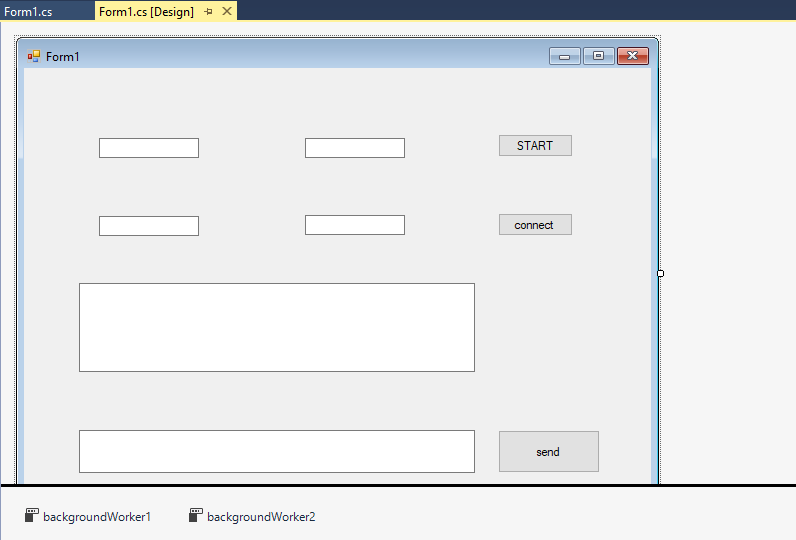
**LAB 10**

A Socket is an End-Point of To and From (Bidirectional) communication link between two programs (Server Program and Client Program ) running on the same network . We need two programs for communicating a socket application in C#. A Server Socket Program ( Server ) and a Client Socket Program ( Client ).  
So let's develop a Client Server Windows Forms C# application (chat application). First let's create this Windows Form:



TextBoxes names : ServerIPtextBox, ServerPorttextBox, ClientIPtextBox, ClientPorttextBox, ChatScreentextBox, MessagetextBox.  
Then add two BackgroundWorker to the project.

**And add these namespaces to the project:**

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 WindowsFormsApplication1

{

public partial class Form1 : Form

{

**After that, declare these variables:**

private TcpClient client;

public StreamReader STR;

public StreamWriter STW;

public string recieve;

public String TextToSend;

public Form1()

{

InitializeComponent();

IPAddress[] localIP = Dns.GetHostAddresses(Dns.GetHostName());

foreach (IPAddress address in localIP)

{

if (address.AddressFamily == AddressFamily.InterNetwork)

{

ServerIPtextBox.Text = address.ToString();

}

}

}

**In StartButton click event add this code:**

private void button1\_Click(object sender, EventArgs e)

{

TcpListener listener = new TcpListener(IPAddress.Any, int.Parse(ServerPorttextBox.Text));

listener.Start();

client = listener.AcceptTcpClient();

STR = new StreamReader(client.GetStream());

STW = new StreamWriter(client.GetStream());

STW.AutoFlush = true;

backgroundWorker1.RunWorkerAsync();

backgroundWorker2.WorkerSupportsCancellation = true;

}

**In ConnectButton click event add this code:**

private void button2\_Click(object sender, EventArgs e)

{

client = new TcpClient();

IPEndPoint IpEnd = new IPEndPoint(IPAddress.Parse(ClientIPtextBox.Text), int.Parse(ClientPorttextBox.Text));

try

{

client.Connect(IpEnd);

if (client.Connected)

{

ChatScreentextBox.AppendText("Connected to server" + "\n");

STW = new StreamWriter(client.GetStream());

STR = new StreamReader(client.GetStream());

STW.AutoFlush = true;

backgroundWorker1.RunWorkerAsync();

backgroundWorker2.WorkerSupportsCancellation = true;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

**In backgroundWorker1 DoWork event add this code:**

private void backgroundWorker1\_DoWork(object sender, DoWorkEventArgs e)

{

while (client.Connected)

{

try

{

recieve = STR.ReadLine();

this.ChatScreentextBox.Invoke(new MethodInvoker(delegate()

{

ChatScreentextBox.AppendText("You:" + recieve + "\n");

}));

recieve = "";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

**And in backgroundWorker2 DoWork event add this code:**

private void backgroundWorker2\_DoWork(object sender, DoWorkEventArgs e)

{

if (client.Connected)

{

STW.WriteLine(TextToSend);

this.ChatScreentextBox.Invoke(new MethodInvoker(delegate()

{

ChatScreentextBox.AppendText("Me:" + TextToSend + "\n");

}));

}

else

{

MessageBox.Show("Sending failed");

}

backgroundWorker2.CancelAsync();

}

**Finally in SendButton click event add this code:**

private void button3\_Click(object sender, EventArgs e)

{

if (MessagetextBox.Text != "")

{

TextToSend = MessagetextBox.Text;

backgroundWorker2.RunWorkerAsync();

}

MessagetextBox.Text = "";

}

}

}

*Then let's run the application and open a second instance to it from (%project directory%\bin\Debug). Open two instances for testing purposes in your computer, then give the IP address and Port number to the two instances. Then connect both Applications and start sending message*

**Finally this is our output:**

