

DotNet-FSE Mandatory Hands-On

WEEK-5

NAME: Sri Ranjani Priya P

EXERCISE 1:

Implement a secure Web API in ASP.NET Core that supports JWT-based authentication and role-based authorization. The API should support login functionality and allow access to specific endpoints based on the user's role (e.g., Admin or User).

CODE:(C#)

File: Models/LoginModel.cs

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

File: Program.cs

```
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.IdentityModel.Tokens;
using System.Text;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddAuthentication(options =>
{
    options.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;
    options.DefaultChallengeScheme = 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();
builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
```

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

```
app.UseHttpsRedirection();
app.UseAuthentication();
app.UseAuthorization();
app.MapControllers();
```

```
app.Run();
```

FILE: appsettings.json:

```
{
  "Jwt": {
    "Key": "ThisIsASecretKeyForJwt",
    "Issuer": "https://localhost:7006",
    "Audience": "https://localhost:7006"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

FILE: Controllers/AuthController.cs:

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.IdentityModel.Tokens;
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;
using System.Text;
using JwtAuthWithRoles.Models;
```

```

namespace JwtAuthWithRoles.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 user)
        {
            if (user.Username == "admin" && user.Password == "password")
            {
                var token = GenerateJwtToken(user.Username, "Admin");
                return Ok(new { token });
            }
            else if (user.Username == "user" && user.Password == "password")
            {
                var token = GenerateJwtToken(user.Username, "User");
                return Ok(new { token });
            }

            return Unauthorized();
        }

        private string GenerateJwtToken(string username, string role)
        {
            var securityKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(_configuration["Jwt:Key"]));
            var credentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256);

            var claims = new[]
            {
                new Claim(ClaimTypes.Name, username),
                new Claim(ClaimTypes.Role, role),
                new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString())
            };

            var token = new JwtSecurityToken(

```

```

        issuer: _configuration["Jwt:Issuer"],
        audience: _configuration["Jwt:Audience"],
        claims: claims,
        expires: DateTime.Now.AddMinutes(60),
        signingCredentials: credentials
    );

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

```

FILE: Controllers/SecureController.cs

```

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;

namespace JwtAuthWithRoles.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class SecureController : ControllerBase
    {
        [HttpGet("test")]
        [Authorize]
        public IActionResult Test()
        {
            return Ok("✅ You are authorized! JWT token works.");
        }

        [HttpGet("admin")]
        [Authorize(Roles = "Admin")]
        public IActionResult AdminOnly()
        {
            return Ok("✅ Welcome, Admin!");
        }

        [HttpGet("user")]
        [Authorize(Roles = "User")]
        public IActionResult UserOnly()
        {
            return Ok("✅ Welcome, User!");
        }
    }
}

```

Login as User:

Responses

Curl

```
curl -X 'POST' \
https://localhost:7006/api/Auth/login' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "username": "sriranjani",
  "password": "sriranjani123"
}
```

Request URL

```
https://localhost:7006/api/Auth/login
```

Server response

Code	Details
200	<div><div>Response body<pre>{ "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IWRkcCkiLCJ0eXAiOiJKV1QiLCJhdWUiOiJSemFvbmVyc3Q6ZnRhbnNlLnR5cyBkaXNjaXNpdjEwLmVudDQyZWYxMDE1MTk2ODg2OTk1MDA1fQ.KUQUKE37vX--rDpYES21j3APWBrrz2VgVwZZsSuLeGEt8" }</pre></div><div><div>Download</div></div></div> <div><div>Response headers<pre>access-control-allow-origin: * content-type: application/json; charset=utf-8 date: Sun, 20 Jul 2025 07:25:20 GMT server: Kestrel</pre></div></div>

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7006/api/Secure/test' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc29udGVudG94eSI6ImFkbG8iLCJ1aWQiOiIwIiwiaWF0IjoxNjU0MjQ0MDA0In0.' \
  -H 'Content-Type: application/json'
```

Request URL

```
https://localhost:7006/api/Secure/test
```

Server response

Code	Details
200	<div><h5>Response body</h5><pre>✔ You are authorized! JWT token works.</pre><h5>Response headers</h5><pre>content-type: text/plain; charset=utf-8 date: Sun, 20 Jul 2025 08:24:18 GMT server: Kestrel</pre></div>

Responses

Code	Description
200	OK

Login as Admin:

```
{
  "username": "admin",
  "password": "password"
}
```

[illegible]

JWT Secured Endpoint – 403 Forbidden Error(for unauthorized role access):

Responses

Curl

```
curl -X 'GET' \
'https://localhost:7006/api/Secure/user' \
-H 'accept: */*' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzbnB2FwLm9yZy93cy8y'

```

Request URL

```
https://localhost:7006/api/Secure/user

```

Server response

Code	Details
403 <i>Undocumented</i>	Error: response status is 403

Response headers

```
content-length: 0
date: Sun, 20 Jul 2025 08:05:45 GMT
server: Kestrel

```

Responses

Code	Description
200	OK