**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No.** | **Practical Name** | **Date** | **Sign** |
| **1** | A client server based program using TCP to find if the number entered is prime. | 14/7/2025 |  |
| **2** | A client server TCP based chatting application. | 28/7/2025 |  |
| **3** | A client server based program using UDP to find if the number entered is even or odd. | 11/8/2025 |  |
| **4** | A client server based program using UDP to find the factorial of the entered number. | 18/8/2025 |  |
| **5** | A program to implement simple calculator operations like addition, subtraction, multiplication and division. | 8/9/2025 |  |
| **6** | A program that finds the square, square root, cube and cube root of the entered number. | 8/9/2025 |  |
| **7** | A RMI based application program to display current date and time. | 15/9/2025 |  |
| **8** | A RMI based application program that converts digits to words, e.g. 123 will be converted to one two three. | 29/9/2025 |  |
| **9** | Implement Xen virtualization and manage with XenCenter. | 6/10/2025 |  |
| **10** | Implement Windows Hyper Virtualization | 13/10/2025 |  |

# Practical No. 1

**Aim: A client server based program using TCP to find if the number entered is prime.**

**Code:**

**1. TCPServer.java** import java.io.\*; import java.net.\*;

public class TCPServer {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

try {

ServerSocket serverSocket = new ServerSocket(8888);

System.out.println("Server started .Waiting for client..");

Socket socket = serverSocket.accept();

System.out.println("Client Connected: ");

DataInputStream dis = new

DataInputStream(socket.getInputStream());

DataOutputStream dos = new

DataOutputStream(socket.getOutputStream());

int number = dis.readInt();

System.out.println("Recieved number: " + number);

boolean isPrime = checkPrime(number);

String result = number + (isPrime ? "is a prime number." : "is not a prime number"); dos.writeUTF(result); dis.close(); dos.close(); socket.close();

serverSocket.close();

} catch (IOException e) { e.printStackTrace();

}

}

public static boolean checkPrime(int num) { if (num <= 1) return false;

for (int i = 2; i <= Math.sqrt(num); i++) { if (num % i == 0) return false;

} return true;

}

}

**2. TCPclient.java** import java.io.\*; import java.net.\*; public class TCPServer { \* @param args public static void main(String[] args) {

try {

ServerSocket serverSocket = new ServerSocket(8888);

System.out.println("Server started .Waiting for client..");

Socket socket = serverSocket.accept();

System.out.println("Client Connected: ");

DataInputStream dis = new DataInputStream(socket.getInputStream()); DataOutputStream dos = new

DataOutputStream(socket.getOutputStream()); int number = dis.readInt();

System.out.println("Recieved number: " + number); boolean isPrime = checkPrime(number);

String result = number + (isPrime ? "is a prime number." : "is not a prime number");

dos.writeUTF(result); dis.close(); dos.close(); socket.close(); serverSocket.close(); } catch (IOException e) {

e.printStackTrace();

}

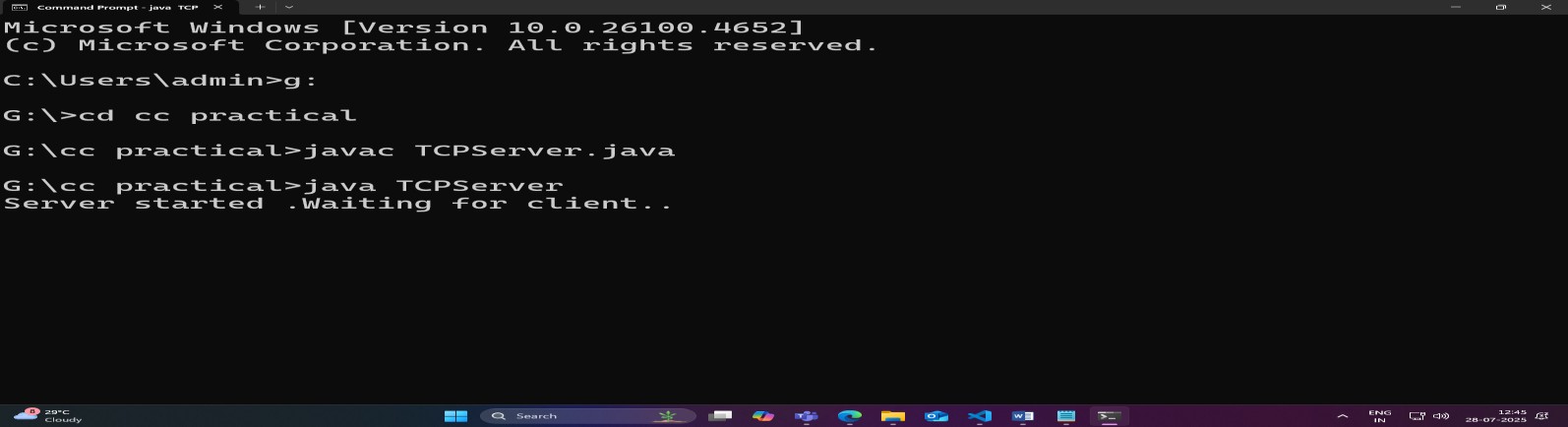
}

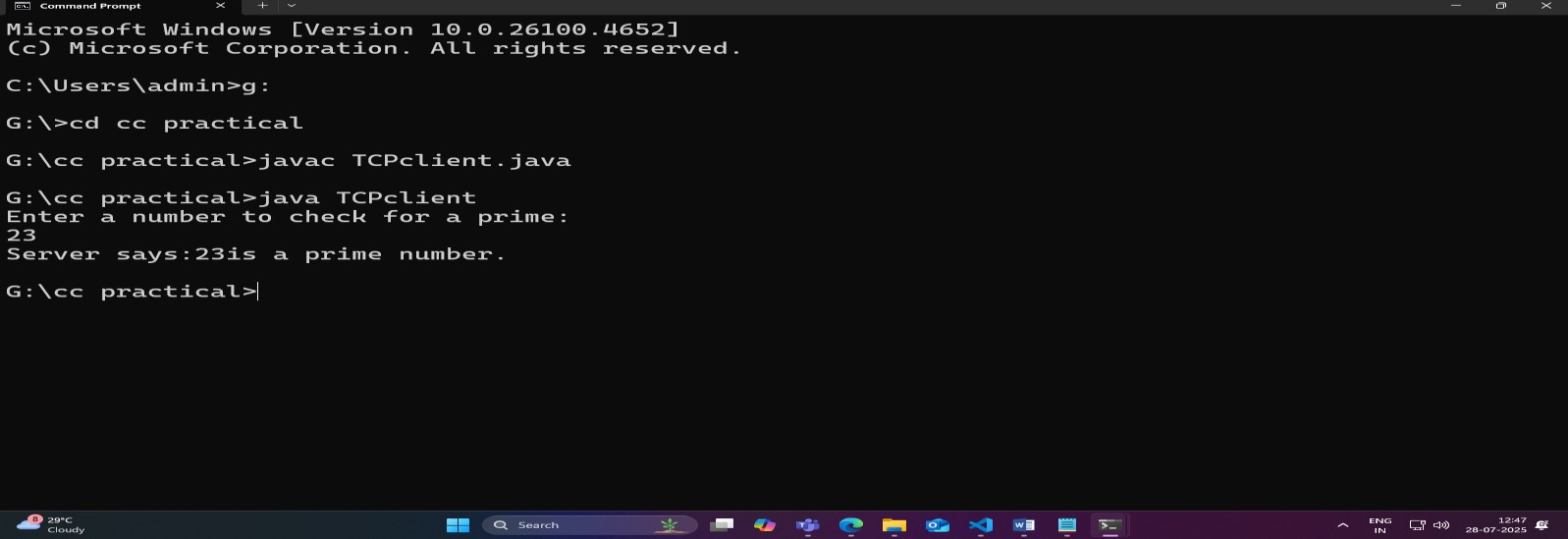
public static boolean checkPrime(int num) { if (num <= 1) return false; for (int i = 2; i <= Math.sqrt(num); i++) { if (num % i == 0) return false;

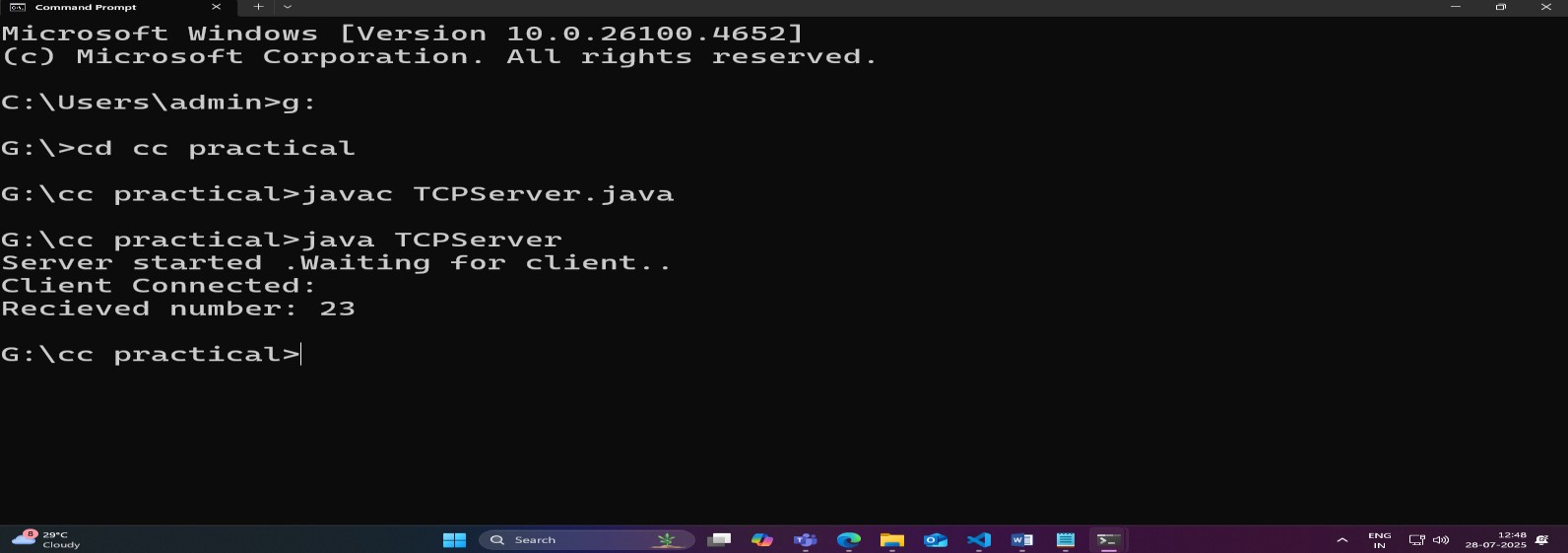
}

return true; }}

**Output:**







**Practical 2 Aim: A client server TCP based chatting application**

**TCPChatServer.java**:

import java.io.\*; import java.net.\*; public class ChatServer { public static void main(String[] args) {

try {

ServerSocket serverSocket = new ServerSocket( 5000);

System.out.println("Server started. Waiting for client...");

Socket socket = serverSocket.accept();

System.out.println("Client connected....");

DataInputStream dis = new DataInputStream(socket.getInputStream());

DataOutputStream dos = new

DataOutputStream(socket.getOutputStream());

BufferedReader reader = new BufferedReader(new

InputStreamReader(System.in));

String clientMessage = "" ,serverMessage = ""; while(!clientMessage.equals("bye")) { clientMessage = dis.readUTF();

System.out.print("Client:" + clientMessage);

System.out.print("Server:"); serverMessage = reader.readLine(); dos.writeUTF (serverMessage);

dos.flush();

} dis.close(); dos.close(); socket.close(); serverSocket.close();

} catch(Exception e){

System.out.println("Error:" +e);

}

}

}

**TCPChatClient.java**

import java.io.\*; import java.net.\*; public class ChatClient { public static void main(String[] args ){

try {

Socket socket = new Socket("localhost", 5000);

System.out.println("Connected to Server..");

DataInputStream dis = new DataInputStream(socket.getInputStream());

DataOutputStream dos = new

DataOutputStream(socket.getOutputStream());

BufferedReader reader = new BufferedReader(new

InputStreamReader(System.in));

String clientMessage = "" ,serverMessage = "" while(!clientMessage.equals("bye")) { System.out.print("Client:"); clientMessage = reader.readLine(); dos.writeUTF(clientMessage); dos.flush(); serverMessage = dis.readUTF();

System.out.println("Server:" +serverMessage);

} dis.close(); dos.close(); socket.close();

} catch(Exception e){

System.out.println("Error:" +e);

}

}

}

**Output:**





# Practical 3

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

import java.io.\*; import java.net.\*; public class EvenOddServer{ public static void main(String[] args){ DatagramSocket serverSocket = null;

try{

serverSocket = new DatagramSocket(9876); byte[] receiveData = new byte [1024]; byte[] sendData = new byte[1024];

System.out.println("UDP Server listening on port 9876...."); while(true){

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length); serverSocket.receive(receivePacket);

String clientMessage = new

String(receivePacket.getData(),0,receivePacket.getLength()); int number = Integer.parseInt(clientMessage.trim()); InetAddress clientAddress = receivePacket.getAddress(); int clientPort = receivePacket.getPort();

String response;

if(number%2 ==0){ response = number+" is Even";

} else { response = number+" is Odd";

}

sendData = response.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort); serverSocket.send(sendPacket);

}

} catch (IOException e){

e.printStackTrace(); } finally { if (serverSocket != null){ serverSocket.close();

}

}

}

}

**EvenOddClient.java**

import java.io.\*; import java.net.\*; import java.util.Scanner; public class EvenOddClient{ public static void main(String[] args){ DatagramSocket clientSocket = null;

try{ clientSocket = new DatagramSocket();

InetAddress serverAddress = InetAddress.getByName("localhost"); int serverPort = 9876

Scanner scanner = new Scanner(System.in); byte[] sendData = new byte [1024]; byte[] receiveData = new byte[1024];

System.out.println("Enter a number...."); String numberString = scanner.nextLine(); sendData = numberString.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress , serverPort); clientSocket.send(sendPacket);

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length); clientSocket.receive(receivePacket);

String serverResponse = new

String(receivePacket.getData(),0,receivePacket.getLength());

System.out.println("Server response :" +serverResponse); } catch (IOException e){

e.printStackTrace();

} finally {

if (clientSocket != null){ clientSocket.close();

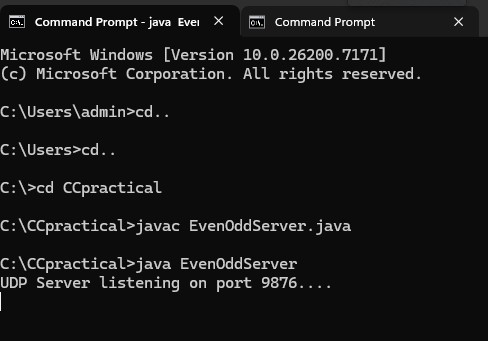
}

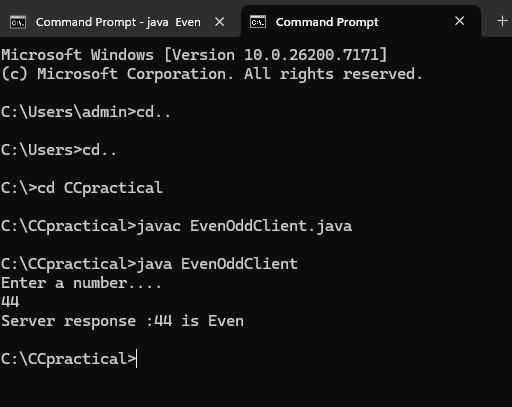
}

}

}

**Output:**





# Practical 4

**Aim : A client server based program using UDP to find the factorial of the entered number. UDPFactorialServer.java** import java.net.\*; import java.io.\*; public class UDPFactorialServer { public static void main(String[] args) { int port = 1234; try {

DatagramSocket socket = new DatagramSocket(port);

System.out.println("Server is running on port " + port + "..."); while (true) {

// Receive number from client byte[] buffer = new byte[1024];

DatagramPacket receivedPacket = new DatagramPacket(buffer, buffer.length); socket.receive(receivedPacket);

String message = new String(receivedPacket.getData(), 0, receivedPacket.getLength()).trim(); int number = Integer.parseInt(message);

// Calculate factorial int result = 1; for (int i = 1; i <= number; i++) { result \*= i;

}

String response = "Factorial of " + number + " is " + result;

// Send result back to client byte[] responseData = response.getBytes();

InetAddress clientAddress = receivedPacket.getAddress(); int clientPort = receivedPacket.getPort();

DatagramPacket responsePacket = new

DatagramPacket(responseData, responseData.length, clientAddress, clientPort); socket.send(responsePacket);

System.out.println("Processed request for number: " + number);

}

} catch (Exception e) {

System.out.println("Server error: " + e.getMessage());

e.printStackTrace();

}

}

} **myclient.java** import java.net.\*; import java.io.\*; class myclient { public static void main(String[] args) { int serverPort = 1234;

try { DatagramSocket socket = new DatagramSocket();

InetAddress serverAddress = InetAddress.getLocalHost();

BufferedReader reader = new BufferedReader(new

InputStreamReader(System.in));

System.out.print("Enter the number to find factorial: ");

String message = reader.readLine(); // Send the number to server byte[] sendData = message.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort); socket.send(sendPacket); // Receive the result from server byte[] receiveData = new byte[1024];

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length); socket.receive(receivePacket);

String response = new String(receivePacket.getData(), 0, receivePacket.getLength());

System.out.println("Response from server: " + response); socket.close();

} catch (IOException e) {

System.out.println("Client error: " + e.getMessage());

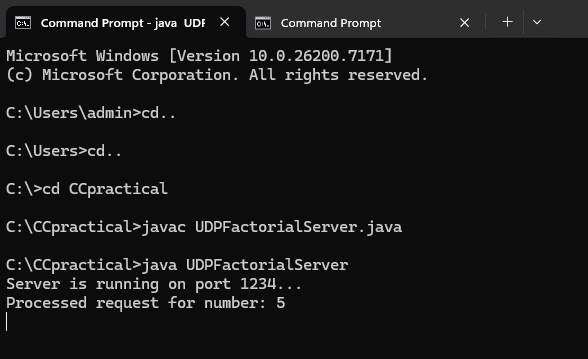
e.printStackTrace();

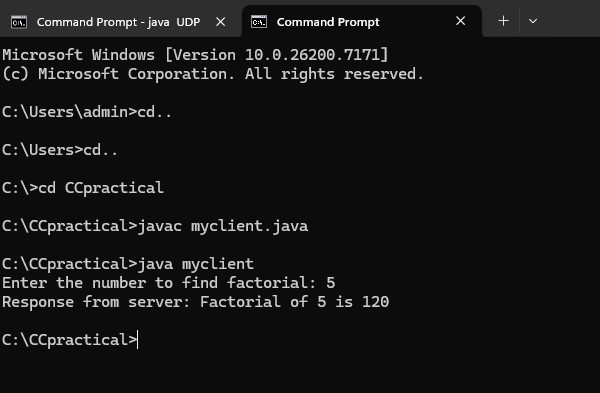
}

}

}

**Output:**

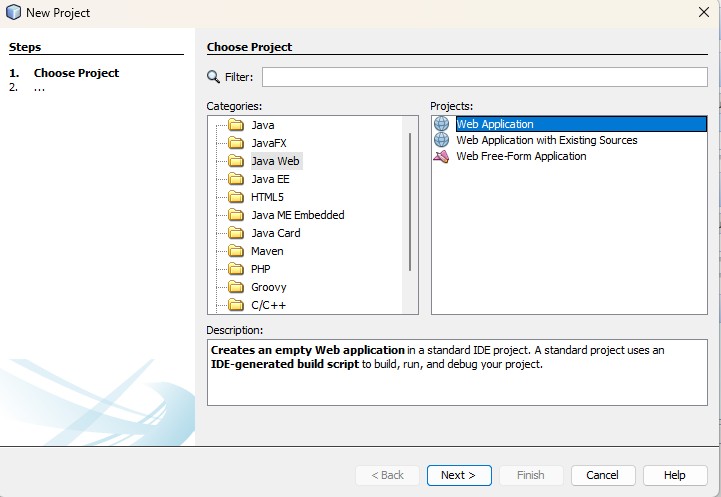
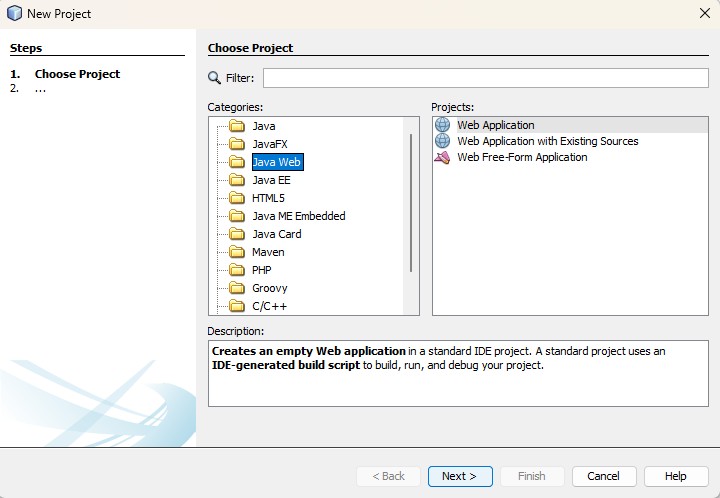


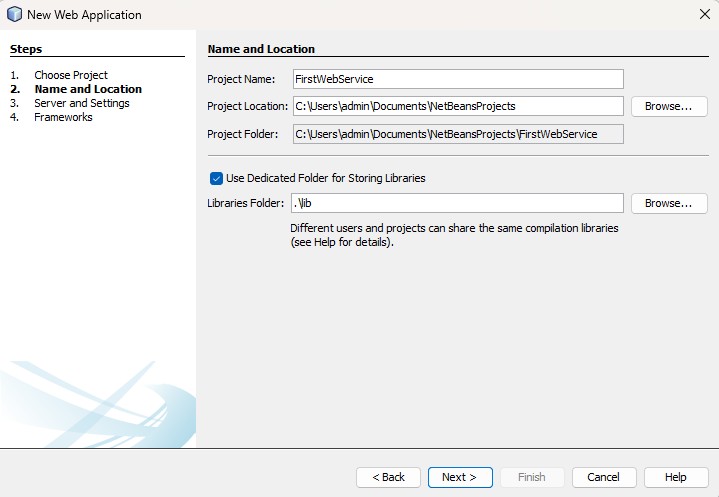


# Practical 5

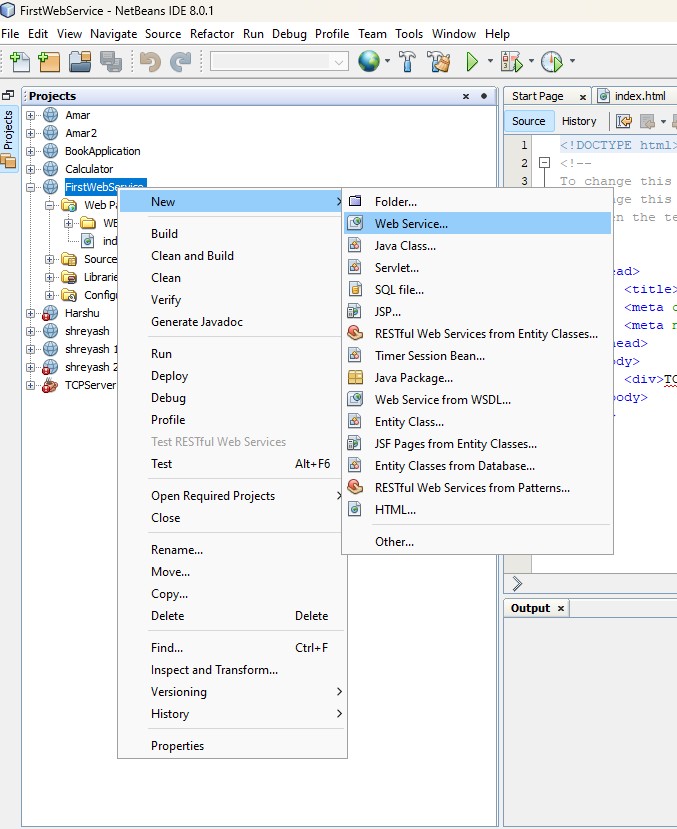
**Aim: A program to implement simple calculator operation like addition, subtraction, multiplication, division.**

STEP1: Choose Project then Java Web and Web Application and then press next.

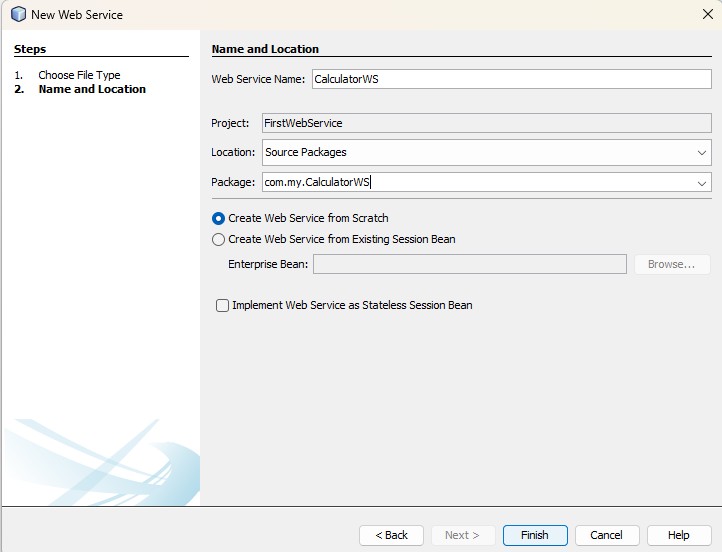




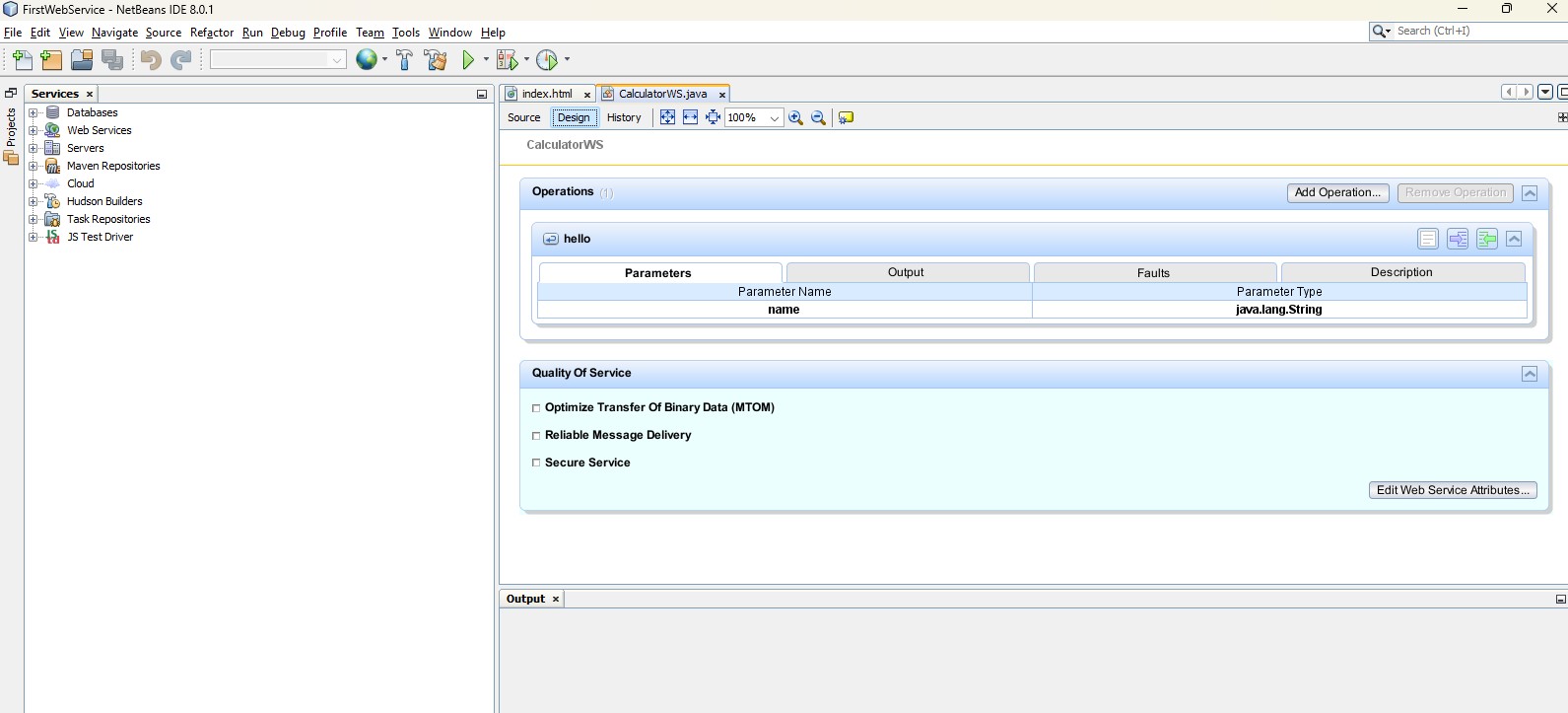
Step2: Select Web Services



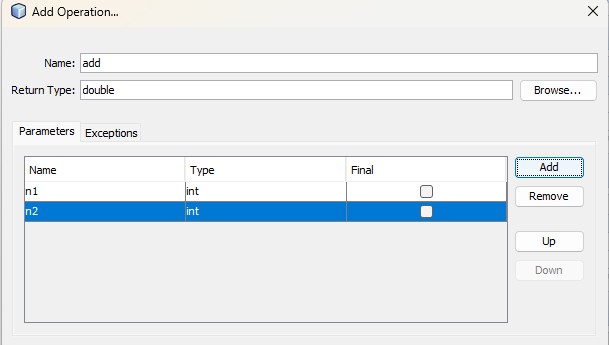
STEP3: Add name to the Web Service and Package name also.

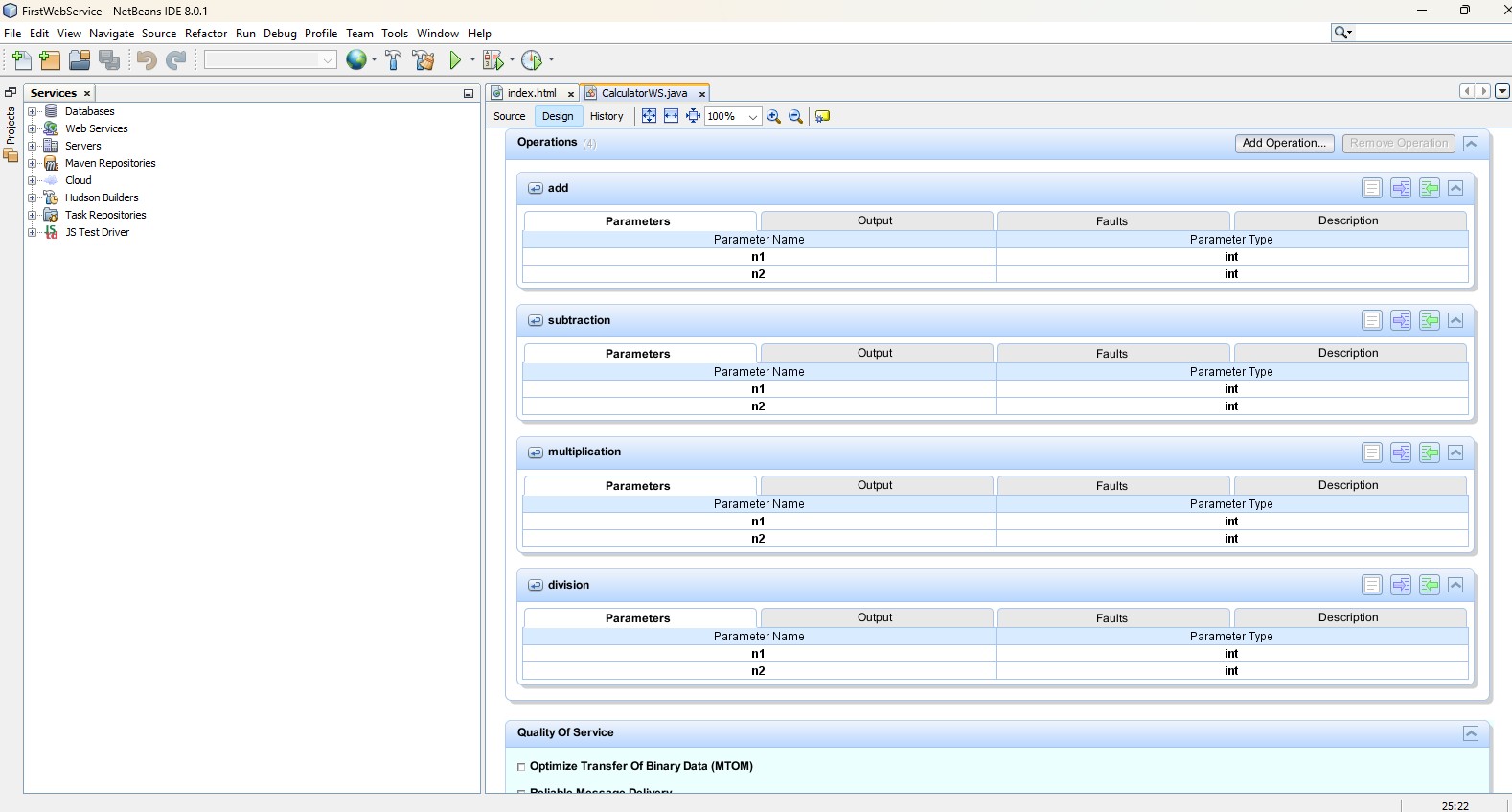


STEP4: After that Select Design option and click on Add Operation.

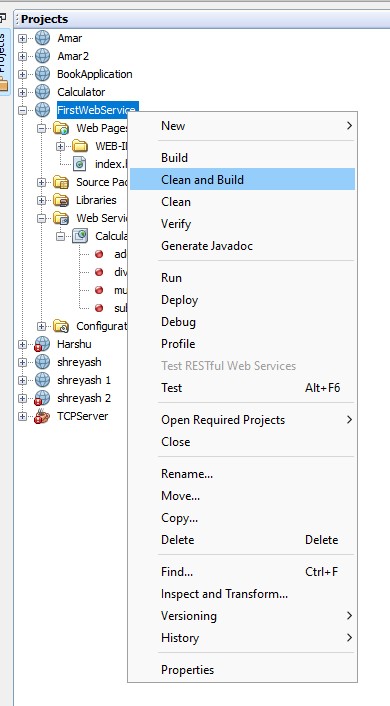


STEP5: Now add the operations(add,sub,multiply,divide).

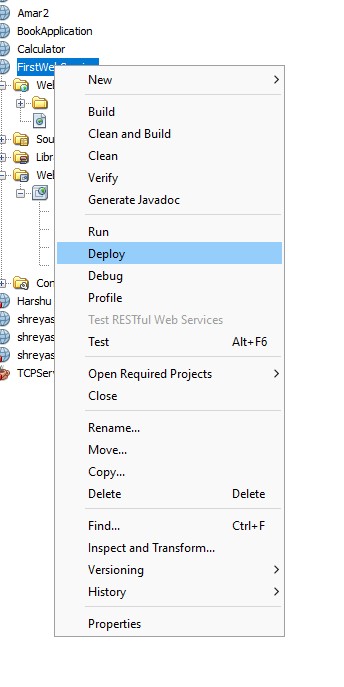


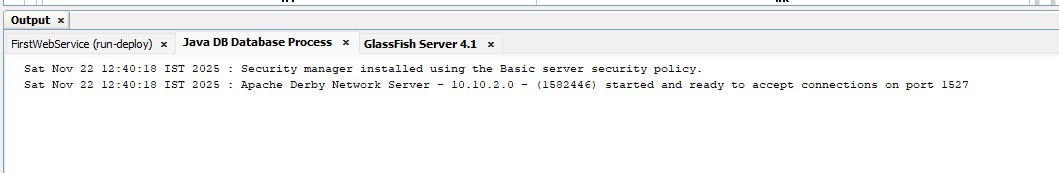


STEP6:Now Clean and Build the Web Servicve.

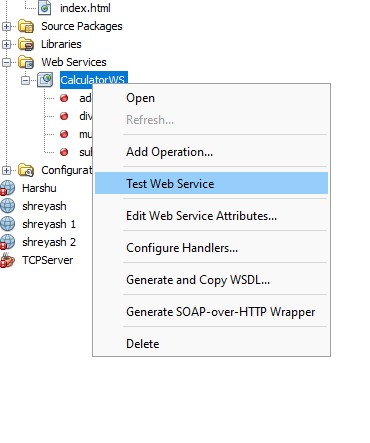


STEP7: Now Deploy the Web Service.

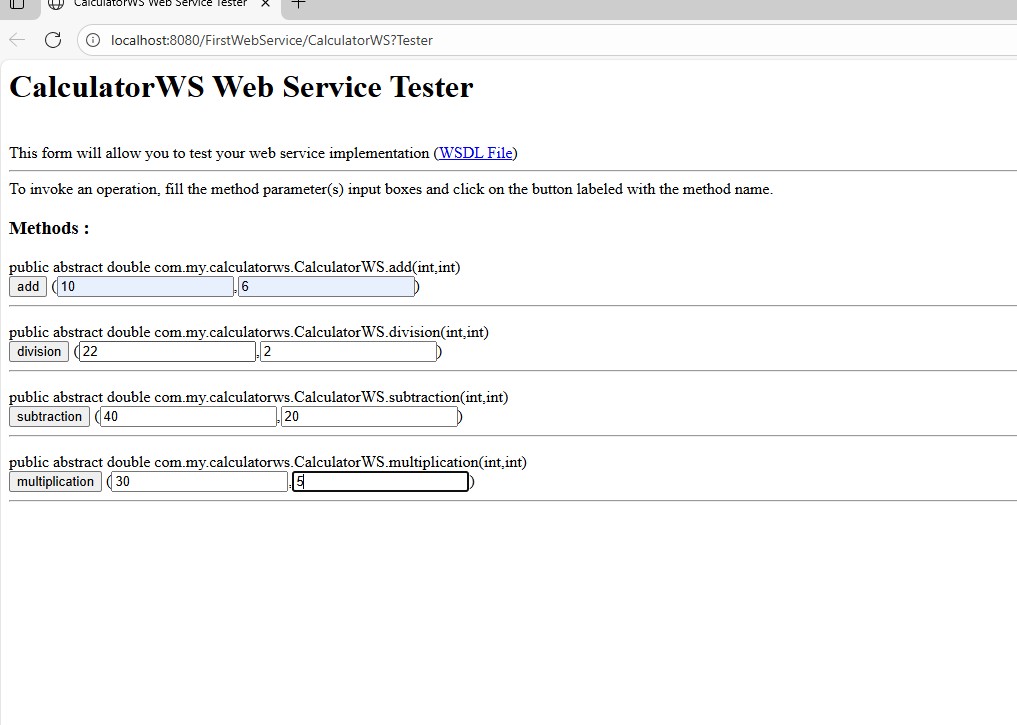


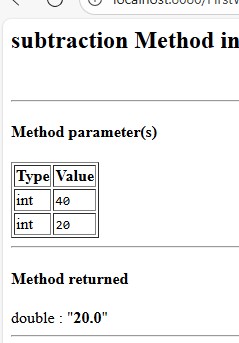
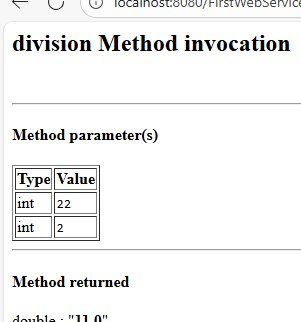
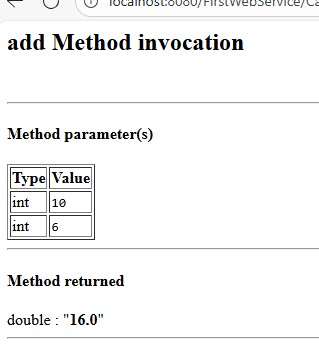


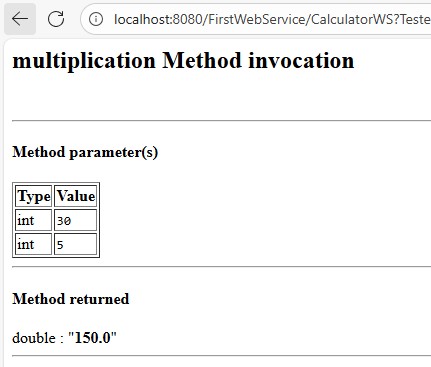
STEP8: Now Test Web Service the CalculatorWS.



**OUTPUT:**







# Practical 6

**Aim: A program that finds the square, square root, cube and cube root of the entered number. Server.java** import java.io.\*; import java.net.\*; public class Server { public static void main(String[] args) {

try {

ServerSocket serverSocket = new ServerSocket(5000);

System.out.println("Server started. Waiting for client...");

Socket socket = serverSocket.accept();

System.out.println("Client connected.");

BufferedReader in = new BufferedReader(new

InputStreamReader(socket.getInputStream()));

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

// Read number from client

String input = in.readLine(); double num = Double.parseDouble(input);

// Calculations double square = num \* num; double sqrt = Math.sqrt(num); double cube = num \* num \* num; double cbrt = Math.cbrt(num);

// Sending results back to client out.println("Square: " + square); out.println("Square Root: " + sqrt); out.println("Cube: " + cube); out.println("Cube Root: " + cbrt); socket.close(); serverSocket.close(); } catch (Exception e) {

e.printStackTrace();

}

}

} **Client.java** import java.io.\*; import java.net.\*; public class Client { public static void main(String[] args) {

try {

Socket socket = new Socket("localhost", 5000);

BufferedReader in = new BufferedReader(new

InputStreamReader(socket.getInputStream()));

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

BufferedReader userInput = new BufferedReader(new

InputStreamReader(System.in));

System.out.print("Enter a number: ");

String number = userInput.readLine();

// Send number to server out.println(number);

// Read results from server

System.out.println("\nResults from Server:");

System.out.println(in.readLine());

System.out.println(in.readLine());

System.out.println(in.readLine()); System.out.println(in.readLine()); socket.close(); } catch (Exception e) {

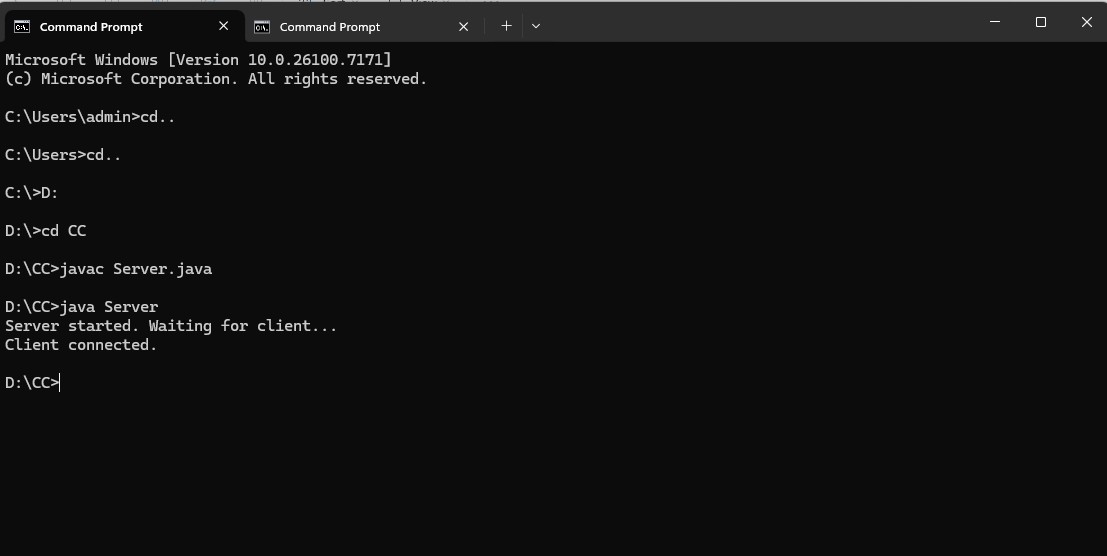
e.printStackTrace();

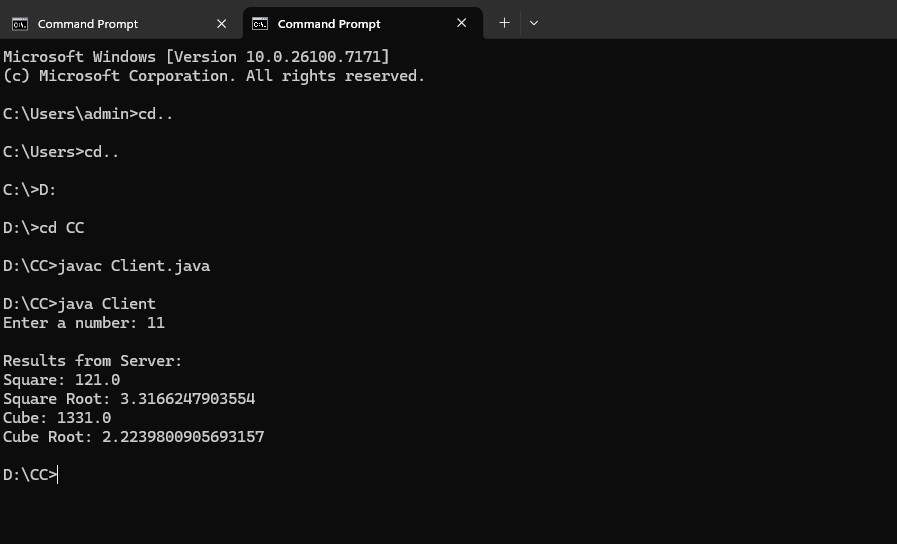
}

}

}

**Output:**





# Practical 7

**Aim : A RMI based application program to display current date and time. RemoteData.java** import java.rmi.Remote; import java.rmi.RemoteException; import java.util.Date;

public interface RemoteDate extends Remote {

Date getRemoteDate() throws RemoteException;

String LOOKUPNAME = "RemoteDateService"; // Name for RMI registry }

**RemoteDateImpl.java** import java.rmi.RemoteException; import java.rmi.server.UnicastRemoteObject; import java.util.Date;

public class RemoteDateImpl extends UnicastRemoteObject implements RemoteDate { protected RemoteDateImpl() throws RemoteException {

super();

}

@Override public Date getRemoteDate() throws RemoteException { return new Date(); // Returns current date and time on the server

}

public static void main(String[] args) {

try {

RemoteDateImpl server = new RemoteDateImpl();

java.rmi.registry.LocateRegistry.createRegistry(1099); // Start RMI registry java.rmi.Naming.rebind(RemoteDate.LOOKUPNAME, server);

System.out.println("Remote Date Server ready.");

} catch (Exception e) {

System.err.println("Remote Date Server exception: " + e.toString());

e.printStackTrace();

}

}

}

**RemoteDateClient.java** import java.rmi.Naming; import java.util.Date;

public class RemoteDateClient { public static void main(String[] args) {

try {

RemoteDate remoteDateService = (RemoteDate)

Naming.lookup("rmi://localhost/" + RemoteDate.LOOKUPNAME);

Date remoteDateTime = remoteDateService.getRemoteDate(); System.out.println("Current date and time from remote server: " + remoteDateTime);

} catch (Exception e) {

System.err.println("Remote Date Client exception: " + e.toString());

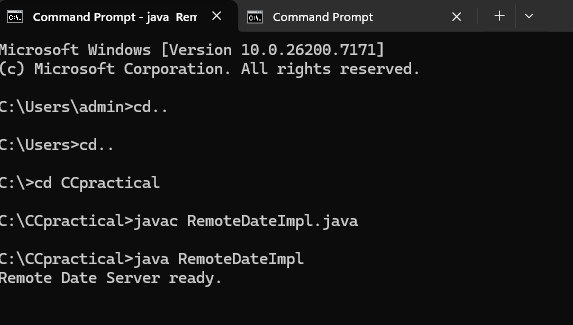
e.printStackTrace();

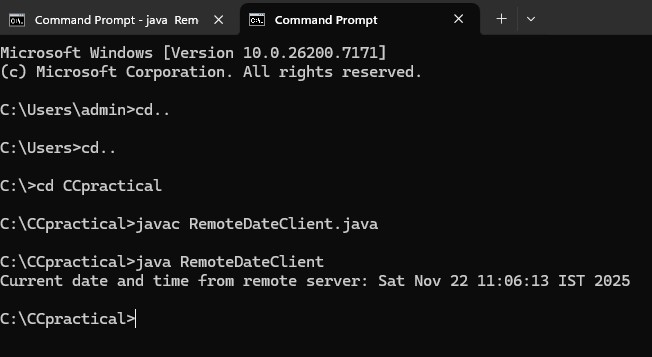
}

}

}

**Output:**





**Practical 8**

**Aim: - A RMI based application program that converts digits to words, e.g 123 will be converted to one two three. InterConvert.java** import java.rmi.\*; public interface InterConvert extends Remote

{ public String convertDigit(String no) throws Exception;

}

**ServerConvert.java** import java.rmi.\*; import java.rmi.server.\*; public class ServerConvert extends UnicastRemoteObject implements InterConvert { public ServerConvert() throws Exception

{ } public String convertDigit(String no) throws Exception

{

String str = ""; for(int i = 0; i < no.length(); i++)

{ int p = no.charAt(i); if( p == 48)

{ str += "zero "; } if( p == 49) { str += "one "; } if( p == 50) { str += "two "; } if( p == 51) { str += "three "; } if( p == 52) { str += "four "; } if( p == 53) { str += "five "; } if( p == 54) { str += "six ";

} if( p == 55) { str += "seven "; } if( p == 56) { str += "eight "; } if( p == 57) { str += "nine ";

} } return str; } public static void main(String args[]) throws

Exception {

ServerConvert s1 = new ServerConvert();

Naming.bind("Wrd",s1);

System.out.println("Object registered....");

}

}

**ClientConvert.java**

import java.rmi.\*; import java.io.\*; public class ClientConvert

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

Exception {

InterConvert h1 =

(InterConvert)Naming.lookup("Wrd"); BufferedReader br = new BufferedReader(new

InputStreamReader(System.in));

System.out.println("Enter a number: \t");

String no = br.readLine();

String ans = h1.convertDigit(no);

System.out.println("The word representation of the entered digit is : " +ans);

}

}

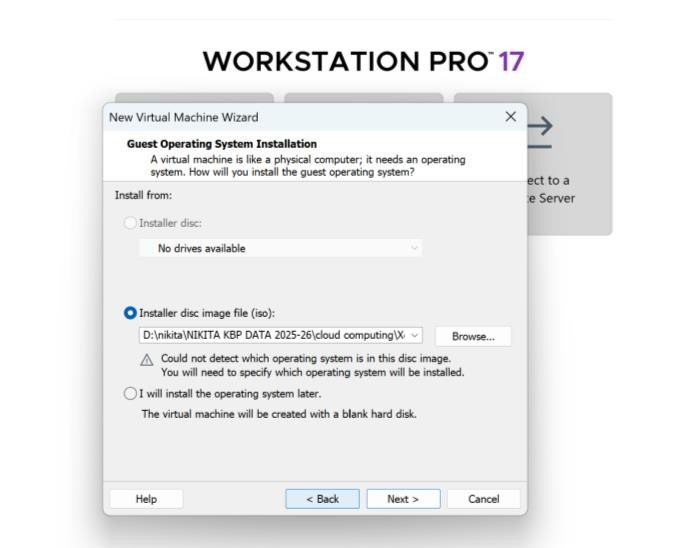


# Practical No. 9

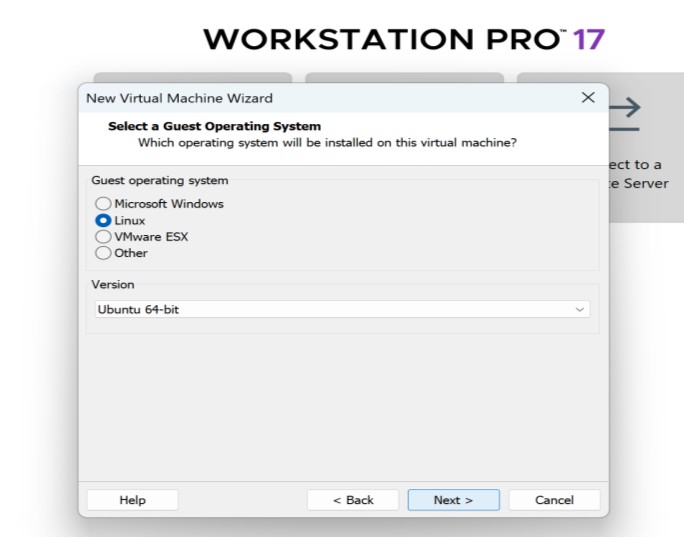
**Aim: Implement Xen virtualization and manage with XenCenter.**

**Part 1: Install the XenServer Hypervisor**

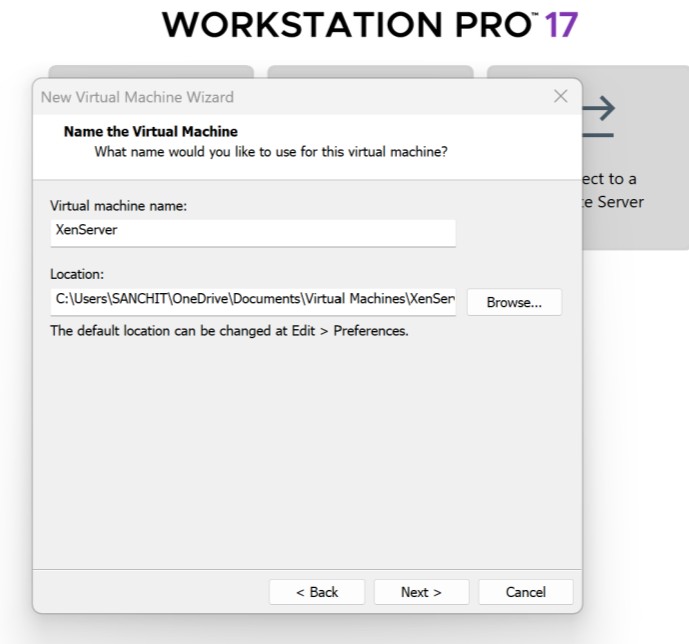
Step 1: Create new virtual machine and then browse the iso image file which you want to install.



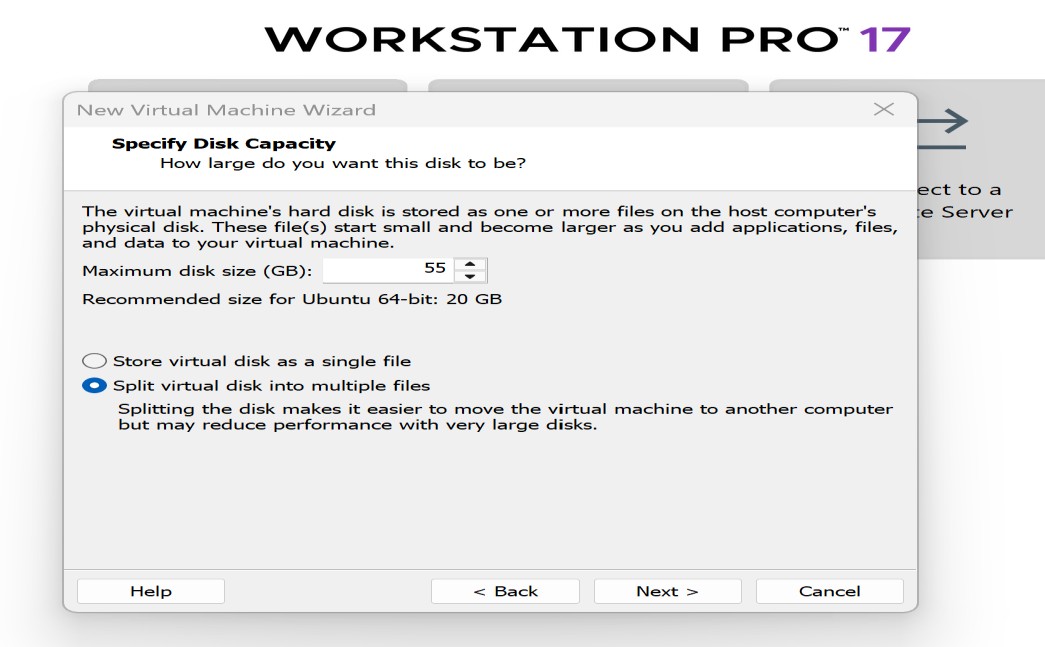
Step 2: Select guest operating system from the given option.



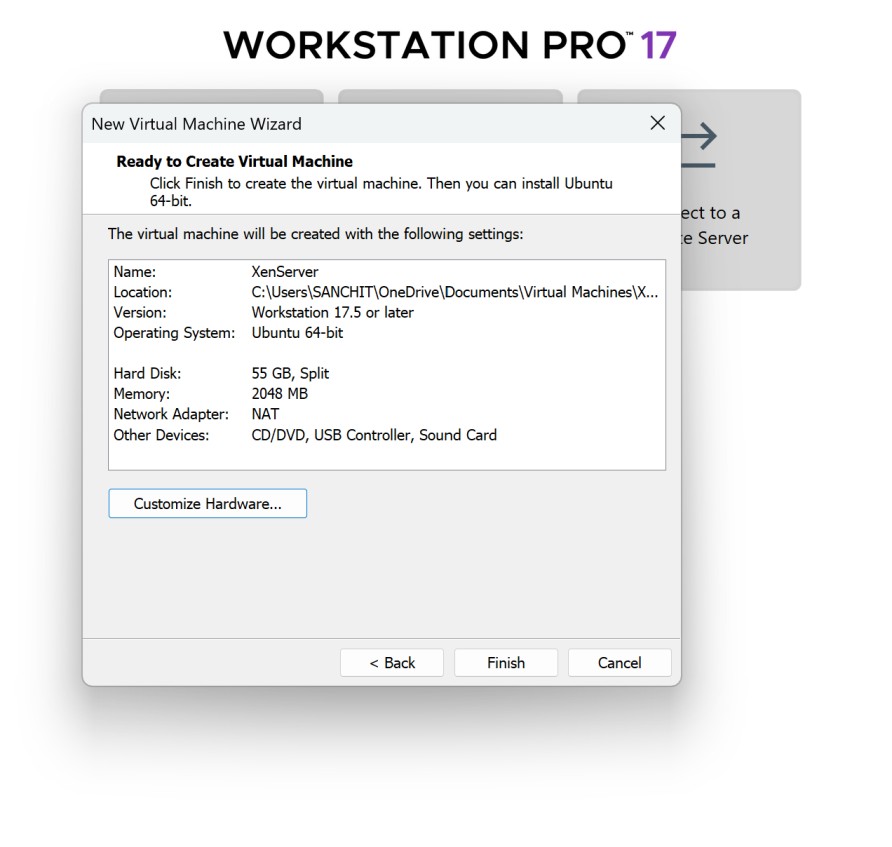
Step 3: Give the name and location to the virtual machine.



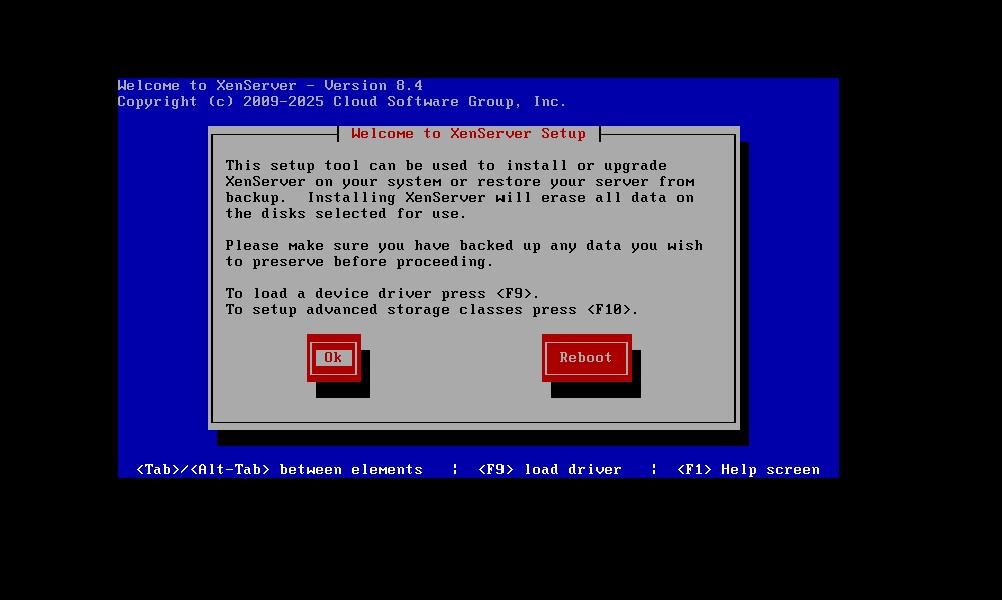
Step 4: Provide disk capacity to install virtual machine.



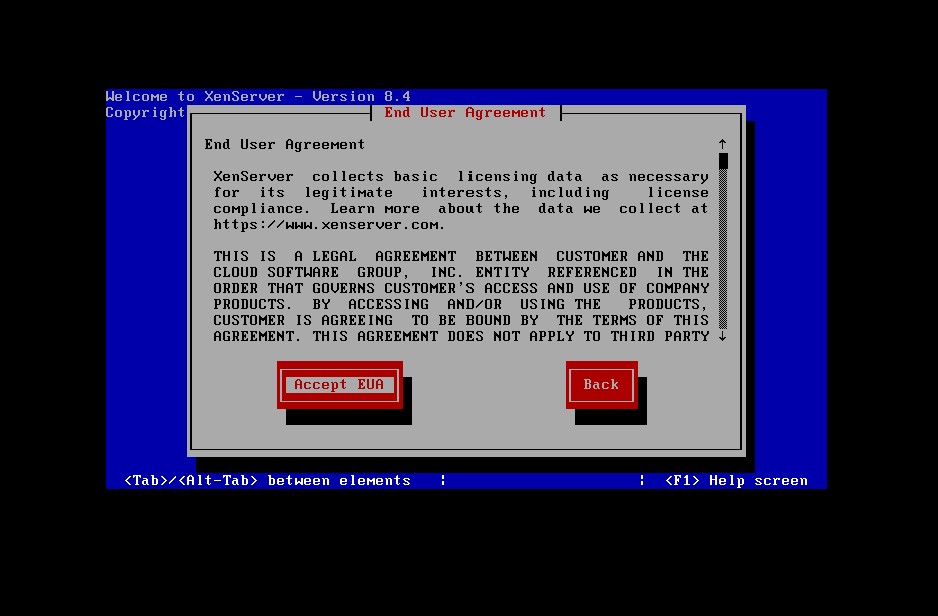
Step 5: Click on finish option for the given window.



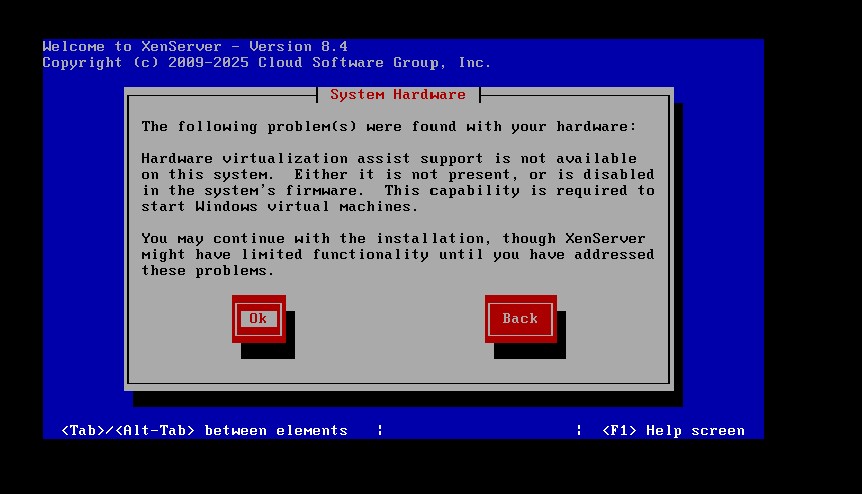
Step 6: From XenServer setup window click on ok button.

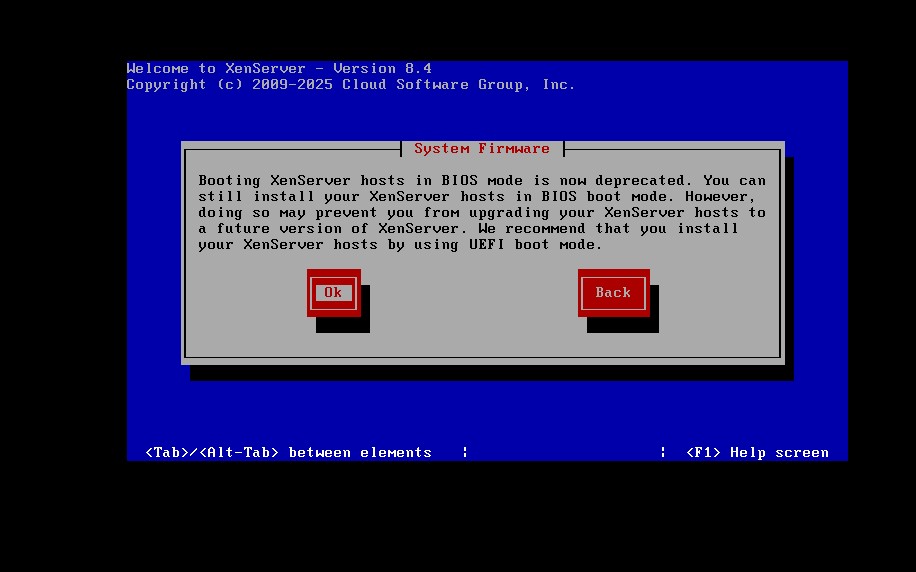


Step 7: Accept the End User License Agreement (EULA).

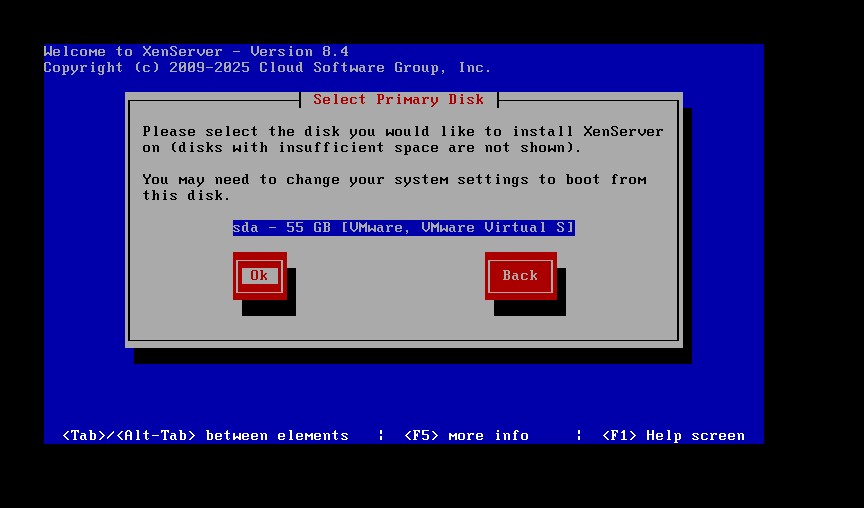


Step 8: Select the destination disk(s) for the installation.

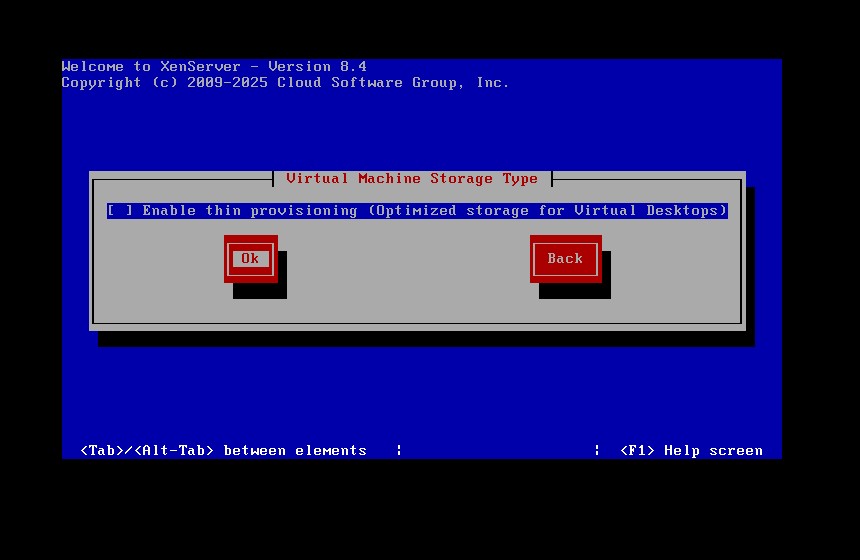


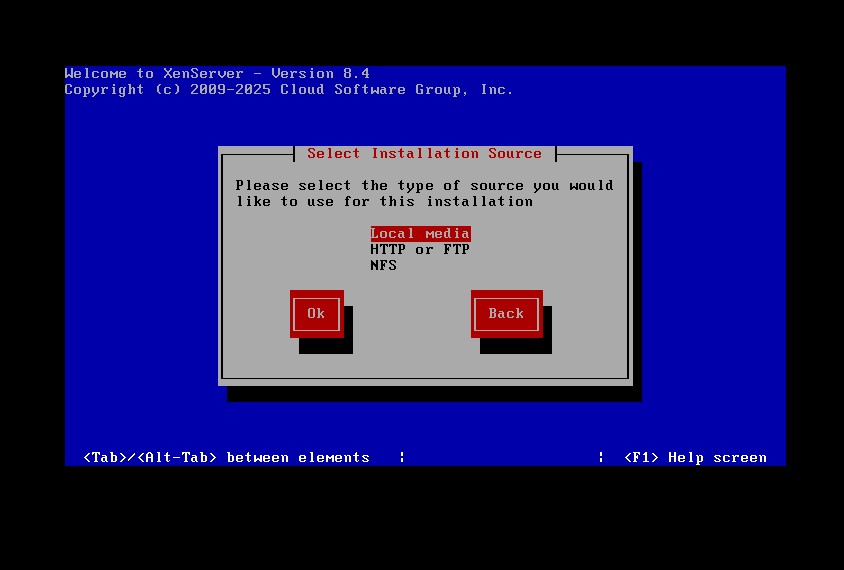


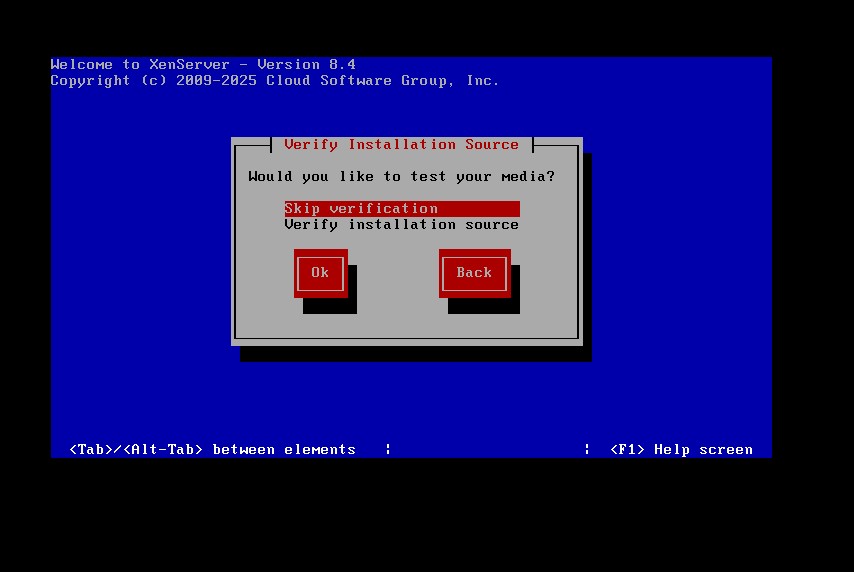
Step 9: Select appropriate installation media and choose whether to install supplemental packs.



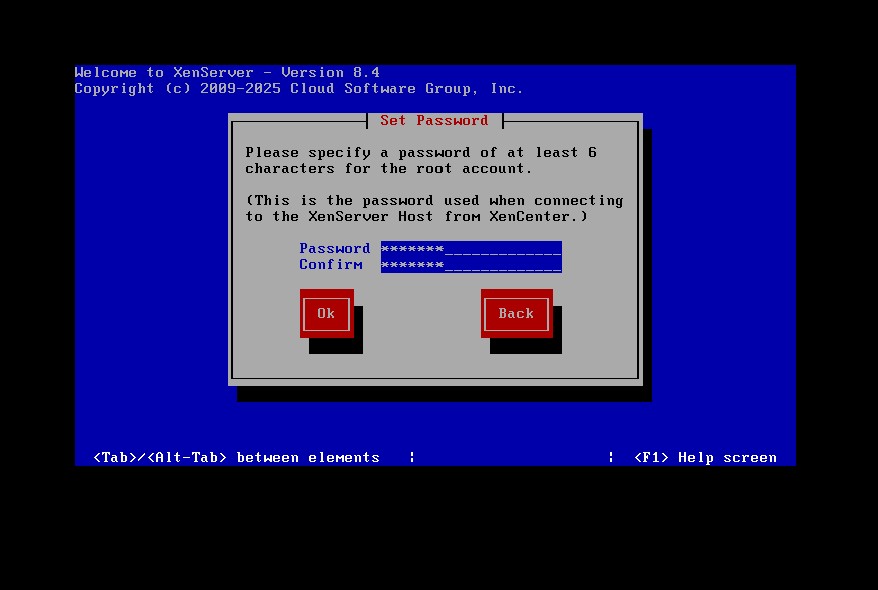




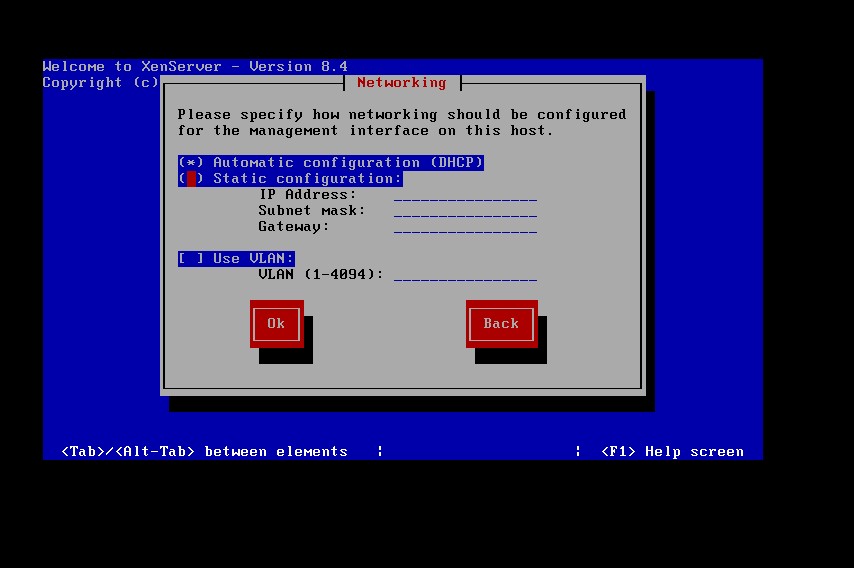


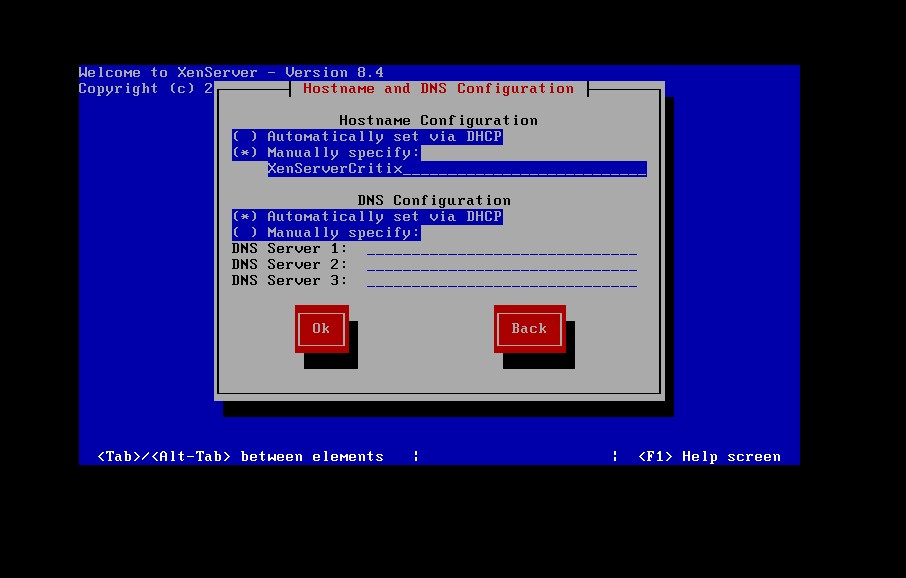


Step 10: Specify and confirm a root123 password for the host.

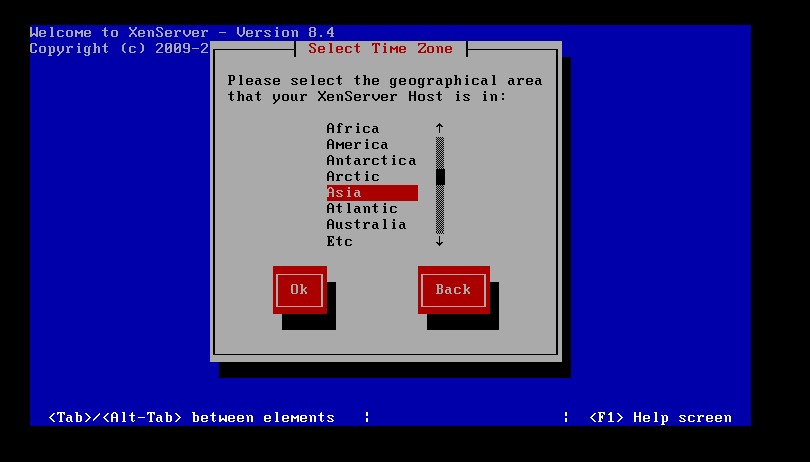


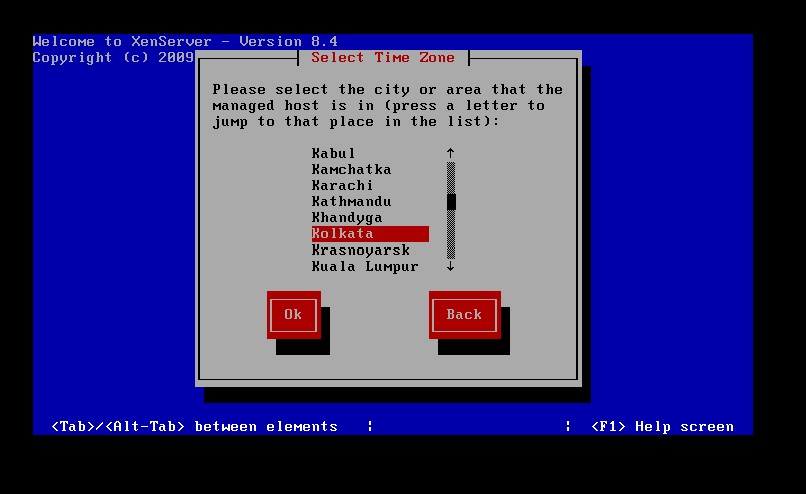
Step 11: Configure the management network interface, setting its IP address (statically or via DHCP), hostname, and DNS server details.

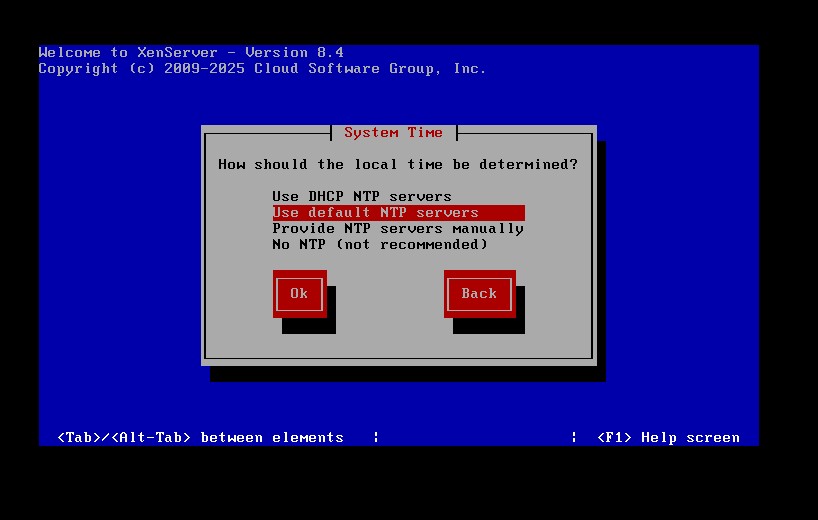




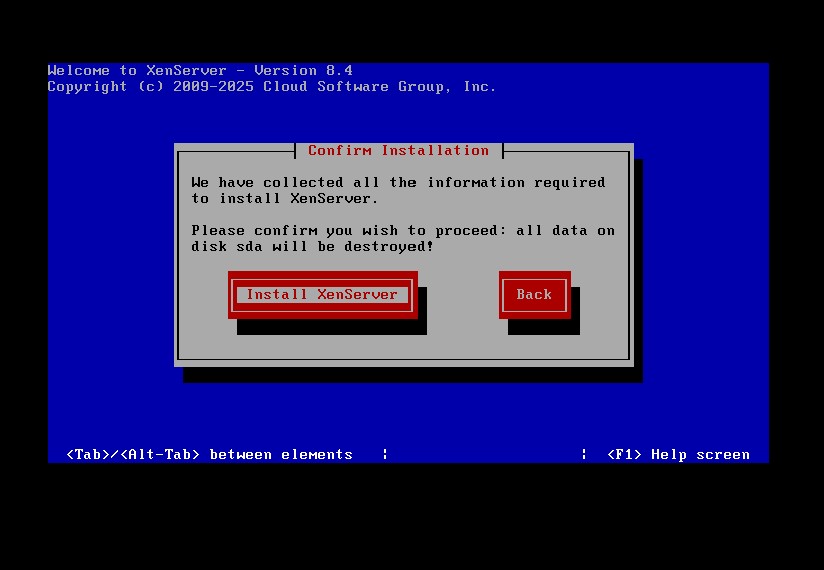
Step 12: Select your time zone and configure NTP servers for time synchronization.

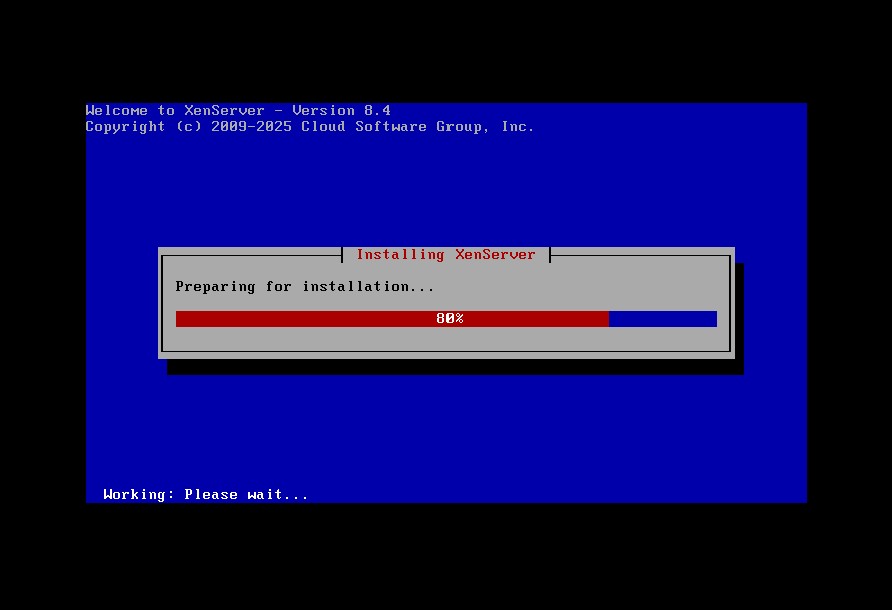


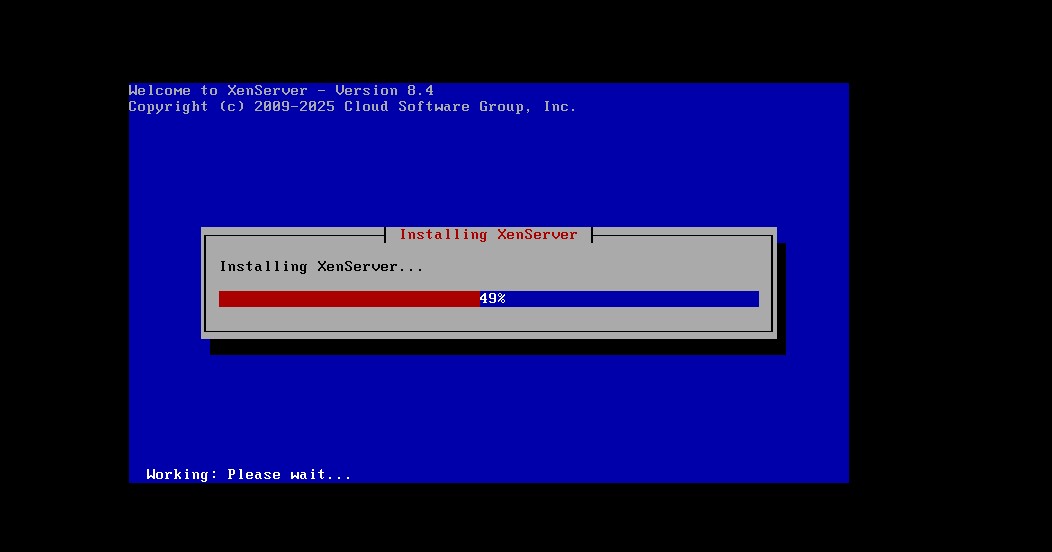


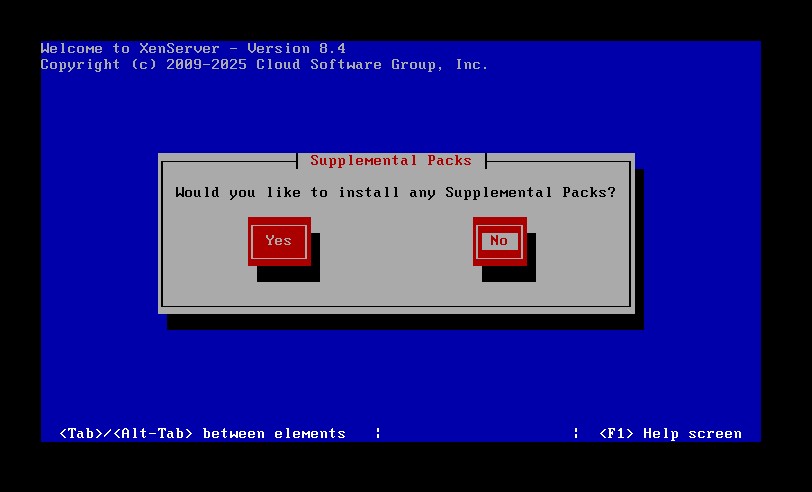


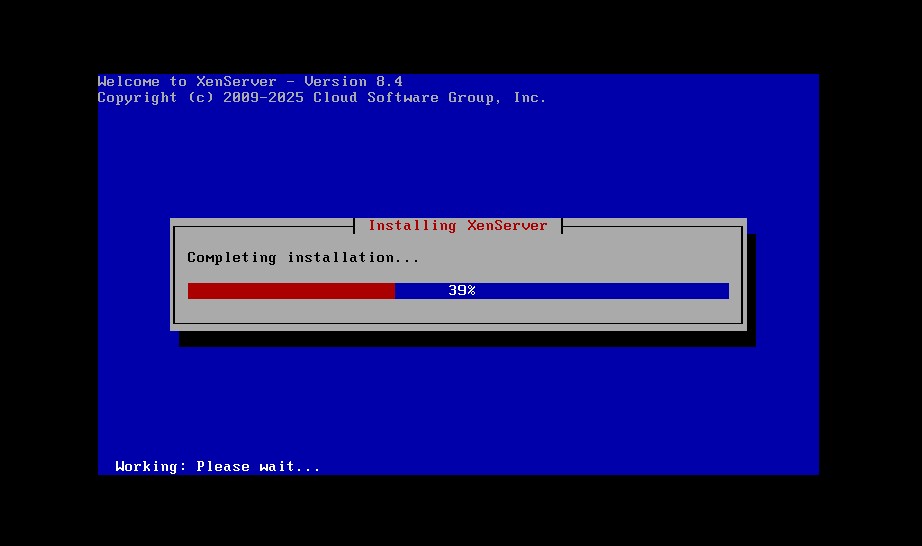
Step 13: Start the installation process.







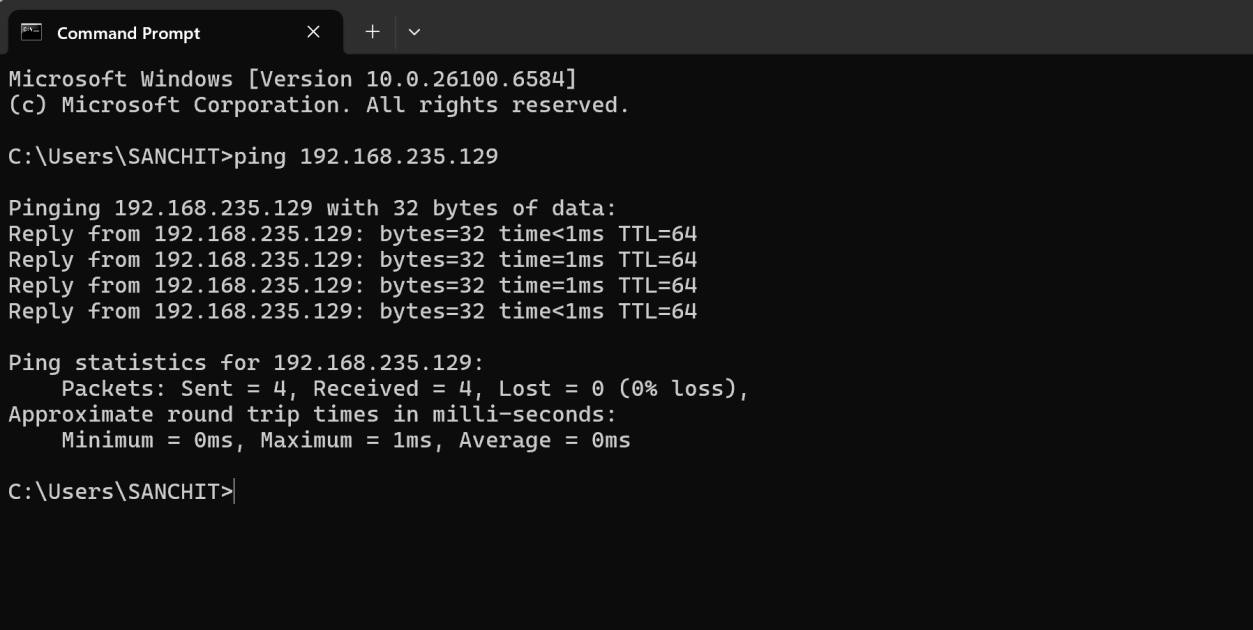




Step 13: Reboot - Once the installation is complete, remove the installation media and reboot the server.





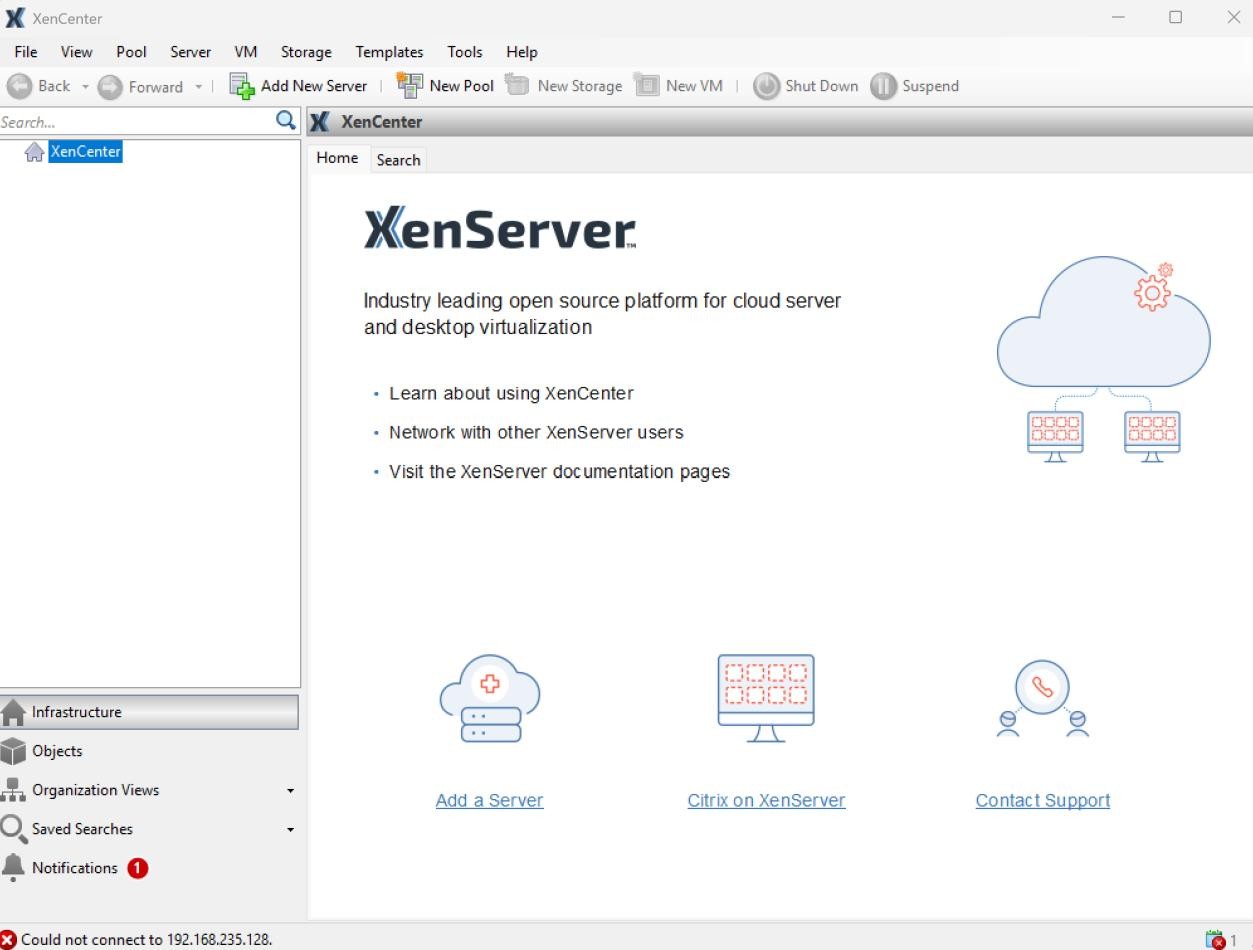


**Part 2: Install and Connect XenCenter**

Step 1: Install XenCenter: On your Windows management machine, run the XenCenter installer executable you downloaded from the website. Follow the installation wizard prompts to complete the setup.

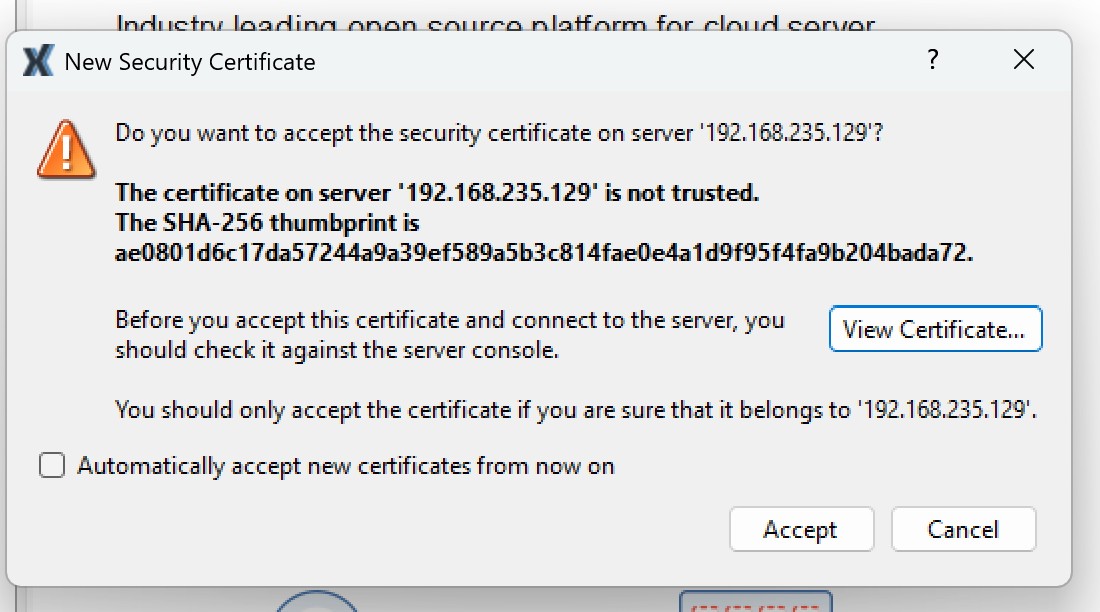
Step 2: Launch XenCenter: Open the XenCenter application after installation.

Step 3: Add New Server: In the XenCenter interface, click on the Add New Server icon or menu option.



Step 4: Enter Connection Details:

1. In the Server field, type the IP address (or hostname) you assigned to the XenServer host during installation.
2. Enter the root username (default).
3. Enter the root password you created during the XenServer installation.



Step 5: Connect: Click Add to connect XenCenter to the XenServer host. The server will now appear in the left-hand pane of the XenCenter console.



**Practical No. 10 Aim: Implement windows Hyper V Virtualization.**

Step 1. **Check if virtualization is enabled:**

* Search for Command Prompt in Windows Start Menu and open it.
* Type systeminfo and press Enter. Wait for the process to finish
* Once the results appear, search for the Hyper-V Requirements section which is usually the last one.
* If it says A hypervisor has been detected. Features required for Hyper-V will not be displayed. that means Hyper-V is already enabled.

**Enable Hyper-V**

* Open File Explorer on windows and Navigate to a folder.
* Right click anywhere in a blank space inside the folder. Select New and then Click Text Document.
* Open the file in Notepad and copy and paste the following text into it. pushd "%~dp0" dir /b %SystemRoot%\servicing\Packages\\*Hyper-V\*.mum >hyper-v.txt

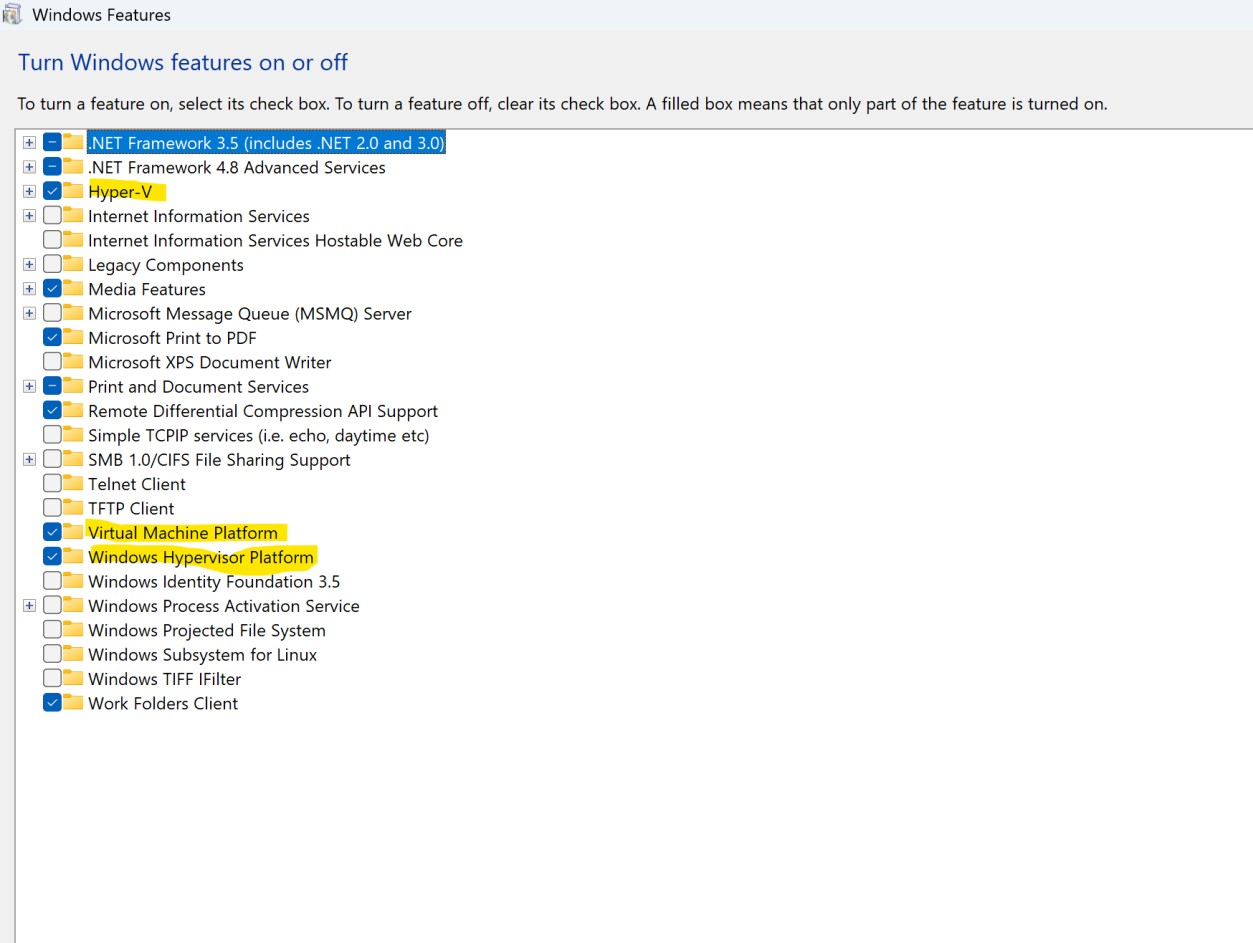
for /f %%i in ('findstr /i . hyper-v.txt 2^>nul') do dism /online /norestart

/add-package:"%SystemRoot%\servicing\Packages\%%i" del hyper-v.txt

Dism /online /enable-feature /featurename:Microsoft-Hyper-V -All

/LimitAccess /ALL pause

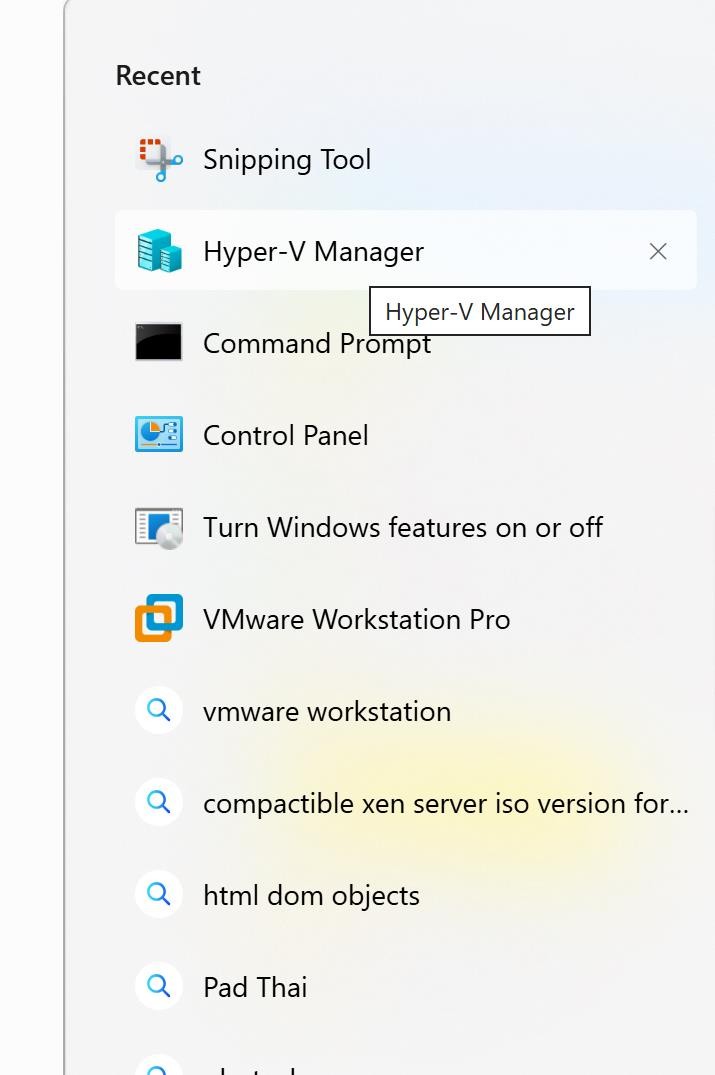
* Click File from the Menu bar in the top of Notepad, then click Save as.... In the Save as Window that appears, Change the File name to "HyperV.bat" and click save.
* Now Double click Hyper-V.bat to run it. This will take some time but will install all features required for Hyper-V. A Restart might be required after it is done.
* After Restarting Windows, search for Turn Windows features on or off in the Start Menu search bar and open it.

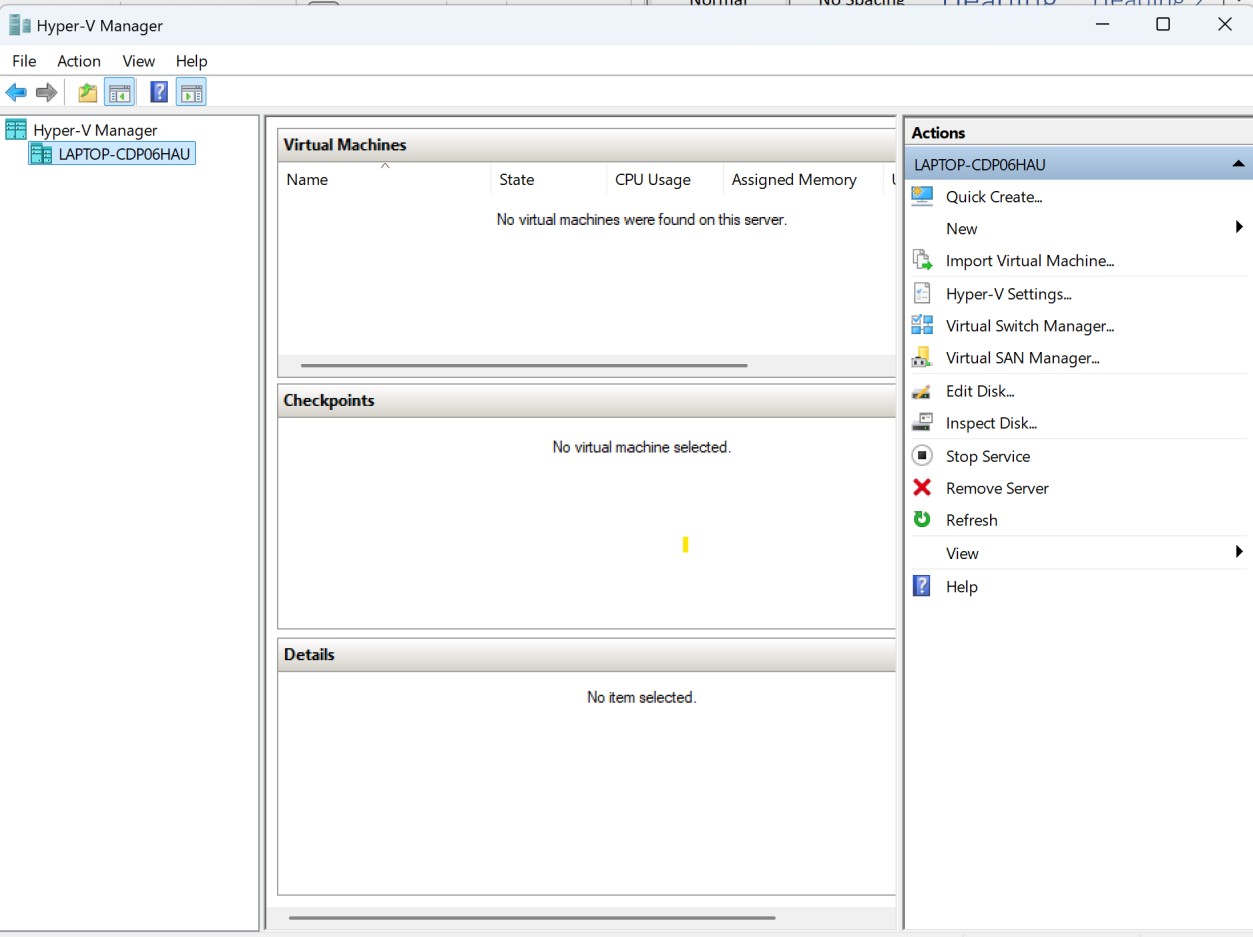


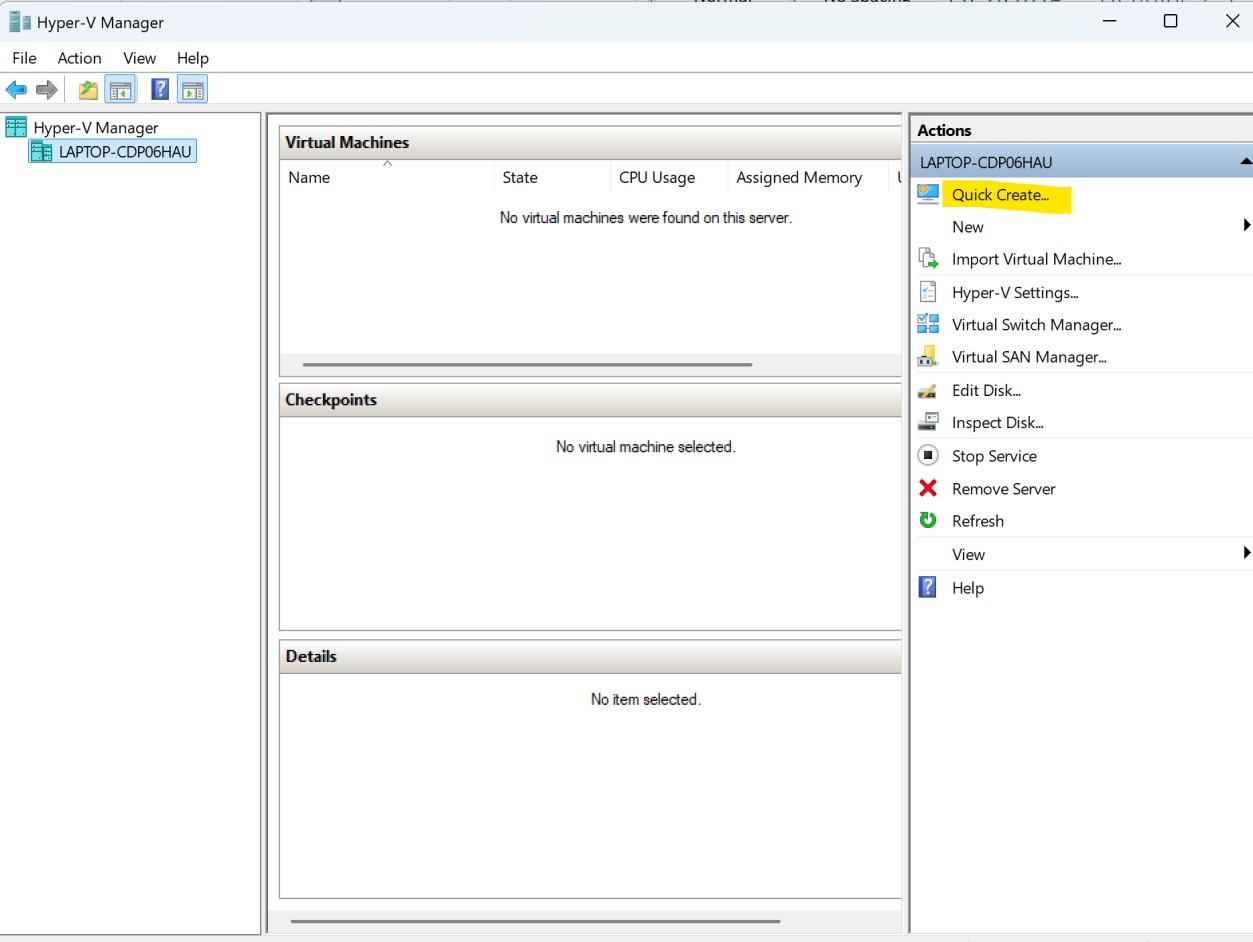
* In the Window, lookout for Hyper-V, Virtual Machine

Platform and Windows Hypervisor Platform. Then check the check boxes before them and click OK. This will also take some time and then a Restart is necessary.

Thats it Hyper-V is enabled.









Then click on create virtual machine option.