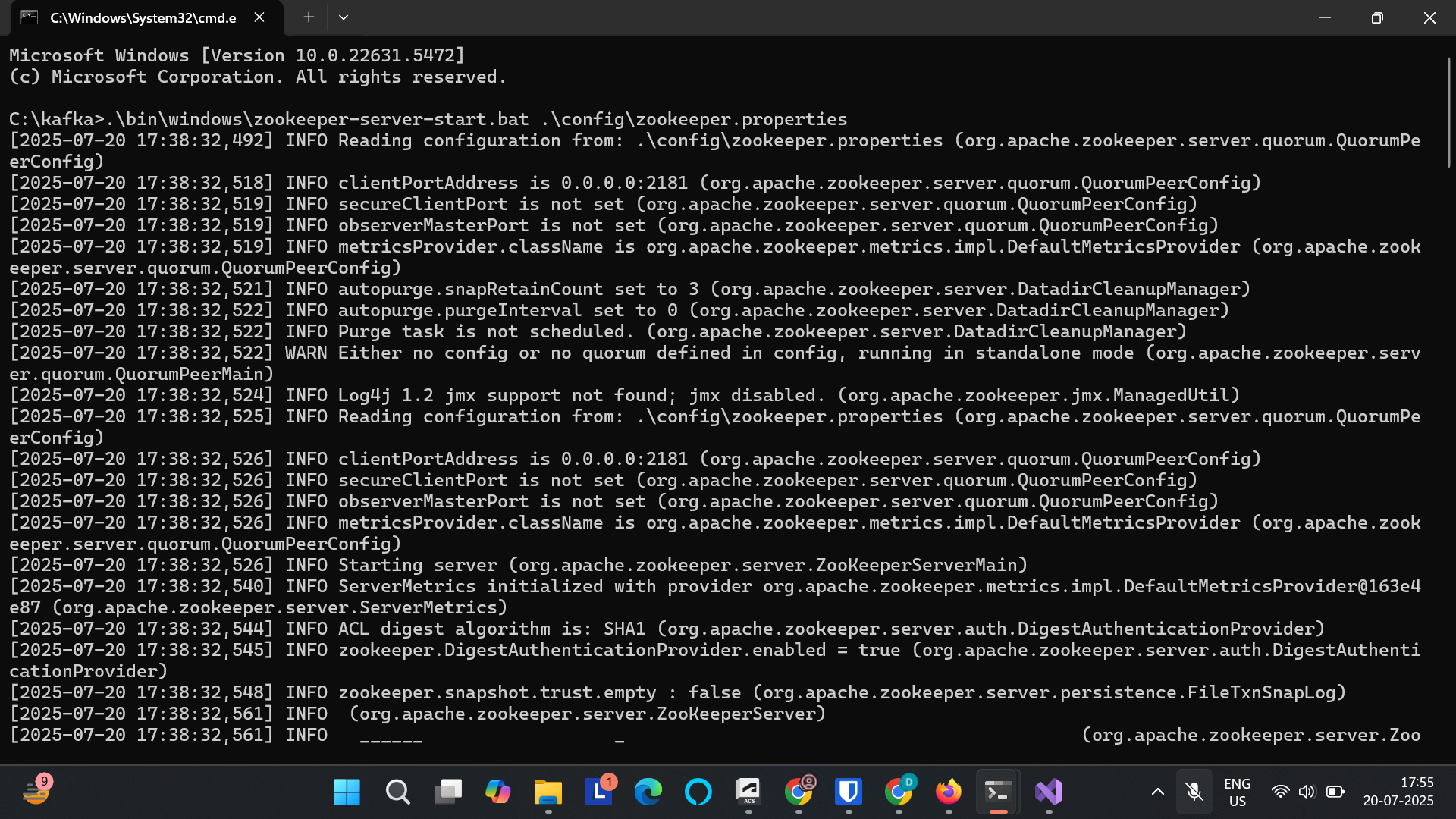
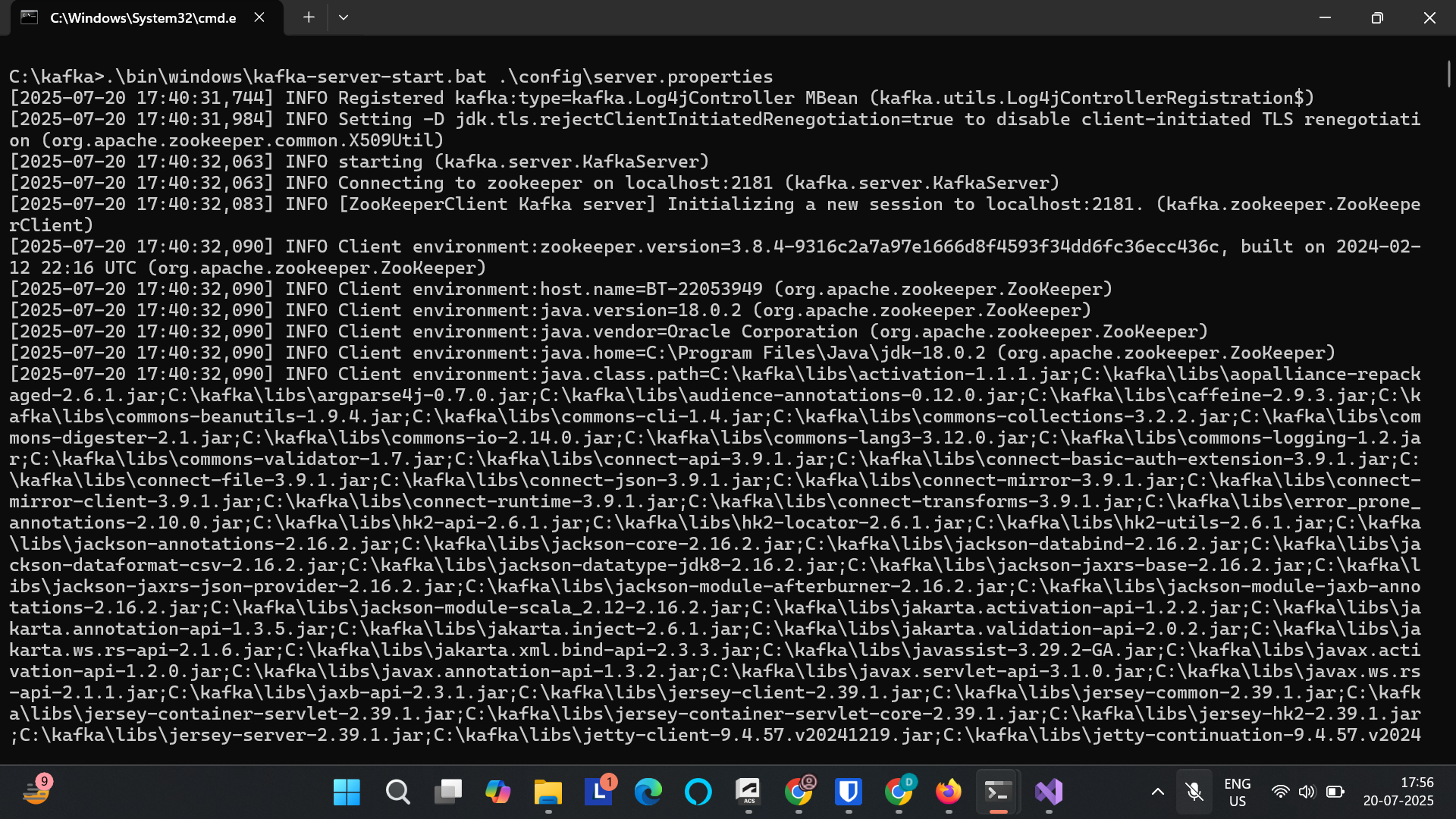
**6. WebApi\_Handson**

1. Create a Chat Application which uses Kafka as a streaming platform and consume the chat messages in the command prompt.

Starting zookeeper :



Starting kafka server:



**Code:**

[KafkaProducer.cs](http://kafkaproducer.cs)

using Confluent.Kafka;

class Program

{

static async Task Main()

{

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

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

Console.WriteLine("Type messages to send to Kafka (type 'exit' to quit):");

while (true)

{

var input = Console.ReadLine();

if (input?.ToLower() == "exit") break;

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

}

}

}

[KafkaConsumer.cs](http://kafkaconsumer.cs)

using Confluent.Kafka;

class Program

{

static void Main()

{

var config = new ConsumerConfig

{

BootstrapServers = "localhost:9092",

GroupId = "chat-consumer-group",

AutoOffsetReset = AutoOffsetReset.Earliest

};

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

consumer.Subscribe("chat-topic");

Console.WriteLine("Receiving messages... (Ctrl+C to quit)");

try

{

while (true)

{

var msg = consumer.Consume();

Console.WriteLine($"Received: {msg.Message.Value}");

}

}

catch (OperationCanceledException)

{

consumer.Close();

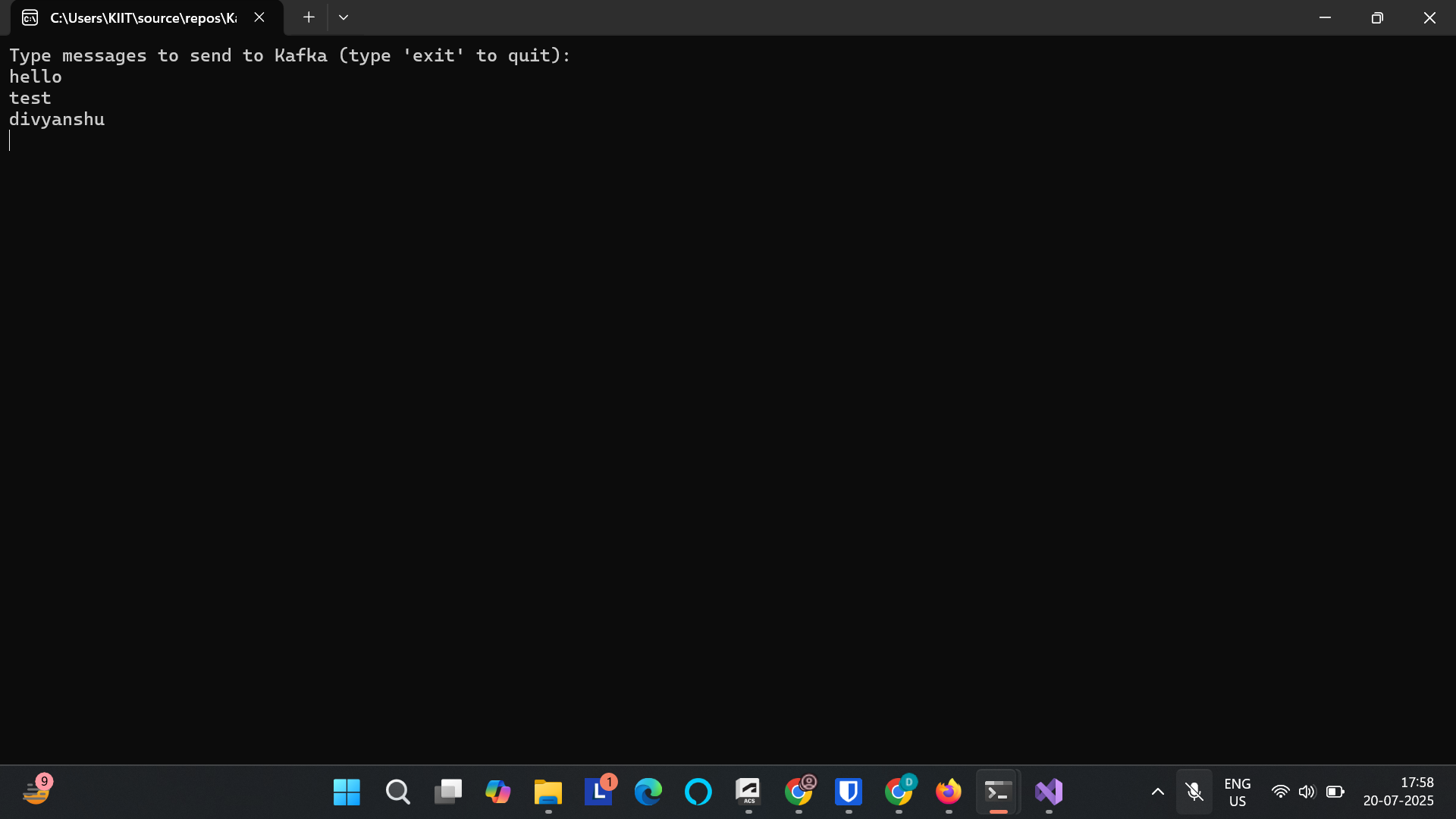
}

}

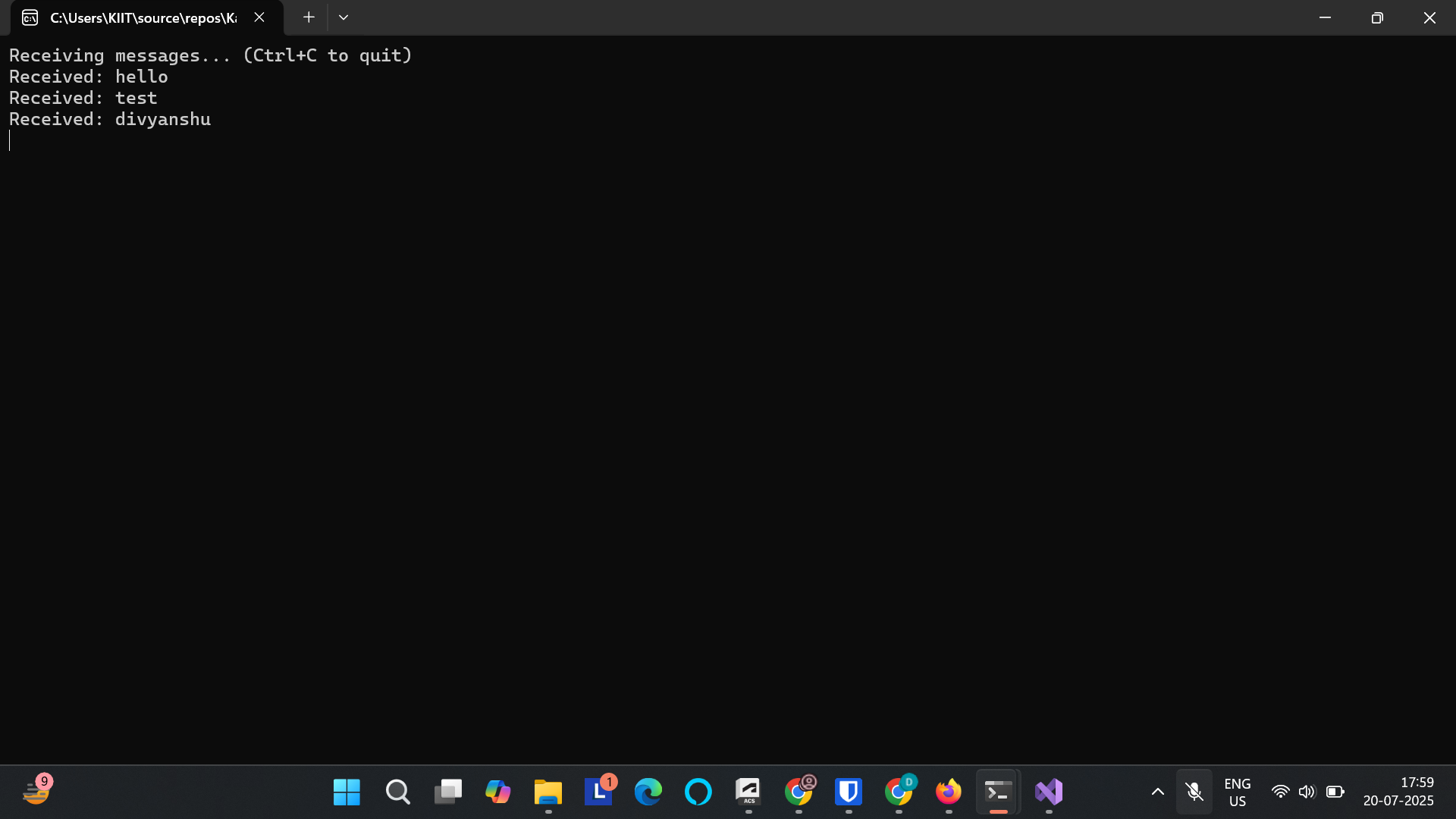
}

**Output:**

Kafka producer:



Kafka Consumer:



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

**Code:**

[**Form1.cs**](http://form1.cs)

using System;

using System.Windows.Forms;

using System.Threading.Tasks;

using Confluent.Kafka;

namespace KafkaWinFormsChat

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent(); // initializes controls defined in the Designer

}

private void Form1\_Load(object sender, EventArgs e)

{

Task.Run(() =>

{

var config = new ConsumerConfig

{

BootstrapServers = "localhost:9092",

GroupId = "chat-group-" + Guid.NewGuid().ToString(),

AutoOffsetReset = AutoOffsetReset.Earliest

};

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

consumer.Subscribe("chat-topic");

while (true)

{

try

{

var cr = consumer.Consume();

Invoke((MethodInvoker)(() =>

{

txtChat.AppendText($"Received: {cr.Message.Value}{Environment.NewLine}");

}));

}

catch (Exception)

{

// Optionally handle exceptions

}

}

});

}

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

{

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 = txtMessage.Text });

}

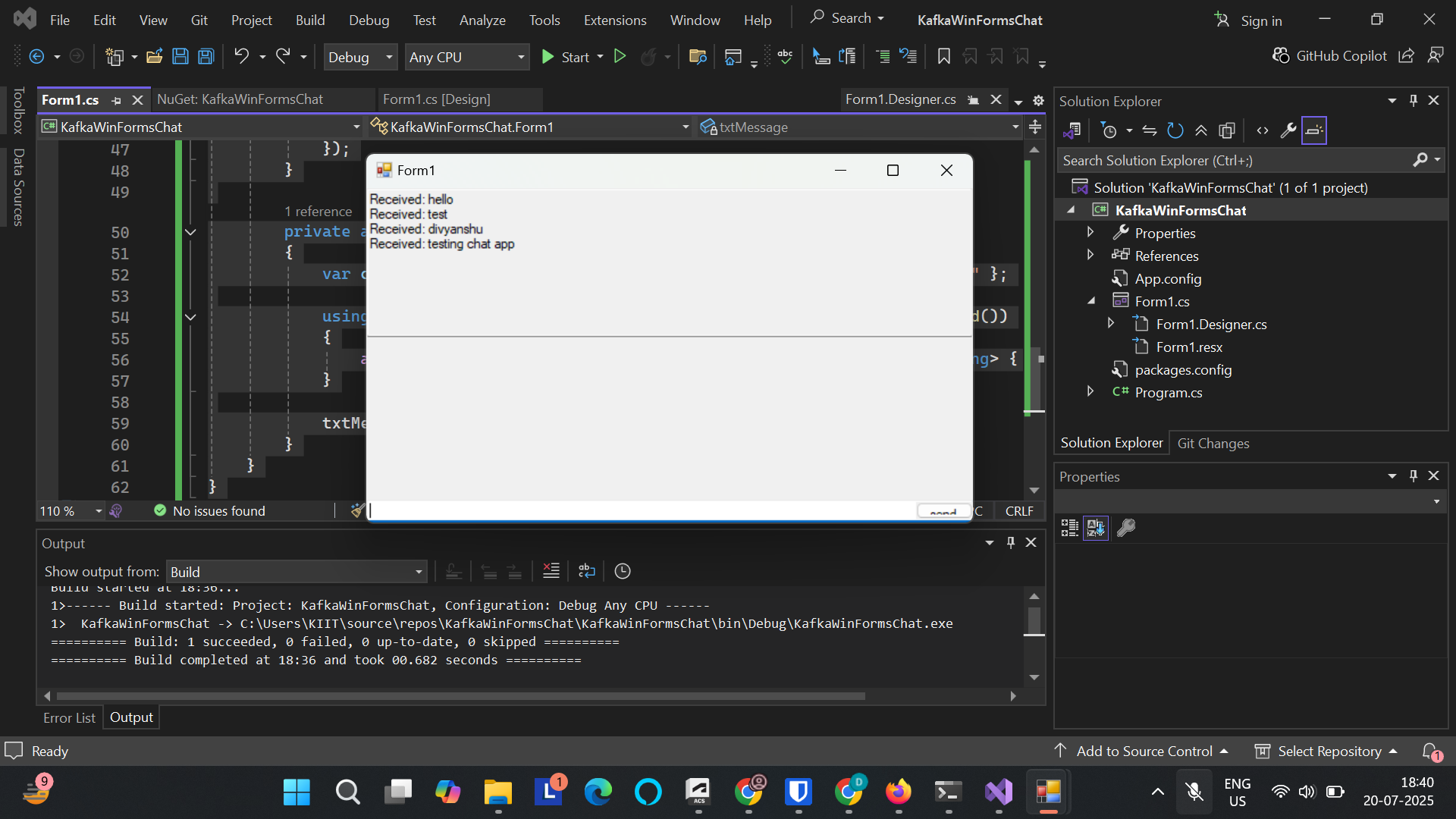
txtMessage.Clear();

}

}

}

**Output:**

****