



*Міський  
сайт Одеси*



*Дорогі друзі!*

*Щиро радий вітати вас на цьому особливому сайті — просторі знань, відкритів та творчості.*

*Тут ви знайдете цікаві матеріали, корисні поради, розваги та підказки, які зроблять навчання легшим і веселішим.*

*Ви — майбутнє нашого міста, розумні, допитливі та добрі серцем. У вас все попереду: перші великі успіхи, нові відкриття і справжні пригоди!*

*А ми, дорослі, зробимо все, щоб вам було комфортно, безпечно та радісно зростати в рідній Одесі.*

*Навчайтесь із задоволенням, мрійте сміливо й пам'ятайте: Одеса пишається вами!*

*З найкращими побажаннями  
Департамент освіти та науки*



# Команда проекту - Одеса для дітей



**Кристіна Черкезян**

Scrum-master  
керівник проекту



**Інна Ночовкіна**

UI/UX  
дизайнер



**Євген Нечипуренко**

Front-End  
developer



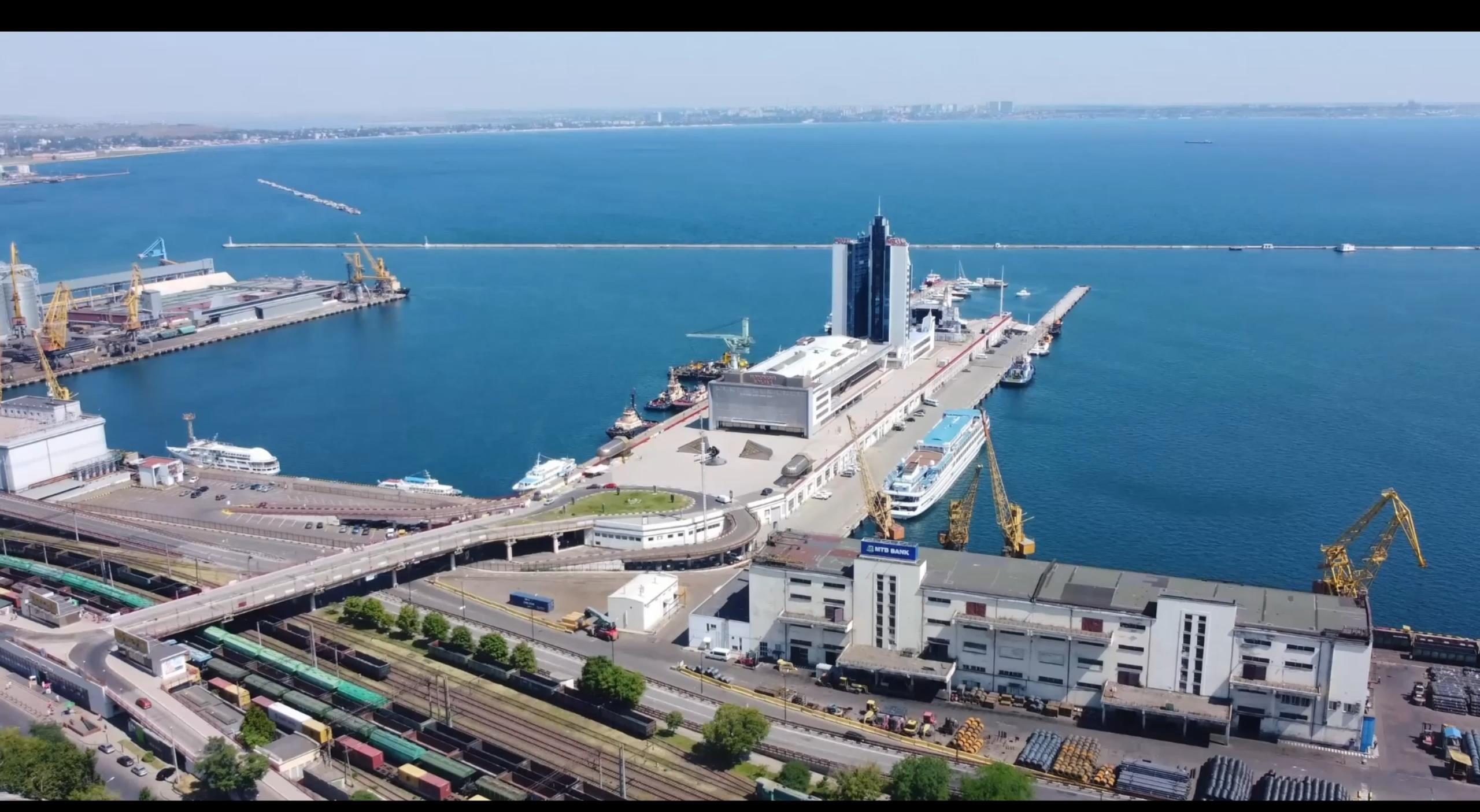
**Олександр Єленок**

Back-End  
developer



**Іван Сливняк**

DevOps  
engineer





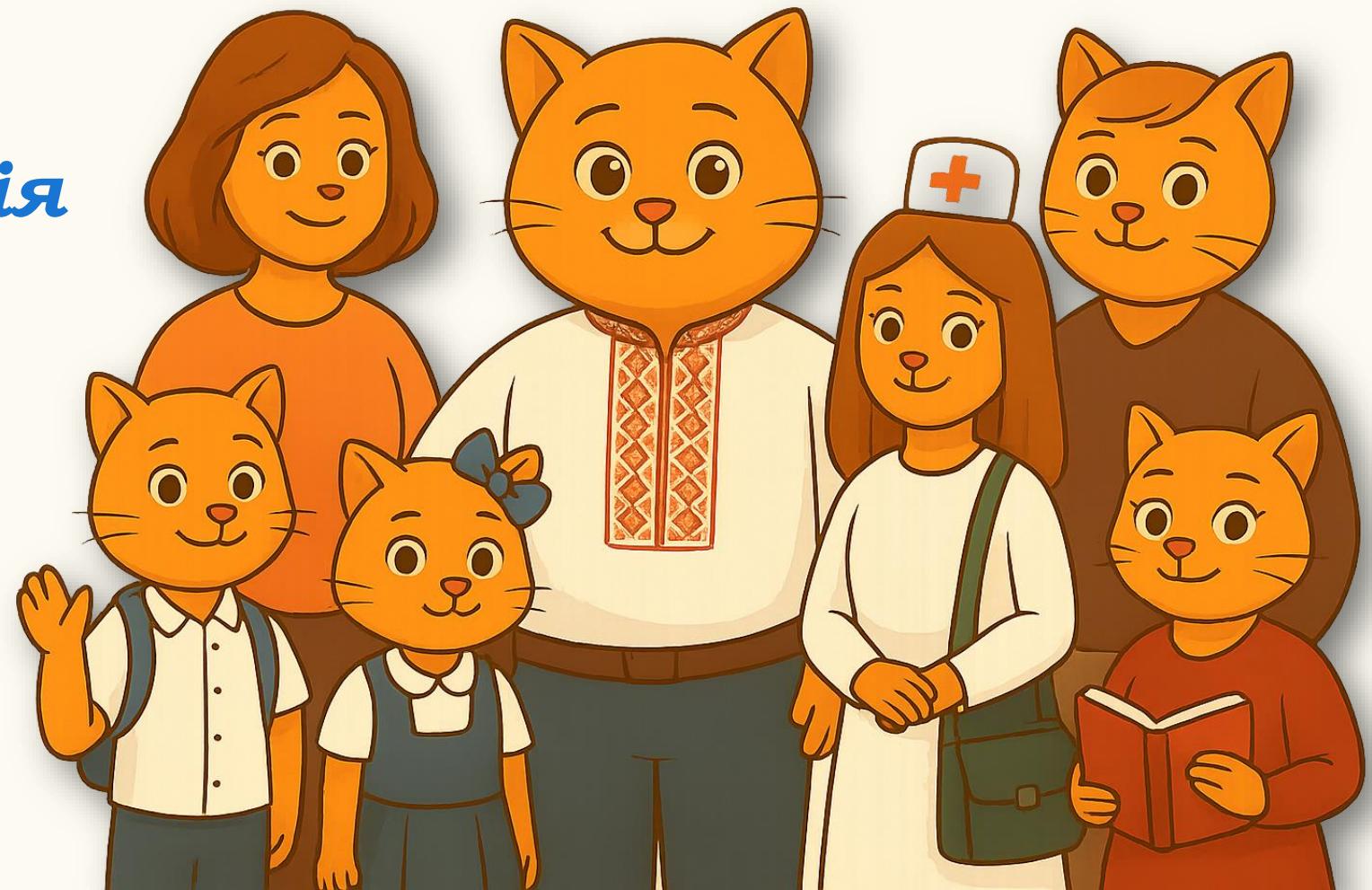
## *Мета нашого проекту*

- ✓ Створити єдиний онлайн-ресурс
- ✓ Об'єднати інформацію про навчання, медицину, відпочинок, розвиток та актуальні події
- ✓ Надати зручний інструмент для швидкого пошуку необхідної інформації
- ✓ Створення зручного API та бази даних



## Цільова аудиторія

- ✓ Батьки школярів та дошкільнят
- ✓ Учні 1 – 11 класів
- ✓ Освітні заклади та вчителі
- ✓ Медичні та культурні установи





## Ключові переваги

- ✓ Єдина точка доступу до всієї важливої інформації
- ✓ Оновлюваний календар подій
- ✓ Доступні онлайн-уроки (зручність для дистанційного навчання)
- ✓ Орієнтація саме на Одесу (локальна цінність)
- ✓ Інтерактивна карта установ
- ✓ Інтегрований YouTube-плеєр

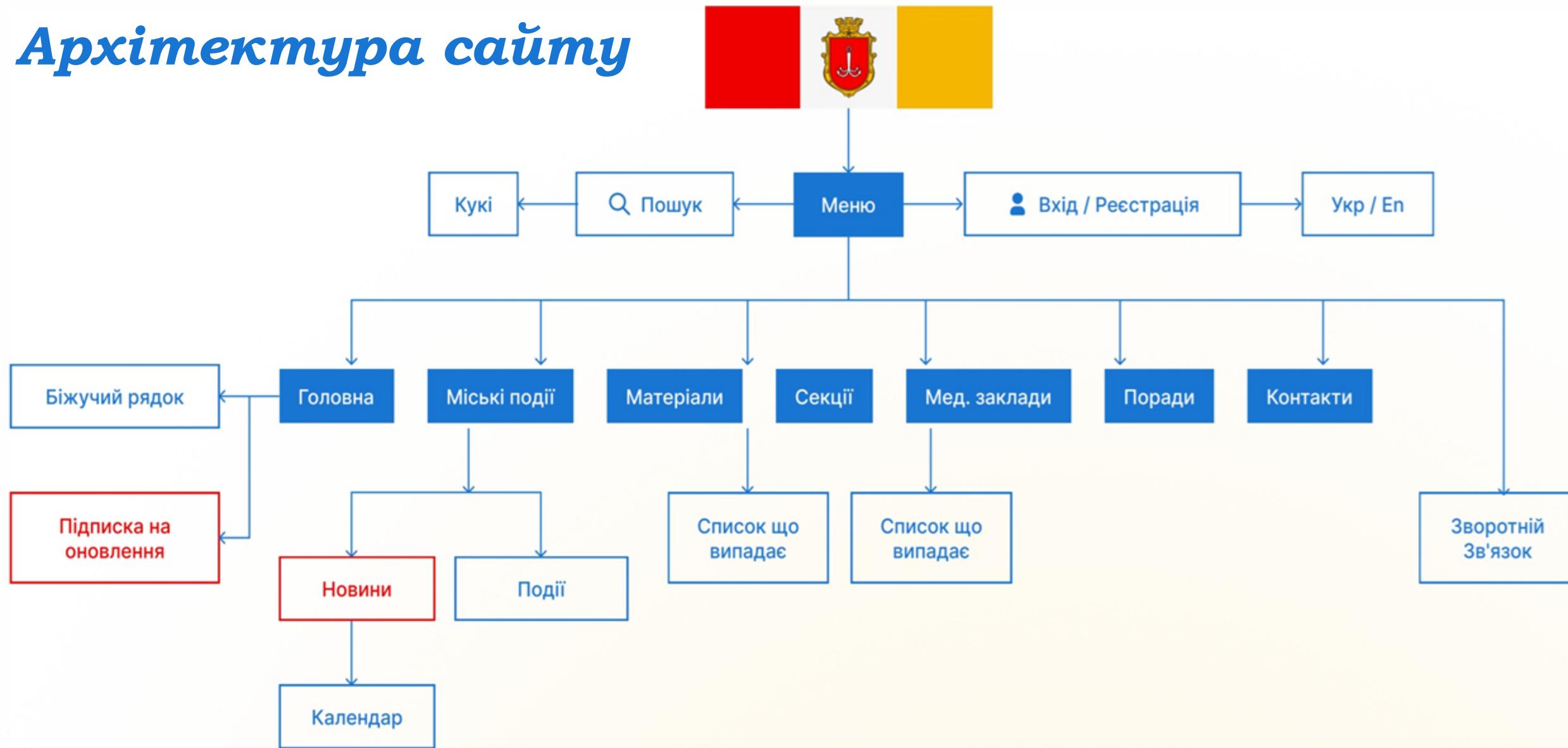




# *Структура проекту*



# Архітектура сайту





## Засоби розробки



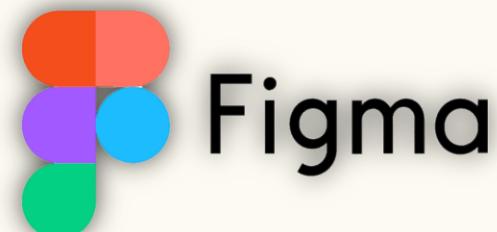
Visual  
Studio



Azure



VS Code



Figma



GitHub





{ REST-API }

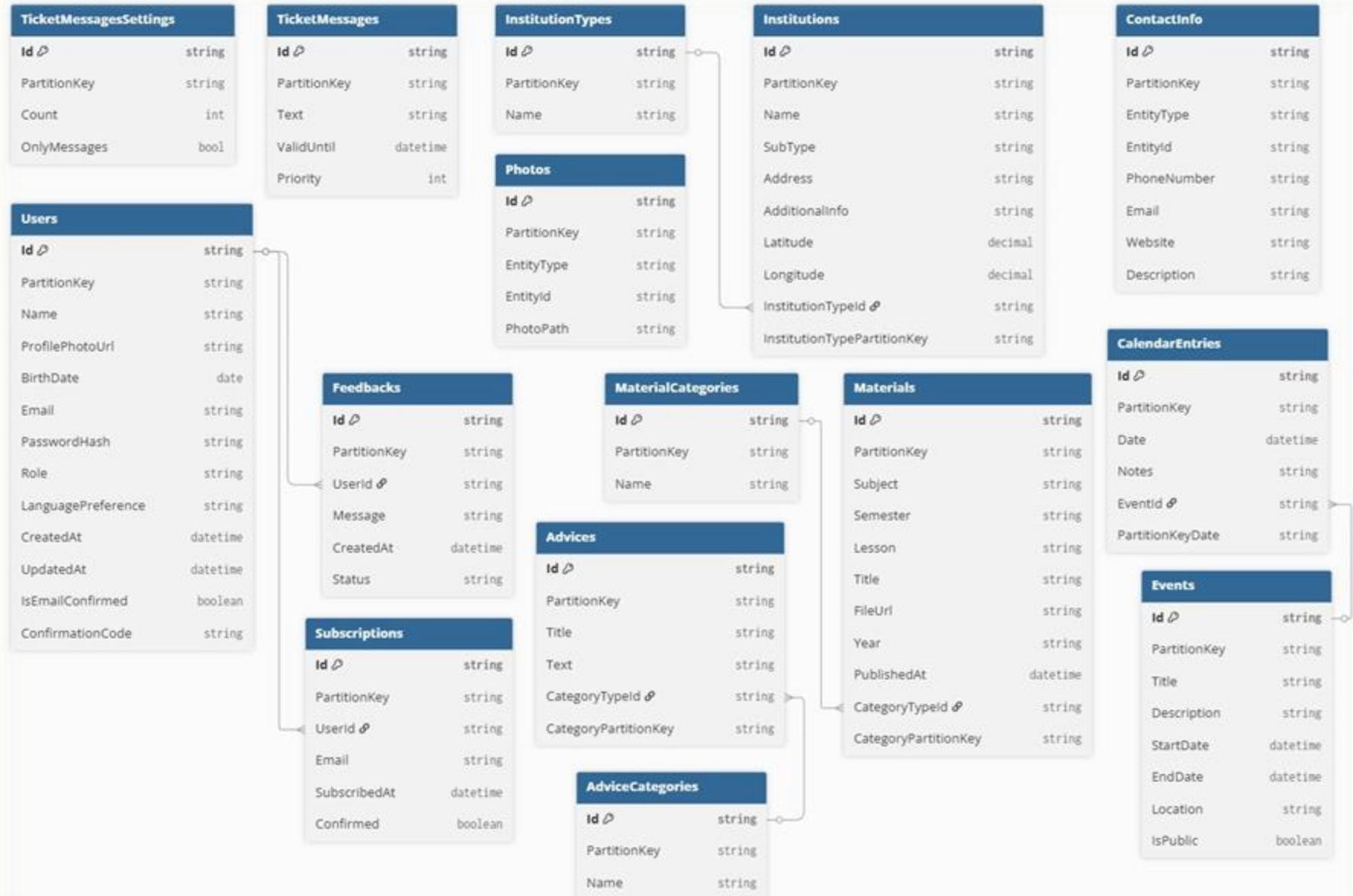


Azure Cosmos DB





# Структура базы данных





# Кешування даних

```
string cacheKey = "AllInstitutionSubTypes";

var cached = await _cacheService.Get<List<InstitutionSubTypesGetDTO>>(cacheKey);
if (cached != null)
    return cached;

_cacheService.Set(cacheKey, result, TimeSpan.FromDays(365));
return result;

private async Task RefreshInstitutionCachesAsync()
{
    _cacheService.Remove("AllInstitutions");
    _cacheService.Remove("AllInstitutionSubTypes");

    var allInstitutions = await _unitOfWork.Institutions.GetAllAsync();
    _cacheService.Set("AllInstitutions", allInstitutions, TimeSpan.FromDays(365));

    var institutionTypes = await _context.InstitutionTypes
        .Where(it => it.PartitionKey == "InstitutionTypePartition")
        .ToListAsync();

    var typeDict = institutionTypes.ToDictionary(t => t.Id, t => t.Name);

    var subTypes = allInstitutions
        .Where(i => !string.IsNullOrEmpty(i.SubType))
        .GroupBy(i => new { i.SubType, i.InstitutionTypeId })
        .Select(g => new InstitutionSubTypesGetDTO
    {
        SubType = g.Key.SubType,
        InstitutionTypeId = g.Key.InstitutionTypeId,
        InstitutionTypeName = typeDict.TryGetValue(g.Key.InstitutionTypeId, out var name) ? name : null
    })
    .ToList();

    _cacheService.Set("AllInstitutionSubTypes", subTypes, TimeSpan.FromDays(365));
}
```

```
namespace ODA_Social_Projeckt.BLL.Services
{
    2 references | AlexLiberty, 60 days ago | 1 author, 2 changes
    public class MemoryCacheService : ICacheService
    {
        private readonly IMemoryCache _cache;

        0 references | AlexLiberty, 103 days ago | 1 author, 1 change
        public MemoryCacheService(IMemoryCache cache)
        {
            _cache = cache;
        }

        1 reference | AlexLiberty, 103 days ago | 1 author, 1 change
        public T GetOrCreate<T>(string key, Func<T> createItem, TimeSpan cacheDuration)
        {
            if (_cache.TryGetValue(key, out T cacheEntry))
            {
                return cacheEntry;
            }

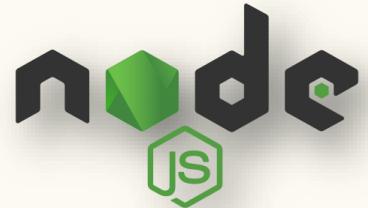
            cacheEntry = createItem();
            _cache.Set(key, cacheEntry, cacheDuration);

            return cacheEntry;
        }

        5 references | AlexLiberty, 103 days ago | 1 author, 1 change
        public async Task<T> Get<T>(string cacheKey)
        {
            _cache.TryGetValue(cacheKey, out T value);
            return value;
        }

        8 references | AlexLiberty, 103 days ago | 1 author, 1 change
        public void Remove(string key)
        {
            _cache.Remove(key);
        }

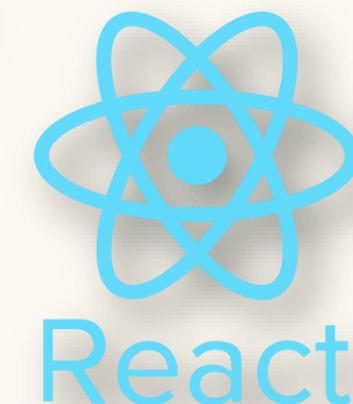
        10 references | AlexLiberty, 103 days ago | 1 author, 1 change
        public void Set<T>(string key, T item, TimeSpan cacheDuration)
        {
            _cache.Set(key, item, cacheDuration);
        }
    }
}
```



Vite



Google Maps



React



A X I O S

## Інструменти бібліотеки Front-End



TypeScript



Tailwind CSS



React Hook Form



*Демонстрація*



## Подальший розвиток

- ❑ Мобільний застосунок
- ❑ Онлайн щоденник школяра
- ❑ Відгуки батьків / рейтинги навчальних закладів
- ❑ Інтеграція у проект "Місто"





*Дякуємо за увагу  
та готові відповісти на питання!*

