namespace KafkaChatWin{

partial class Form1

    {

        private System.ComponentModel.IContainer components = null;

        private System.Windows.Forms.TextBox txtChat;

        private System.Windows.Forms.TextBox txtMessage;

        private System.Windows.Forms.Button btnSend;

        protected override void Dispose(bool disposing)

        {

            if (disposing && (components != null))

                components.Dispose();

            base.Dispose(disposing);

        }

        private void InitializeComponent()

        {

            this.txtChat = new System.Windows.Forms.TextBox();

            this.txtMessage = new System.Windows.Forms.TextBox();

            this.btnSend = new System.Windows.Forms.Button();

            this.SuspendLayout();

            // txtChat

            this.txtChat.Dock = System.Windows.Forms.DockStyle.Top;

            this.txtChat.Multiline = true;

            this.txtChat.ReadOnly = true;

            this.txtChat.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;

            this.txtChat.Height = 300;

            this.txtChat.Name = "txtChat";

            // txtMessage

            this.txtMessage.Dock = System.Windows.Forms.DockStyle.Bottom;

            this.txtMessage.Multiline = false;

            this.txtMessage.Name = "txtMessage";

            // btnSend

            this.btnSend.Dock = System.Windows.Forms.DockStyle.Bottom;

            this.btnSend.Text = "Send";

            this.btnSend.Name = "btnSend";

            this.btnSend.Click += new System.EventHandler(this.btnSend\_Click);

            // Form1

            this.ClientSize = new System.Drawing.Size(600, 400);

            this.Controls.Add(this.txtChat);

            this.Controls.Add(this.btnSend);

            this.Controls.Add(this.txtMessage);

            this.Name = "Form1";

            this.Text = "Kafka Chat";

            this.ResumeLayout(false);

            this.PerformLayout();

        }

    }

}

namespace KafkaChatWin;

static class Program

{

    /// <summary>

    ///  The main entry point for the application.

    /// </summary>

    [STAThread]

    static void Main()

    {

        // To customize application configuration such as set high DPI settings or default font,

        // see https://aka.ms/applicationconfiguration.

        ApplicationConfiguration.Initialize();

        Application.Run(new Form1());

    }

}

using System;

using System.Windows.Forms;

using System.Threading.Tasks;

using Confluent.Kafka;

namespace KafkaChatWin

{

    public partial class Form1 : Form

    {

        private readonly string kafkaTopic = "chat-topic";

        private readonly string bootstrapServers = "localhost:9092";

        private string username;

        public Form1()

        {

            InitializeComponent();

            PromptForUsername();

            StartConsumer(); // Updated to ignore old messages

        }

        private void PromptForUsername()

        {

            username = Microsoft.VisualBasic.Interaction.InputBox("Enter your name:", "Username", "User");

            if (string.IsNullOrWhiteSpace(username))

            {

                MessageBox.Show("Username is required to start chat.");

                Environment.Exit(0);

            }

        }

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

        {

            string message = txtMessage.Text.Trim();

            if (string.IsNullOrEmpty(message))

                return;

            string fullMessage = $"{username}: {message}";

            var config = new ProducerConfig { BootstrapServers = bootstrapServers };

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

            try

            {

                await producer.ProduceAsync(kafkaTopic, new Message<Null, string> { Value = fullMessage });

                txtChat.AppendText("[SENT] " + fullMessage + Environment.NewLine);

                txtMessage.Clear();

            }

            catch (Exception ex)

            {

                MessageBox.Show("Error sending message: " + ex.Message);

            }

        }

        private void StartConsumer(){

            var config = new ConsumerConfig{

    BootstrapServers = bootstrapServers,

    GroupId = "chat-client-" + Guid.NewGuid(), // make group ID unique every time

    AutoOffsetReset = AutoOffsetReset.Latest,

    EnableAutoCommit = false};

            Task.Run(() =>{

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

                consumer.Subscribe(kafkaTopic);

                try{

                    while (true){

                        var cr = consumer.Consume();

                        this.Invoke((MethodInvoker)(() =>{

                            txtChat.AppendText("[RECEIVED] " + cr.Message.Value + Environment.NewLine);

                        }));

                    }

                }

                catch (OperationCanceledException) { }

                finally

                {

                    consumer.Close();

                }

            });

        }

    }

}











