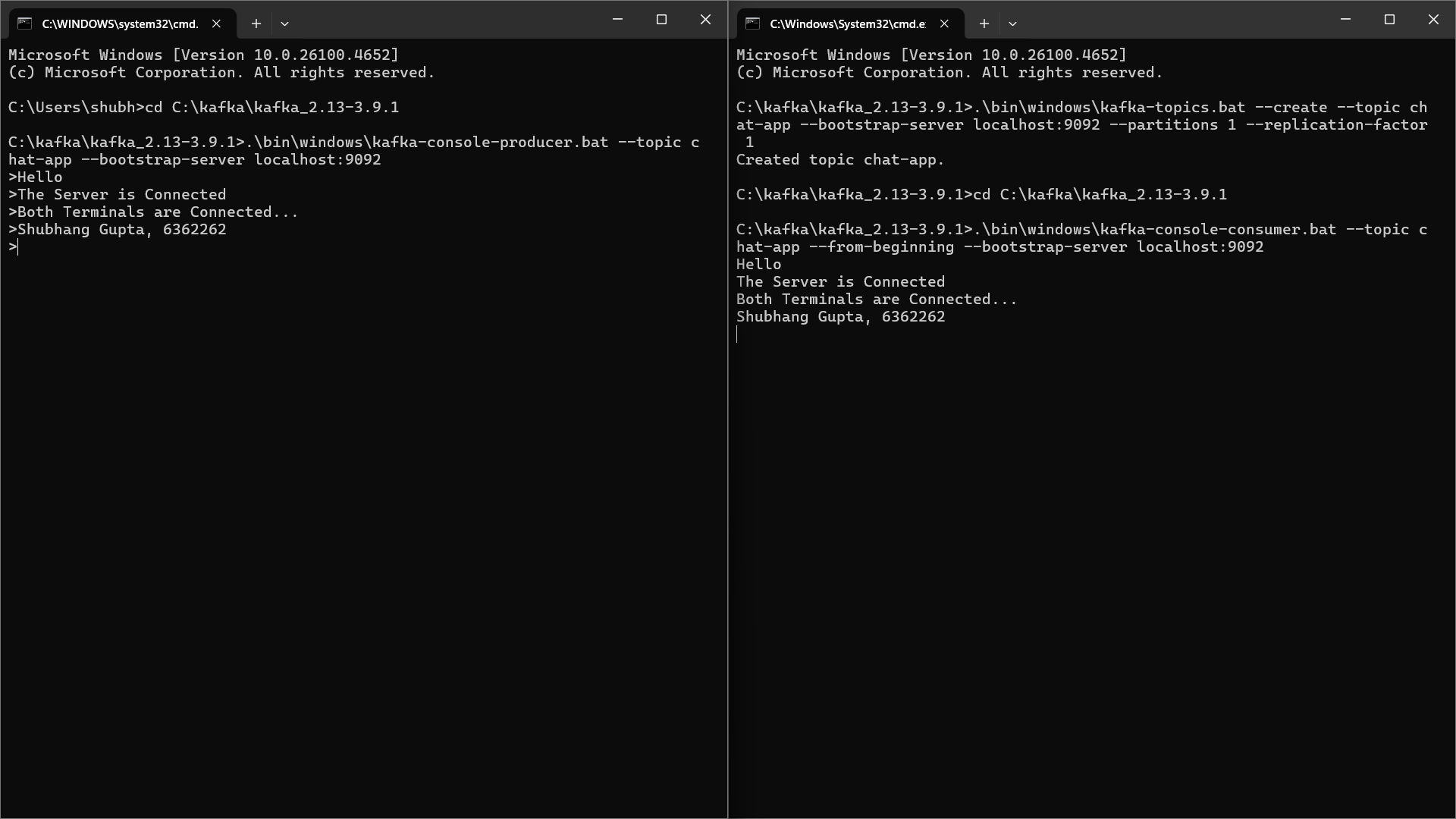
Week 5

Web\_API\_HandsOn\_6 (Kafka Integration with C#)

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

Chat Message Screenshot :



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

Producer Code (Program.cs) :

using System;

using System.Threading.Tasks;

using Confluent.Kafka;

class Program

{

public static async Task Main(string[] args)

{

var config = new ProducerConfig

{

BootstrapServers = "localhost:9092"

};

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

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

while (true)

{

var messageValue = Console.ReadLine();

if (messageValue == "exit") break;

var result = await producer.ProduceAsync("test-topic", new Message<Null, string> { Value = messageValue });

Console.WriteLine($"Message sent to: {result.TopicPartitionOffset}");

}

}

}

Producer Code Terminal Screenshot :

A screenshot of a computer

AI-generated content may be incorrect.

Consumer Code (Program.cs) :

using System;

using Confluent.Kafka;

class Program

{

static void Main(string[] args)

{

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("test-topic");

Console.WriteLine("Listening to chat messages...");

try

{

while (true)

{

var consumeResult = consumer.Consume();

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

}

}

catch (OperationCanceledException)

{

consumer.Close();

}

}

}

Consumer Code Terminal Screenshot :

A screenshot of a computer

AI-generated content may be incorrect.