PRACTICAL 1

AIM: Create Chat application using either TCP or UDP protocol

❖ TCP

```
Server Side:
import java.net.*;
import java.io.*;
public class P1MySerever {
   public static void main(String[] args){
        ServerSocket ss = new ServerSocket(8070);
        Socket s = ss.accept();
        OutputStream obj = s.getOutputStream();
        PrintStream ps = new PrintStream(obj);
        ps.println("Hello clients");
        ps.println("How are you?");
        ps.println("This is from serever side");
        ss.close();
        s.close();
        ps.close();
    }
    catch(IOException ex){
       ex.printStackTrace();
    }
O/P:
☐ ∰ JavaFXApplication7
☐ ∰ jsf
                            Output %
                            AJ (run) x AJ (run) #2 x
 ⊕ 🎳 mcm
 🗈 🍃 new
                            java.net.BindException: Address already in use: JVM_Bind
newservlet
newservlet1
                                      at java.net.DualStackPlainSocketImpl.bind0(Native Method)
                                      at java.net.DualStackFlainSocketImpl.socketBind(DualStackFlainSocketImpl.java:106) at java.net.AbstractFlainSocketImpl.bind(AbstractFlainSocketImpl.java:376)
 ⊕ 🦣 p18
                                      at java.net.PlainSocketImpl.bind(PlainSocketImpl.java:190 at java.net.ServerSocket.bind(ServerSocket.java:376)
  -- 🜦 p20
  at java.net.ServerSocket.<init>(ServerSocket.java:237
 ⊕ 🦣 pr19
                                      at java.net.ServerSocket.<init>(ServerSocket.java:128
 ⊕-p prac8
⊕-🍃 practical 10
                                      at aj.PlMySerever.main(PlMySerever.java:16)
                               BUILD SUCCESSFUL (total time: 0
 🗎 🦣 practical 18
```

O # 🧿 🙍 💽 🖫 🖽 🗗

 ${\cal P}$ Type here to search

⊕ b practical2.java ⊕ 🌐 pratik

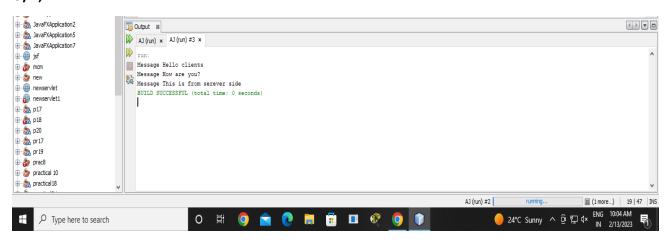
№ 19 | 48 INS

— 22℃ Sunny ^ @ 딮 ♥

• Client Side:

```
import java.net.*;
import java.io.*;
public class P1MyClients {
  public static void main(String[] args){
  try{
   Socket s = new Socket("localhost",8070);
   InputStream is = s.getInputStream();
   BufferedReader br = new BufferedReader(new InputStreamReader(is));
   String receivedMessage = br.readLine();
    System.out.println("Message "+receivedMessage);
    String receivedMessage1 = br.readLine();
    System.out.println("Message "+receivedMessage1);
    String receivedMessage2 = br.readLine();
    System.out.println("Message "+receivedMessage2);
    br.close();
    s.close();
  }
  catch(IOException ex){
    ex.printStackTrace();
}
}
```

O/P;



EN NO: 200760107043 [Savani Zarana]

❖ <u>UDP</u>

```
• Server:
import java.net.*;
import java.io.*;
public class P1UDPsender {
  public static void main(String[] args) {
try {
DatagramSocket ds=new DatagramSocket();
String str="Message from Sender";
InetAddress ip=InetAddress.getByName("localhost");
DatagramPacket dp=new DatagramPacket(str.getBytes(),
str.length(), ip, 6666);
ds.send(dp);
ds.close();
} catch (Exception ex) {
ex.printStackTrace();
}
```

O/P:



• Client:

```
import java.net.*;
import java.io.*;
public class P1UDPreceiver {
   public static void main(String[] args) {
   try {
```

EN NO: 200760107043 [Savani Zarana]

```
DatagramSocket ds = new DatagramSocket(6666);
byte buffer[] = new byte[1024];
DatagramPacket dp = new DatagramPacket(buffer, 1024);
ds.receive(dp);
String str =new String(dp.getData(),0,dp.getLength());
System.out.println("Receive: "+str);
ds.close();
} catch (Exception ex) {
ex.printStackTrace();
}
}
```

O/P:

