Звіт

З виконання

Практичної роботи №2

Студента ІТШІ 21-5

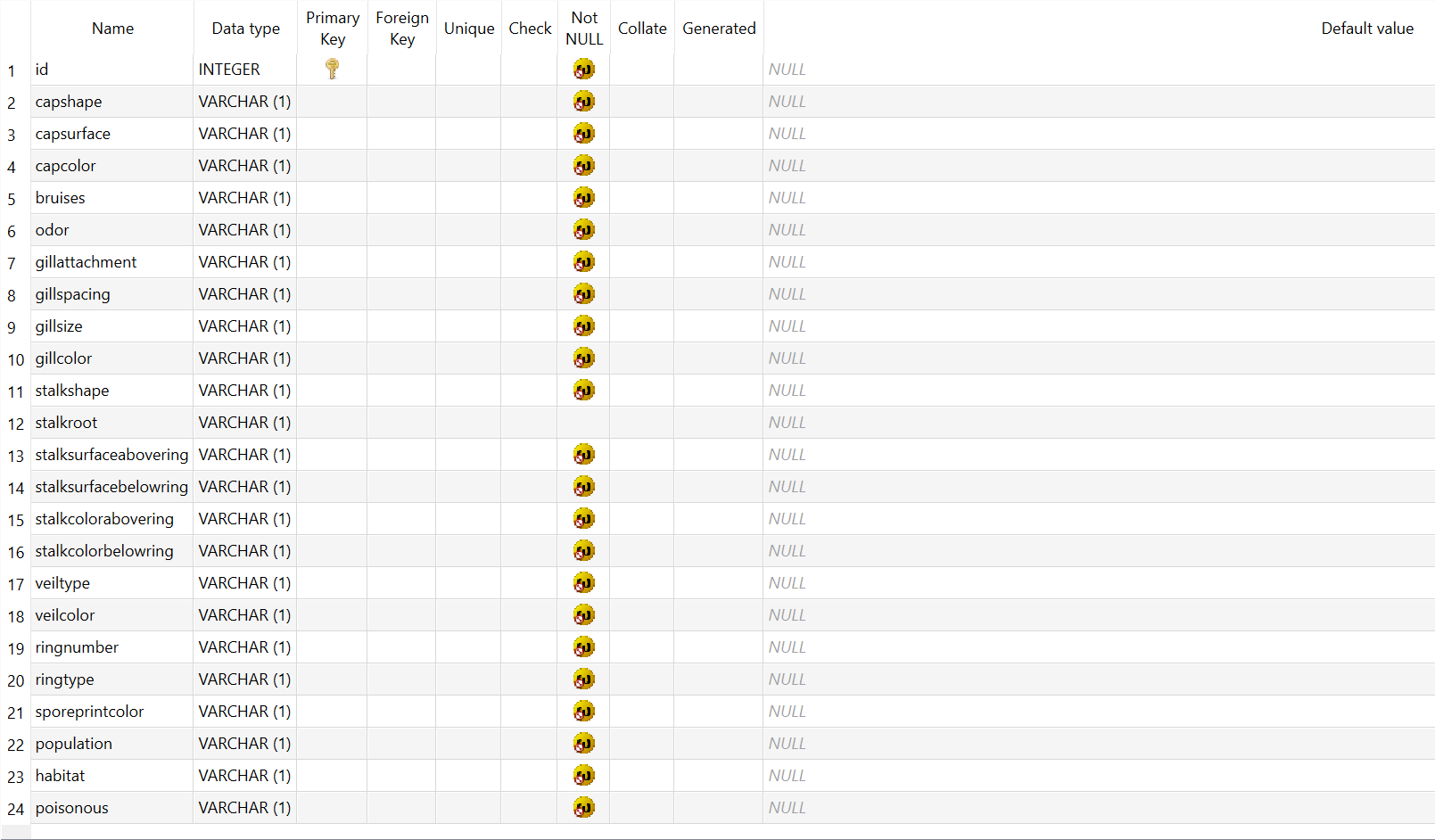
Сарміні Абделя

Мета: Вивчення особливостей використання технології Entity Framework Core для організації доступу до баз даних з .NET-застосунків.

Використання технології Entity Framework Core для організації доступу до баз даних з .NET-застосунків.

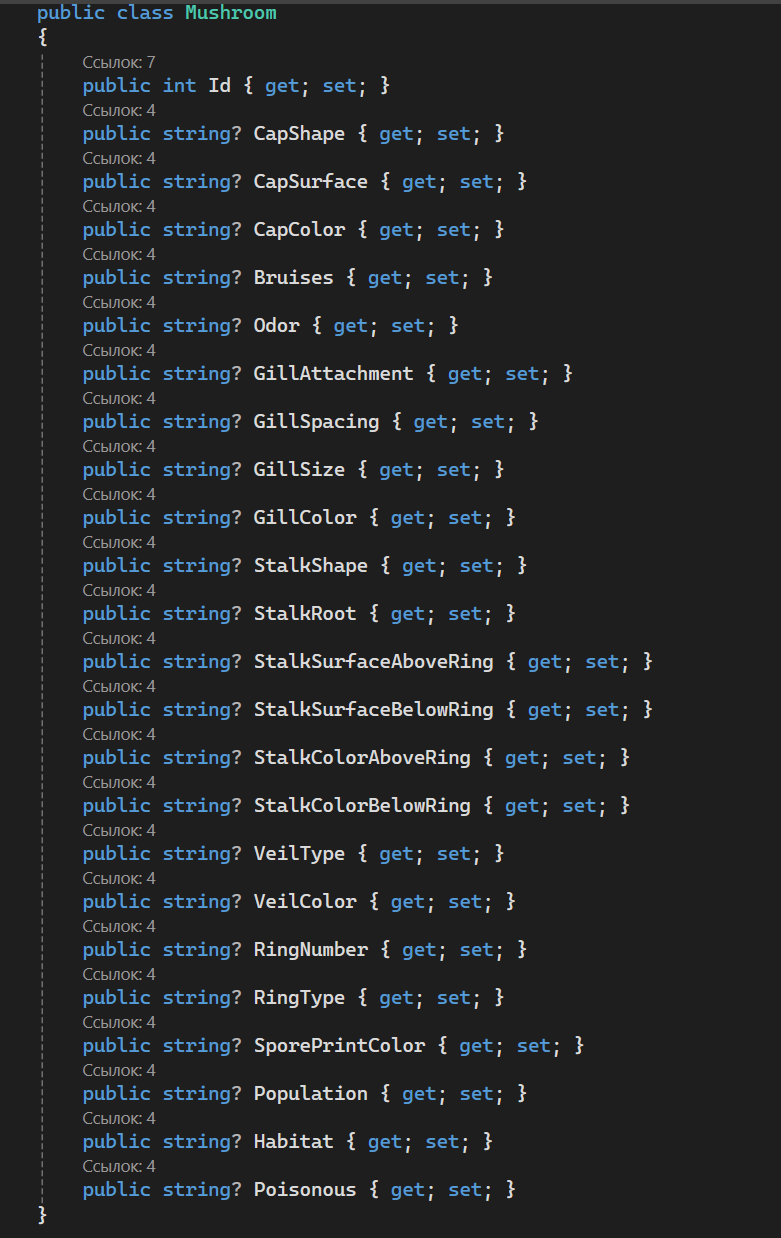
Хід виконання:

Схема бази даних:

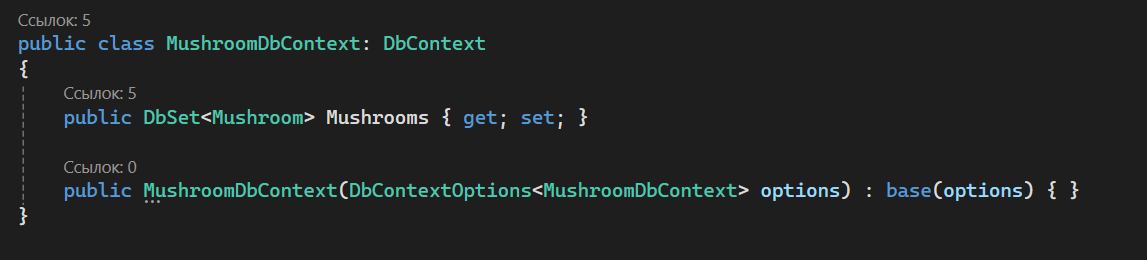


Інтерфейси класів, зокрема. класів сутностей та контексту даних:

Клас гриба:



Клас контексту даних:



Вихідні тексти програмної реалізації:

Застосунок виконан за допомогою ASP.Net Core, тому майже вся взаємодія із даними проходить через MushroomController:

CRUD операції:

public MushroomController(MushroomDbContext \_context)

{

dbContext = \_context;

}

public async Task<IActionResult> Index()

{

var mushroom = await dbContext.Mushrooms.ToListAsync();

return View(mushroom);

}

public IActionResult Create()

{

return View();

}

[HttpPost]

public async Task<IActionResult> Create(Mushroom mushroom)

{

if (ModelState.IsValid)

{

dbContext.Add(mushroom);

await dbContext.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

return View(mushroom);

}

public async Task<IActionResult> Edit(int id)

{

var mushroom = await dbContext.Mushrooms.FindAsync(id);

if (mushroom == null)

{

return NotFound();

}

return View(mushroom);

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Edit(int id, Mushroom mushroom)

{

bool MushroomExists(int id)

{

return dbContext.Mushrooms.Any(e => e.Id == id);

}

if (id != mushroom.Id)

{

return BadRequest();

}

if (ModelState.IsValid)

{

try

{

dbContext.Entry(mushroom).State = EntityState.Modified;

await dbContext.SaveChangesAsync();

return RedirectToAction("Index");

}

catch (DbUpdateConcurrencyException)

{

if (!MushroomExists(mushroom.Id))

{

return NotFound();

}

throw;

}

}

return View(mushroom);

}

[HttpPost]

public async Task<IActionResult> Delete(int id)

{

var mushroom = await dbContext.Mushrooms.FindAsync(id);

if (mushroom != null)

{

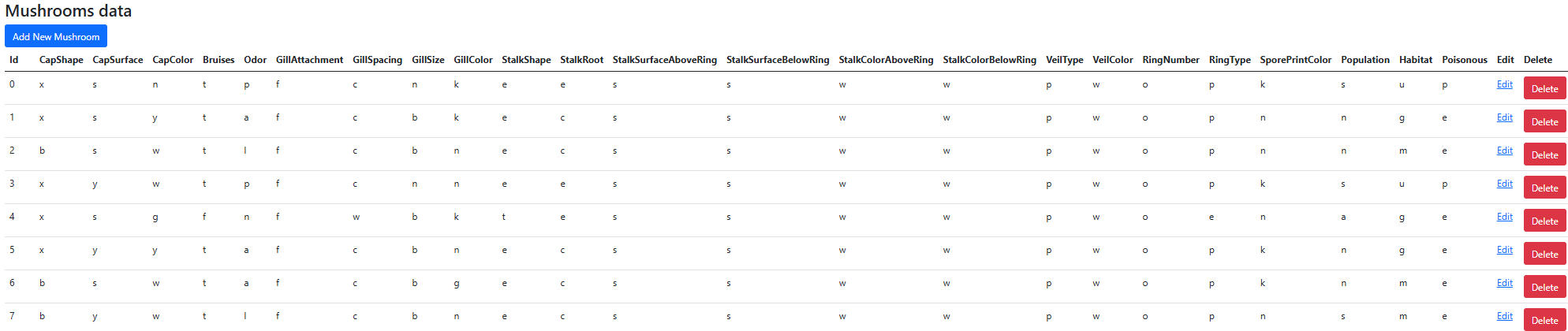
dbContext.Mushrooms.Remove(mushroom);

await dbContext.SaveChangesAsync();

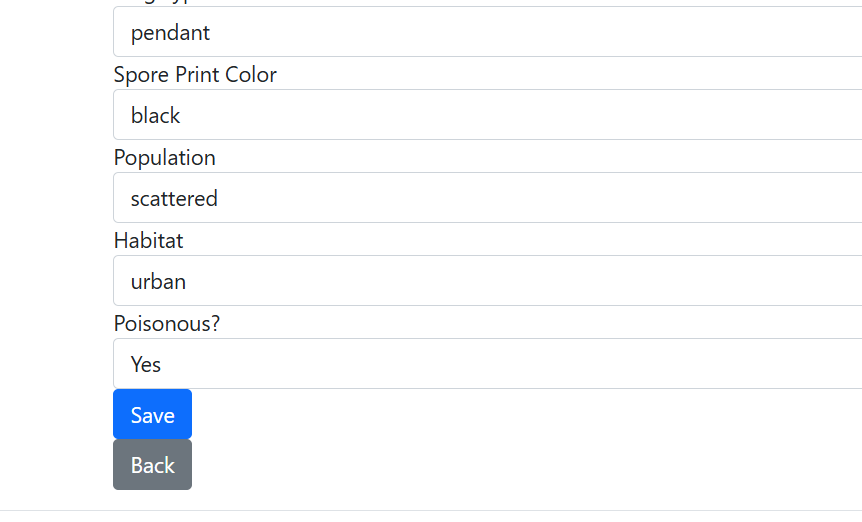
}

return RedirectToAction("Index");

}

Скріни візуального інтерфейсу:

Сторінка Edit:



Висновки: Ми створили сторінку ASP.Net Core для виконання CRUD операцій на датасету грибів, та вивчили особливості виокристання Entity Framework Core.