

PRACTICAL 1

AIM :- WRITE A PROGRAM FOR IMPLEMENTING CLIENT SERVER COMMUNICATION MODEL USING TCP.

PRACTICAL 1A :-

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

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. tcpClientPrime.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 :-

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java tcpServerPrime
Server Started.....
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java tcpClientPrime
Enter a number :
7
7 is Prime
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java tcpServerPrime
Server Started.....
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java tcpClientPrime
Enter a number :
10
10 is not Prime
```

PRACTICAL 1B :-**A client server TCP based chatting application.****Code :-****1. ChatServer.java**

```
import java.net.*;
import java.io.*;
class ChatServer
{
    public static void main(String args[])
    {
        try
        {
            ServerSocket ss = new ServerSocket(8000);
            System.out.println("Waiting for client to connect..");
            Socket s = ss.accept();
            BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
            DataOutputStream out = new DataOutputStream(s.getOutputStream());
            DataInputStream in = new DataInputStream(s.getInputStream());
            String receive, send;
            while((receive = in.readLine()) != null)
            {
                if(receive.equals("STOP"))
                    break;
                System.out.println("Client Says : "+receive);
                System.out.print("Server Says : ");
                send = br.readLine();
                out.writeBytes(send+"\n");
            }
            br.close();
            in.close();
            out.close();
            s.close();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

2. ChatClient.java

```
import java.net.*;
import java.io.*;
class ChatClient
{
    public static void main(String args[])
    {
        try
        {
            Socket s = new Socket("Localhost",8000);
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            DataOutputStream out = new DataOutputStream(s.getOutputStream());
            DataInputStream in = new DataInputStream(s.getInputStream());
            String msg;
            System.out.println("To stop chatting with server type STOP");
            System.out.print("Client Says: ");
            while((msg = br.readLine()) != null)
            {
                out.writeBytes(msg+"\n");
                if(msg.equals("STOP"))
                    break;
                System.out.println("Server Says : "+in.readLine());
                System.out.print("Client Says : ");
            }
            br.close();
            in.close();
            out.close();
            s.close();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Output :-

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java ChatServer
Waiting for client to connect..
Client Says : Hii
Server Says : Hello
Client Says : How are you ?
Server Says : I am Fine.
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java ChatClient
To stop chatting with server type STOP
Client Says: Hii
Server Says : Hello
Client Says : How are you ?
Server Says : I am Fine.
Client Says : STOP
```

PRACTICAL 02**AIM :- WRITE A PROGRAM FOR IMPLEMENTING CLIENT SERVER COMMUNICATION MODEL USING UDP.****PRACTICAL 2A :-**

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

Code :-

1. udpServerEO.java

```
/*Program which finds entered number is even or odd */
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 :-

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac udpServerE0.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java udpServerE0  
12  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac udpServerE0.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java udpServerE0  
19
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac udpClientE0.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java udpClientE0  
Enter a number :  
12  
Number is even  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac udpClientE0.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java udpClientE0  
Enter a number :  
19  
Number is odd
```

PRACTICAL 2B :-

A client server based program using UDP to find the factorial of the entered number.

Code :-

1. udpServerFact.java

```
/*Program which calculate factorial of a number*/
import java.io.*;
import java.net.*;
public class udpServerFact
{
    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);
            int f = 1, i;
            String s= new String();
            for(i=1;i<=a;i++)
            {
                f=f*i;
            }
            s=Integer.toString(f);
            String str1 = "The Factorial of " + str + " is : " +
f; byte b1[] = new byte[1024]; b1 =
str1.getBytes();
            DatagramPacket dp1 = new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000);
            ds.send(dp1);
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

2. udpClientFact.java

```
/*Program which calculate factorial of a number*/
import java.io.*;
import java.net.*;
public class udpClientFact
{
    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 :-

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac udpServerFact.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java udpServerFact  
9  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java udpServerFact  
11
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac udpClientFact.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java udpClientFact  
Enter a number :  
9  
The Factorial of 9 is : 362880  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java udpClientFact  
Enter a number :  
11  
The Factorial of 11 is : 39916800
```

PRACTICAL 2C :-

A program to implement simple calculator operations like addition, subtraction, multiplication and division.

Code :-

1. RPCServer.java

```

import java.util.*;
import java.net.*;
class RPCServer
{
    DatagramSocket ds;
    DatagramPacket dp;
    String str,methodName,result;
    int val1,val2;
    RPCServer()
    {
        try
        {
            ds=new DatagramSocket(1200);
            byte b[]={};byte[4096];
            while(true)
            {
                dp=new DatagramPacket(b,b.length);
                ds.receive(dp);
                str=new String(dp.getData(),0,dp.getLength());
                if(str.equalsIgnoreCase("q"))
                {
                    System.exit(1);
                }
                else
                {
                    StringTokenizer st = new StringTokenizer(str, " ");
                    int i=0;
                    while(st.hasMoreTokens())
                    {
                        String token=st.nextToken();
                        methodName=token;
                        val1 = Integer.parseInt(st.nextToken());
                        val2 = Integer.parseInt(st.nextToken());
                    }
                }
                System.out.println(str);
                InetAddress ia = InetAddress.getLocalHost();
                if(methodName.equalsIgnoreCase("add"))
                {
                    result= "" + add(val1,val2);
                }
            }
        }
    }
}

```

```
else if(methodName.equalsIgnoreCase("sub"))
{
    result= "" + sub(val1,val2);
}
else if(methodName.equalsIgnoreCase("mul"))
{
    result= "" + mul(val1,val2);
}
else if(methodName.equalsIgnoreCase("div"))
{
    result= "" + div(val1,val2);
}
byte b1[]=result.getBytes();
DatagramSocket ds1 = new DatagramSocket();

DatagramPacket dp1 = new
    DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),
1300); System.out.println("result : "+result+"\n"); ds1.send(dp1);
}

public int add(int val1, int val2)
{
    return val1+val2;
}
public int sub(int val3, int val4)
{
    return val3-val4;
}
public int mul(int val3, int val4)
{
    return val3*val4;
}
public int div(int val3, int val4)
{
    return val3/val4;
}
public static void main(String[] args)
{
    new RPCServer();
}
```

2. RPCClient.java

```
import java.io.*;
import java.net.*;
class RPCClient
{
    RPCClient()
    {
        try
        {
            InetAddress ia = InetAddress.getLocalHost();
            DatagramSocket ds = new DatagramSocket();
            DatagramSocket ds1 = new DatagramSocket(1300);
            System.out.println("\nRPC Client\n");
            System.out.println("Enter method name and parameter like add 3
4\n");
            while (true)
            {
                BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
                String str = br.readLine();
                byte b[] = str.getBytes();
                DatagramPacket dp = new
DatagramPacket(b,b.length,ia,1200);
                ds.send(dp);
                dp = new DatagramPacket(b,b.length);
                ds1.receive(dp);
                String s = new String(dp.getData(),0,dp.getLength());
                System.out.println("\nResult = " + s + "\n");
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
    public static void main(String[] args)
    {
        new RPCClient();
    }
}
```

Output :-

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac RPCServer.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java RPCServer  
add 3 4  
result : 7  
  
sub 10 8  
result : 2  
  
mul 3 10  
result : 30  
  
div 22 2  
result : 11
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac RPCClient.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java RPCClient  
  
RPC Client  
  
Enter method name and parameter like add 3 4  
  
add 3 4  
  
Result = 7  
  
sub 10 8  
  
Result = 2  
  
mul 3 10  
  
Result = 30  
  
div 22 2  
  
Result = 11  
.
```

PRACTICAL 2D :-

A program that finds the square, square root, cube and cube root of the entered number.

Code :-

1. RPCNumServer.java

```
import java.util.*;
import java.net.*;
import java.io.*;
class RPCNumServer
{
    DatagramSocket ds;
    DatagramPacket dp;
    String str,methodName,result;
    int val;
    RPCNumServer()
    {
        try
        {
            ds=new DatagramSocket(1200);
            byte b[]=new byte[4096];
            while(true)
            {
                dp=new DatagramPacket(b,b.length);
                ds.receive(dp);
                str=new String(dp.getData(),0,dp.getLength());
                if(str.equalsIgnoreCase("q"))
                {
                    System.exit(1);
                }
                else
                {
                    StringTokenizer st = new StringTokenizer(str, " ");
                    int i=0;
                    while(st.hasMoreTokens())
                    {
                        String token=st.nextToken();
                        methodName=token;
                        val = Integer.parseInt(st.nextToken());
                    }
                }
                System.out.println(str);
                InetAddress ia = InetAddress.getLocalHost();
                if(methodName.equalsIgnoreCase("square"))
                {
                    result= "" + square(val);
                }
                else if(methodName.equalsIgnoreCase("squareroot"))
                {
```

```
        result= "" + squareroot(val);
    }
    else if(methodName.equalsIgnoreCase("cube"))
    {
        result= "" + cube(val);
    }
    else if(methodName.equalsIgnoreCase("cuberoott"))
    {
        result= "" + cuberoott(val);
    }
    byte b1[]=result.getBytes();
    DatagramSocket ds1 = new DatagramSocket();
    DatagramPacket dp1 = new
    DatagramPacket(b1,b1.length,InetAddress.getLocalHost(), 1300);
    System.out.println("result : "+result+"\n"); ds1.send(dp1);
}
catch (Exception e)
{
    e.printStackTrace();
}
}
public double square(int a) throws Exception
{
    double ans;
    ans = a*a;
    return ans;
}
public double squareroot(int a) throws Exception
{
    double ans;
    ans = Math.sqrt(a);
    return ans;
}
public double cube(int a) throws Exception
{
    double ans;
    ans = a*a*a;
    return ans;
}
public double cuberoott(int a) throws Exception
{
    double ans;
    ans = Math.cbrt(a);
    return ans;
}
public static void main(String[] args)
{
    new RPCNumServer();
}
```

```
}
```

2. RPCNumClient.java

```
import java.io.*;
import java.net.*;
class RPCNumClient
{
RPCNumClient()
{
try
{
InetAddress ia = InetAddress.getLocalHost(); DatagramSocket ds = new DatagramSocket();
DatagramSocket ds1 = new DatagramSocket(1300); System.out.println("\nRPC Client\n");
System.out.println("1. Square of the number - square\n2. Square root of the number - squareroot\n3. Cube of the number - cube\n4. Cube root of the number - cuberoot");
System.out.println("Enter method name and the number\n");
while (true)
{
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
String str = br.readLine();
byte b[] = str.getBytes();
DatagramPacket dp = new
DatagramPacket(b,b.length,ia,1200);
ds.send(dp);
dp = new DatagramPacket(b,b.length);
ds1.receive(dp);
String s = new String(dp.getData(),0,dp.getLength()); System.out.println("\nResult = " + s +
"\n");
}
}
catch (Exception e)
{
e.printStackTrace();
}
}
public static void main(String[] args)
{
new RPCNumClient();
}
```

Output :-

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac RPCNumServer.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java RPCNumServer  
square 7  
result : 49.0  
  
squareroot 25  
result : 5.0  
  
cube 3  
result : 27.0  
  
cuberoot 27  
result : 3.0  
|
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac RPCNumClient.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java RPCNumClient  
RPC Client  
  
1. Square of the number - square  
2. Square root of the number - squareroot  
3. Cube of the number - cube  
4. Cube root of the number - cuberoot  
Enter method name and the number  
  
square 7  
  
Result = 49.0  
  
squareroot 25  
  
Result = 5.0  
  
cube 3  
  
Result = 27.0  
  
cuberoot 27  
  
Result = 3.0
```

PRACTICAL 03

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;
        // set the multicast address to your local subnet
        address = InetAddress.getByName("239.1.2.3");
        socket = new MulticastSocket();
        // join a Multicast group and send the group messages
        socket.joinGroup(address);
        byte[] data = null;
        for(;;)
        {
            Thread.sleep(10000);
            System.out.println("Sending ");
            String str = ("This is Pushpa Calling... ");
            data = str.getBytes();
            packet = new DatagramPacket(data, str.length(),address,PORT);
            // Sends the packet
            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;
        // set the multicast address to your local subnet
        address = InetAddress.getByName("239.1.2.3");
        socket = new MulticastSocket(PORT);
        //join a Multicast group and wait for a message
        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 :-

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac BroadcastServer.java
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java BroadcastServer
Sending
Sending
Sending
Sending
Sending
Sending
Sending
Sending
Sending
|
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac BroadcastClient.java
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java BroadcastClient
Message received from /127.0.0.1Message is : This is Pushpa Calling....
Message received from /127.0.0.1Message is : This is Pushpa Calling....
Message received from /127.0.0.1Message is : This is Pushpa Calling....
Message received from /127.0.0.1Message is : This is Pushpa Calling....
Message received from /127.0.0.1Message is : This is Pushpa Calling....
Message received from /127.0.0.1Message is : This is Pushpa Calling....
Message received from /127.0.0.1Message is : This is Pushpa Calling....
Message received from /127.0.0.1Message is : This is Pushpa Calling....
Message received from /127.0.0.1Message is : This is Pushpa Calling....
Message received from /127.0.0.1Message is : This is Pushpa Calling....
|
```

PRACTICAL 04**AIM :- WRITE A PROGRAM TO SHOW THE OBJECT COMMUNICATION USING RMI.****PRACTICAL 4A :-**

A RMI based application program to display current date and time.

Code :-

1. InterDate.java

```
import java.rmi.*;
public interface InterDate extends Remote
{
    public String display() throws Exception;
}
```

2. ServerDate.java

```
import java.rmi.*;
import java.rmi.server.*;
import java.util.*;
public class ServerDate extends UnicastRemoteObject implements
InterDate {
    public ServerDate() throws Exception
    {
    }
    public String display() throws Exception
    {
        String str = "";
        Date d = new Date();
        str = d.toString();
        return str;
    }
    public static void main(String args[]) throws
Exception {
    ServerDate s1 = new ServerDate();
    Naming.bind("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 :-

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac ServerDate.java
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac ClientDate.java
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>rmic ServerDate
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>rmiregistry
|
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac ServerDate.java
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java ServerDate
Object registered.....
|
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac ClientDate.java
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java ClientDate
Sat Jan 14 00:10:09 IST 2023
|
```

PRACTICAL 4B :-

A RMI based application program that converts digits to words, e.g. 123 will be converted to one two three.

Code :-

1. InterConvert.java

```
import java.rmi.*;
public interface InterConvert extends Remote
{
    public String convertDigit(String no) throws Exception;
}
```

2. 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... ");
}
```

3. 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 : ");
String no = br.readLine();
String ans = h1.convertDigit(no);
System.out.println("The word representation of the entered digit is : " +ans);
}
}
```

Output :-

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac ServerConvert.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac ClientConvert.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>rmic ServerConvert  
Warning: generation and use of skeletons and static stubs for JRMP  
is deprecated. Skeletons are unnecessary, and static stubs have  
been superseded by dynamically generated stubs. Users are  
encouraged to migrate away from using rmic to generate skeletons and static  
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>rmiregistry  
|
```

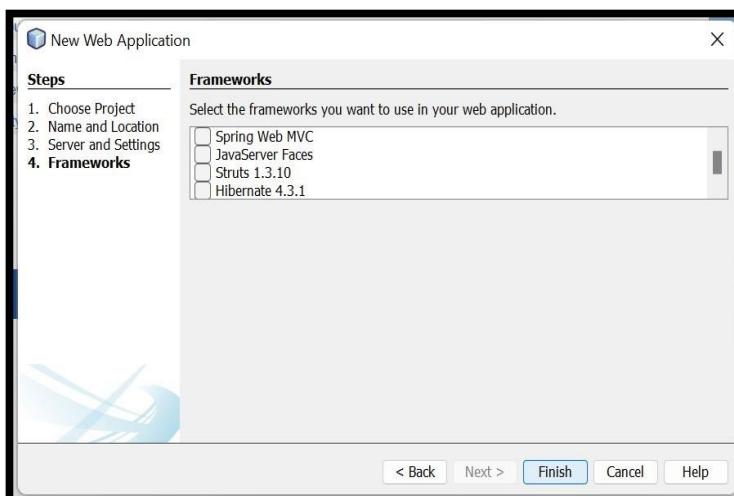
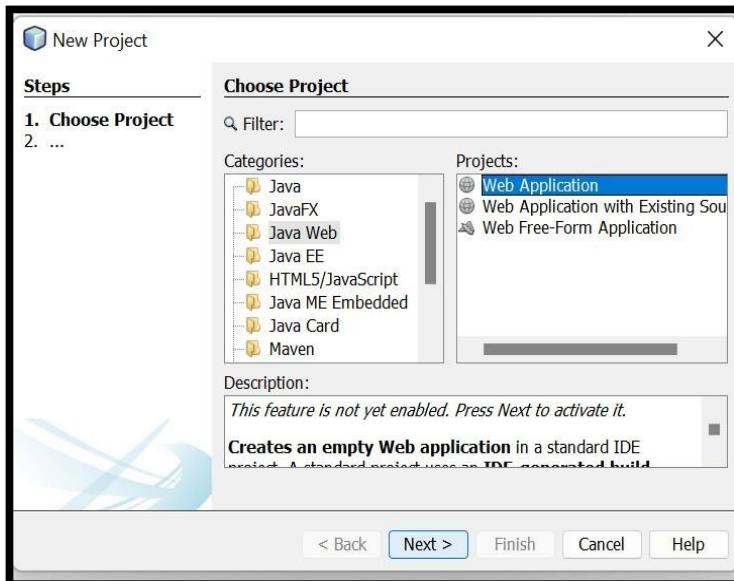
```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac ServerConvert.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java ServerConvert  
Object registered....  
|
```

```
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>javac ClientConvert.java  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java ClientConvert  
Enter a number :  
456  
The word representation of the entered digit is : four five six  
C:\College\KIRTI\Msc IT Part 1 Sem 1\Practicals\CLOUD COMPUTING\java file>java ClientConvert  
Enter a number :  
789  
The word representation of the entered digit is : seven eight nine
```

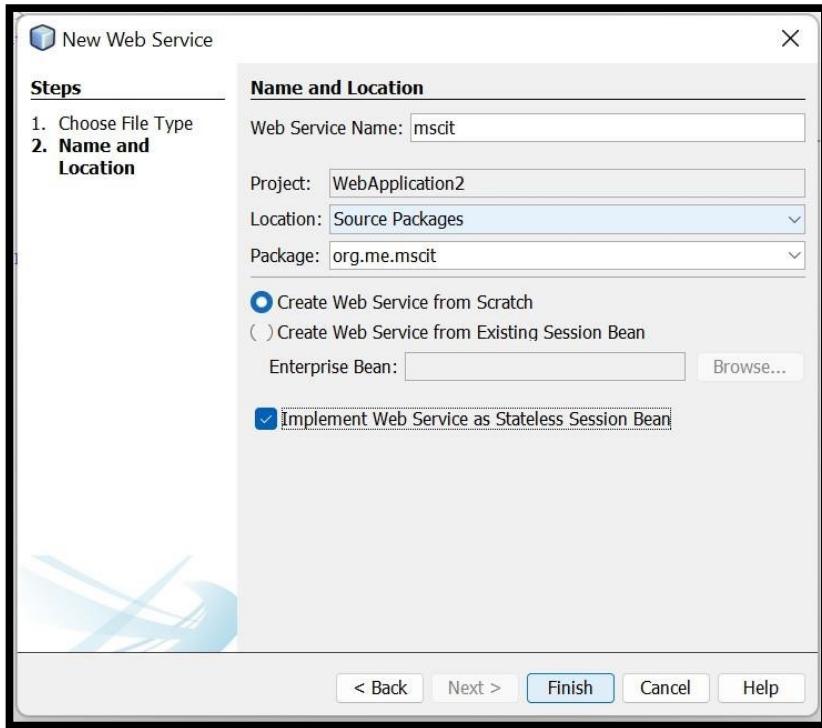
PRACTICAL 05

AIM :- SHOW THE IMPLEMENTATION OF WEB SERVICES.

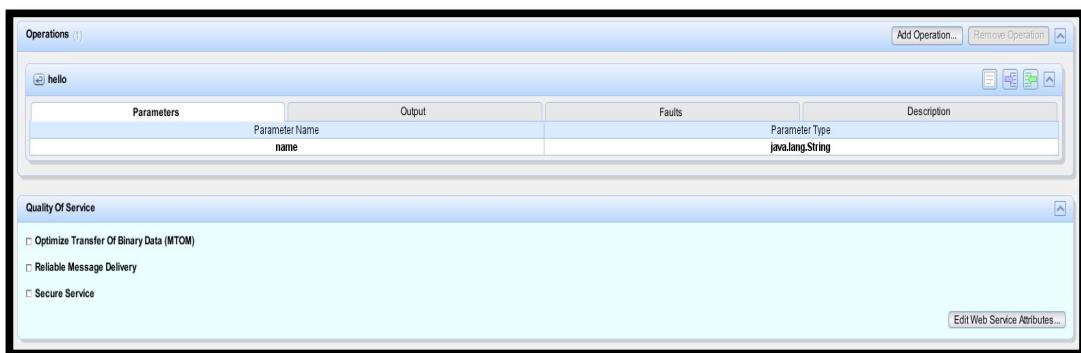
Step 1 :- File >> New Project >> Choose Project >> Java Web >> Web Application >> Next >> Give Project Name >> Next >> Next >> Finish

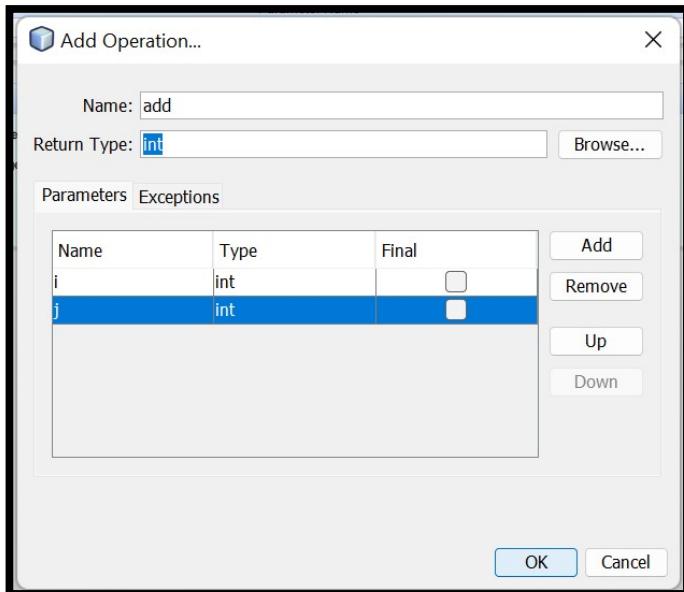


Step 2 :- Right Click on Project >> New >> Web Service >> Give name and package name >> Finish



Step 3 :- Go on Design Section of .java file >> add operation as add and return type as int >> addtwo parameters i & j >> OK





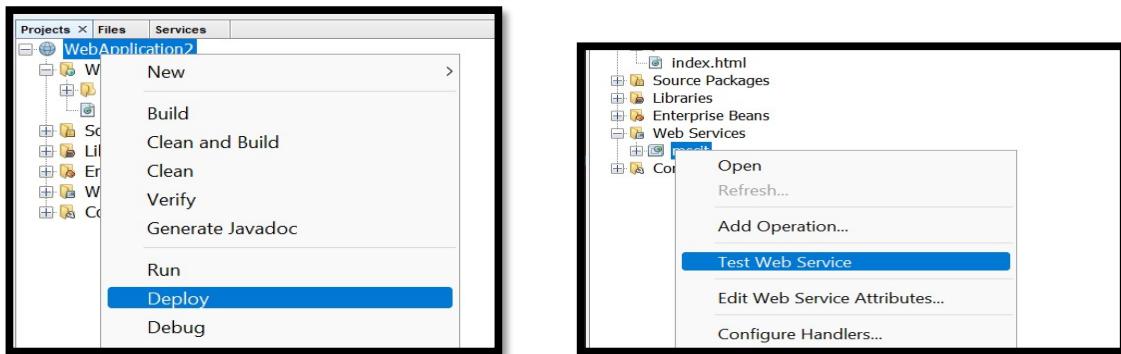
Step 4 :- Add following code inside .java file

The screenshot shows the NetBeans IDE interface with the following details:

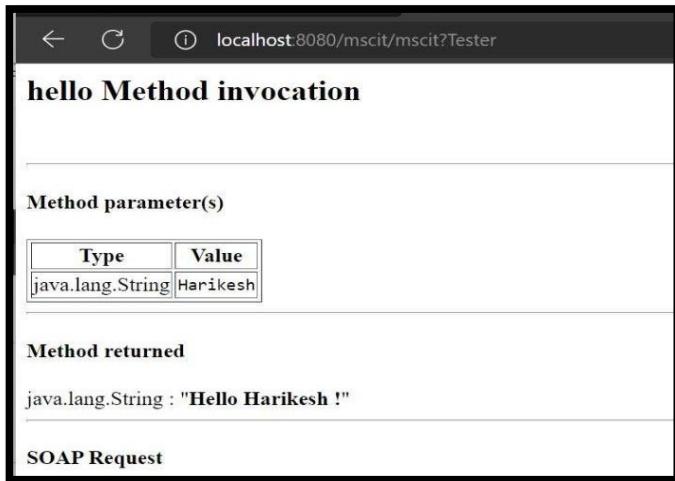
- Title Bar:** Shows tabs for "Start Page", "index.html", and "mscit.java".
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Find, Replace, Cut, Copy, Paste, Undo, Redo), a search bar, and other project-related tools.
- Code Editor:** Displays Java code for a Web service named "mscit". The code includes annotations for WebService, WebMethod, and WebParam, as well as sample implementation logic for the "hello" and "add" methods.
- Code Structure View:** A tree view on the left side of the editor pane, showing the hierarchical structure of the code, including comments and imports.

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6 package org.me.mscit;
7
8 import javax.jws.WebService;
9 import javax.jws.WebMethod;
10 import javax.jws.WebParam;
11 import javax.ejb.Stateless;
12
13 /**
14  *
15  * @author Acer
16  */
17 @WebService(serviceName = "mscit")
18 @Stateless()
19 public class mscit {
20
21     /**
22      * This is a sample web service operation
23      */
24     @WebMethod(operationName = "hello")
25     public String hello(@WebParam(name = "name") String txt) {
26         return "Hello " + txt + " !";
27     }
28
29     /**
30      * Web service operation
31      */
32     @WebMethod(operationName = "add")
33     public int add(@WebParam(name = "i") int i, @WebParam(name = "j") int j) {
34         //TODO write your implementation code here:
35         int k = i + j;
36         return k;
37     }
38 }
39
```

Step 5 :- Right click on the project name and Deploy the Web Service and finally Test the WebService

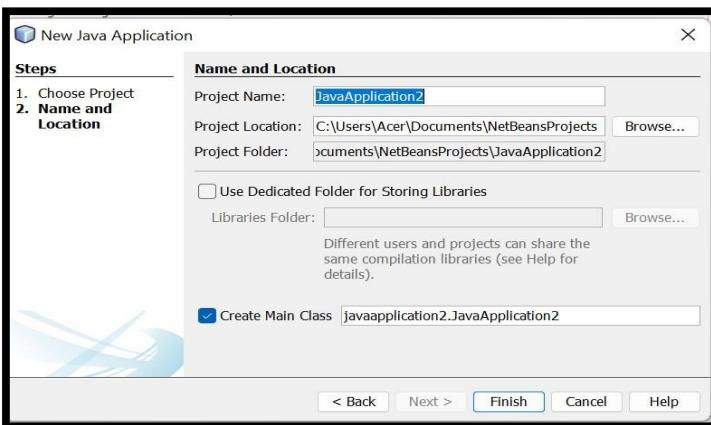
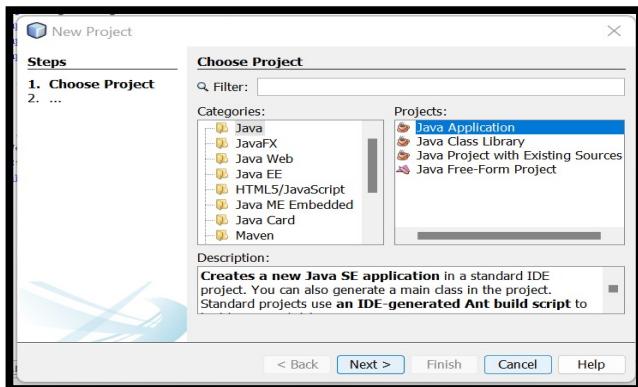


Output :-

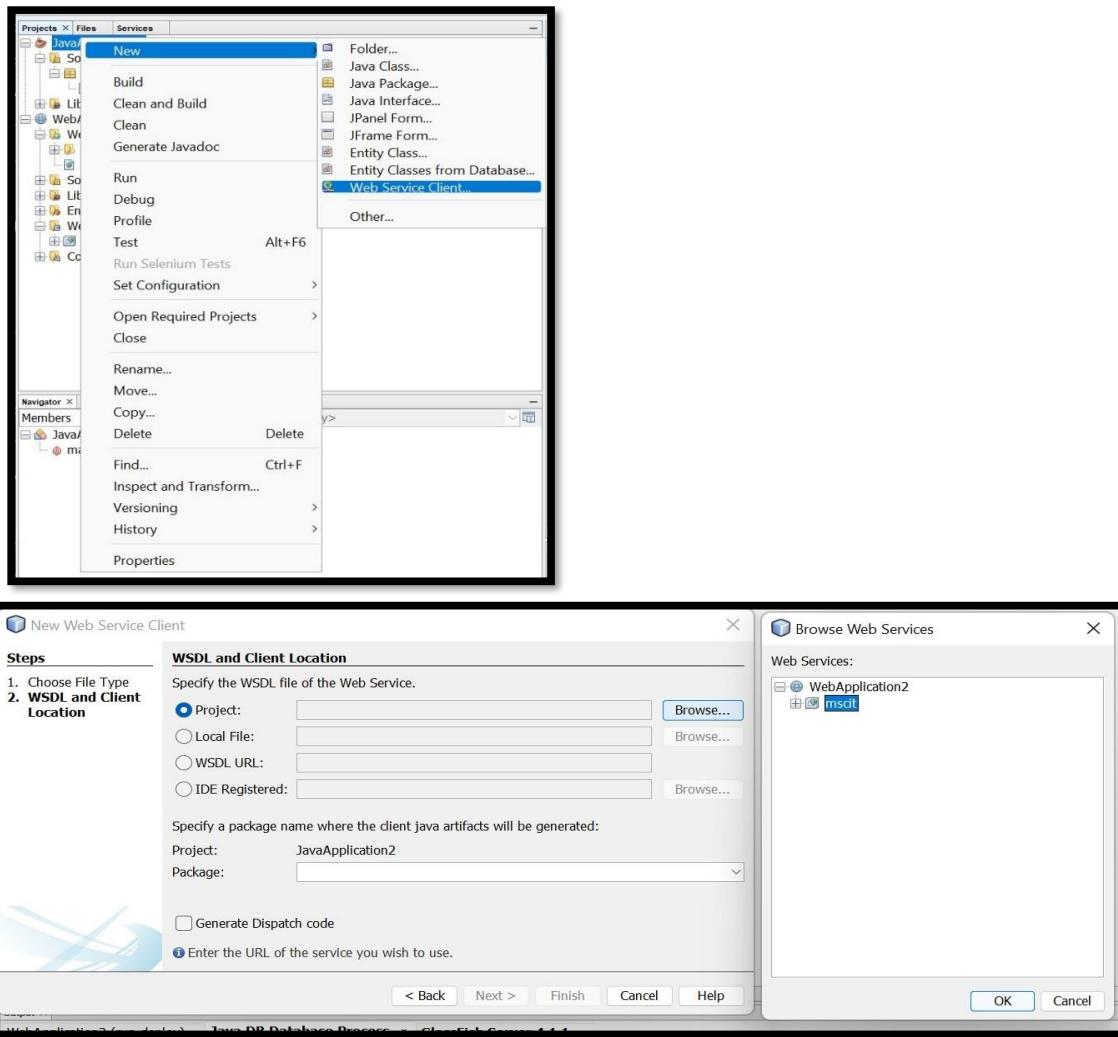


Steps for Consuming the Web Service :-

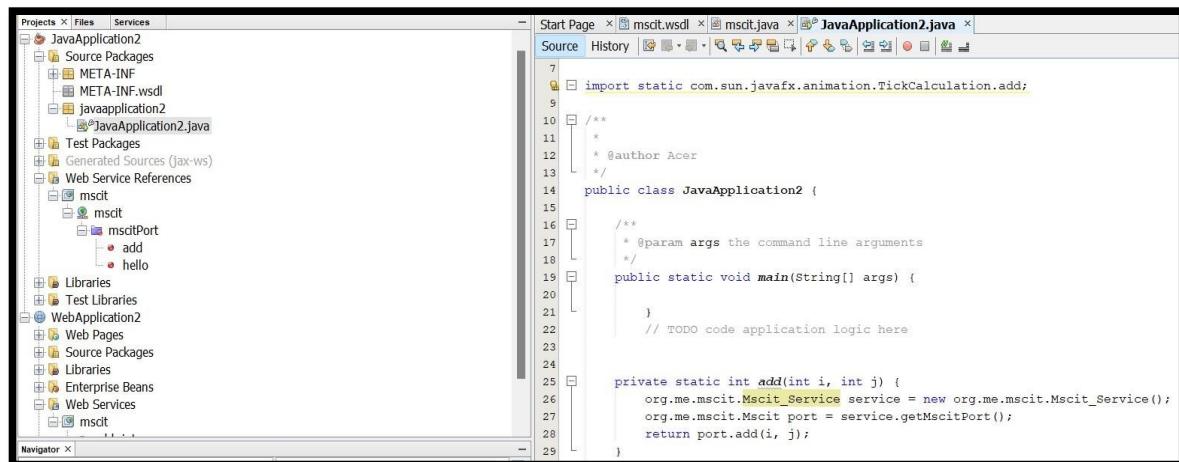
Step 1 :- Create new java application >> File >> New Project >> Choose Project >> Java >> Java Application >> Next >> Give Project Name >> Next >> Next >> Finish



Step 2 :- Right click on project >> new >> Web Service Client >> browse Web Service >> select the deployed project >> OK



Step 3 :- drag add() node from Web Service References below the Main() method



use try/catch block for printing exception

```
14  public class JavaApplication2 {
15
16      /**
17      * @param args the command line arguments
18      */
19      public static void main(String[] args) {
20          try {
21              {
22                  int i =66;
23                  int j =44;
24                  long result = add(i,j);
25                  System.out.println("Result = " + result);
26              }
27              catch (Exception ex){
28                  System.out.println("Exception= " + ex);
29              }
30          }
31          // TODO code application logic here
32
33
34      private static int add(int i, int j) {
35          org.me.mscit.Mscit_Service service = new org.me.mscit.Mscit_Service();
36          org.me.mscit.Mscit port = service.getMscitPort();
37          return port.add(i, j);
38      }
39  }
```

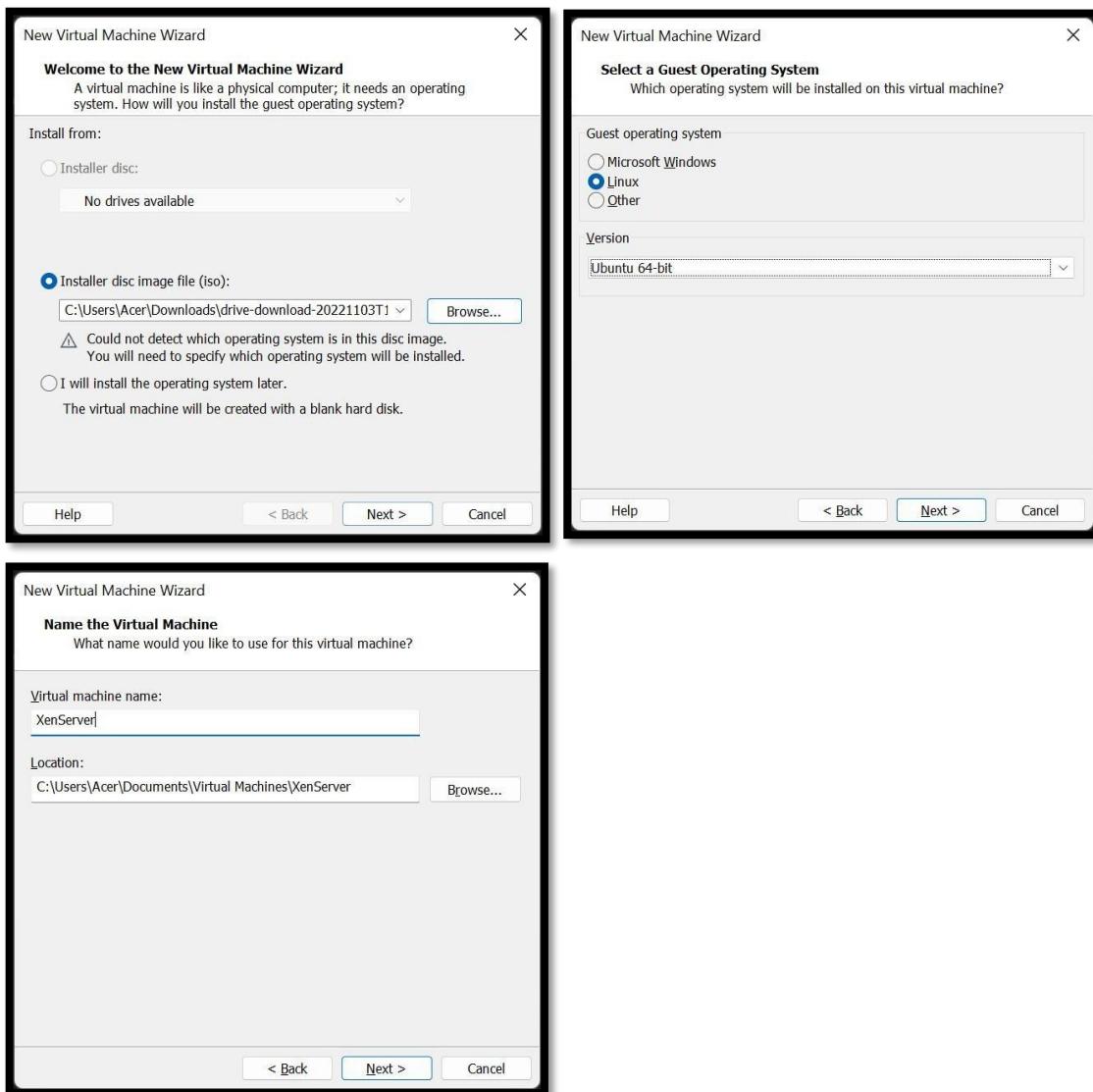
Output :-

```
Updating property file: C:\Users\Acer\Documents\NetBeansProjects\JavaApplication2\build\built-jar.properties
wsimport-init:
wsimport-client-mscit:
files are up to date
wsimport-client-generate:
compile:
run:
Result = 110
BUILD SUCCESSFUL (total time: 0 seconds)
```

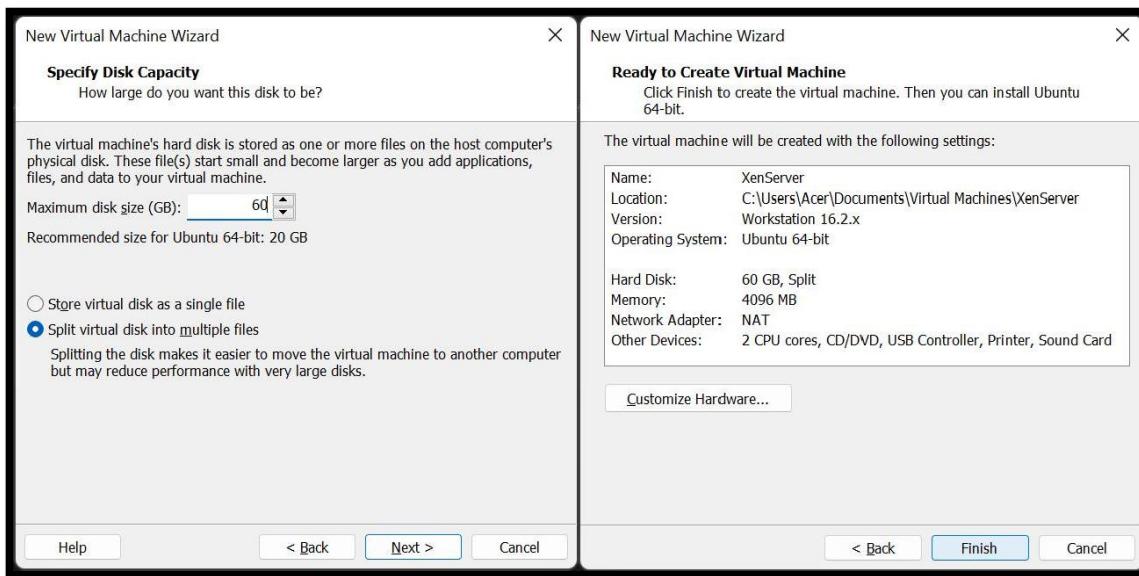
PRACTICAL 06

AIM :- IMPLEMENT XEN VIRTUALIZATION AND MANAGE WITH XEN CENTER

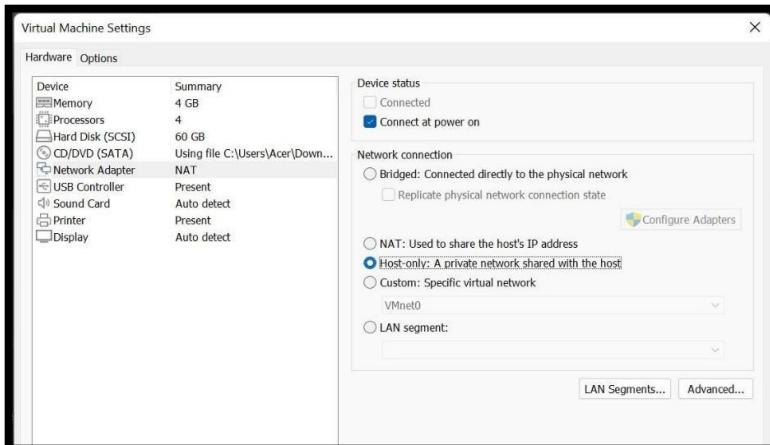
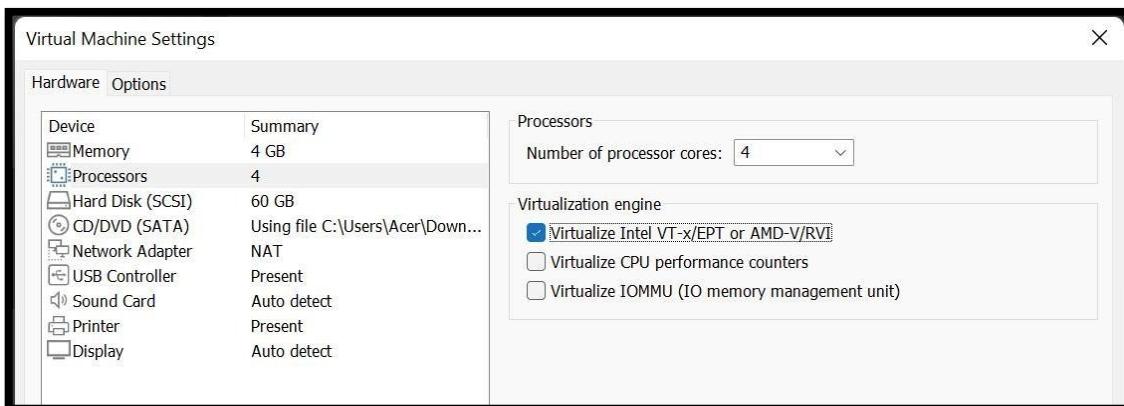
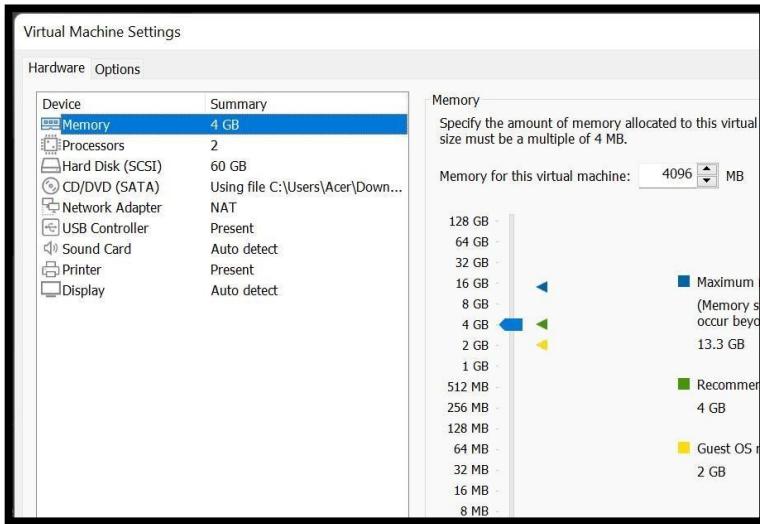
Step1 :- Select “New Virtual Machine” >> Next >> Select iso file >> Next >> Select guestOperating system >> Next >> give name as XenServer >> Next



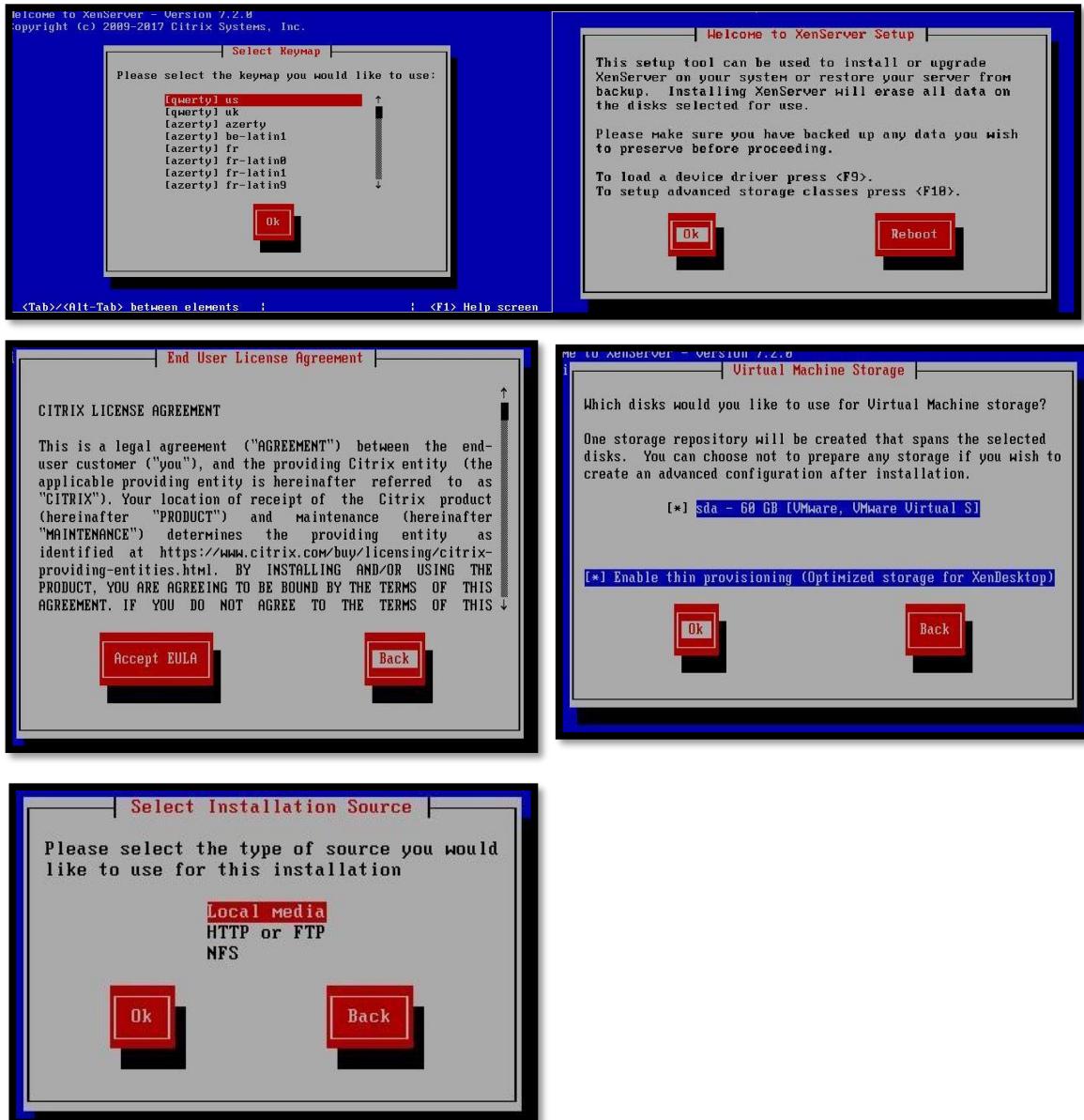
>> Select Disk Capacity as “60Gb”>> Next >> Finish.



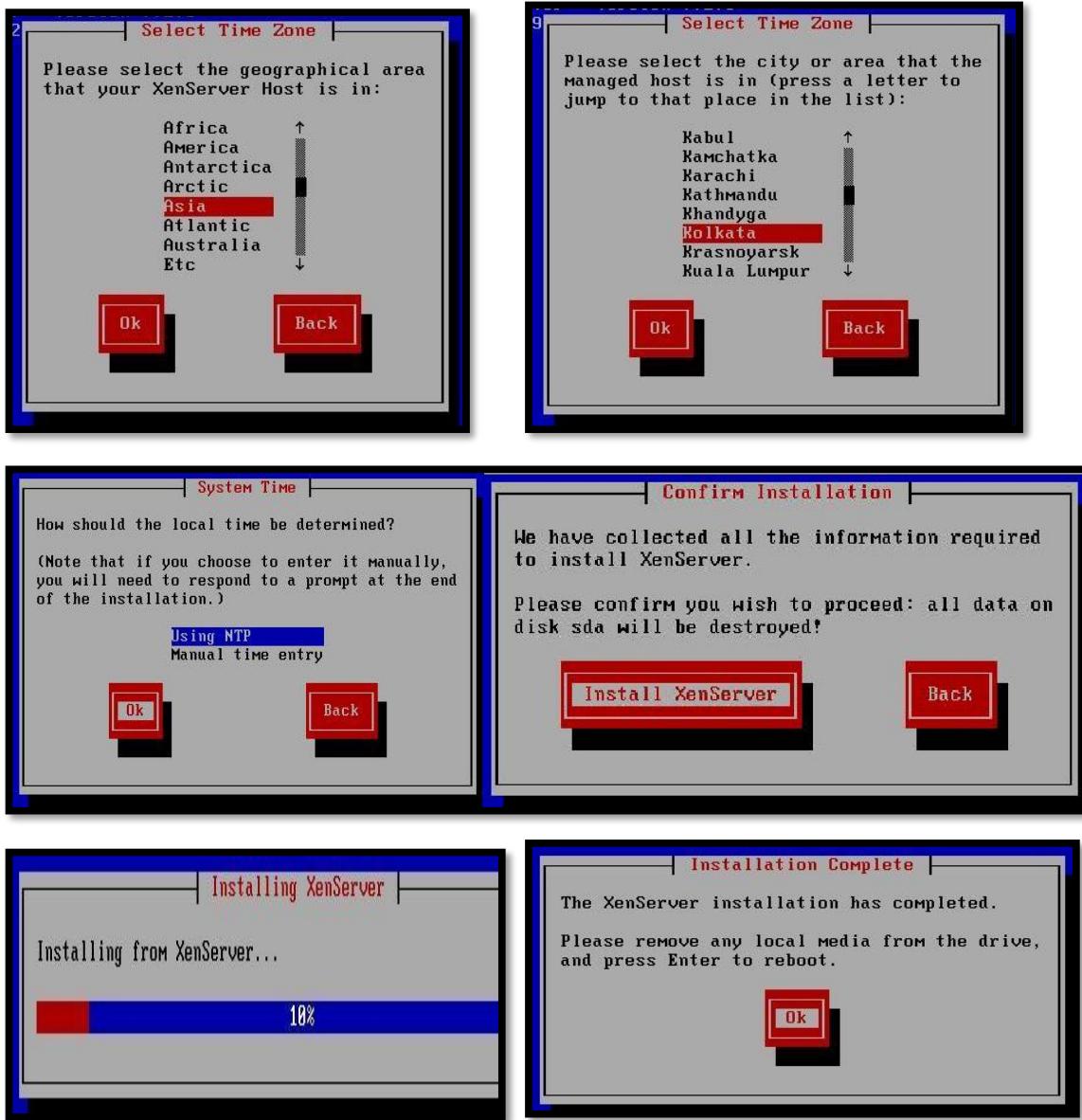
Step 2 :- Under edit Virtual machine Settings set RAM as 4GB and select no. of processor as 4 and under Processor set Intel-VT >> OK >> under Network Adapter >> select Host-only >> Ok.



Step 3:- Restart The Virtual Machine and click on “OK” on the Pop-Up to continue and click on“Accept EULA” to accept the EULA >> press OK .



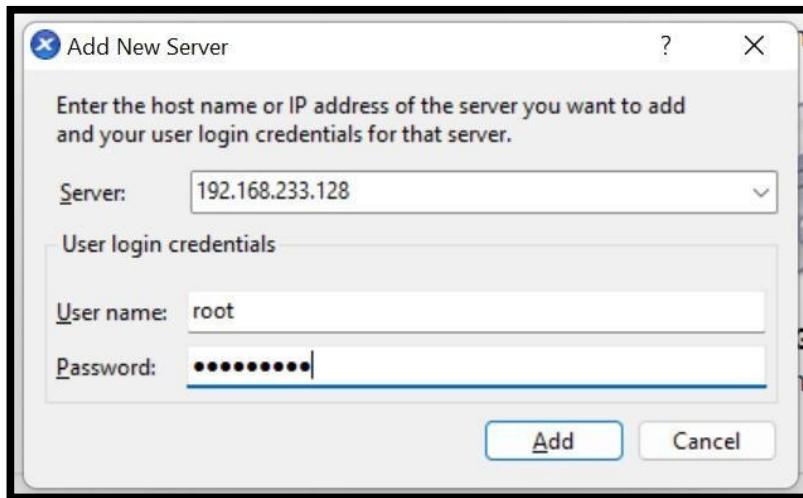
Select Time Zone >> Asia >> Kolkata >> System Time >> OK >> Click on “Install XenServer” >> OK



Step 4 :- note down the IP address and reboot the XenServer enter username and password. And open Critix Xen Center



Step 5 :- open Xen Center add server -> add the IP address “192.168.233.128”and enter usernameand password as “root” and “admin@123” respectively



Output :-

```
192.168.233.128

XenServer Host SSL certificate fingerprint:
4E:02:75:29:5A:2A:C7:04:8A:2E:8C:BA:43:6C:50:85:AB:09:E8:DC

Password:
Last failed login: Fri Nov  4 06:23:19 IST 2022 on pts/0
There was 1 failed login attempt since the last successful login.

XenServer dom0 configuration is tuned for maximum performance and reliability.

Configuration changes which are not explicitly documented or approved by Citrix
Technical Support, may not have been tested and are therefore not supported. In
addition, configuration changes may not persist after installation of a hotfix
or upgrade, and could also cause a hotfix or upgrade to fail.

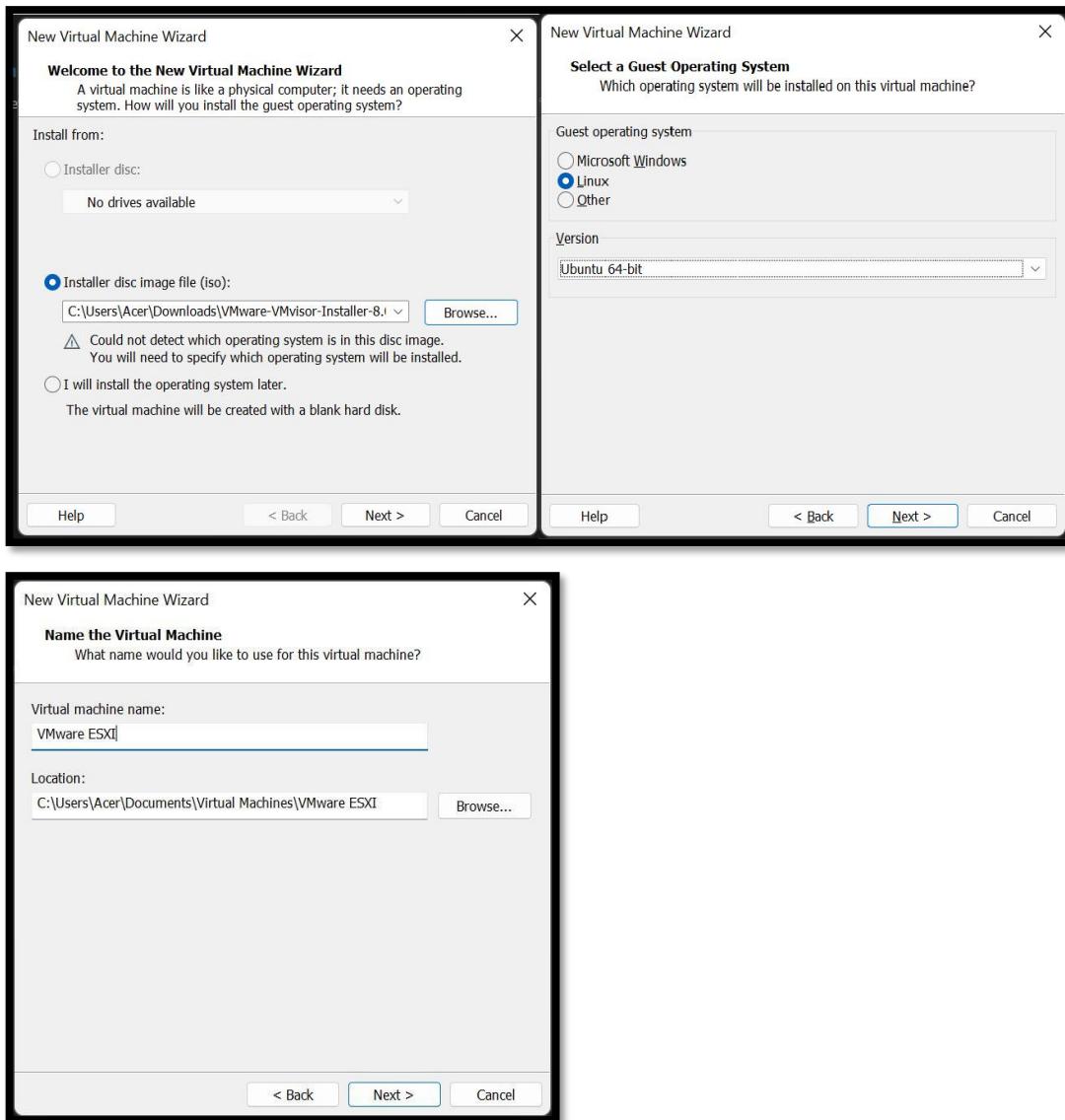
Third party tools, which require modification to dom0 configuration, or
installation into dom0, may cease to function correctly after upgrade or hotfix
installation. Please consult Citrix Technical Support for advice regarding
specific tools.

Type "xsconsole" for access to the management console.
[root@xenserver-punseb.jr ~]#
```

PRACTICAL 07

AIM :- IMPLEMENT VIRTUALIZATION USING VMWARE ESXI SERVER AND MANAGING WITH VCENTER

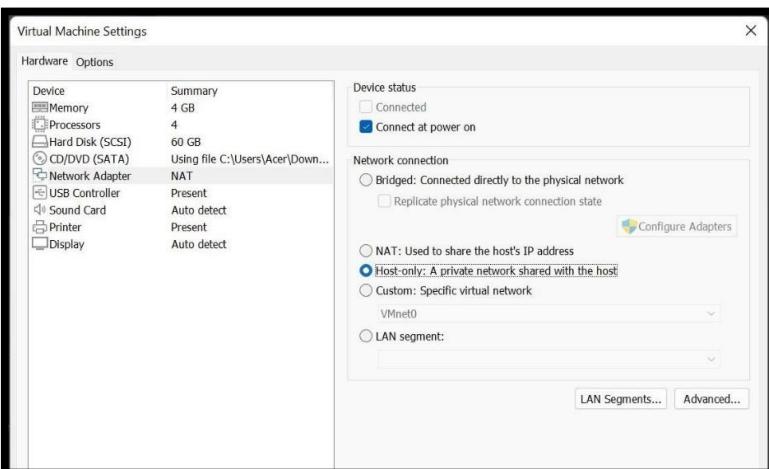
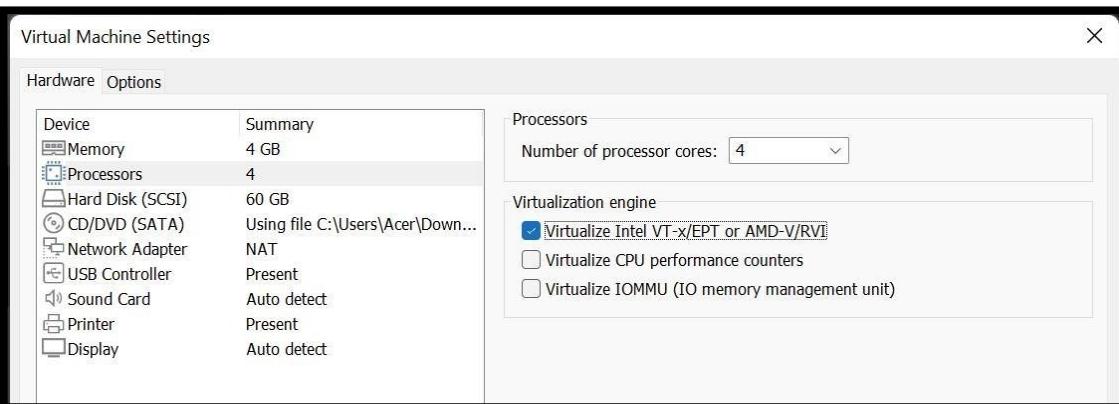
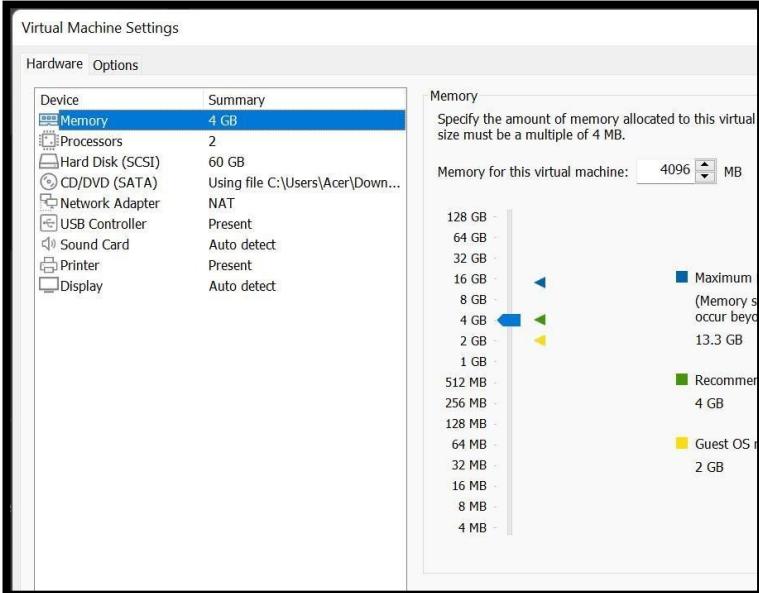
Step1 : Select “New Virtual Machine” >> Next >> Select iso file >> Next >> Select guestOperating system >> Next >> give name as VMware ESXI>> Next



>> Select Disk Capacity as “60Gb” >> Next >> Finish.



Step 2:- Under edit Virtual machine Settings set RAM as 4GB and select no. of processor as 4 and under Processor set Intel-VT >> OK >> under Network Adapter >> selectHost-only >> Ok.



Step 3 :- Restart the virtual machine and click on “F11” key to accept the EULA and click on“Enter” on the Pop-Up to continue .



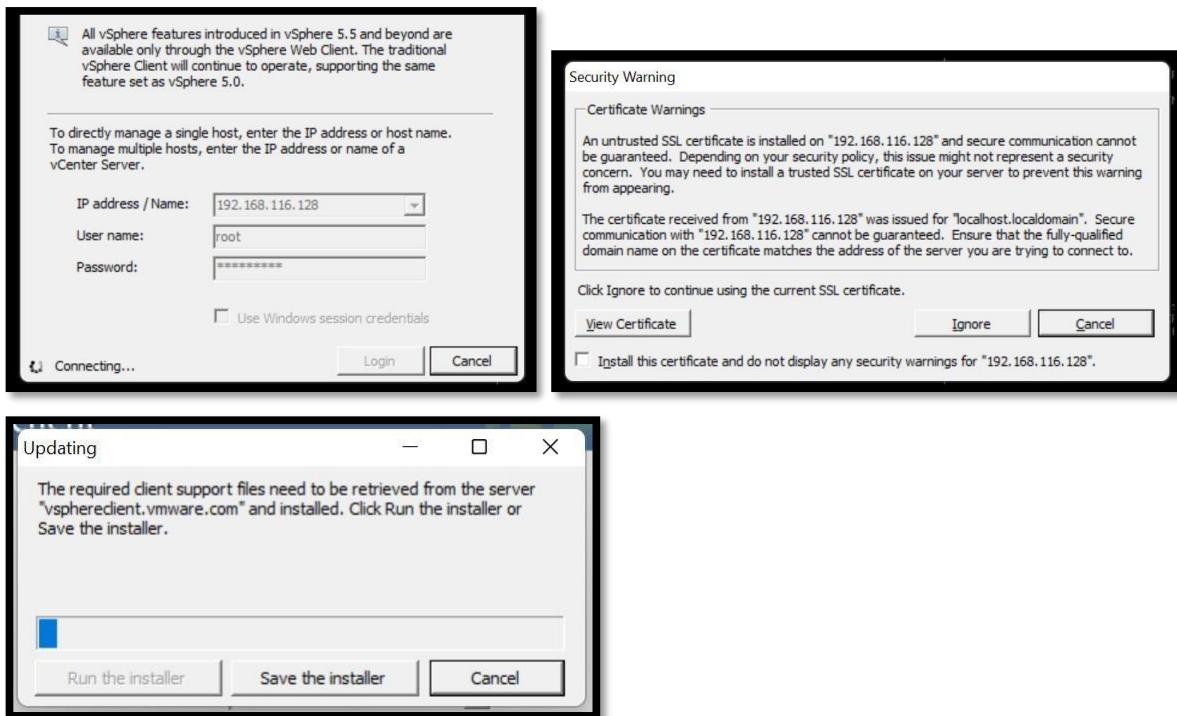
Step 4 :- press “Enter” >> Select US Default >> Enter >> Enter password as “admin@123”>>Press ‘F11’ key .



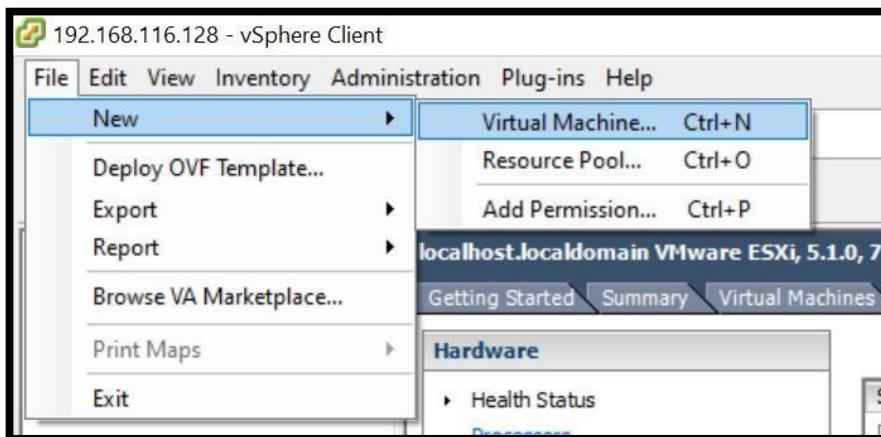
Step 5 :- Server will reboot after that Give IP Address “192.168.146.130” upon Pop-Up give username as “Root” and Passowrd as “admin@123” >> Click on “OK” >> press “F11” Key to restart



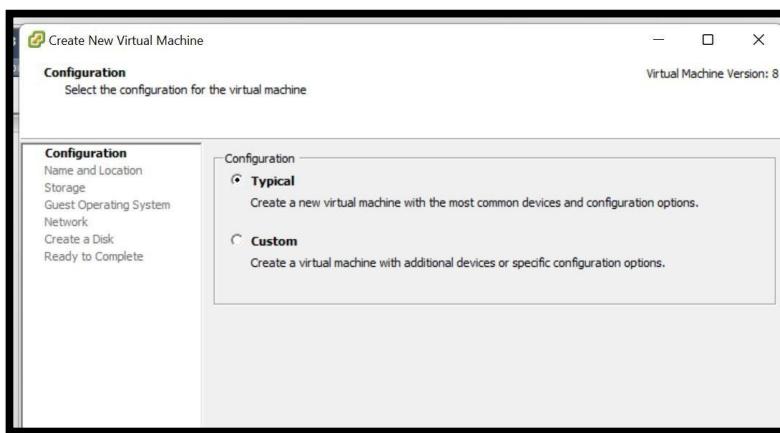
Step 6:-Open the VMware vSphere Client Give IP address “192.168.146.128”, User name =“root”Password = “admin@123” and Click on Login if warning POP-UP appers then click on “Ignore”



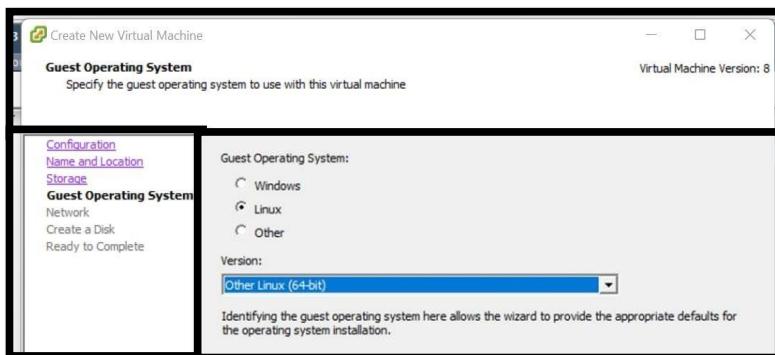
Step 7: - Under Configuration Select Host >> New Virtual Machine



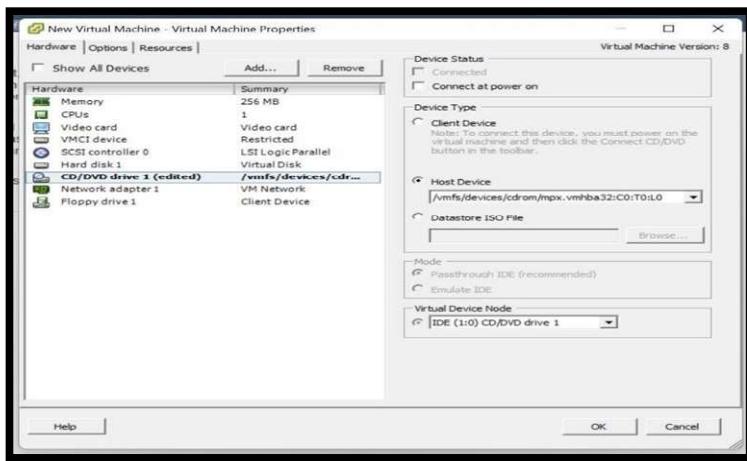
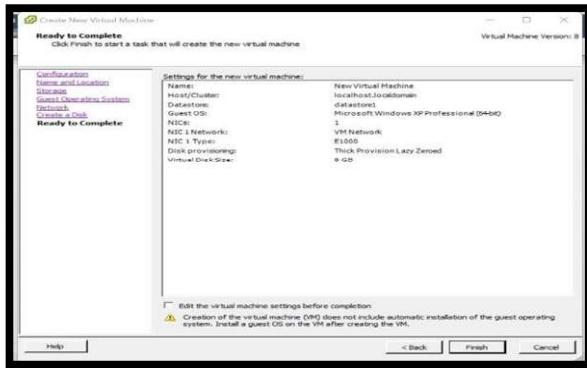
Configuration window select checkbox “typical” >>



Step 8:Select the Linux and give other Linux(64 bit) And Click on “Next”.



Click on “Finish” >> OK



STEP 9:- GO UNDER “CONSOLE” SECTION FOR “OUTPUT”

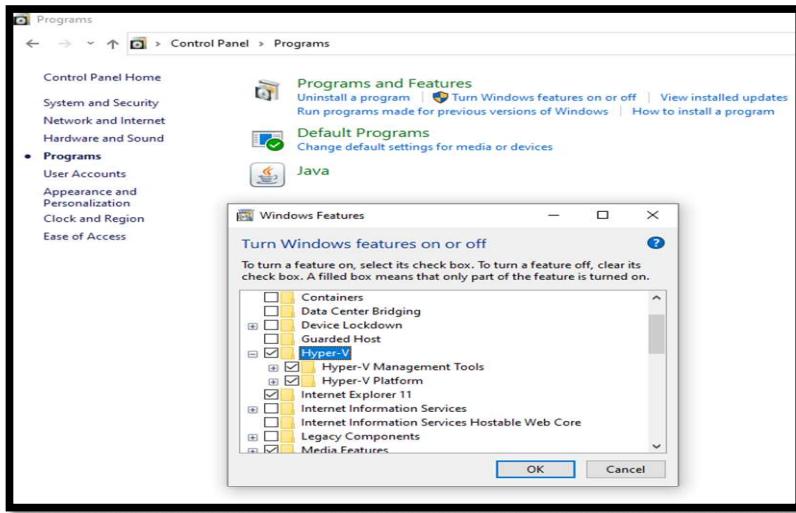


PRACTICAL NO.8

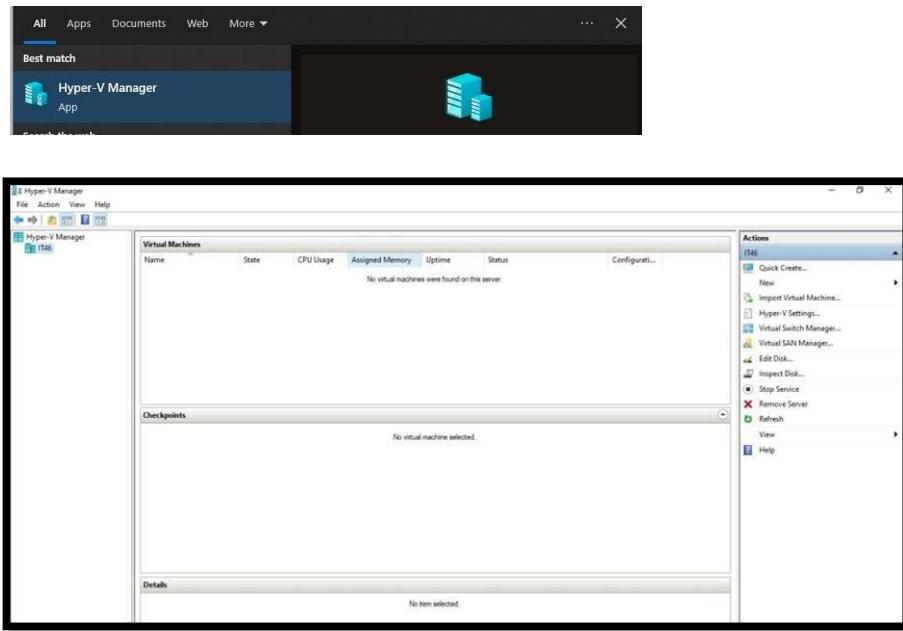
AIM :- IMPLEMENT WINDOWS HYPER V VIRTUALIZATION.

Firstly we turn off vmware software and enable hyper v through control panel

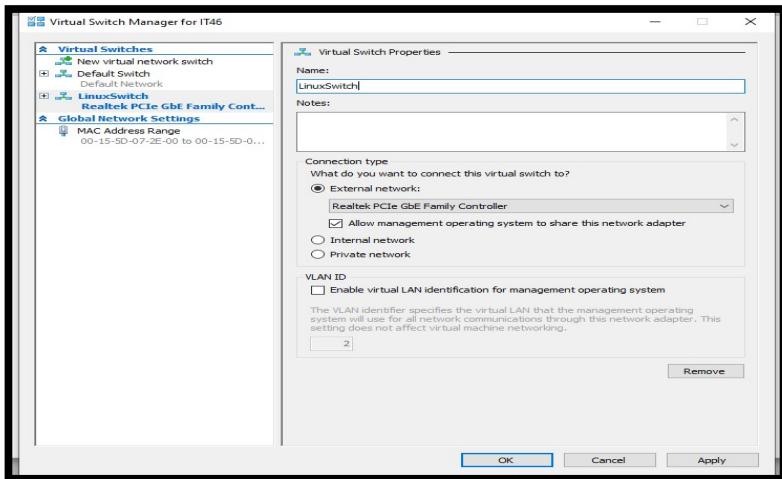
Control panel -> program file -> turn windows feature on or Off -> click on hyper v and enable it.



Then click on start button and search hyper v manager and click on it.



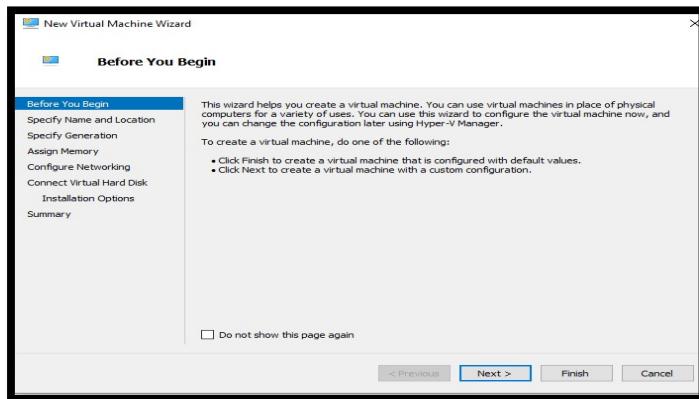
Click virtual switch manager and create virtual switch



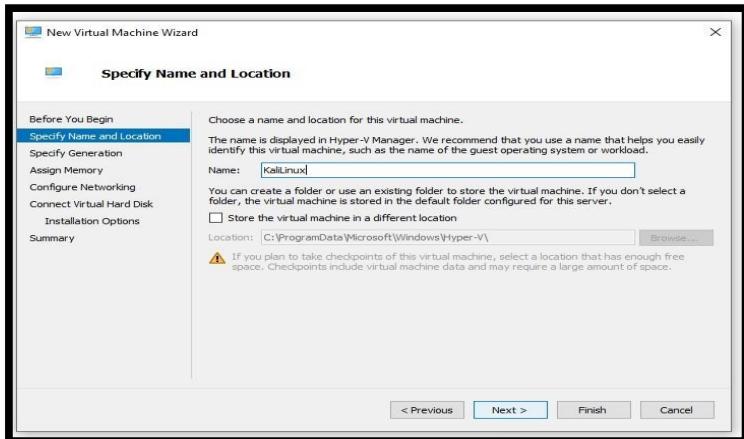
Click new and create new virtual machine



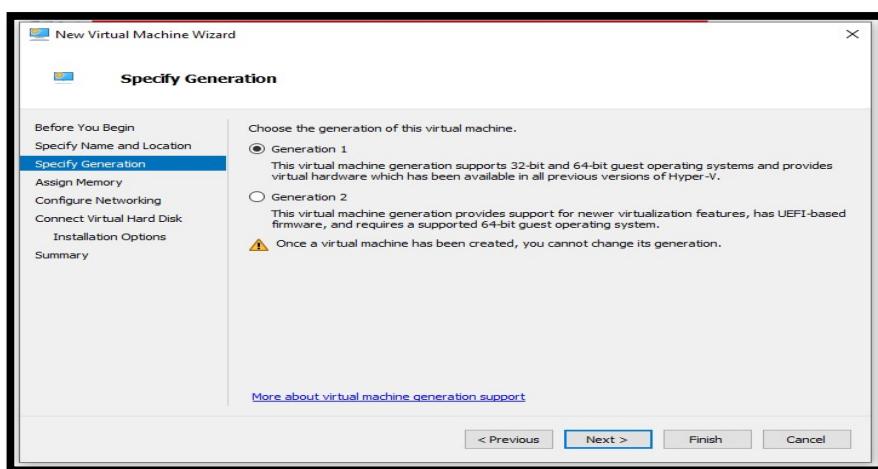
After click, wizard page open n follow next step as show on screen.



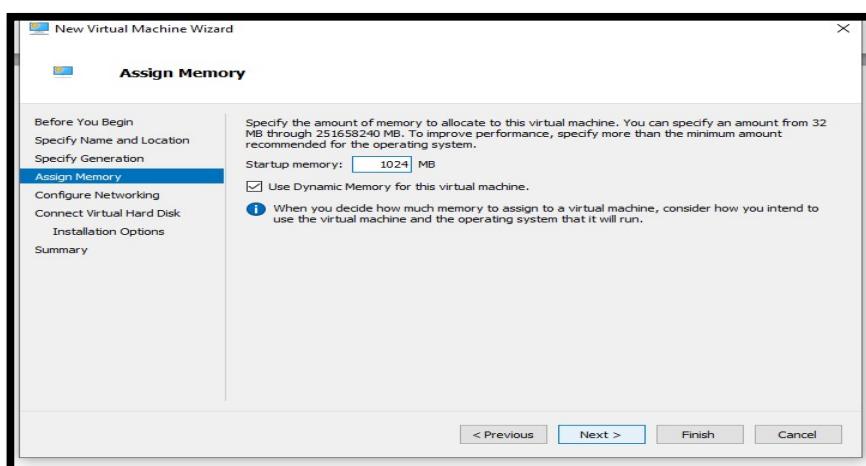
Give name to your VM and click on Next



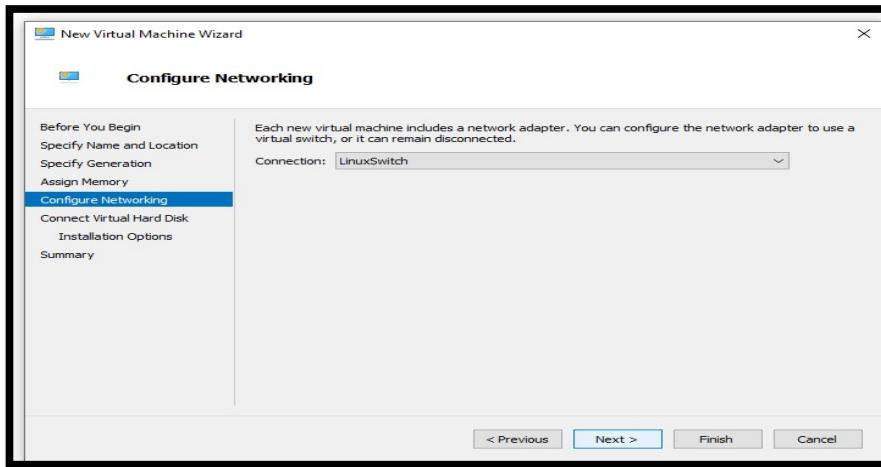
Then we need to specify Generation, select Generation 1 And next.



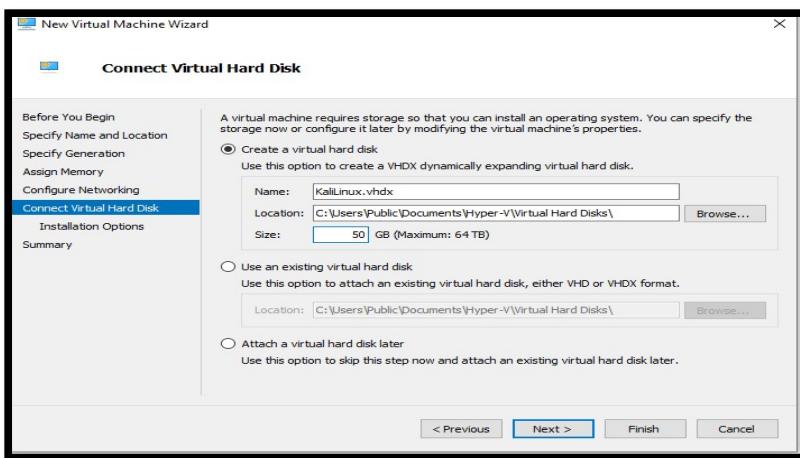
Next, here we assign memory to it and next.



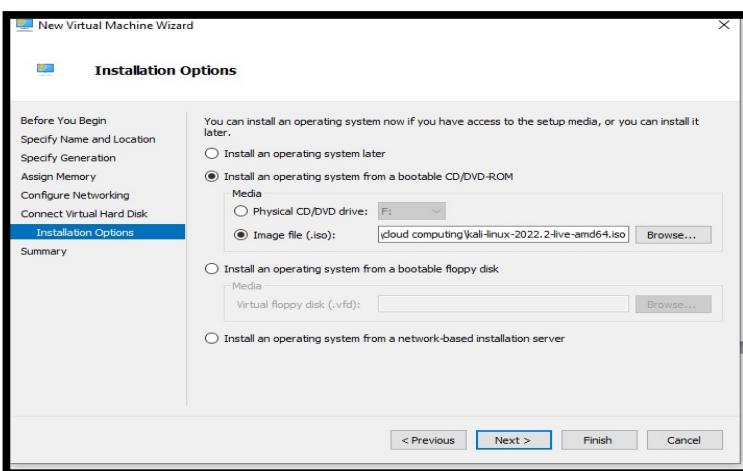
Here we select connection that we already create i.e LinuxSwitch and next.



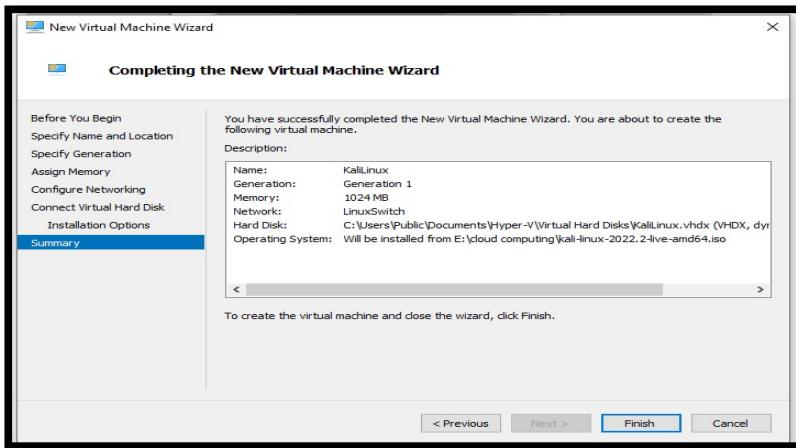
Give hard disk space and next



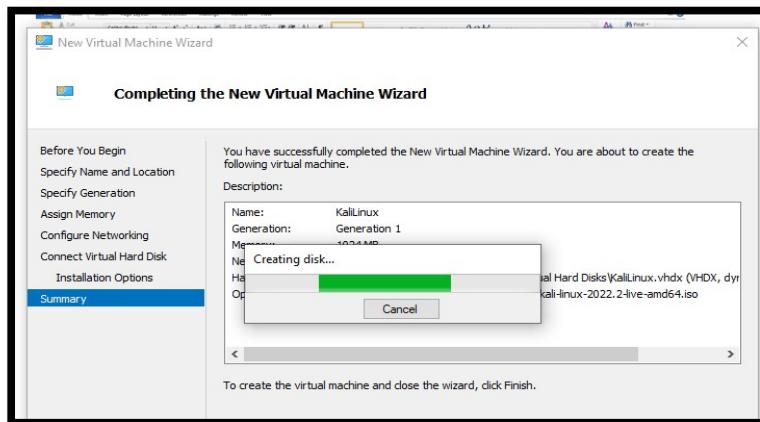
Here we give our ISO file path and next



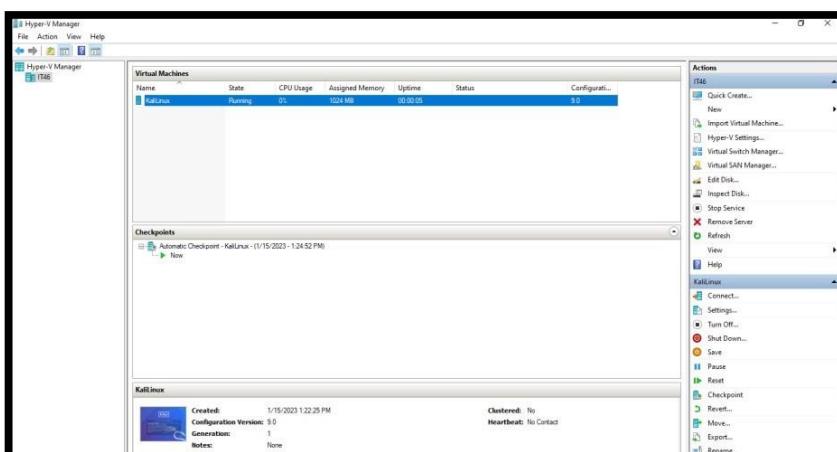
Summary of all details.



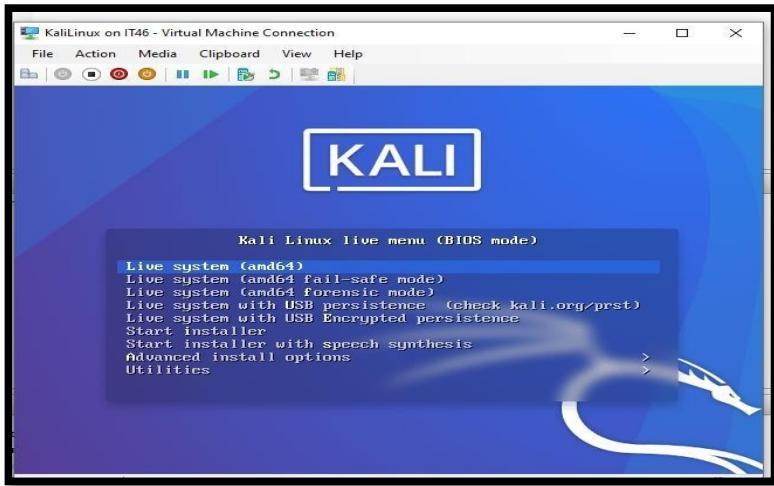
create and finish



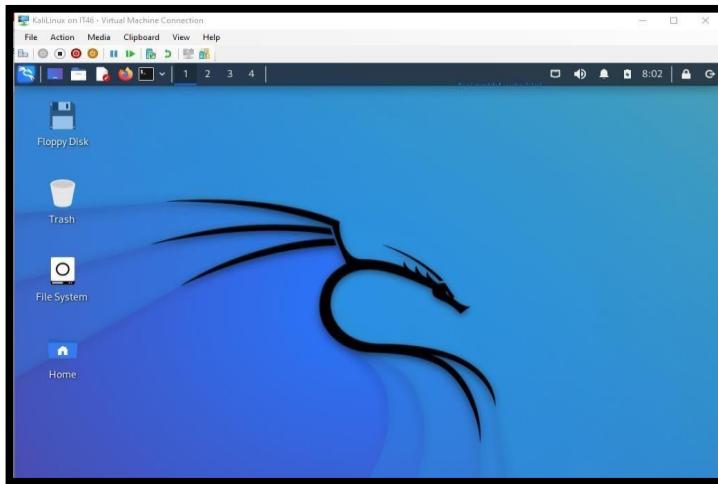
Your main kaliLinux VM create and now start that VM.



Double click on bottom side small kali linux window. And enter Live system



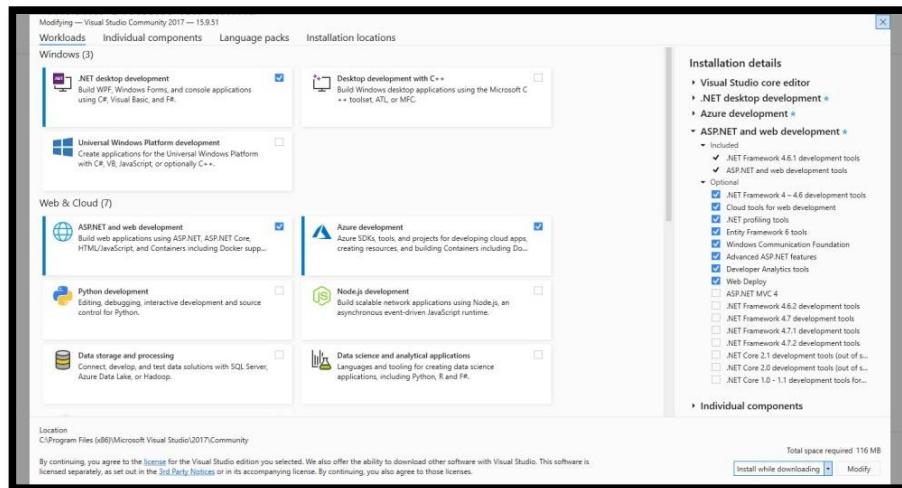
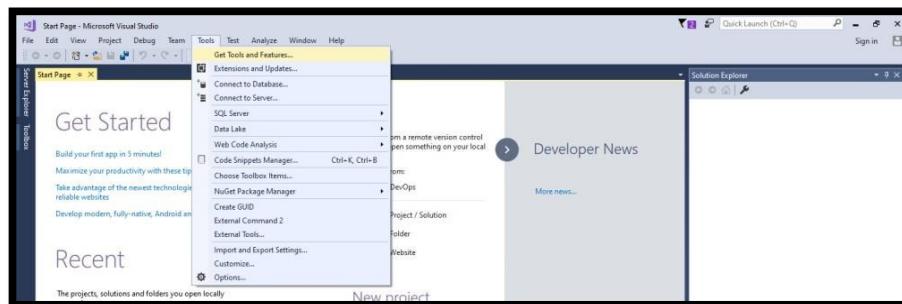
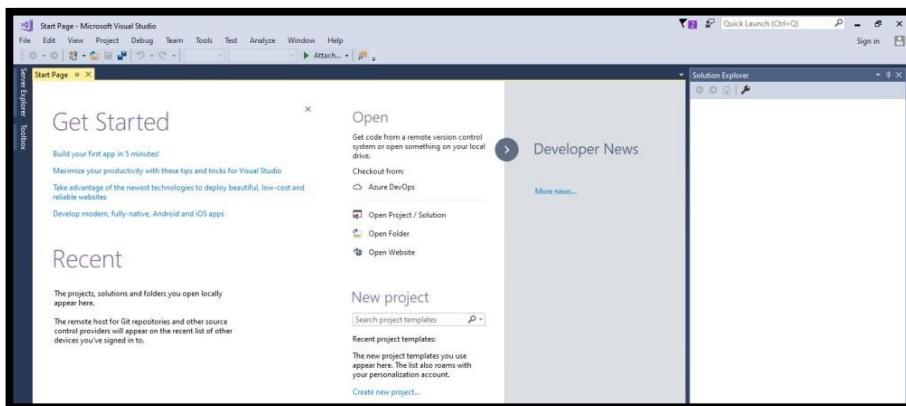
After click first option, we can see kali linux OS interface that's means we install kali linux successfully.

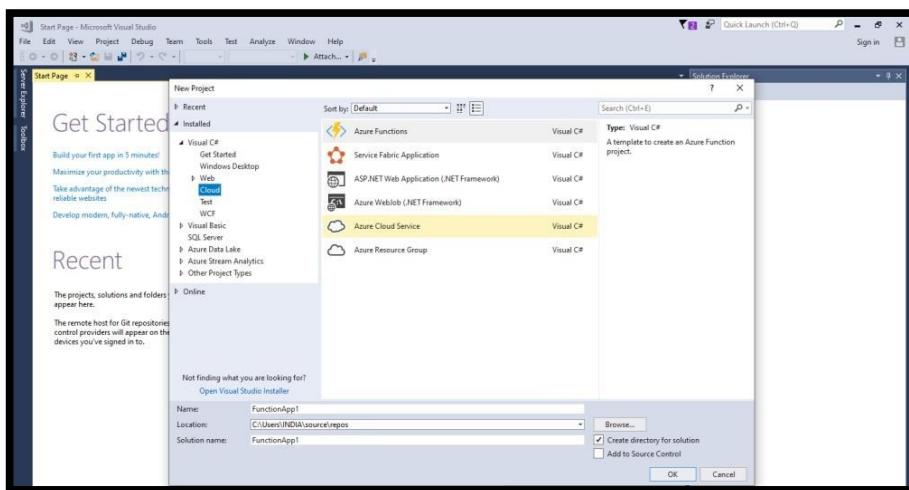
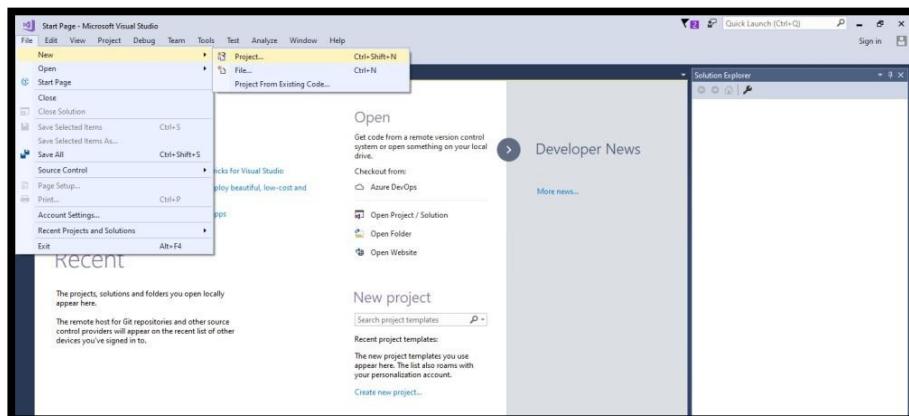
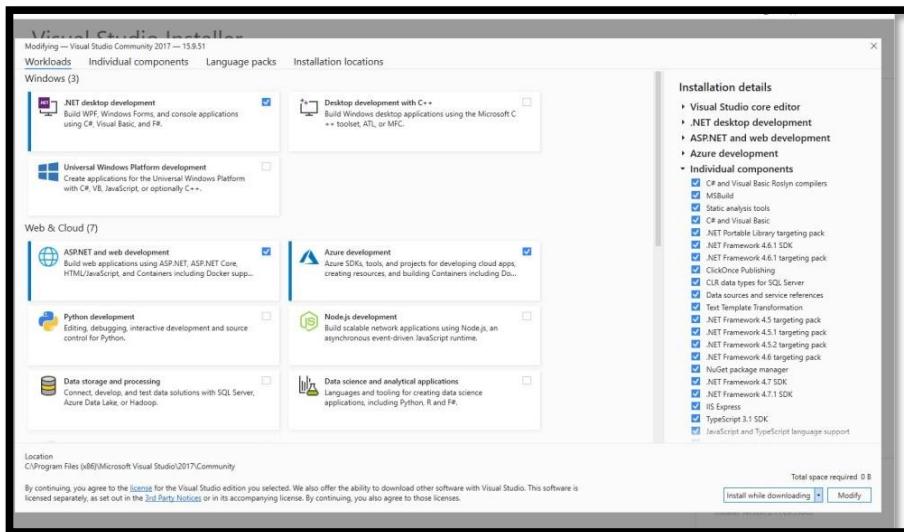


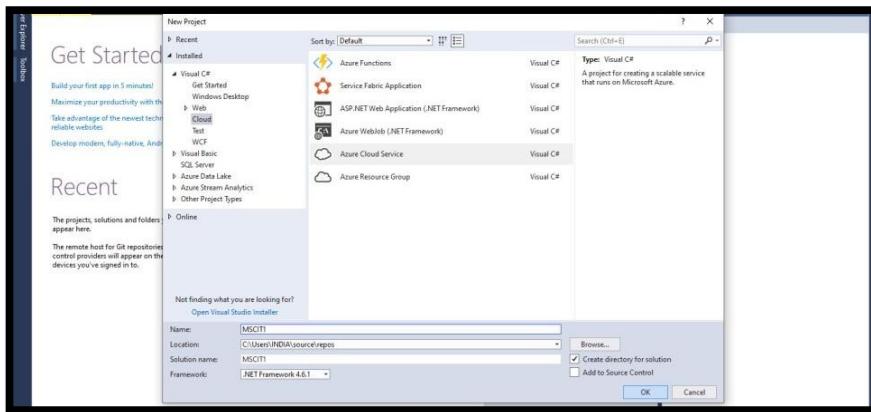
PRACTICAL 09

AIM :- DEVELOP APPLICATION FOR MICROSOFT AZURE.

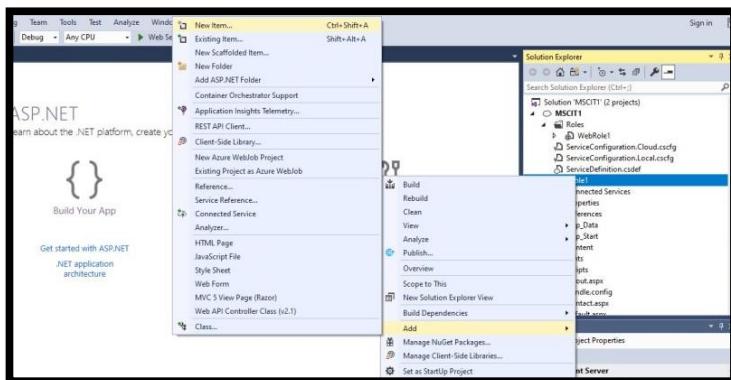
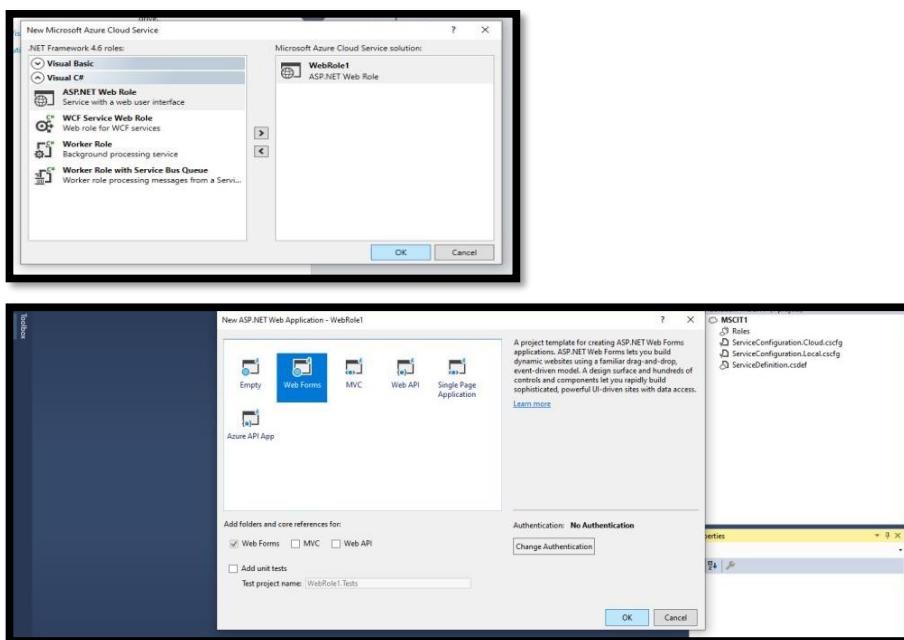
Step 1: - To Develop an application for Windows Azure open Visual Studio Community 2019 Select “Create New Project” -> select “Azure Cloud Service (Classic)” -> Click on “Create”

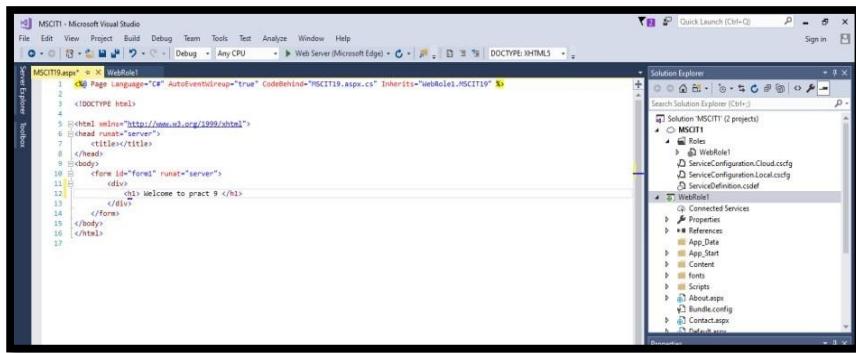
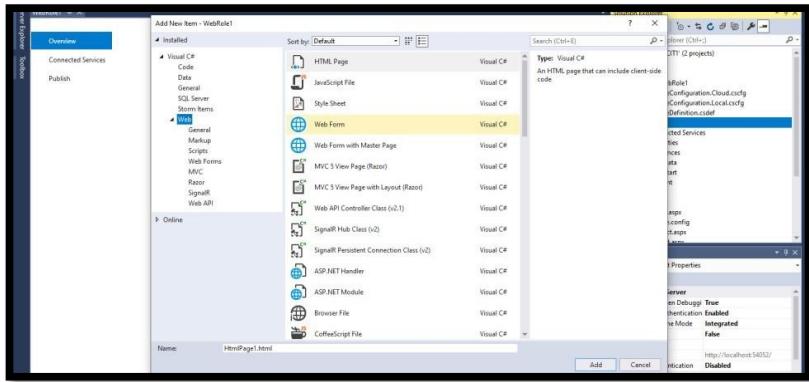






Step 2 :- In the Pop-up window select “ASP.NET Web Role” and add to the solution->Click on “OK”





Output :-

