**1.Web Api using .Net core with Swagger**

Program.cs :

using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo

{

Title = "Swagger Demo",

Version = "v1",

Description = "TBD",

TermsOfService = new Uri("https://example.com/terms"),

Contact = new OpenApiContact

{

Name = "John Doe",

Email = "john@xyzmail.com",

Url = new Uri("https://www.example.com")

},

License = new OpenApiLicense

{

Name = "License Terms",

Url = new Uri("https://www.example.com")

}

});

});

var app = builder.Build();

app.UseSwagger();

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1/swagger.json", "Swagger Demo");

});

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

EmployeeController.cs :

namespace APIDemo.Controllers

{

    [ApiController]

    [Route("api/Emp")]  *// Custom route as per your assignment*

*public* class EmployeeController : ControllerBase

    {

*// Simulated data source*

*private* *static* *readonly* List<string> employees = new()

        {

            "Tapabrata Das",

            "Sumit Sen",

            "Tina Gupta"

        };

*// GET: api/Emp*

        [HttpGet]

*public* IActionResult GetEmployees()

        {

            return Ok(employees);

        }

*// GET: api/Emp/1*

        [HttpGet("{id}")]

*public* IActionResult GetEmployee(int id)

        {

            if (id < 0 || id >= employees.Count)

                return NotFound("Employee not found");

            return Ok(employees[id]);

        }

*// POST: api/Emp*

        [HttpPost]

*public* IActionResult AddEmployee([FromBody] string name)

        {

            employees.Add(name);

            return CreatedAtAction(nameof(GetEmployee), new { id = employees.Count - 1 }, name);

        }

*// DELETE: api/Emp/1*

        [HttpDelete("{id}")]

*public* IActionResult DeleteEmployee(int id)

        {

            if (id < 0 || id >= employees.Count)

                return NotFound("Employee not found");

            employees.RemoveAt(id);

            return NoContent();

        }

    }

}

Output :



