SuperSet ID: 6365365

1. WebApi\_Handson

KafkaConsumer

KafkaConsumer.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>

Program.cs

using System;

using System.Threading;

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("chat-messages");

        Console.WriteLine("🎧 Kafka Chat Consumer Started!");

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

        var cts = new CancellationTokenSource();

        Console.CancelKeyPress += (\_, e) => {

            e.Cancel = true;

            cts.Cancel();

        };

        try

        {

            while (!cts.Token.IsCancellationRequested)

            {

                try

                {

                    var consumeResult = consumer.Consume(cts.Token);

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

                }

                catch (ConsumeException e)

                {

                    Console.WriteLine($"❌ Consume error: {e.Error.Reason}");

                }

            }

        }

        catch (OperationCanceledException)

        {

            Console.WriteLine("🛑 Consumer canceled.");

        }

        finally

        {

            consumer.Close();

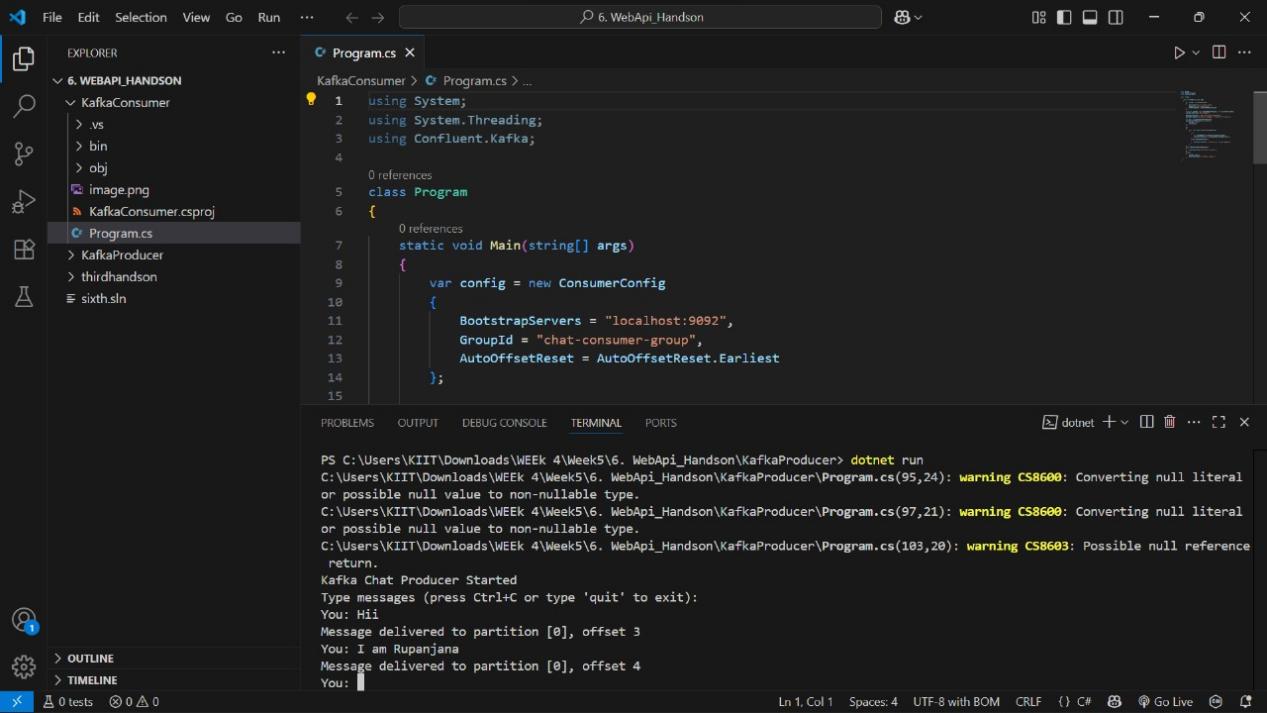
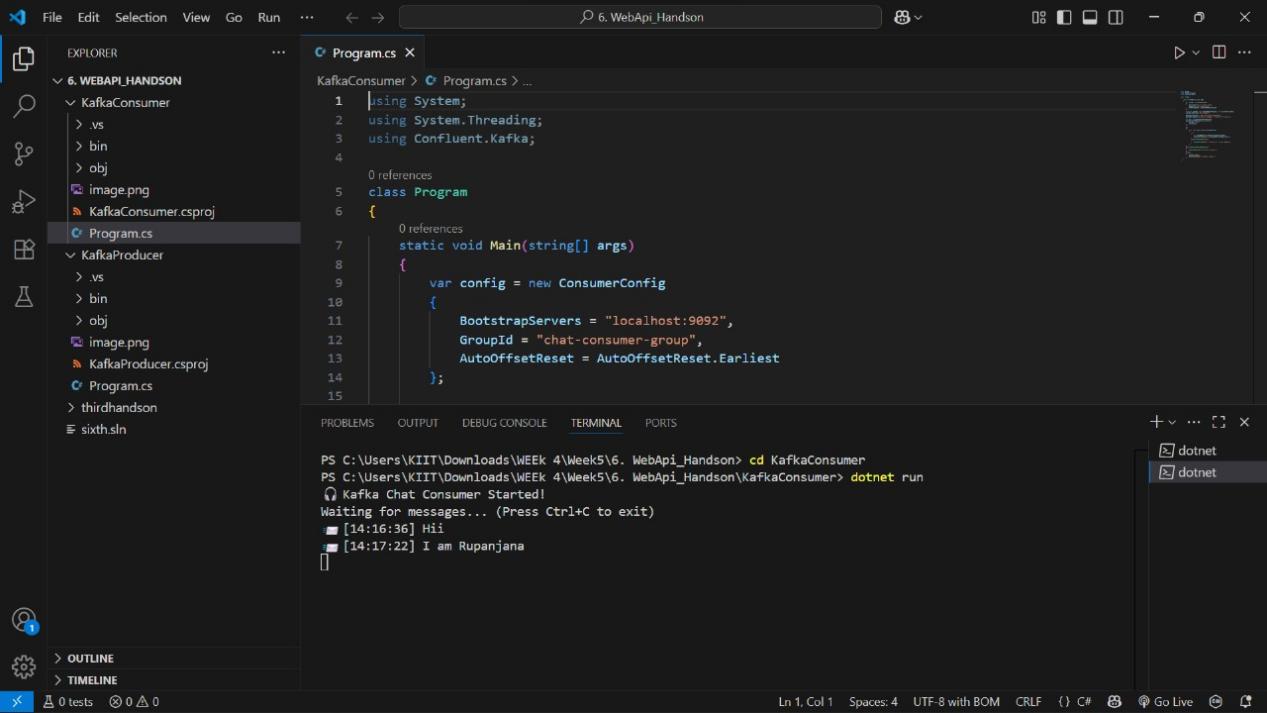
            Console.WriteLine("✅ Consumer closed.");

        }

    }

}

Output



MicroServices JWT

Fifth.csproj

<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>

    <TargetFramework>net9.0</TargetFramework>

    <Nullable>enable</Nullable>

    <ImplicitUsings>enable</ImplicitUsings>

  </PropertyGroup>

  <ItemGroup>

    <PackageReference Include="Microsoft.AspNetCore.Authentication.JwtBearer" Version="9.0.7" />

    <PackageReference Include="Microsoft.AspNetCore.Cors" Version="2.3.0" />

    <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="9.0.6" />

    <PackageReference Include="Swashbuckle.AspNetCore" Version="9.0.3" />

    <PackageReference Include="System.IdentityModel.Tokens.Jwt" Version="8.12.1" />

  </ItemGroup>

</Project>

Program.cs

using Microsoft.AspNetCore.Authentication.JwtBearer;

using Microsoft.IdentityModel.Tokens;

using System.Text;

var builder = WebApplication.CreateBuilder(args);

// Add JWT Authentication

builder.Services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)

    .AddJwtBearer(options =>

    {

        options.TokenValidationParameters = new TokenValidationParameters

        {

            ValidateIssuer = true,

            ValidateAudience = true,

            ValidateLifetime = true,

            ValidateIssuerSigningKey = true,

            ValidIssuer = builder.Configuration["Jwt:Issuer"],

            ValidAudience = builder.Configuration["Jwt:Audience"],

            IssuerSigningKey = new SymmetricSecurityKey(

                Encoding.UTF8.GetBytes(builder.Configuration["Jwt:Key"]))

        };

    });

builder.Services.AddAuthorization();

var app = builder.Build();

// Middleware

app.UseAuthentication();

app.UseAuthorization();

app.Run();

Output