1. **JsonWebToken**

**Employee.cs**

**using System;**

**using System.Collections.Generic;**

**namespace MyFirstWebAPI.Models**

**{**

**public class Employee**

**{**

**public int Id { get; set; }**

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

**public int Salary { get; set; }**

**public bool Permanent { get; set; }**

**public Department Department { get; set; }**

**public List<Skill> Skills { get; set; }**

**public DateTime DateOfBirth { get; set; }**

**}**

**}**

**Department.cs**

**namespace MyFirstWebAPI.Models**

**{**

**public class Department**

**{**

**public int Id { get; set; }**

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

**}**

**}**

**Skill.cs**

**namespace MyFirstWebAPI.Models**

**{**

**public class Skill**

**{**

**public int Id { get; set; }**

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

**}**

**}**

**AuthController**

**using Microsoft.AspNetCore.Mvc;**

**using Microsoft.IdentityModel.Tokens;**

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

**using System.Security.Claims;**

**using System.Text;**

**namespace MyFirstWebAPI.Controllers**

**{**

**[ApiController]**

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

**[AllowAnonymous]**

**public class AuthController : ControllerBase**

**{**

**[HttpGet]**

**public ActionResult<string> GetToken()**

**{**

**var token = GenerateJSONWebToken(1, "Admin"); // or "POC"**

**return Ok(token);**

**}**

**private string GenerateJSONWebToken(int userId, string userRole)**

**{**

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

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

**var claims = new List<Claim>**

**{**

**new Claim(ClaimTypes.Role, userRole),**

**new Claim("UserId", userId.ToString())**

**};**

**var token = new JwtSecurityToken(**

**issuer: "mySystem",**

**audience: "myUsers",**

**claims: claims,**

**expires: DateTime.Now.AddMinutes(2),**

**signingCredentials: credentials);**

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

**}**

**}**

**}**

**EmployeeController**

**using Microsoft.AspNetCore.Authorization;**

**using Microsoft.AspNetCore.Mvc;**

**using MyFirstWebAPI.Models;**

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**namespace MyFirstWebAPI.Controllers**

**{**

**[ApiController]**

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

**[Authorize(Roles = "Admin,POC")] // Only Admin/POC allowed**

**public class EmployeeController : ControllerBase**

**{**

**private static List<Employee> \_employees = new List<Employee>**

**{**

**new Employee**

**{**

**Id = 1,**

**Name = "John",**

**Salary = 50000,**

**Permanent = true,**

**DateOfBirth = new DateTime(1990, 1, 1),**

**Department = new Department { Id = 1, Name = "IT" },**

**Skills = new List<Skill>**

**{**

**new Skill { Id = 1, Name = "C#" },**

**new Skill { Id = 2, Name = "SQL" }**

**}**

**}**

**};**

**[HttpGet]**

**public ActionResult<List<Employee>> Get()**

**{**

**return Ok(\_employees);**

**}**

**[HttpPut("{id}")]**

**public ActionResult<Employee> Put(int id, [FromBody] Employee updatedEmployee)**

**{**

**if (id <= 0)**

**return BadRequest("Invalid employee id");**

**var employee = \_employees.FirstOrDefault(e => e.Id == id);**

**if (employee == null)**

**return BadRequest("Invalid employee id");**

**employee.Name = updatedEmployee.Name;**

**employee.Salary = updatedEmployee.Salary;**

**employee.Permanent = updatedEmployee.Permanent;**

**employee.DateOfBirth = updatedEmployee.DateOfBirth;**

**employee.Department = updatedEmployee.Department;**

**employee.Skills = updatedEmployee.Skills;**

**return Ok(employee);**

**}**

**}**

**}**

**Startup**

**using Microsoft.AspNetCore.Builder;**

**using Microsoft.AspNetCore.Hosting;**

**using Microsoft.Extensions.DependencyInjection;**

**using Microsoft.Extensions.Hosting;**

**using Microsoft.IdentityModel.Tokens;**

**using System.Text;**

**namespace MyFirstWebAPI**

**{**

**public class Startup**

**{**

**public void ConfigureServices(IServiceCollection services)**

**{**

**string securityKey = "mysuperdupersecret";**

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

**services.AddAuthentication(x =>**

**{**

**x.DefaultAuthenticateScheme = "JwtBearer";**

**x.DefaultChallengeScheme = "JwtBearer";**

**x.DefaultSignInScheme = "JwtBearer";**

**})**

**.AddJwtBearer("JwtBearer", x =>**

**{**

**x.TokenValidationParameters = new Microsoft.IdentityModel.Tokens.TokenValidationParameters**

**{**

**ValidateIssuer = true,**

**ValidateAudience = true,**

**ValidateLifetime = true,**

**ValidateIssuerSigningKey = true,**

**ValidIssuer = "mySystem",**

**ValidAudience = "myUsers",**

**IssuerSigningKey = symmetricSecurityKey**

**};**

**});**

**services.AddControllers();**

**services.AddSwaggerGen();**

**}**

**public void Configure(IApplicationBuilder app, IWebHostEnvironment env)**

**{**

**if (env.IsDevelopment())**

**app.UseDeveloperExceptionPage();**

**app.UseSwagger();**

**app.UseSwaggerUI();**

**app.UseRouting();**

**app.UseAuthentication();**

**app.UseAuthorization();**

**app.UseEndpoints(endpoints =>**

**{**

**endpoints.MapControllers();**

**});**

**}**

**}**

**}**