c:b941:204d/64 scope global temporary dynamic
       valid_lft 86221sec preferred_lft 14221sec
    inet6 fd17:625c:f037:2:a00:27ff:fecd:310c/64 scope global dynamic mngtmpaddr
       valid_lft 86221sec preferred_lft 14221sec
    inet6 fe80::a00:27ff:fecd:310c/64 scope link
       valid_lft forever preferred_lft forever
ubuntuu@ubuntuu:~$
```

Step 7: Map Domains in Host OS (hosts file)

On Windows:

1. Open Notepad **as Administrator**
2. Open file: C:\Windows\System32\drivers\etc\hosts
3. Add at the bottom:

192.168.68.128 example.com

192. 168.68.128 test.com

4. Save and exit.

On Linux/macOS:

sudo nano /etc/hosts

Add:

192.168.56.101 example.com

192.168.56.101 test.com

Step 8: Test in Browser (from Host OS)

- Open your browser
- Visit: http://example.com

Output: Hello from example.com

- Visit: http://test.com

Output: Hello from test.com

Practical no-6

Aim: SSH Key Login.

On a cloud server (like AWS EC2), if **anyone knows the password**, they can login and **hack the server**. So we do 3 things:

1. **Create a new admin user** (instead of using root)
2. **Use SSH Key Login** (so no one can hack using passwords)
3. **Disable Root Login** (so no one can target root)

This is the **basic cloud security rule**.

Step 1: Create a new admin user
Run this command:
sudo adduser adminuser
What happens?

- Computer asks: Enter password → **Create any password**
- Then it asks name, phone etc → **Just press Enter for all**

Now give this user permission to run admin commands:
sudo usermod -aG sudo adminuser
Meaning:
We are making adminuser part of the **sudo group**, which means this user can run commands like an administrator.
Result:
We now have a **safe admin user** who will manage the server.
We will **stop using the root user**.

```
ubuntuu@ubuntuu:~$ sudo adduser adminuser
[sudo] password for ubuntuu:
Sorry, try again.
[sudo] password for ubuntuu:
info: Adding user `adminuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `adminuser' (1001) ...
info: Adding new user `adminuser' (1001) with group `adminuser (1001)' ...
info: Creating home directory `/home/adminuser' ...
info: Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for adminuser
Enter the new value, or press ENTER for the default
        Full Name []: admin
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `adminuser' to supplemental / extra groups `users' ...
info: Adding user `adminuser' to group `users' ...
ubuntuu@ubuntuu:~$ sudo usermod -aG sudo adminuser
ubuntuu@ubuntuu:~$
```

If You Are On Windows

Step 1: Download the Tool (for Windows)
MobaXterm
This is the tool that makes SSH easy (no ssh-copy-id needed).
Download Link: https://mobaxterm.mobatek.net/download.html
Click **"MobaXterm Home Edition → Installer Edition"**
Then install it normally.

This is on linux:

Step 1 — Make Sure SSH Server Is Installed
Inside your **VM terminal** run:
sudo apt update
sudo apt install openssh-server -y

```
ubuntuu@ubuntuu:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:6 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [71.5 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:8 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu noble-updates/main Icons (64x64) [51.0 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:12 http://in.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:14 http://in.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7,140 B]
Get:15 http://in.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
Get:16 http://in.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [29.2 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [17.6 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.0 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Fetched 1,142 kB in 19s (59.7 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
102 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntuu@ubuntuu:~$ sudo apt install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-client openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 4 newly installed, 0 to remove and 101 not upgraded.
```

Now start + enable SSH:
sudo systemctl enable ssh
sudo systemctl start ssh
Check status:
sudo systemctl status ssh
If it says **active (running)**

```
ubuntuu@ubuntuu:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.
ubuntuu@ubuntuu:~$ sudo systemctl start ssh
ubuntuu@ubuntuu:~$ sudo systemctl status ssh
Warning: The unit file, source configuration file or drop-ins of ssh.service changed on disk. Run 'systemctl daemon-reload' to reload units.
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-11-30 09:03:32 IST; 46s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 5627 (sshd)
      Tasks: 1 (limit: 9205)
     Memory: 1.2M (peak: 1.6M)
        CPU: 121ms
     CGroup: /system.slice/ssh.service
             └─5627 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 30 09:03:31 ubuntuu systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 30 09:03:32 ubuntuu sshd[5627]: Server listening on 0.0.0.0 port 22.
Nov 30 09:03:32 ubuntuu sshd[5627]: Server listening on :: port 22.
Nov 30 09:03:32 ubuntuu systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
ubuntuu@ubuntuu:~$
```

Step 3 — Allow SSH in the Firewall (If Needed)

Inside VM:

sudo ufw allow OpenSSH
sudo ufw enable
sudo ufw status

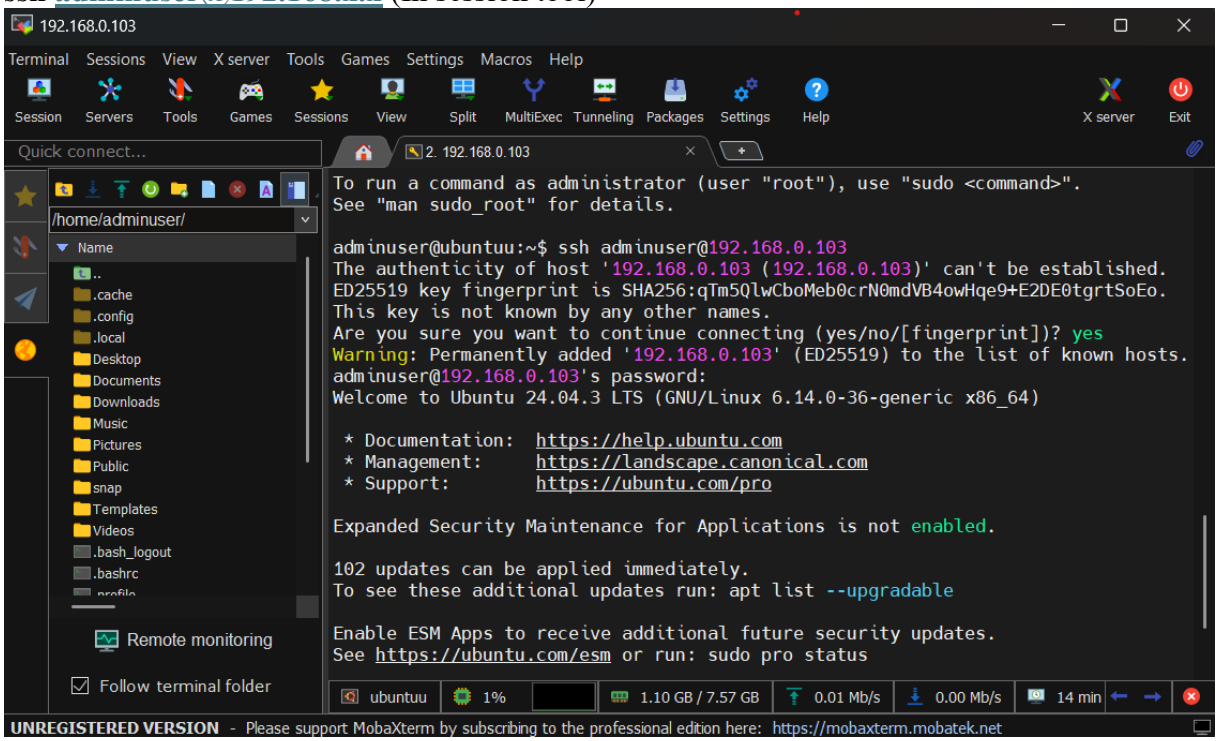
```
ubuntuu@ubuntuu:~$ sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 101 not upgraded.
ubuntuu@ubuntuu:~$ sudo ufw allow OpenSSh
Rules updated
Rules updated (v6)
ubuntuu@ubuntuu:~$ sudo ufw allow OpenSSH
Skipping adding existing rule
Skipping adding existing rule (v6)
ubuntuu@ubuntuu:~$ sudo ufw enable
Firewall is active and enabled on system startup
ubuntuu@ubuntuu:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)

ubuntuu@ubuntuu:~$
```

```
ubuntuu@ubuntuu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:cd:31:0c brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.103/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3
       valid_lft 6972sec preferred_lft 6972sec
    inet6 fe80::a00:27ff:fecd:310c/64 scope link
       valid_lft forever preferred_lft forever
ubuntuu@ubuntuu:~$
```

Test SSH Again from Host (MobaXterm / CMD)
ssh adminuser@192.168.x.x (In session tool)



Step 4 — Copy Your Public Key to the Server
From your **host machine** (MobaXterm terminal):
ssh-copy-id adminuser@<server-ip>
Example:
ssh-copy-id adminuser@192.168.1.45
You will be asked for **adminuser password (the one you created)**.

```
adminuser@ubuntuu:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/adminuser/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/adminuser/.ssh/id_rsa
Your public key has been saved in /home/adminuser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:7CdL+873aHbvNcBlW/huuMmHVJ2Lh0jsJqckPKYSqiY adminuser@ubuntuu
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|               . |
|          .   .o+|
|       .   o. oo=|
|      . S o .oooo|
|   .   * o = oo= |
|  . . o B *  .oo+|
|Eo . . . B  +oo++|
|=    .   oo++.o=+o|
+----[SHA256]-----+
adminuser@ubuntuu:~$ ssh-copy-id adminuser@192.168.0.103
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/adminuser/.
ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
adminuser@192.168.0.103's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'adminuser@192.168.0.103'"
and check to make sure that only the key(s) you wanted were added.

adminuser@ubuntuu:~$
```

After success, test login:
ssh adminuser@192.168.1.45
If it logs in **without asking password → success** 🎯

```
adminuser@ubuntuu:~$ ssh adminuser@192.168.0.103
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

102 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Nov 30 10:00:34 2025 from 192.168.0.103
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

adminuser@ubuntuu:~$
```

Step 6: Disable Root Login & Password Login
Inside VM, open the SSH config file:
sudo nano /etc/ssh/sshd_config
Change (or add) the following lines:
PermitRootLogin no
PasswordAuthentication no
Save & exit:
CTRL + O → Enter → CTRL + X
Restart SSH:
sudo systemctl restart ssh

```
ubuntuu@ubuntuu:~$ sudo nano /etc/ssh/sshd_config
[sudo] password for ubuntuu:
ubuntuu@ubuntuu:~$ sudo systemctl restart ssh
ubuntuu@ubuntuu:~$
```

Practical no-7

Aim: Network & Compliance Hardening for a Linux Server.

To secure a Linux server by:

- Allowing only required network ports,
- Monitoring if system files are changed,
- Preventing any user from overusing system resources.

PART 1 - Firewall Configuration Using UFW

Step 1: Install UFW

sudo apt install ufw -y

```
ubuntuu@ubuntuu:~$ sudo apt install ufw -y
[sudo] password for ubuntuu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
0 upgraded, 0 newly installed, 0 to remove and 101 not upgraded.
ubuntuu@ubuntuu:~$
```

Step 2: Set Default Rules

- **Block all incoming requests** (safer)
- **Allow outgoing traffic**

sudo ufw default deny incoming
sudo ufw default allow outgoing

```
ubuntuu@ubuntuu:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
ubuntuu@ubuntuu:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
ubuntuu@ubuntuu:~$
```

Step 3: Allow Only Important Ports

Port	*Service*	*Why?*
22	SSH	For remote login
443	HTTPS	For secure websites
80	HTTP	We will **deny** (to practice control)

sudo ufw allow 22/tcp
sudo ufw allow 443/tcp
sudo ufw deny 80/tcp

```
ubuntuu@ubuntuu:~$ sudo ufw allow 22/tcp
Rule added
Rule added (v6)
ubuntuu@ubuntuu:~$ sudo ufw allow 443/tcp
Rule added
Rule added (v6)
ubuntuu@ubuntuu:~$ sudo ufw allow 80/tcp
Rule added
Rule added (v6)
ubuntuu@ubuntuu:~$
```

Step 4: Enable and Check Firewall

sudo ufw enable

sudo ufw status

```
ubuntuu@ubuntuu:~$ sudo ufw enable
Firewall is active and enabled on system startup
ubuntuu@ubuntuu:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
443/tcp                    ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
443/tcp (v6)               ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)

ubuntuu@ubuntuu:~$
```

Outcome:

Only allowed services can be accessed ,reduces hacking attacks.

PART 2 - Compliance Logging Using auditd

Step 1: Install auditd

sudo apt install auditd -y

```
ubuntuu@ubuntuu:~$ sudo apt install auditd -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libauparse0t64
Suggested packages:
  audispd-plugins
The following NEW packages will be installed:
  auditd libauparse0t64
0 upgraded, 2 newly installed, 0 to remove and 101 not upgraded.
```

Step 2: Start the audit service

sudo systemctl start auditd

Step 3: Monitor Important File (User Account File)

sudo auditctl -w /etc/passwd -p wa -k passwd_changes

- -w → watch file
- -p wa → watch for Write and Attribute changes
- -k passwd_changes → give it a name (tag)

Step 4: View Logs When Someone Modifies Users

sudo ausearch -k passwd_changes

```
ubuntuu@ubuntuu:~$ sudo systemctl start auditd
ubuntuu@ubuntuu:~$ sudo auditctl -w /etc/passwd -p wa -k passwd_changes
ubuntuu@ubuntuu:~$ sudo ausearch -k passwd_changes
----
time->Sun Nov 30 10:56:00 2025
type=PROCTITLE msg=audit(1764480360.888:189): proctitle=617564697463746C002D77002F6574632F706173737764002D70007761002D6B007061737377645F6368616E676573
type=SYSCALL msg=audit(1764480360.888:189): arch=c000003e syscall=44 success=yes exit=1084 a0=4 a1=7ffffdb0d2a0 a2=43c a3=0 items=0 ppid=5034 pid=5035 auid=1000 uid=0 gid=0 euid=0 suid
=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=3 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1764480360.888:189): auid=1000 ses=3 subj=unconfined op=add_rule key="passwd_changes" list=4 res=1
ubuntuu@ubuntuu:~$
```

Outcome:

If any user is added, modified, or deleted → It gets recorded → Useful in investigations.

How to Verify auditd is Working

Step 1 - Make a Small Change (So audit logs get created)

We will **add a temporary user** (just for testing):
sudo adduser testuser
Enter a password → You can give any temporary password.
This action **modifies /etc/passwd**, so auditd will record it.

```
ubuntuu@ubuntuu:~$ sudo adduser testuser
info: Adding user `testuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `testuser' (1002) ...
info: Adding new user `testuser' (1002) with group `testuser (1002)' ...
info: Creating home directory `/home/testuser' ...
info: Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for testuser
Enter the new value, or press ENTER for the default
        Full Name []: test
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `testuser' to supplemental / extra groups `users' ...
info: Adding user `testuser' to group `users' ...
ubuntuu@ubuntuu:~$
```

Step 2 — Now Check the audit Logs

Run: sudo ausearch -k passwd changes

```
type=CWD msg=audit(1764424111.606:128): cwd="/home/ubuntu"
type=SYSCALL msg=audit(1764424111.606:128): arch=c000003e syscall=82 success=yes exit=0 a
0=7ffd0d256b00 a1=61c0bf02fea0 a2=7ffd0d256a70 a3=100 items=9 ppid=6922 pid=6947 auid=100
0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=2 comm="chfn" exe=
"/usr/bin/chfn" subj=unconfined key="passwd_changes"
```

This means auditd **caught and logged** the change.
To view in readable form, use:
sudo aureport -f -i
This will show logs in **human-readable** format (file names instead of inode numbers).

```
ubuntu@ubuntu:~$ sudo aureport -f -i

File Report
===============================================
# date time file syscall success exe auid event
===============================================
1. 11/29/2025 13:48:15 /etc/passwd openat yes /usr/sbin/useradd ubuntu 121
2. 11/29/2025 13:48:15 /etc/ rename yes /usr/sbin/useradd ubuntu 123
3. 11/29/2025 13:48:31 /etc/passwd openat yes /usr/bin/chfn ubuntu 127
4. 11/29/2025 13:48:31 /etc/ rename yes /usr/bin/chfn ubuntu 128
ubuntu@ubuntu:~$
```

sudo lastlog
(to check the last log)

```
testuser                                         **Never logged in**
```

Step 1: Open Limits Configuration File
sudo nano /etc/security/limits.conf
Step 2: Add These Lines at Bottom
* hard nproc 200
* soft nofile 100
Meaning

Rule	Controls	Effect
nproc	Number of running processes	Stops users from flooding the system
nofile	Number of files a program can open	Prevents crashes / overload

Step 3: Check Limits
ulimit -a

```
ubuntu@ubuntu:~$ sudo nano /etc/security/limits.conf
ubuntu@ubuntu:~$ ulimit -a
real-time non-blocking time  (microseconds, -R) unlimited
core file size              (blocks, -c) 0
data seg size               (kbytes, -d) unlimited
scheduling priority                 (-e) 0
file size                   (blocks, -f) unlimited
pending signals                     (-i) 7324
max locked memory           (kbytes, -l) 245692
max memory size             (kbytes, -m) unlimited
open files                          (-n) 1024
pipe size                (512 bytes, -p) 8
POSIX message queues         (bytes, -q) 819200
real-time priority                  (-r) 0
stack size                  (kbytes, -s) 8192
cpu time                   (seconds, -t) unlimited
max user processes                  (-u) 7324
virtual memory              (kbytes, -v) unlimited
file locks                          (-x) unlimited
ubuntu@ubuntu:~$
```

Outcome:
No single user/application can misbehave , prevents system slowdown or high cloud bills.

PART 3 Checking Open Network Ports (Monitoring Active Services)

Aim: To identify which network ports and services are currently running on the Linux server.
Why This Is Important
Every open port is a **possible entry point for attackers**.
Only necessary services should be running.
Command Used: sudo ss -tuln
(Alternative command): sudo netstat -tuln

Explanation of Options

Option	Meaning
-t	TCP connections
-u	UDP connections
-l	Listening services
-n	Show port numbers

Output

```
ubuntuu@ubuntuu:~$ sudo ss -tuln
Netid            State             Recv-Q            Send-Q                       Local Address:Port                       Peer Address:Port           Process
udp              UNCONN            0                 0                              127.0.0.54:53                            0.0.0.0:*
udp              UNCONN            0                 0                           127.0.0.53%lo:53                            0.0.0.0:*
udp              UNCONN            0                 0                                0.0.0.0:5353                          0.0.0.0:*
udp              UNCONN            0                 0                                0.0.0.0:43467                         0.0.0.0:*
udp              UNCONN            0                 0                                   [::]:47986                            [::]:*
udp              UNCONN            0                 0                                   [::]:5353                             [::]:*
tcp              LISTEN            0                 4096                             0.0.0.0:22                            0.0.0.0:*
tcp              LISTEN            0                 4096                           127.0.0.54:53                           0.0.0.0:*
tcp              LISTEN            0                 4096                           127.0.0.1:631                           0.0.0.0:*
tcp              LISTEN            0                 4096                        127.0.0.53%lo:53                           0.0.0.0:*
tcp              LISTEN            0                 511                                 *:80                                    *:*
tcp              LISTEN            0                 4096                             [::]:22                                 [::]:*
tcp              LISTEN            0                 4096                            [::1]:631                                [::]:*
ubuntuu@ubuntuu:~$
```

Shows:

- Open ports (22, 443, etc.)
- Services listening on those ports

One can verify that **only authorized services and ports are active**, reducing security risks.

PART 4 Enable Automatic Security Updates

Aim: To ensure the system automatically installs critical security updates.

Why This Is Important

Unpatched systems are vulnerable to known attacks.
Automatic updates keep the server **safe without manual work**.

Step 1: Install Unattended Upgrades

sudo apt install unattended-upgrades -y

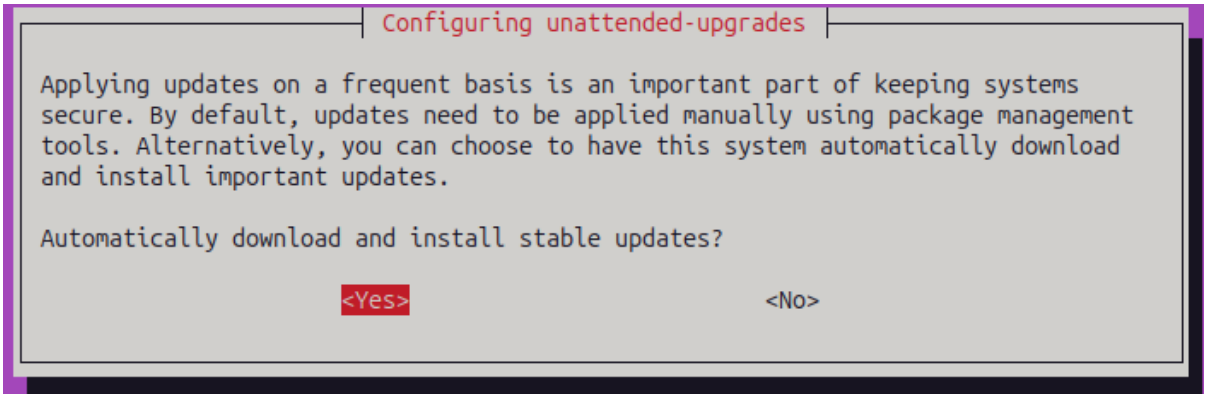
```
ubuntu@ubuntu:~$ sudo apt install unattended-upgrades -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unattended-upgrades is already the newest version (2.9.1+nmu4ubuntu1).
unattended-upgrades set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 211 not upgraded.
ubuntu@ubuntu:~$
```

Step 2: Enable Automatic Updates

sudo dpkg-reconfigure unattended-upgrades

Select **Yes** when prompted.

```
ubuntu@ubuntu:~$ sudo dpkg-reconfigure unattended-upgrades
```



Outcome
The system will automatically install important security updates, protecting it from new vulnerabilities.

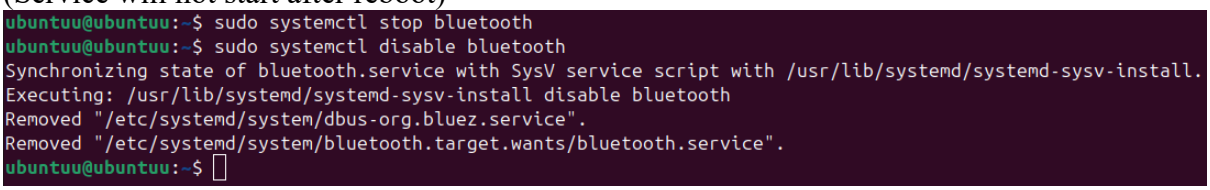
PART 5 Disable Unnecessary Services

Aim: To reduce attack surface by stopping and disabling unused services.
Why This Is Important
Unneeded services consume resources and increase security risks.

Step 1: List All Services
systemctl list-unit-files --type=service

Step 2: Stop an Unused Service (Example: Bluetooth)
sudo systemctl stop Bluetooth

Step 3: Disable the Service Permanently
sudo systemctl disable bluetooth
(Service will not start after reboot)



Outcome
Only required services remain active → improved security and performance.

Practical no-8

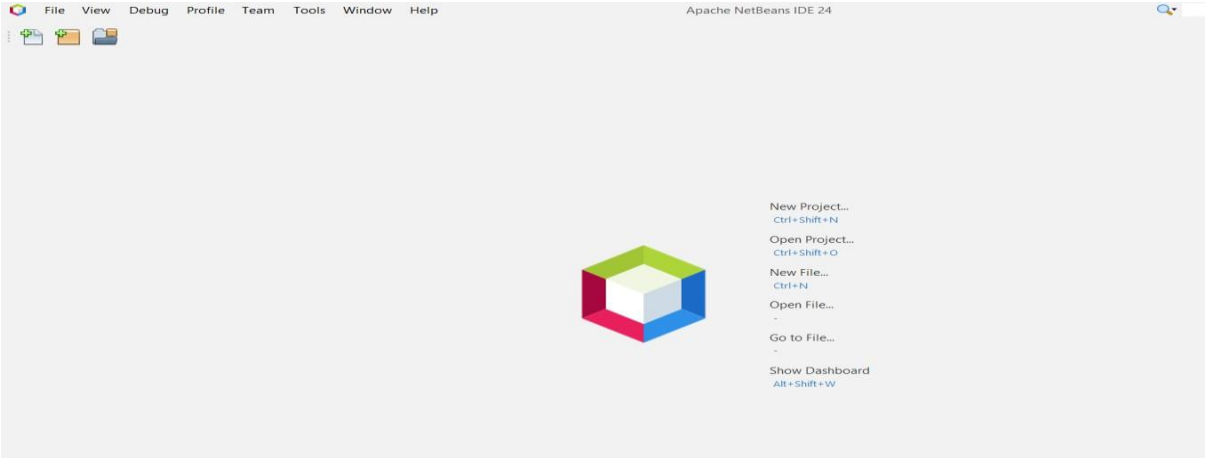
Aim: Show the implementation of web services.

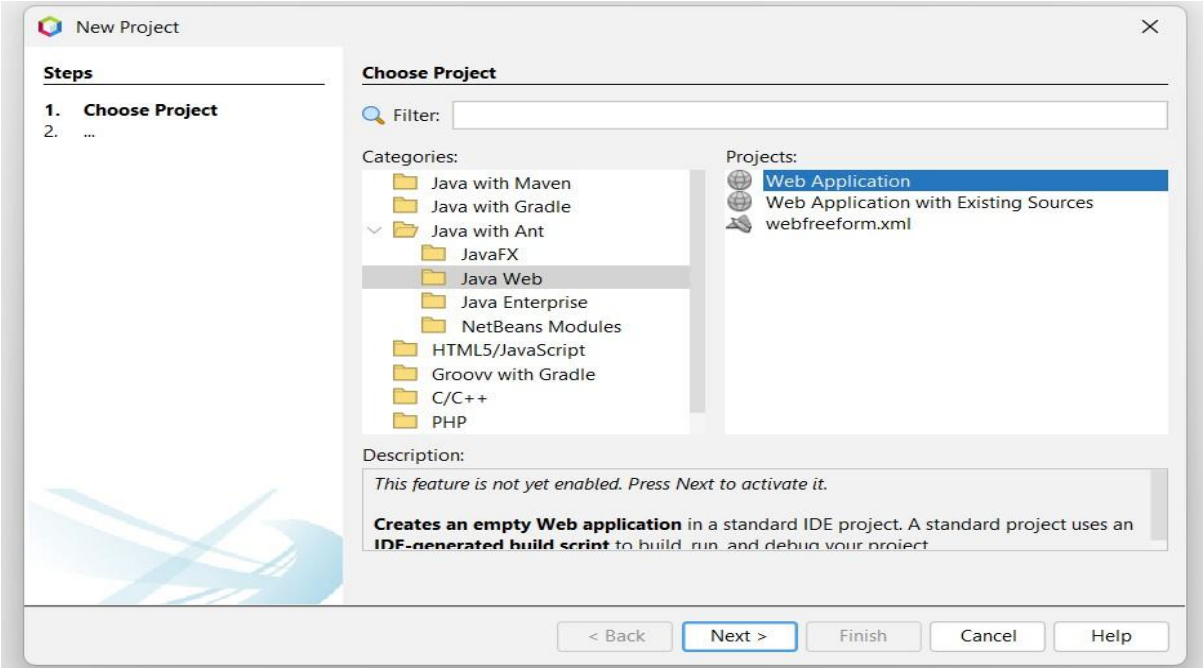
What Are Web Services?

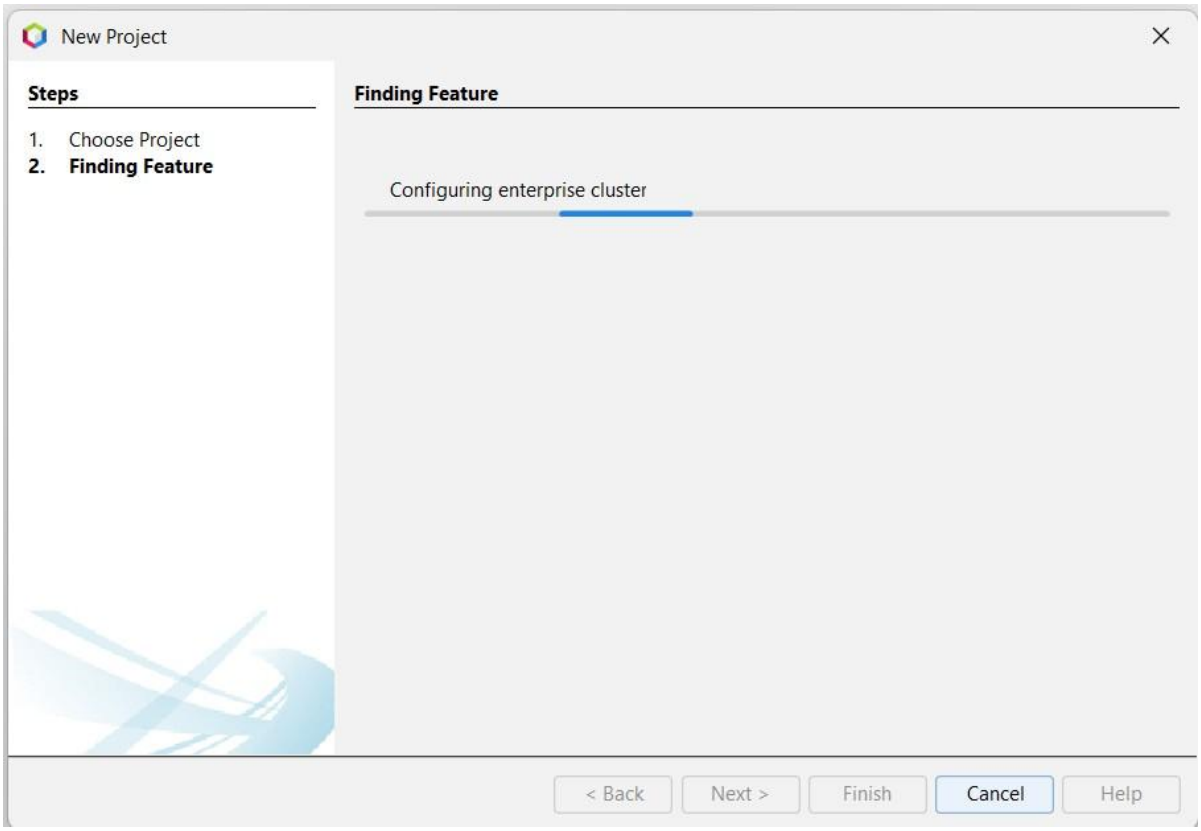
Web services are client and server applications that communicate over the World Wide Web's (WWW) HyperText Transfer Protocol (HTTP). As described by the World Wide Web Consortium (W3C), web services provide a standard means of interoperating between software applications running on a variety of platforms and frameworks. Web services are characterized by their great interoperability and extensibility, as well as their machine-process able descriptions, thanks to the use of XML. Web services can be combined in a loosely coupled way to achieve complex operations. Programs providing simple services can interact with each other to deliver sophisticated added-value services.

Create a calculator web service

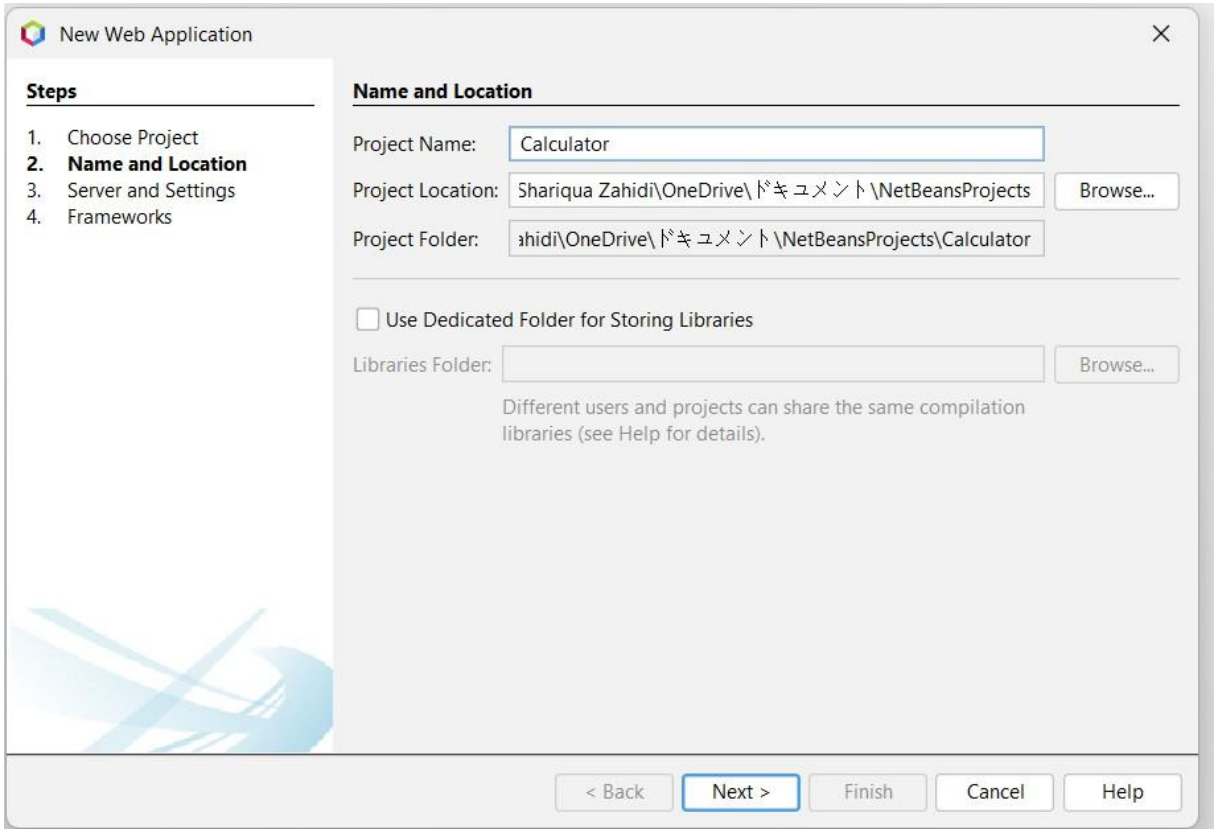
1. **Choose File > New Project. Select Web Application from the Java Web.**



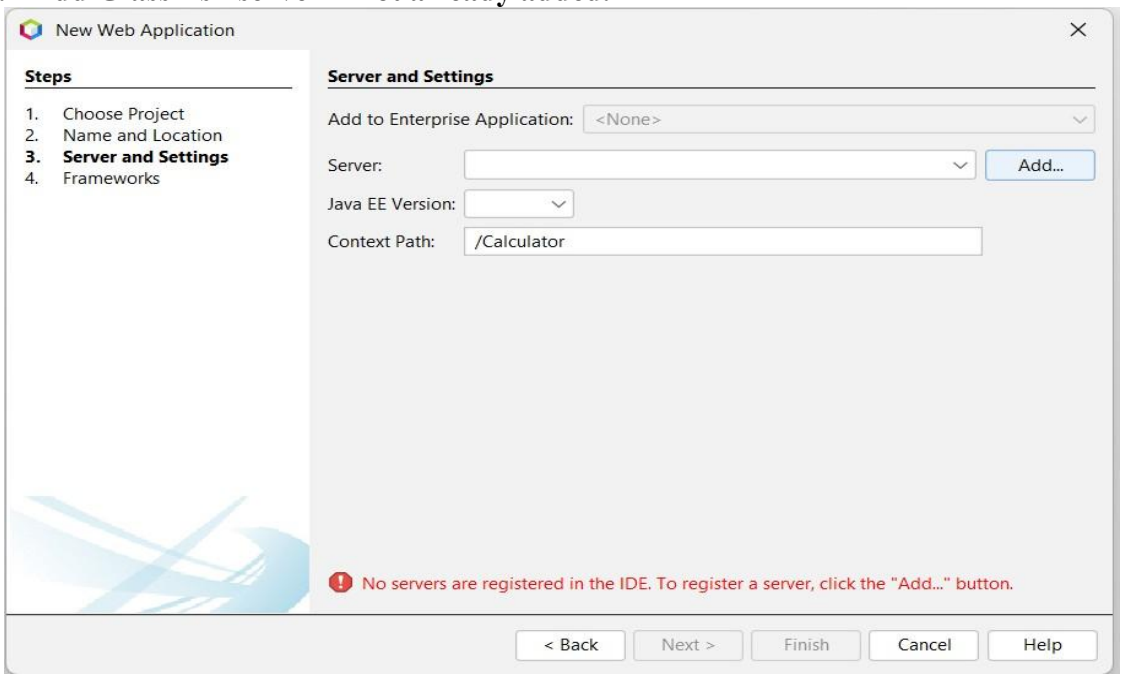


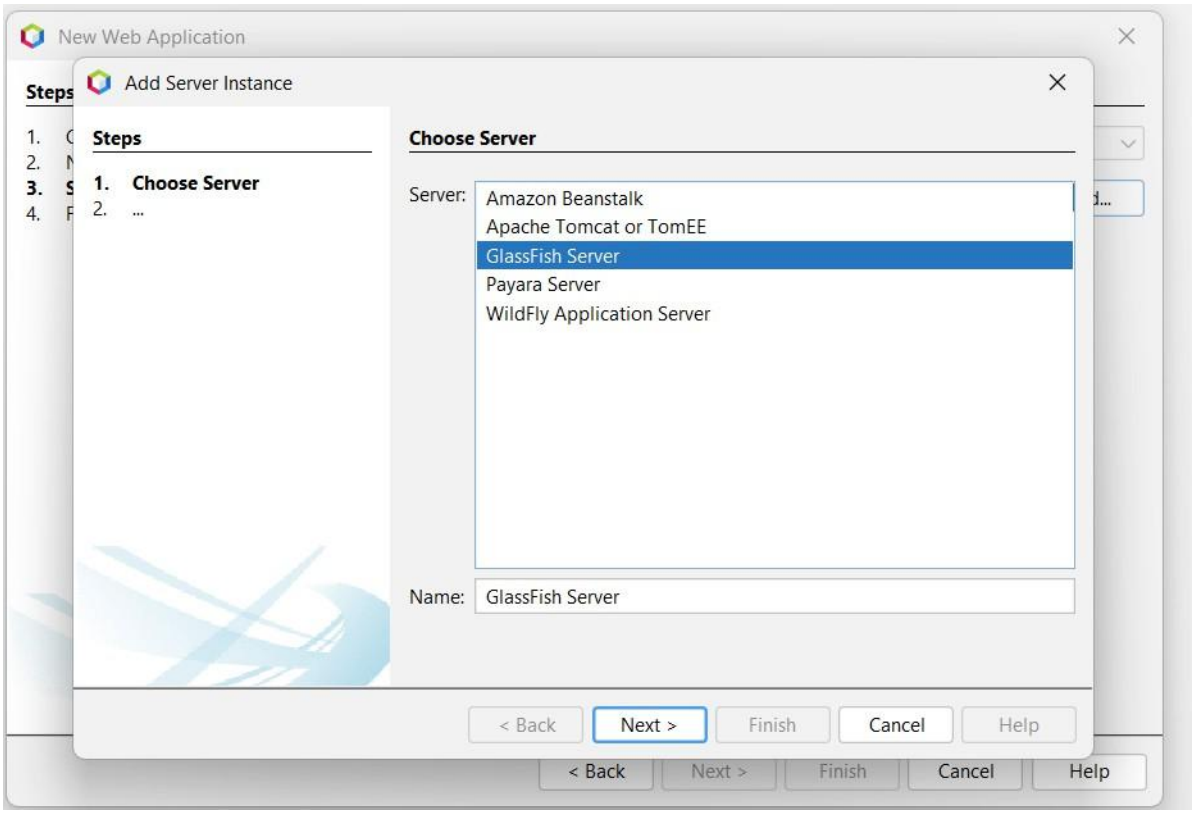


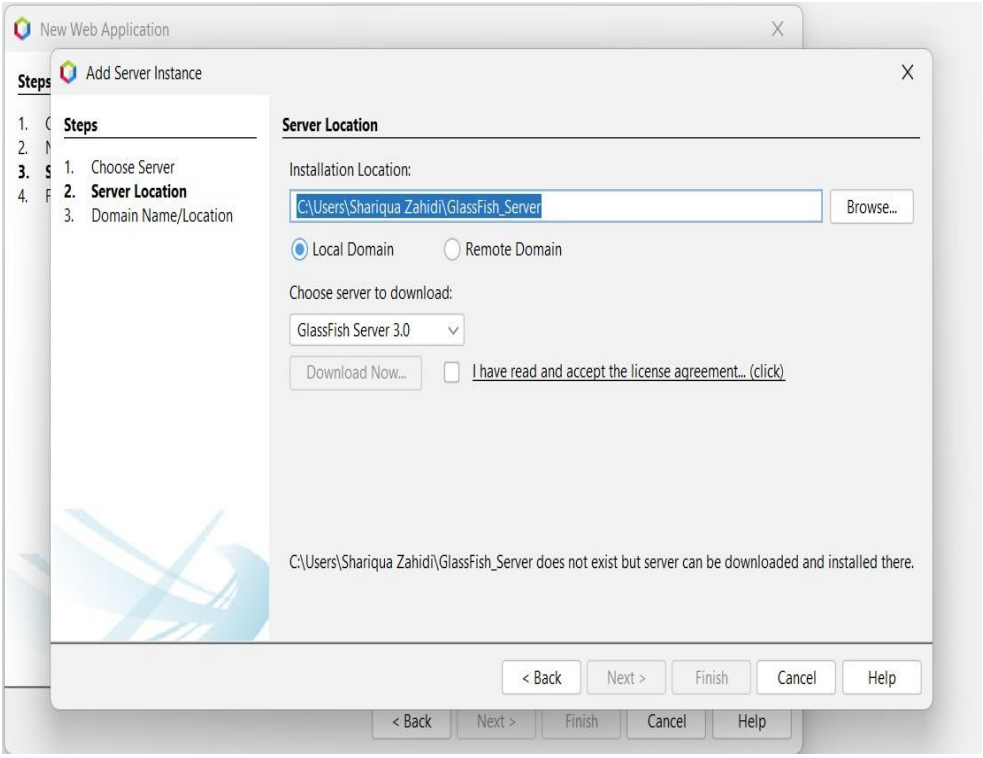
2. **Name the project Calculator. Select a location for the project. Click Next.**

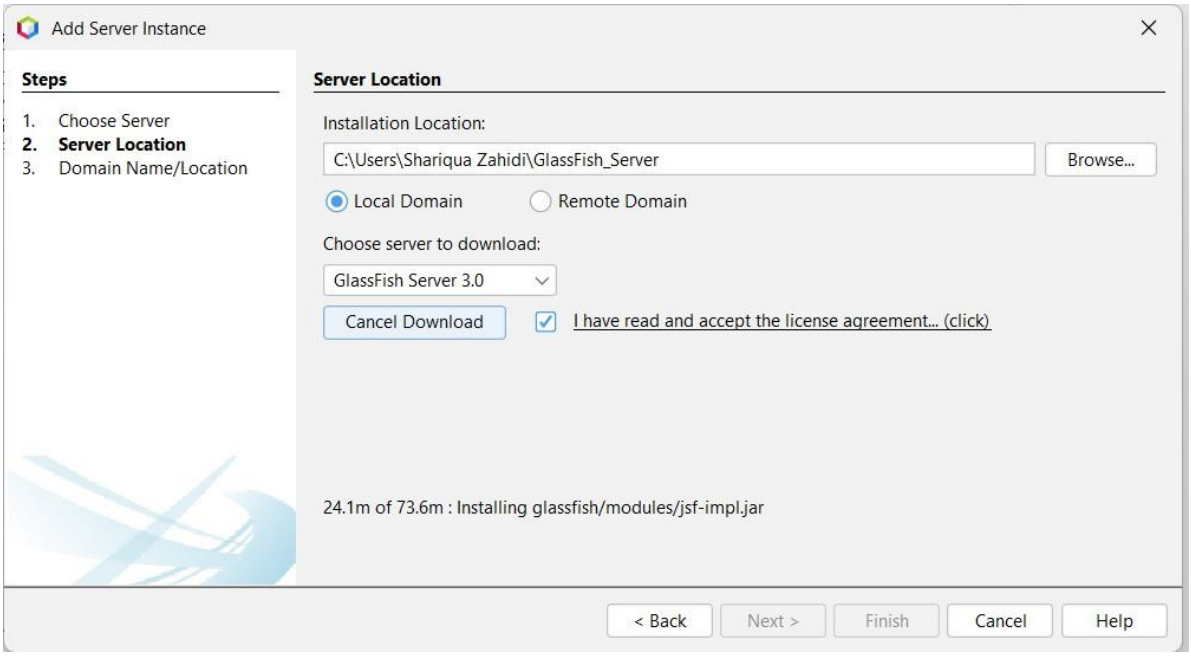


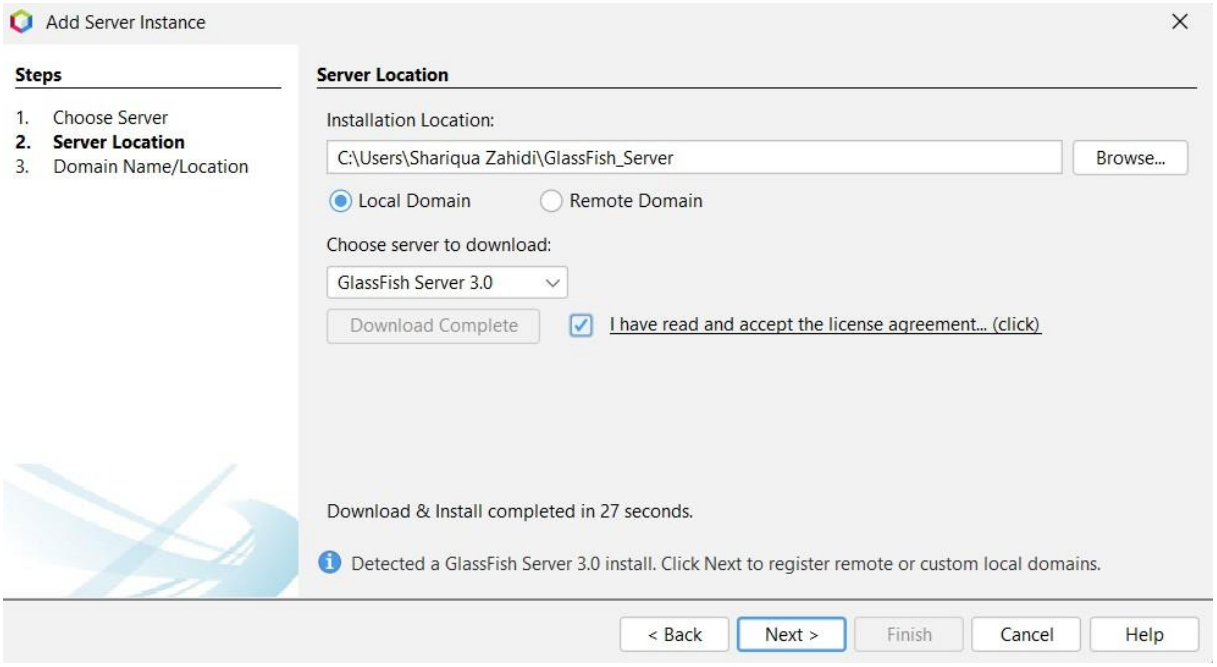
3. Add GlassFish server if not already added.

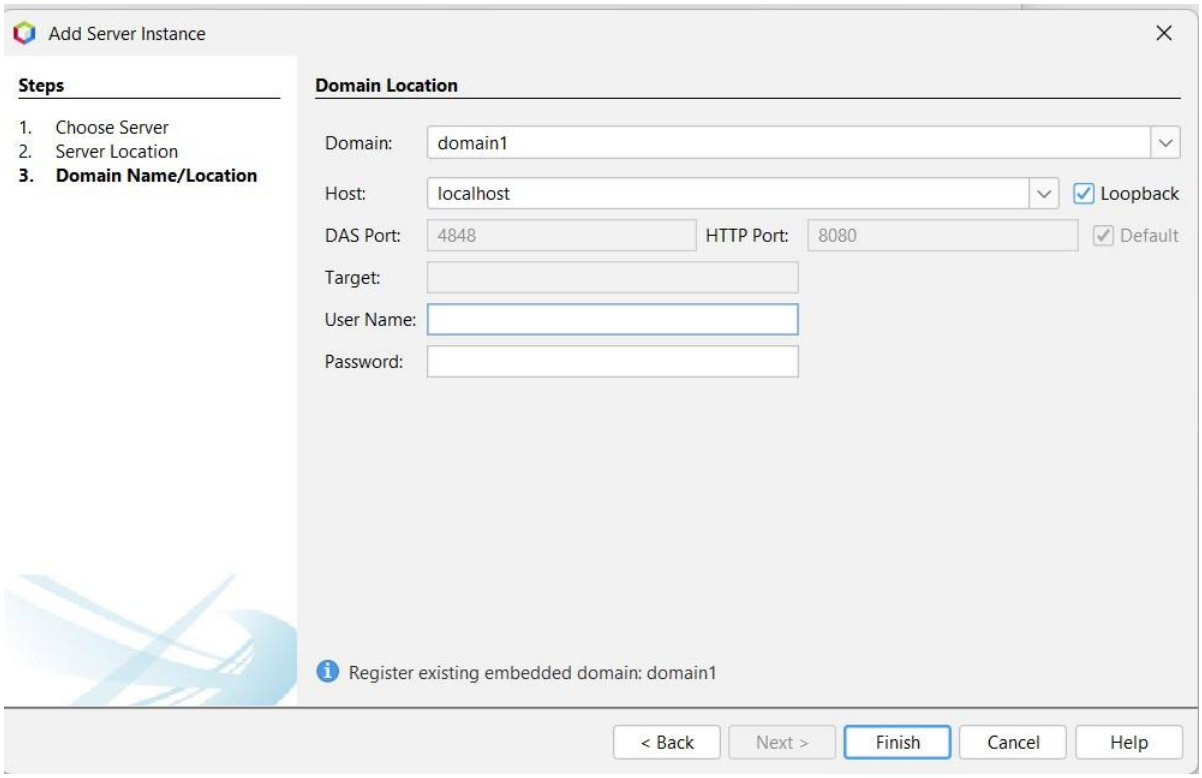


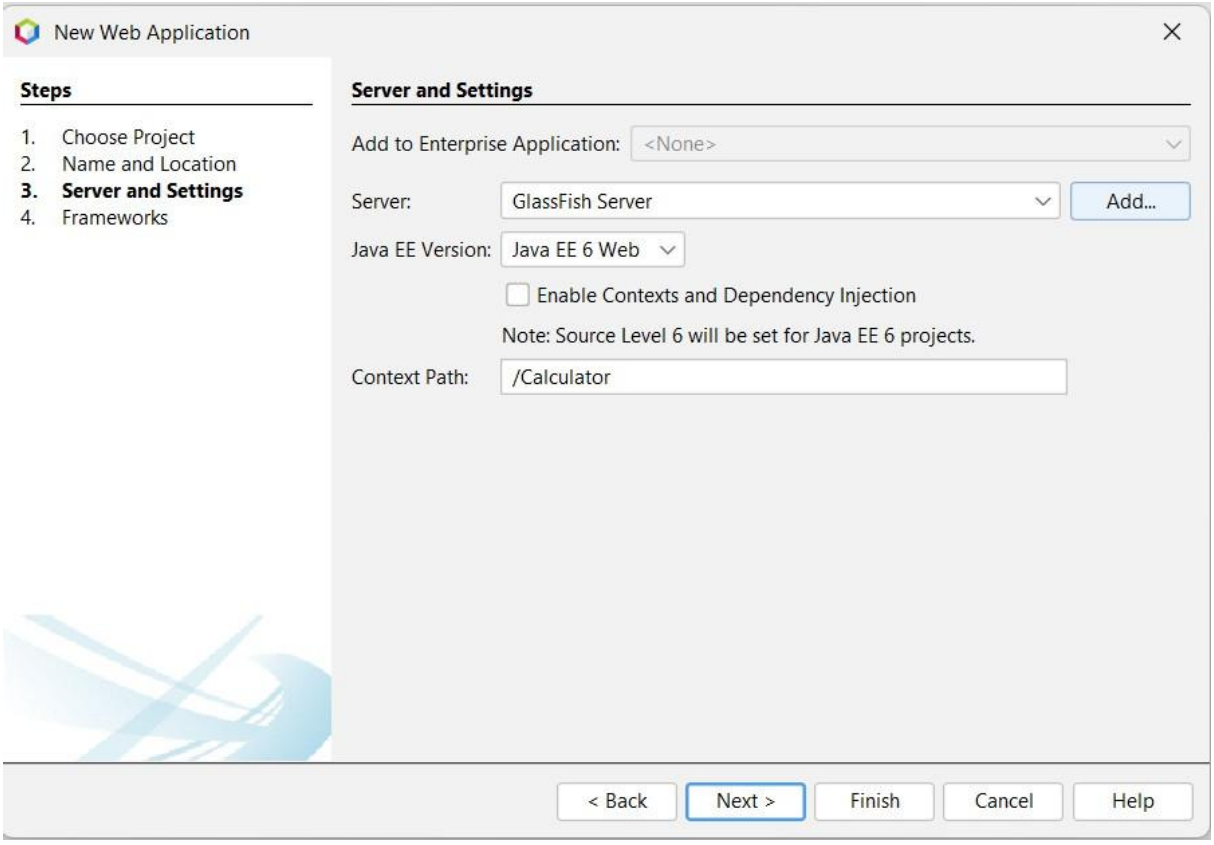

New Web Application
Add Server Instance
Steps
1. Choose Server
2. ...
Choose Server
Server:
Amazon Beanstalk
Apache Tomcat or TomEE
GlassFish Server
Payara Server
WildFly Application Server
Name: GlassFish Server
< Back
Next >
Finish
Cancel
Help

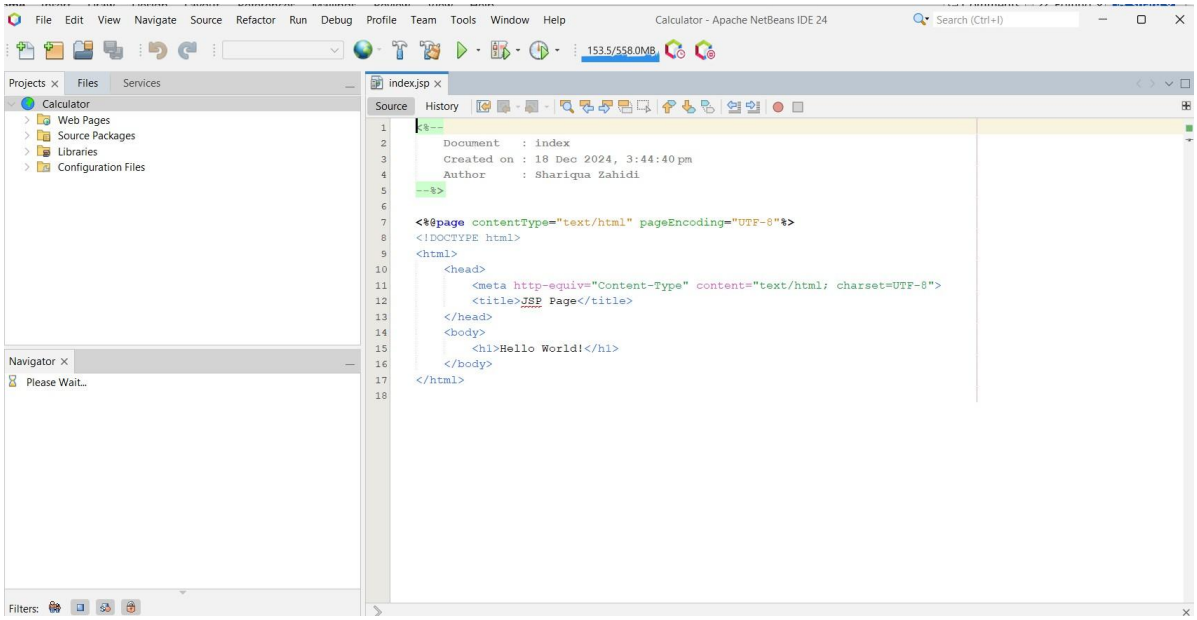

New Web Application
Add Server Instance
Steps
1. Choose Server
2. Server Location
3. Domain Name/Location
Server Location
Installation Location:
C:\Users\Shariqua Zahidi\GlassFish_Server
Browse...
Local Domain
Remote Domain
Choose server to download:
GlassFish Server 3.0
Download Now...
I have read and accept the license agreement... (click)
C:\Users\Shariqua Zahidi\GlassFish_Server does not exist but server can be downloaded and installed there.
< Back
Next >
Finish
Cancel
Help


Add Server Instance
Steps
1. Choose Server
2. Server Location
3. Domain Name/Location
Server Location
Installation Location:
C:\Users\Shariqua Zahidi\GlassFish_Server
Browse...
Local Domain
Remote Domain
Choose server to download:
GlassFish Server 3.0
Cancel Download
I have read and accept the license agreement... (click)
24.1m of 73.6m : Installing glassfish/modules/jsf-impl.jar
< Back
Next >
Finish
Cancel
Help

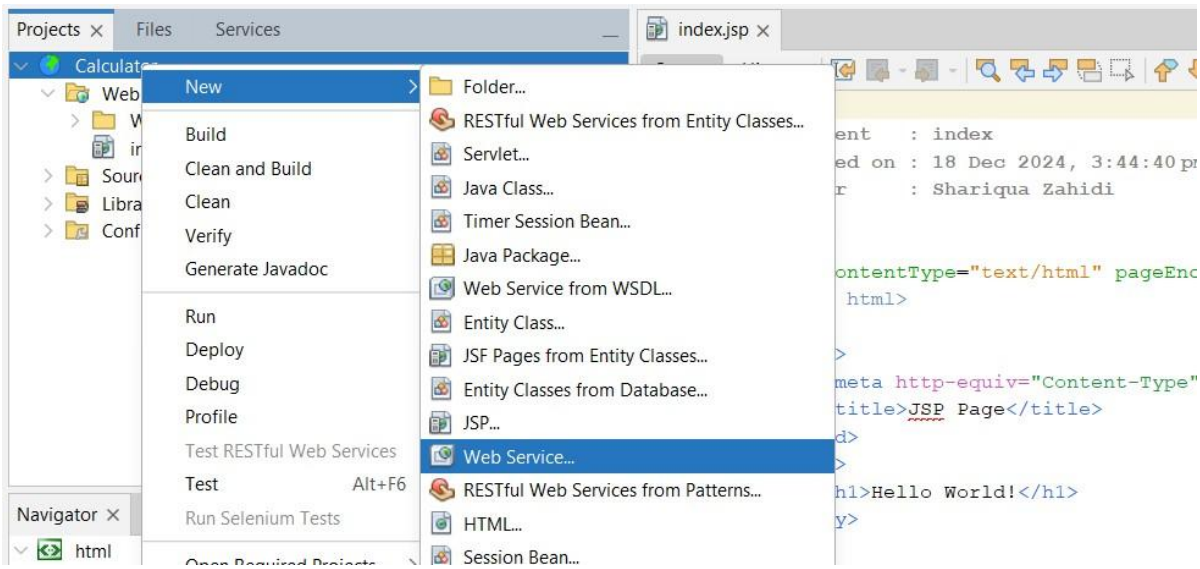

Add Server Instance
Steps
1. Choose Server
2. Server Location
3. Domain Name/Location
Server Location
Installation Location:
C:\Users\Shariqua Zahidi\GlassFish_Server
Browse...
Local Domain
Remote Domain
Choose server to download:
GlassFish Server 3.0
Download Complete
I have read and accept the license agreement... (click)
Download & Install completed in 27 seconds.
Detected a GlassFish Server 3.0 install. Click Next to register remote or custom local domains.
< Back
Next >
Finish
Cancel
Help


Add Server Instance
Steps
1. Choose Server
2. Server Location
3. Domain Name/Location
Domain Location
Domain: domain1
Host: localhost
Loopback
DAS Port: 4848
HTTP Port: 8080
Default
Target:
User Name:
Password:
Register existing embedded domain: domain1
< Back
Next >
Finish
Cancel
Help

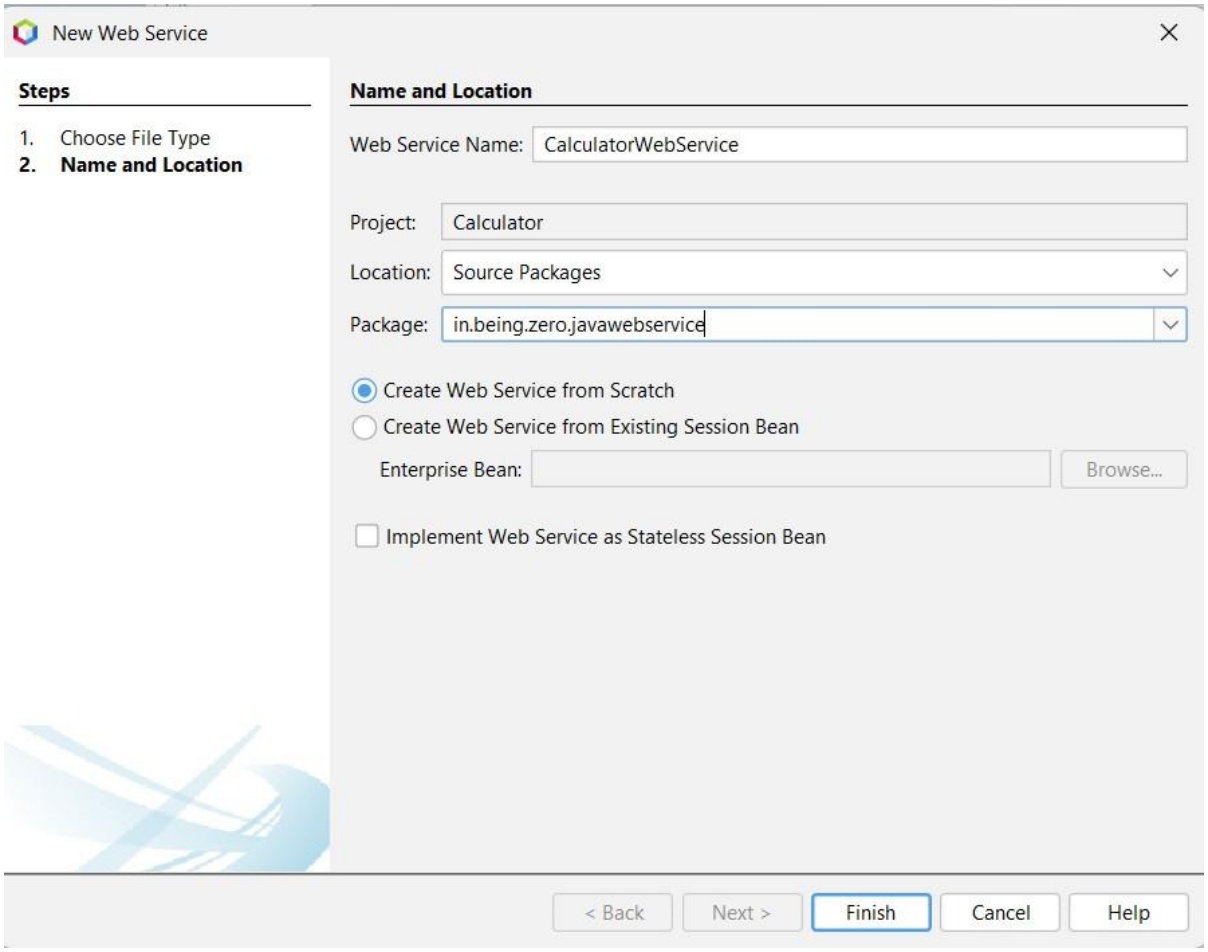

New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Server and Settings
Add to Enterprise Application: <None>
Server: GlassFish Server
Add...
Java EE Version: Java EE 6 Web
Enable Contexts and Dependency Injection
Note: Source Level 6 will be set for Java EE 6 projects.
Context Path: /Calculator
< Back
Next >
Finish
Cancel
Help



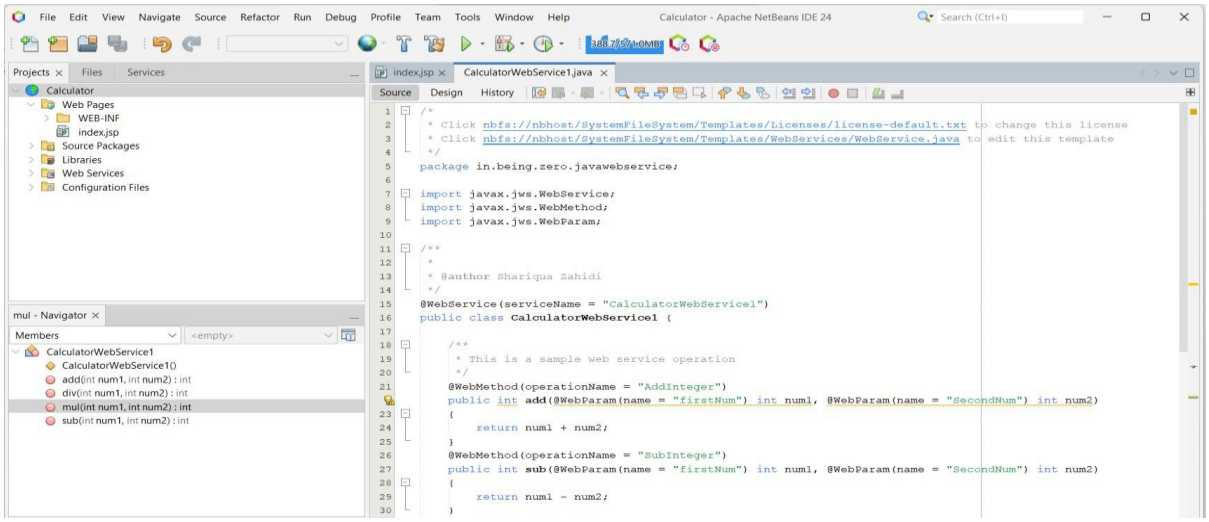
4. **Right click on the Project (i.e. Calculator), click New → Web Service**

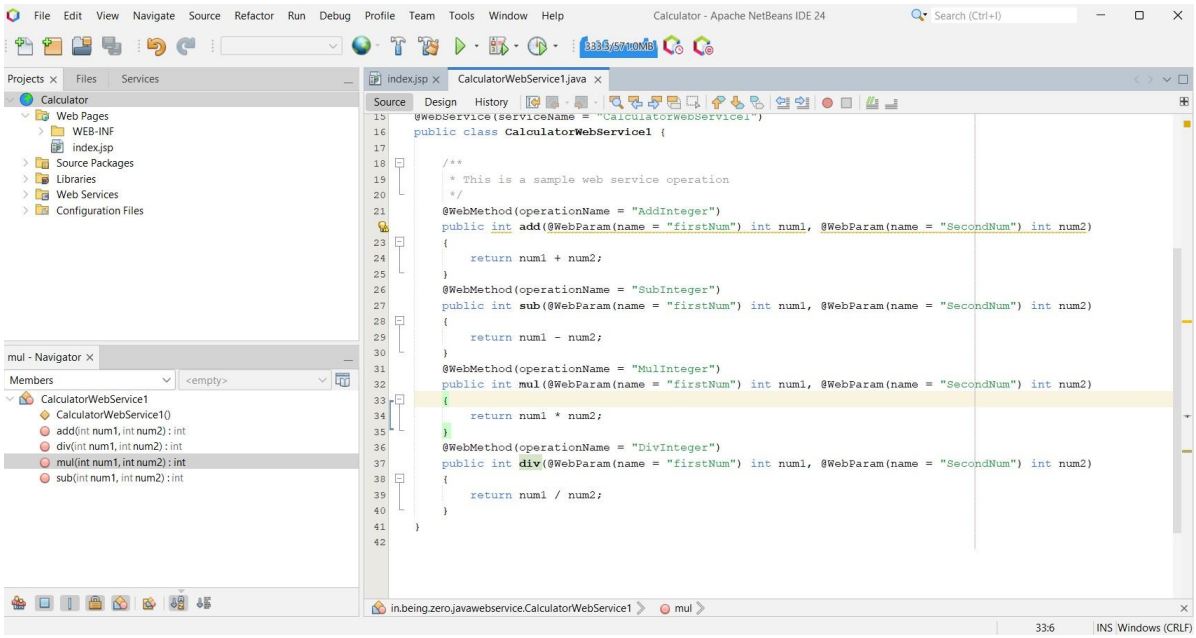


5. **Enter web service name and package name then click on Finish**

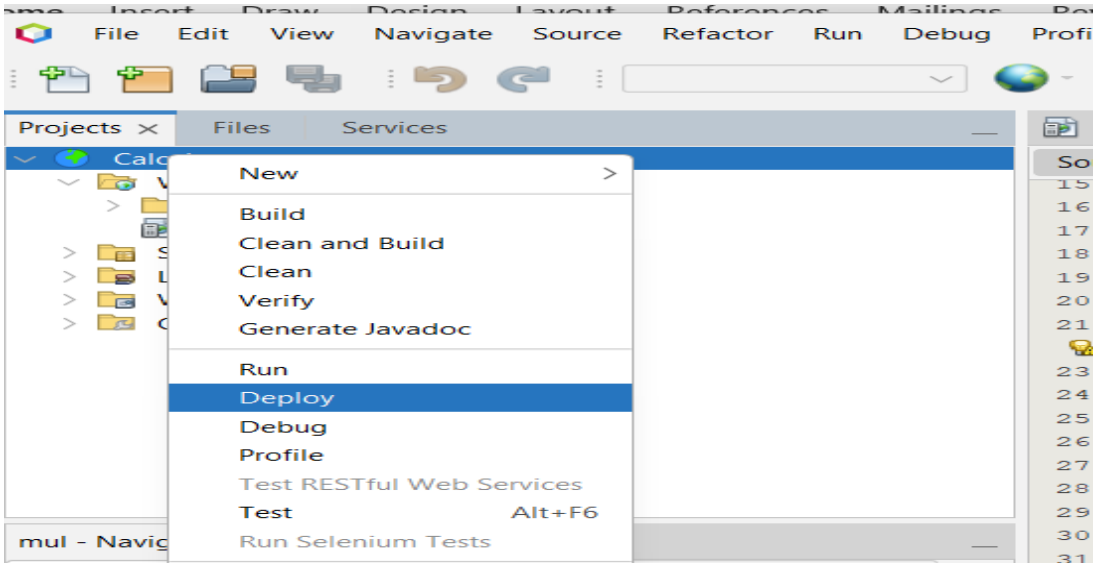


6. **A new page Web service code page should be displayed. Update the code by following the below screenshot**

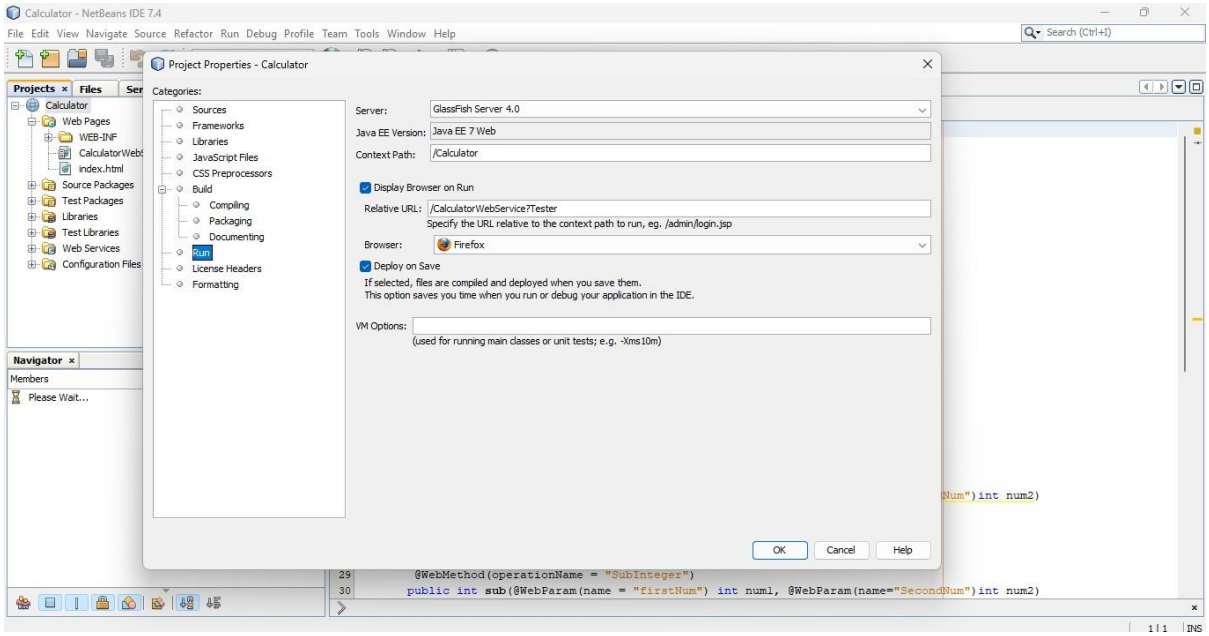




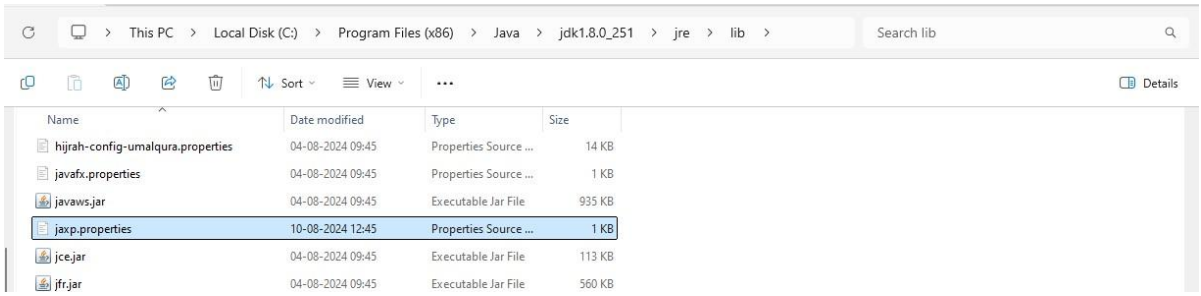
7. **Again, right click on project and click on deploy option**

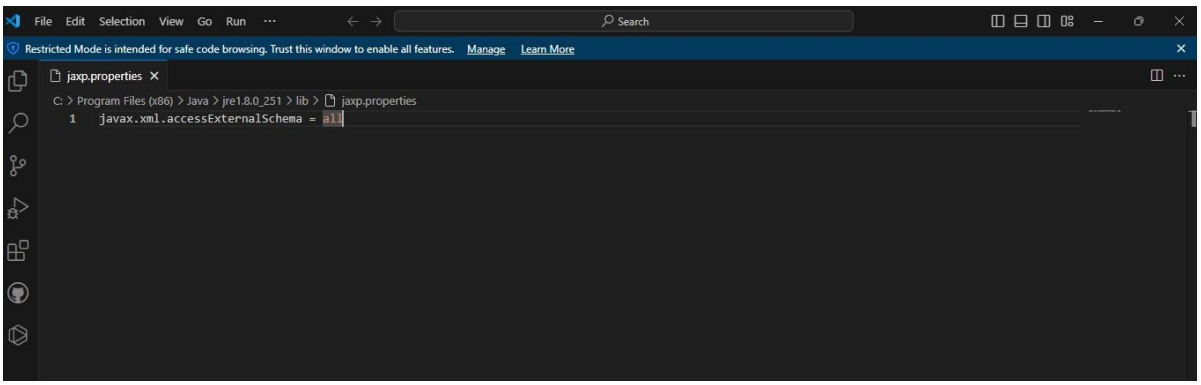


8. **Click on Ok**



9. **If deployment fails, then do the below steps**





10. Again, deploy the java file