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

**Producer.cs:**

using System;

using Confluent.Kafka;

class Producer

{

public static async Task Start()

{

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

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

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

while (true)

{

var message = Console.ReadLine();

if (message == "exit") break;

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

}

}

}

**Consumer.cs:**

using System;

using Confluent.Kafka;

class Consumer

{

public static void Start()

{

var config = new ConsumerConfig

{

BootstrapServers = "localhost:9092",

GroupId = "chat-group",

AutoOffsetReset = AutoOffsetReset.Earliest

};

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

consumer.Subscribe("chat-topic");

Console.WriteLine("Waiting for messages... (Ctrl+C to exit)");

try

{

while (true)

{

var cr = consumer.Consume();

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

}

}

catch (OperationCanceledException)

{

consumer.Close();

}

}

}

**Program.cs:**

using System;

class Program

{

static async Task Main(string[] args)

{

Console.WriteLine("Type 'p' for producer or 'c' for consumer:");

var choice = Console.ReadLine();

if (choice == "p")

await Producer.Start();

else if (choice == "c")

Consumer.Start();

else

Console.WriteLine("Invalid choice.");

}

}

**Output:**



