**2. WebApi\_Handson**

**Objectives:**

1. Demonstrate Swagger installation to WebAPI and WebAPI listing on browser

Swagger can be integrated into a WebAPI project using the Swashbuckle.AspNetCore NuGet package. This allows automatic generation of interactive API documentation. First, install the package using NuGet. Then, in Startup.cs, add `services.AddSwaggerGen()` inside `ConfigureServices()` and `app.UseSwagger()` and `app.UseSwaggerUI()` inside `Configure()`. The `ProducesResponseType` attribute can be used on API methods to document expected response types. Once set up, navigating to `/swagger` on the browser will display the API listing in an interactive interface.

2. Demonstrate the usage of Postman tool to hit WebAPI methods

Postman is a powerful tool used to test WebAPI endpoints. In Postman, you can create a collection to group related requests. Each request can specify HTTP method (GET, POST, etc.), endpoint URL, headers (including Authorization), and request body in JSON format. Postman allows you to add authentication headers such as Bearer tokens and visualize responses in various formats. Tabs in the central pane allow navigation between requests. You can add new requests to a collection and save them for reuse.

3. Demonstrate the usage of Route and explain Name attribute in Http requests

In Web API, the Route attribute can define custom URIs for controller actions, making them user-friendly and easier to consume. The Name attribute can be used to assign a readable name to the route, useful in link generation. Additionally, the ActionName attribute allows you to define multiple methods with the same HTTP verb in the same controller, but with different method names. This helps in logically organizing actions while keeping RESTful principles intact.

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

**Program.cs**  
using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();

// Add Swagger services

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo

{

Title = "Swagger Demo",

Version = "v1",

Description = "TBD",

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

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();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI(c =>

{

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

});

}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

**EmployeeController.cs**  
  
using Microsoft.AspNetCore.Mvc;

namespace SwaggerDemoAPI.Controllers

{

[ApiController]

[Route("Emp")] // Access this controller via: https://localhost:<port>/Emp

public class EmployeeController : ControllerBase

{

// GET: /Emp

[HttpGet]

public IActionResult GetEmployees()

{

var employees = new[]

{

new { Id = 1, Name = "Alice", Department = "HR" },

new { Id = 2, Name = "Bob", Department = "IT" },

new { Id = 3, Name = "Charlie", Department = "Finance" }

};

return Ok(employees);

}

}

}

**Output:**



