**WEEK – 5-Microservices Architecture using ASP.NET Core WebAPI**

1. **Microservices – JWT**

**Question 1: Implement JWT Authentication in ASP.NET Core Web API**

**appsettings.json**

{  
 "Logging": {  
 "LogLevel": {  
 "Default": "Information",  
 "Microsoft.AspNetCore": "Warning"  
 }  
 },  
 "Jwt": {  
 "Key": "ThisIsMySuperSecureJwtKey\_ThatIsLongEnough123!",  
 "Issuer": "MyAuthServer",  
 "Audience": "MyApiUsers",  
 "DurationInMinutes": 60  
 },  
 "AllowedHosts": "\*"  
}

**Program.cs**

using Microsoft.AspNetCore.Authentication.JwtBearer;  
using Microsoft.IdentityModel.Tokens;  
using System.Text;  
  
var builder = WebApplication.CreateBuilder(args);  
  
// Add services to the container  
builder.Services.AddControllers();  
  
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();  
  
app.UseHttpsRedirection();  
app.UseAuthentication();  
app.UseAuthorization();  
app.MapControllers();  
app.Run();

**Models/LoginModel.cs**

namespace JwtAuthExample.Models  
{  
 public class LoginModel  
 {  
 public string Username { get; set; }  
 public string Password { get; set; }  
 }  
}

**Controllers/AuthController.cs**

using JwtAuthExample.Models;  
using Microsoft.AspNetCore.Mvc;  
using Microsoft.IdentityModel.Tokens;  
using System.IdentityModel.Tokens.Jwt;  
using System.Security.Claims;  
using System.Text;  
  
namespace JwtAuthExample.Controllers  
{  
 [ApiController]  
 [Route("api/[controller]")]  
 public class AuthController : ControllerBase  
 {  
 private readonly IConfiguration \_configuration;  
  
 public AuthController(IConfiguration configuration)  
 {  
 \_configuration = configuration;  
 }  
  
 [HttpPost("login")]  
 public IActionResult Login([FromBody] LoginModel model)  
 {  
 if (IsValidUser(model))  
 {  
 var token = GenerateJwtToken(model.Username);  
 return Ok(new { Token = token });  
 }  
 return Unauthorized();  
 }  
  
 private bool IsValidUser(LoginModel model)  
 {  
 return model.Username == "varsha" && model.Password == "1234";  
 }  
  
 private string GenerateJwtToken(string username)  
 {  
 var claims = new[]  
 {  
 new Claim(ClaimTypes.Name, username)  
 };  
  
 var key = new SymmetricSecurityKey(  
 Encoding.UTF8.GetBytes(\_configuration["Jwt:Key"]));  
 var creds = new SigningCredentials(key, SecurityAlgorithms.HmacSha256);  
  
 var token = new JwtSecurityToken(  
 issuer: \_configuration["Jwt:Issuer"],  
 audience: \_configuration["Jwt:Audience"],  
 claims: claims,  
 expires: DateTime.Now.AddMinutes(60),  
 signingCredentials: creds  
 );  
  
 return new JwtSecurityTokenHandler().WriteToken(token);  
 }  
 }  
}

**Controllers/HelloController.cs**

using Microsoft.AspNetCore.Authorization;  
using Microsoft.AspNetCore.Mvc;  
  
namespace JwtAuthExample.Controllers  
{  
 [ApiController]  
 [Route("api/[controller]")]  
 public class HelloController : ControllerBase  
 {  
 [HttpGet]  
 [Authorize]  
 public IActionResult GetSecretMessage()  
 {  
 return Ok("This is a protected message. You are authorized!");  
 }  
 }  
}

**Output:**  
  


