**Week-5 Hands-on**

**Lab-6**

**Sender.cs:**

﻿using Confluent.Kafka;

class Sender

{

static async Task Main(string[] args)

{

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

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

Console.WriteLine("Type a message :");

while (true)

{

string? input = Console.ReadLine();

if (string.IsNullOrWhiteSpace(input)) continue;

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

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

Console.WriteLine($"Sent: {result.Value}");

}

}

}

﻿**Receiver.cs:**

using Confluent.Kafka;

class Receiver

{

static void Main(string[] args)

{

var config = new ConsumerConfig

{

BootstrapServers = "localhost:9092",

GroupId = "chat-consumer-group",

AutoOffsetReset = AutoOffsetReset.Earliest

};

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

receiver.Subscribe("chat-topic");

Console.WriteLine("Listening for messages....");

while (true)

{

var consumeResult = receiver.Consume();

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

}

}

}

**Kafkareceiver.csproj:**

﻿<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>net9.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Confluent.Kafka" Version="2.11.0" />

</ItemGroup>

</Project>

**Kafkasender.csproj:**

﻿<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>net9.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Confluent.Kafka" Version="2.11.0" />

</ItemGroup>

</Project>

