МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерної інженерії та системного програмування

**Лабораторна робота № 7**

з дисципліни

«Програмування веб-сервісів»

на тему:

«**ASP.NET Core** »

ЛРКІ.21048.21.02.18 ПЗ

Виконав:

студент 3 курсу, групи КІ-2-21-2 Сурма В.А.

(Підпис)

Перевірив: Лисенко С.М.

(Підпис)

**Хмельницький – 2023**

public class Task

{

public int Id { get; set; }

public string Title { get; set; }

public string Description { get; set; }

public bool IsCompleted { get; set; }

}

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

using System.Linq;

namespace YourAppName.Controllers

{

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

[ApiController]

public class TasksController : ControllerBase

{

private static List<Task> \_tasks = new List<Task>

{

new Task { Id = 1, Title = "Task 1", Description = "Description for Task 1", IsCompleted = false },

new Task { Id = 2, Title = "Task 2", Description = "Description for Task 2", IsCompleted = true }

};

[HttpGet]

public ActionResult<IEnumerable<Task>> GetTasks()

{

return Ok(\_tasks);

}

[HttpGet("{id}")]

public ActionResult<Task> GetTaskById(int id)

{

var task = \_tasks.FirstOrDefault(t => t.Id == id);

if (task == null)

{

return NotFound();

}

return Ok(task);

}

[HttpPost]

public IActionResult CreateTask(Task task)

{

task.Id = \_tasks.Count + 1;

\_tasks.Add(task);

return CreatedAtAction(nameof(GetTaskById), new { id = task.Id }, task);

}

[HttpPut("{id}")]

public IActionResult UpdateTask(int id, Task task)

{

var existingTask = \_tasks.FirstOrDefault(t => t.Id == id);

if (existingTask == null)

{

return NotFound();

}

existingTask.Title = task.Title;

existingTask.Description = task.Description;

existingTask.IsCompleted = task.IsCompleted;

return NoContent();

}

[HttpDelete("{id}")]

public IActionResult DeleteTask(int id)

{

var task = \_tasks.FirstOrDefault(t => t.Id == id);

if (task == null)

{

return NotFound();

}

\_tasks.Remove(task);

return NoContent();

}

}

}

using Microsoft.OpenApi.Models;

public class Startup

{

public void ConfigureServices(IServiceCollection services)

{

services.AddControllers();

services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo { Title = "YourAppName API", Version = "v1" });

});

}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

app.UseRouting();

app.UseSwagger();

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1/swagger.json", "YourAppName API v1");

c.RoutePrefix = string.Empty;

});

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers();

});

}

}