**1b.Create a Chat Application using C# Windows Application using Kafka and consume the message in different client applications.**

**Chatsender:**

using System;

using System.Drawing;

using System.Threading.Tasks;

using System.Windows.Forms;

using Confluent.Kafka;

namespace KafkaWinChatSender

{

public class SenderForm : Form

{

private TextBox txtMessage;

private Button btnSend;

public SenderForm()

{

this.Text = "Kafka Chat - Sender";

this.Size = new Size(400, 150);

txtMessage = new TextBox { Location = new Point(20, 20), Size = new Size(250, 25) };

btnSend = new Button { Text = "Send", Location = new Point(280, 20), Size = new Size(75, 25) };

btnSend.Click += BtnSend\_Click;

Controls.Add(txtMessage);

Controls.Add(btnSend);

}

private async void BtnSend\_Click(object sender, EventArgs e)

{

var message = txtMessage.Text.Trim();

if (!string.IsNullOrEmpty(message))

{

await SendMessage("Sender: " + message);

txtMessage.Clear();

}

}

private async Task SendMessage(string message)

{

var config = new ProducerConfig { BootstrapServers = "localhost:9092" };

using var producer = new ProducerBuilder<Null, string>(config).Build();

await producer.ProduceAsync("chat-topic", new Message<Null, string> { Value = message });

}

[STAThread]

public static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new SenderForm());

}

}

}

**ChatReceiver:**

using System;

using System.Drawing;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

using Confluent.Kafka;

namespace KafkaWinChatReceiver

{

public class ReceiverForm : Form

{

private ListBox lstMessages;

public ReceiverForm()

{

this.Text = "Kafka Chat - Receiver";

this.Size = new Size(400, 500);

lstMessages = new ListBox

{

Location = new Point(20, 20),

Size = new Size(340, 400)

};

Controls.Add(lstMessages);

StartConsumer();

}

private void StartConsumer()

{

Task.Run(() =>

{

var config = new ConsumerConfig

{

GroupId = "receiver-group",

BootstrapServers = "localhost:9092",

AutoOffsetReset = AutoOffsetReset.Earliest

};

using var consumer = new ConsumerBuilder<Ignore, string>(config).Build();

consumer.Subscribe("chat-topic");

while (true)

{

var result = consumer.Consume(CancellationToken.None);

Invoke((MethodInvoker)(() =>

{

lstMessages.Items.Add(result.Message.Value);

}));

}

});

}

[STAThread]

public static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new ReceiverForm());

}

}

}

**Output:**



