


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:

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	id	INTEGER								NULL
2	capshape	VARCHAR (1)								NULL
3	capsurface	VARCHAR (1)								NULL
4	capcolor	VARCHAR (1)								NULL
5	bruises	VARCHAR (1)								NULL
6	odor	VARCHAR (1)								NULL
7	gillattachment	VARCHAR (1)								NULL
8	gillspacing	VARCHAR (1)								NULL
9	gillsize	VARCHAR (1)								NULL
10	gillcolor	VARCHAR (1)								NULL
11	stalkshape	VARCHAR (1)								NULL
12	stalkroot	VARCHAR (1)								NULL
13	stalksurfaceabovering	VARCHAR (1)								NULL
14	stalksurfacebelowring	VARCHAR (1)								NULL
15	stalkcolorabovering	VARCHAR (1)								NULL
16	stalkcolorbelowring	VARCHAR (1)								NULL
17	veiltype	VARCHAR (1)								NULL
18	veilcolor	VARCHAR (1)								NULL
19	ringnumber	VARCHAR (1)								NULL
20	ringtype	VARCHAR (1)								NULL
21	sporeprintcolor	VARCHAR (1)								NULL
22	population	VARCHAR (1)								NULL
23	habitat	VARCHAR (1)								NULL
24	poisonous	VARCHAR (1)								NULL

Інтерфейси класів, зокрема. класів сутностей та контексту даних:

Клас гриба:

```
public class Mushroom
{
    Ссылка: 7
    public int Id { get; set; }
    Ссылка: 4
    public string? CapShape { get; set; }
    Ссылка: 4
    public string? CapSurface { get; set; }
    Ссылка: 4
    public string? CapColor { get; set; }
    Ссылка: 4
    public string? Bruises { get; set; }
    Ссылка: 4
    public string? Odor { get; set; }
    Ссылка: 4
    public string? GillAttachment { get; set; }
    Ссылка: 4
    public string? GillSpacing { get; set; }
    Ссылка: 4
    public string? GillSize { get; set; }
    Ссылка: 4
    public string? GillColor { get; set; }
    Ссылка: 4
    public string? StalkShape { get; set; }
    Ссылка: 4
    public string? StalkRoot { get; set; }
    Ссылка: 4
    public string? StalkSurfaceAboveRing { get; set; }
    Ссылка: 4
    public string? StalkSurfaceBelowRing { get; set; }
    Ссылка: 4
    public string? StalkColorAboveRing { get; set; }
    Ссылка: 4
    public string? StalkColorBelowRing { get; set; }
    Ссылка: 4
    public string? VeilType { get; set; }
    Ссылка: 4
    public string? VeilColor { get; set; }
    Ссылка: 4
    public string? RingNumber { get; set; }
    Ссылка: 4
    public string? RingType { get; set; }
    Ссылка: 4
    public string? SporePrintColor { get; set; }
    Ссылка: 4
    public string? Population { get; set; }
    Ссылка: 4
    public string? Habitat { get; set; }
    Ссылка: 4
    public string? Poisonous { get; set; }
}
```

Клас контексту даних:

```
Ссылка: 5
public class MushroomDbContext: DbContext
{
    Ссылка: 5
    public DbSet<Mushroom> Mushrooms { get; set; }

    Ссылка: 0
    public MushroomDbContext(DbContextOptions<MushroomDbContext> options) : base(options) { }
}
```

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)
{
}
```


Edit Mushroom

Cap Shape

convex

Cap Surface

smooth

Cap Color

brown

Bruises?

Yes

Odor

pungent

Gill Attachment

free

Gill Spacing

close

Gill Size

narrow

Gill Color

black

Stalk Shape

enlarging

Stalk Root

equal

Stalk Surface Above Ring

smooth

Stalk Surface Below Ring

smooth

Stalk Color Above Ring

white

Spore Shape	pendant
Spore Print Color	black
Population	scattered
Habitat	urban
Poisonous?	Yes
Save	
Back	

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