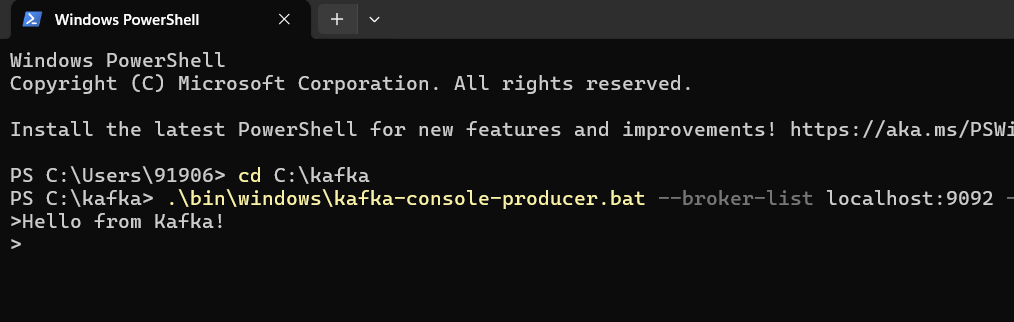
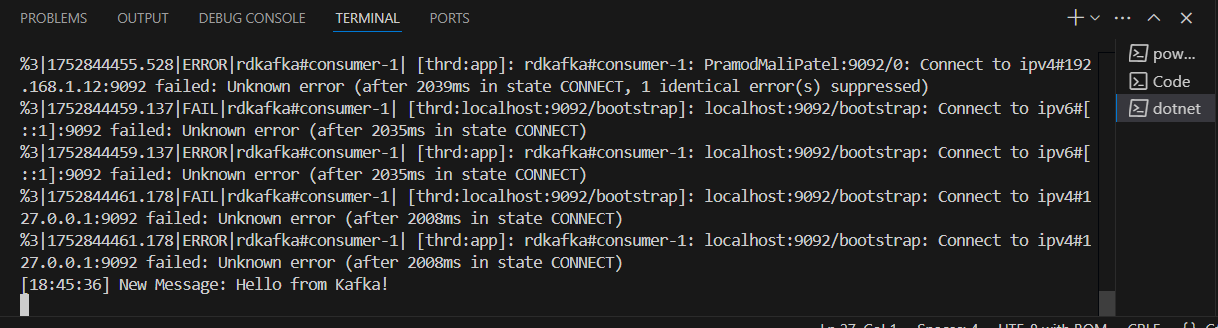
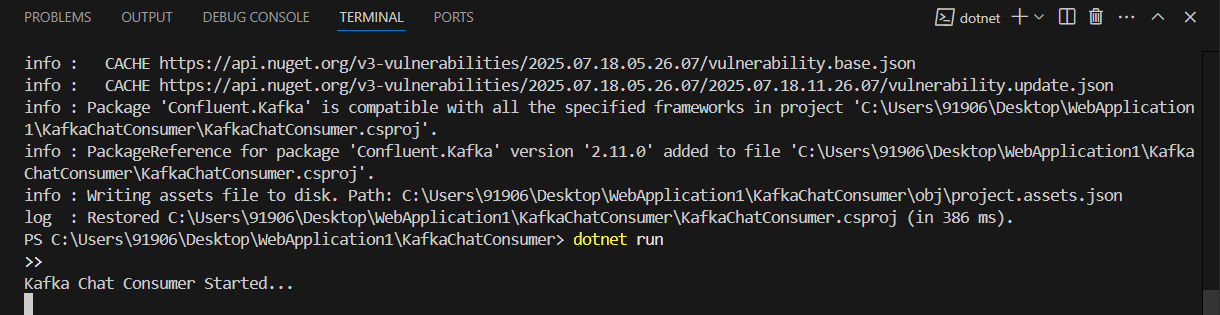
**WEB API 6:**

****

**CODE:**

**using Confluent.Kafka;**

**Console.WriteLine("Kafka Chat Consumer Started...");**

**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");**

**try**

**{**

**while (true)**

**{**

**var cr = consumer.Consume();**

**Console.WriteLine($"[{DateTime.Now:T}] New Message: {cr.Message.Value}");**

**}**

**}**

**catch (OperationCanceledException)**

**{**

**consumer.Close();**

**}**

**MICROSERVICES:**

**auth**

**using Microsoft.AspNetCore.Mvc;**

**using Microsoft.IdentityModel.Tokens;**

**using System.IdentityModel.Tokens.Jwt;**

**using System.Security.Claims;**

**using System.Text;**

**using JwtAuthDemo.Models;**

**namespace JwtAuthDemo.Controllers**

**{**

**[ApiController]**

**[Route("api/[controller]")]**

**public class AuthController : ControllerBase**

**{**

**private readonly IConfiguration \_config;**

**public AuthController(IConfiguration config)**

**{**

**\_config = config;**

**}**

**[HttpPost("login")]**

**public IActionResult Login([FromBody] LoginModel login)**

**{**

**if (login.Username == "admin" && login.Password == "password")**

**{**

**var token = GenerateJwtToken(login.Username);**

**return Ok(new { token });**

**}**

**return Unauthorized("Invalid credentials");**

**}**

**private string GenerateJwtToken(string username)**

**{**

**var key = \_config["Jwt:Key"];**

**var issuer = \_config["Jwt:Issuer"];**

**var audience = \_config["Jwt:Audience"];**

**var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(key));**

**var credentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256);**

**var claims = new[]**

**{**

**new Claim(ClaimTypes.Name, username),**

**new Claim(ClaimTypes.Role, "Admin")**

**};**

**var token = new JwtSecurityToken(**

**issuer: issuer,**

**audience: audience,**

**claims: claims,**

**expires: DateTime.Now.AddMinutes(30),**

**signingCredentials: credentials**

**);**

**return new JwtSecurityTokenHandler().WriteToken(token);**

**}**

**}**

**}**

**Loginmodel.cs:**

**namespace JwtAuthDemo.Models**

**{**

**public class LoginModel**

**{**

**public string Username { get; set; }**

**public string Password { get; set; }**

**}**

**}**

**Program.cs:**

**using Microsoft.AspNetCore.Authentication.JwtBearer;**

**using Microsoft.IdentityModel.Tokens;**

**using Microsoft.OpenApi.Models;**

**using System.Text;**

**var builder = WebApplication.CreateBuilder(args);**

**// ======== JWT Authentication Configuration ========**

**var key = builder.Configuration["Jwt:Key"] ?? "ThisIsASuperSecretKeyForJwtToken";**

**builder.Services.AddAuthentication(options =>**

**{**

**options.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;**

**options.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;**

**})**

**.AddJwtBearer(options =>**

**{**

**options.RequireHttpsMetadata = false;**

**options.SaveToken = true;**

**options.TokenValidationParameters = new TokenValidationParameters**

**{**

**ValidateIssuer = true,**

**ValidateAudience = true,**

**ValidateLifetime = true,**

**ValidateIssuerSigningKey = true,**

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

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

**IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(key))**

**};**

**});**

**// ======== Add Controllers ========**

**builder.Services.AddControllers();**

**// ======== Add Swagger ========**

**builder.Services.AddEndpointsApiExplorer();**

**builder.Services.AddSwaggerGen(c =>**

**{**

**// Add JWT Auth in Swagger UI**

**c.SwaggerDoc("v1", new OpenApiInfo { Title = "JwtAuthDemo API", Version = "v1" });**

**c.AddSecurityDefinition("Bearer", new OpenApiSecurityScheme**

**{**

**Name = "Authorization",**

**Type = SecuritySchemeType.ApiKey,**

**Scheme = "Bearer",**

**BearerFormat = "JWT",**

**In = ParameterLocation.Header,**

**Description = "Enter 'Bearer' [space] and then your token in the text input below.\n\nExample: `Bearer eyJhbGci...`"**

**});**

**c.AddSecurityRequirement(new OpenApiSecurityRequirement**

**{**

**{**

**new OpenApiSecurityScheme**

**{**

**Reference = new OpenApiReference { Type = ReferenceType.SecurityScheme, Id = "Bearer" }**

**},**

**Array.Empty<string>()**

**}**

**});**

**});**

**var app = builder.Build();**

**// ======== Middleware Pipeline ========**

**if (app.Environment.IsDevelopment())**

**{**

**app.UseSwagger();**

**app.UseSwaggerUI();**

**}**

**app.UseHttpsRedirection();**

**app.UseAuthentication();**

**app.UseAuthorization();**

**app.MapControllers();**

**app.Run();**

**Appsettings.json:**

**{**

**"Jwt": {**

**"Key": "ThisIsMySuperSecretKeyForJwt",**

**"Issuer": "JwtAuthDemo",**

**"Audience": "JwtAuthDemoUser",**

**"Subject": "JwtAuthAccessToken"**

**},**

**"Logging": {**

**"LogLevel": {**

**"Default": "Information",**

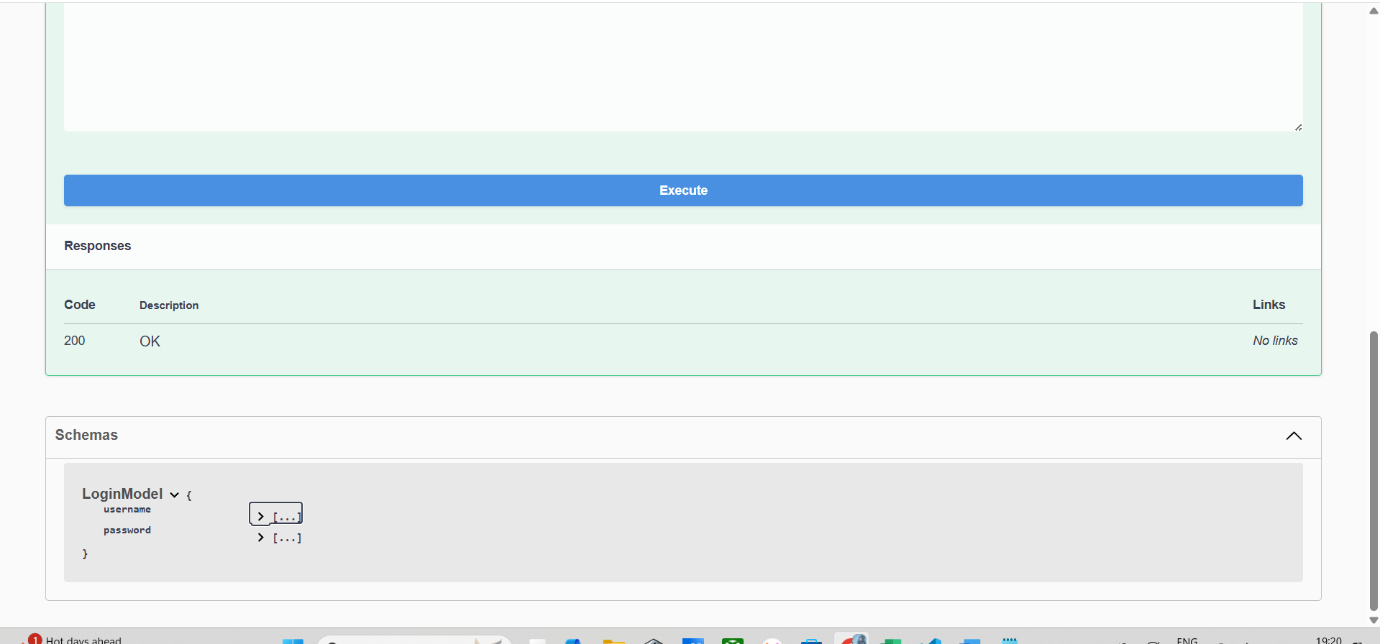
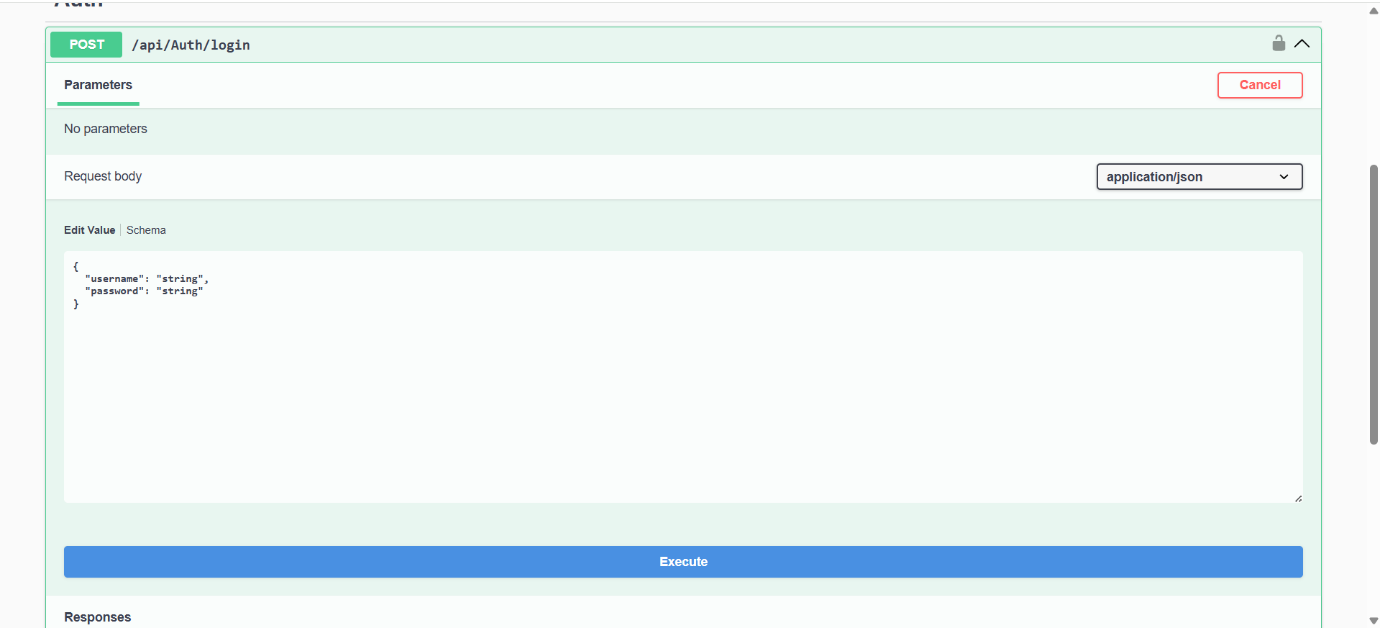
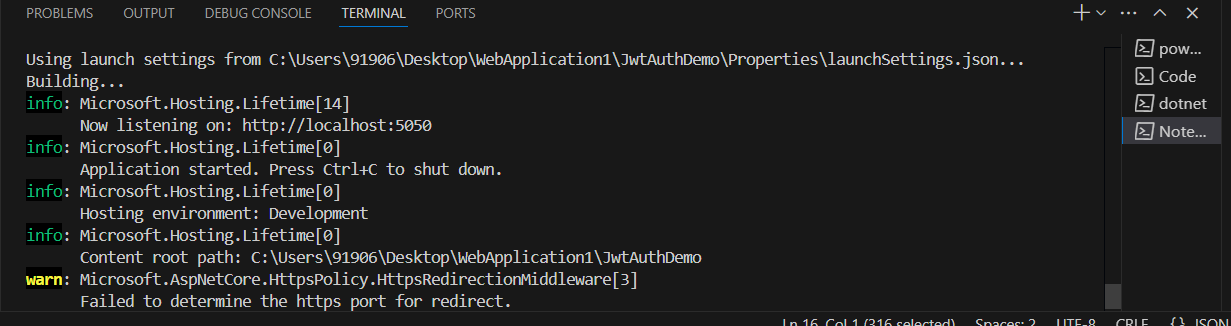
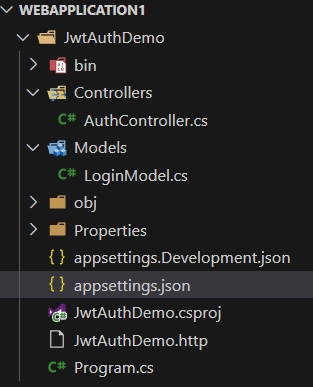
**"Microsoft.AspNetCore": "Warning"**

**}**

**},**

**"AllowedHosts": "\*"**

**}**

****