```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Net;
using System.Net.Sockets;
using System.IO;


namespace ServerClient
{
    public partial class Server : Form
    {
        private TcpClient client;  //provides client connections for TCP netweok services
        public StreamReader STR;  //read text from a file
        public StreamWriter STW;  //designed for character output in particular coding
        public string receive;
        public string text_to_send;

        public Server()
        {
            InitializeComponent();

            IPAddress[] localIP = Dns.GetHostAddresses(Dns.GetHostName());   //get my own IP
            foreach(IPAddress address in localIP)
            {
                if(address.AddressFamily == AddressFamily.InterNetwork)
                {
                    txtIPServer.Text = address.ToString();
                }
            }
        }



        private void start_Click(object sender, EventArgs e)  //Start Server
        {
            TcpListener listener = new TcpListener(IPAddress.Any, 5000);
            listener.Start();
            client = listener.AcceptTcpClient();
            STR = new StreamReader(client.GetStream());
            STW = new StreamWriter(client.GetStream());
            STW.AutoFlush = true;
            backgroundWorker1.RunWorkerAsync();                       //Start receiving Data in
    background
            backgroundWorker2.WorkerSupportsCancellation = true;     //Ability to cancel this thread
        }



        private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e) //receive data
        {
            while(client.Connected)
            {
                try
```

```
            {
                receive = STR.ReadLine();
                this.txtStatus.Invoke(new MethodInvoker(delegate() { txtStatus.AppendText("You: " + receive + "\n"); }));
                receive = "";
            }
            catch(Exception x)
            {
                MessageBox.Show(x.Message.ToString());
            }
        }
    }



    private void backgroundWorker2_DoWork(object sender, DoWorkEventArgs e) //send data
    {
        if(client.Connected)
        {
            STW.WriteLine(text_to_send);
            this.txtStatus.Invoke(new MethodInvoker(delegate() { txtStatus.AppendText("Me: " + text_to_send + "\n"); }));
        }
        else
        {
            MessageBox.Show("Send failed!");
        }
        backgroundWorker2.CancelAsync();
    }



    private void connect_Click(object sender, EventArgs e)  //connect to server
    {
        client = new TcpClient();
        IPEndPoint IP_End = new IPEndPoint(IPAddress.Parse(txtIPClient.Text), int.Parse(txtPortClient.Text));

        try
        {
            client.Connect(IP_End);
            if(client.Connected)
            {
                txtStatus.AppendText("Connected to Server" + "\n");
                STW = new StreamWriter(client.GetStream());
                STR = new StreamReader(client.GetStream());
                STW.AutoFlush = true;

                backgroundWorker1.RunWorkerAsync();                     //Start receiving Data in background
                backgroundWorker2.WorkerSupportsCancellation = true;    //Ability to cancel this thread
            }
        }catch (Exception x)
        {
            MessageBox.Show(x.Message.ToString());
        }
    }
```

```
        private void send_Click(object sender, EventArgs e)  //send button
        {
            if(txtType.Text != "")
            {
                text_to_send = txtType.Text;
                backgroundWorker2.RunWorkerAsync();
            }
            txtType.Text = "";
        }
    }
}
```