

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

**Факультет прикладної математики та інформатики**

(повне найменування назва факультету)

**Кафедра інформаційних систем**

(повна назва кафедри)

## **Дипломна робота**

**Розробка веб-аплікації по моніторингу об'єктів міста**

Виконав: студент групи ПМі-42с

спеціальності

122 «Комп'ютерні науки»

(шифр і назва спеціальності)

**Кондера С.Ю**

(підпис)

(прізвище та ініціали)

Керівник **Бернакевич І. Є**

(підпис)

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(підпис)

(прізвище та ініціали)

2023

## ЗМІСТ

ВСТУП.....	4
РОЗДІЛ1.ВИКОРИСТАННЯ ВЕБ АПЛІКАЦІЙ ДЛЯ МОНИТОРНГУ ОБ’ЄКТІВ МІСТА .....	5
1.1 Опис предметної області .....	5
1.2 Огляд існуючих програмних рішень.....	5
1.3 Підсумок розділу.....	7
РОЗДІЛ2.ВИМОГИ ТА ЗАВДАННЯ ВЕБ АПЛІКАЦІЇ ДЛЯ МОНИТОРНГУ ОБ’ЄКТІВ МІСТА.....	8
2.1 Постановка завдання.....	8
2.2 Вибір інструментів та підходів для реалізації програмного забезпечення.....	9
2.3 Специфікація вимог .....	10
2.3.1 Середовище функціонування .....	10
2.3.2 Вимоги зовнішніх інтерфейсів .....	10
2.4 Підсумок розділу.....	12
РОЗДІЛ3.ПРОЕКТУВАННЯ ВЕБ АПЛІКАЦІЇ ДЛЯ МОНИТОРНГУ ОБ’ЄКТІВ МІСТА .....	13
3.1 Проектування архітектури .....	13
3.2 Проектування бази даних .....	14
3.2.1 Логічна модель бази даних .....	14
3.2.2 Опис сутностей.....	15
3.2.3 Опис зв’язків між таблицями.....	17
3.3 Підсумок розділу.....	17

РОЗДІЛ 4. РОЗРОБКА ТА ІМПЛЕМЕНТАЦІЯ ВЕБ АПЛІКАЦІЇ ДЛЯ МОНІТОРИНГУ ОБ'ЄКТІВ МІСТА .....	18
4.1 Імплементация веб аплікації для моніторингу об'єктів міста.....	18
4.2 Підсумок розділу.....	18
ВИСНОВКИ.....	30
СПИСОК ЛІТЕРАТУРИ.....	31

## ВСТУП

На сьогоднішній день в Україні використовують застарілі системи для моніторингу об'єктів у містах. Це часто веб-додатки, в яких об'єкти розташовані у хаотичному порядку, не можна фільтрувати за певною категорією, а іноді навіть є просто документ-таблиця з даними. Все це дуже незручно. Крім того, такі веб-додатки не містять інформації про бомбосховища як окрему категорію. Щоб знайти бомбосховища у своєму місті, потрібно використовувати інший сервіс. А в моєму додатку є можливість додати категорію бомбосховища та слідкувати за їхнім станом і дізнатись, чи працює зараз, чи, можливо на стадії ремонту, чи зовсім не працює бомбосховище.

З метою забезпечення системного й оперативного доступу до моніторингової інформації про об'єкти міста, я вирішив розробити власний веб-додаток, який відповідатиме усім вимогам користувача. Моя система дозволяє створювати різні категорії об'єктів міста з різними типами полів, такими як текст, великий текст, посилання, електронна пошта, номер телефону тощо. Після переходу на сторінку моніторингу, користувач може використовувати фільтри, такі як пошук за частковим збігом назви, вибір категорій, які не показувати, а також встановлення радіусу для відображення об'єктів у певній відстані від користувача, якщо він дозволив програмі доступ до своєї позиції. Біля карти Google відображаються маркери фільтрованих об'єктів. Основний функціонал полягає у можливості створювати об'єкти міста, вводити значення полів, додавати локацію та фотографії. Для кожного об'єкту доступні такі функції, як детальний перегляд, редагування та видалення. Також не менш важливою фішкою програми є головна інформаційна панель, на якій користувач може бачити різні графіки та зміну статусів за останні 24 години.

Наразі цей тип додатків не є дуже популярним в Україні, але це, швидше за все, через відсутність якісних продуктів. Я вважаю, що з відповідальним підходом до розробки ми могли б привернути більше користувачів. Тому темою моєї роботи стала розробка веб-додатків для моніторингу об'єктів у містах.

## РОЗДІЛ 1. ВИКОРИСТАННЯ ВЕБ АПЛІКАЦІЙ ДЛЯ МОНІТОРИНГУ ОБ'ЄКТІВ МІСТА

### 1.1 Опис предметної області

Раніше ,для моніторингу об'єктів міста використовували журнали з таблицями , згодом таблиці появились в електронному форматі. Зараз почали вже використовувати моніторингові веб аплікації. Це суттєво спростило процедуру додавання нових об'єктів, редагування та їх моніторингу.

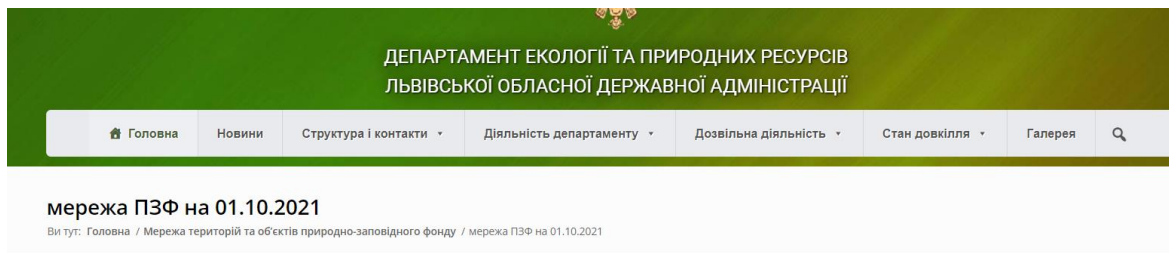
Для того, щоб веб аплікація для моніторингу об'єктів міста користувалась популярністю, вона має відповідати вимогам користувача. На даний момент, на українському ринку взагалі не має аплікацій для моніторингу об'єктів міста, які б відповідали вимогам.

### 1.2 Огляд існуючих програмних рішень

Програмні рішення для моніторинг саме об'єктів міста знайти майже не можливо, тому будемо проводити наш аналіз на прикладі схожих систем. А саме : моніторинг мережі територій та об'єктів природньо-заповідного фонду, що розташований на сайті департаменту екології та природних ресурсів львівської обласної державної адміністрації [12] та систему City Monitor [9].

#### **Департамент екології та природних ресурсів**

На цьому сайті можна провести моніторинг стану довкілля, якості води, мережі територій та об'єктів природньо-заповідного фонду, водні ресурси, біорізноманіття та ще багато іншого. Із плюсів, це те, що тут можна провести моніторинг різних сфер діяльності департаменту , але дуже великий мінус - це якість викладу моніторингового матеріалу. Весь моніторинговий матеріал подано або у таблицях, або у величезних файлах, які читати не зручно. Графічний інтерфейс системи зображено на *рис 1.1* , а сам моніторинговий матеріал на *рис 1.2*.



[мережа ПЗФ на 01.10.2021](#)

МИ В FACEBOOK



Рис 1.1

№ п/п	Назва об'єкта ПЗФ	Тип	Площа, га	Адміністративне розташування та місцезнаходження об'єкта ПЗФ	Назва підприємства, установи - землекористувача (землевласника), у відані якого знаходиться об'єкт ПЗФ	Рішення, згідно з яким створено (оголошено) даний об'єкт ПЗФ, змінено його площу тощо
11	3 Північне Поділля		15587,9200	Бродівський, Буський, Золочівський райони	Мінприроди України	Указ президента України від 10.02.2010 №156/2010
12	4 Бойківщина		12240,0000	Турківський р-н, ДП "Боринське ЛП", ДП "Турківське ЛП" Турківське ДЛПП ЛПП "Галсільніс"	Державне лісове агентство України	Указ президента України від 11.04.2019 №130/2019
13	5 Королівські Бескиди		8997,0000	Старосамбірський р-н, Старосамбірське ЛДП "Галсільніс", Старосамбірське л-во (кв. 1-38, 4207 га) Дністриське л-во (кв.1-37, 4700 га)	Міндовкілля України	Указ Президента України від 30.11.2020 № 526/2020
14	Разом НПП - 5		79587,5200			
15	Заказники					
16	ландшафтні					
				Скопівський р-н, Верхняцьке л-во, кв.1 (п.23,26,32,33); кв.2 (п.6,14,16); кв.3 (п.36-39); кв.4 (п.21,24); Климцьке л-во кв.9 (13-17); кв.11 (п.3-		

Рис 1.2

## City Monitor

City Monitor [9] – це доволі зручна веб-аплікація для моніторингу широкого спектру категорії. А саме : моніторинг населення, динаміки бізнесу, працевлаштування, житло та інші. Веб аплікація має дуже приємний користувацький інтерфейс та широкий функціонал. Користувач має можливість фільтрувати міста та фільтрувати категорії . Також є можливість посортувати за зростанням або за спаданням якогось з критеріїв. Із мінусів можу відзначити те, що у системі не має всіх міст. Користувацький інтерфейс зображений на *рис 1.3* та *рис 1.4*

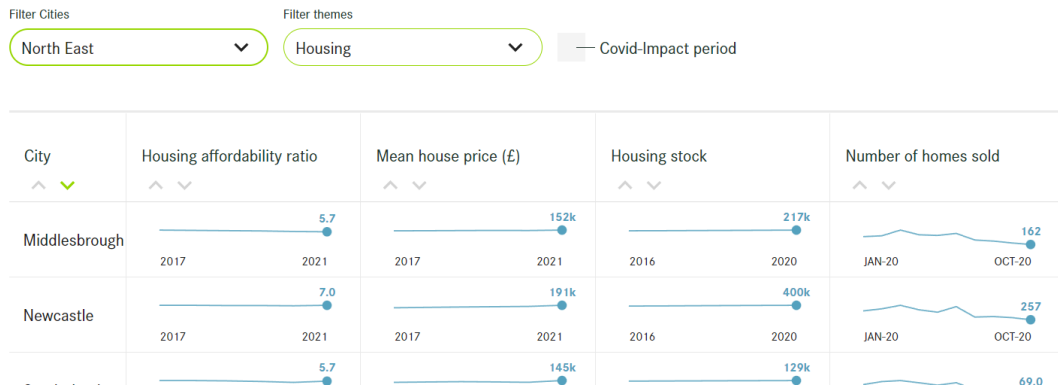


рис 1.3

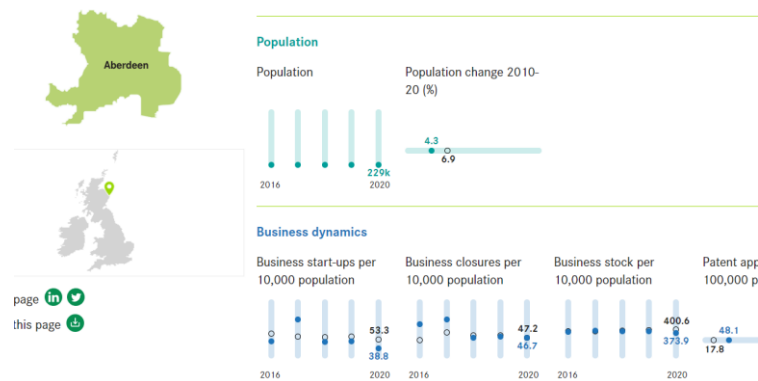


рис 1.4

### 1.3 Підсумок розділу

Проаналізувавши вищезгадані речі, можна зробити підсумок, що наразі не має зручних веб-аплікацій для моніторингу об'єктів міста саме для українського користувача, а якщо користувач використає не зручні існуючі, то він зіштовхнеться з труднощами з оглядом моніторингової інформації.

## РОЗДІЛ 2. ВИМОГИ ТА ЗАВДАННЯ ВЕБ АПЛІКАЦІЇ ДЛЯ МОНІТОРНГУ ОБ'ЄКТІВ МІСТА

### 2.1 Постановка завдання

Метою даної веб аплікації є можливість проведення моніторингу об'єктів міста користувачем. Всі користувачі мають мати змогу оглядати всі об'єкти та дивитись детальніше по кожному з них. Також має бути реалізовано зручна пошукова система для пошуку об'єкта міста за назвою, категорією та за відстанню до них. Окрім цього було б добре, щоб була можливість бачити реальну динаміку статусів об'єктів міста та кількість об'єктів за по категоріях та статусах. Дуже зручно для користувачів було б відображення інформаційної панелі, яка включатиме вище сказане та відобразатиме у вигляді графіків.

Система матиме наступні типи користувачів: гість та адміністратор. Гість матиме такі можливості: огляд інформаційної панелі, огляд сторінки для моніторингу, користування фільтрами пошуку, огляд об'єктів міста та створення звернення. Адміністратор матиме всі можливі доступи, а саме: оглядати інформаційну панель, створювати, редагувати, видаляти категорії, створювати, редагувати та видаляти властивості до категорії, створювати, редагувати, видаляти об'єкти міста, огляд сторінки для моніторингу, користування фільтрами пошуку, огляд об'єктів міста. Діаграми прецедентів для кожної з ролей подано нижче.



Рис 2.1 Діаграма прецедентів для ролі Гість



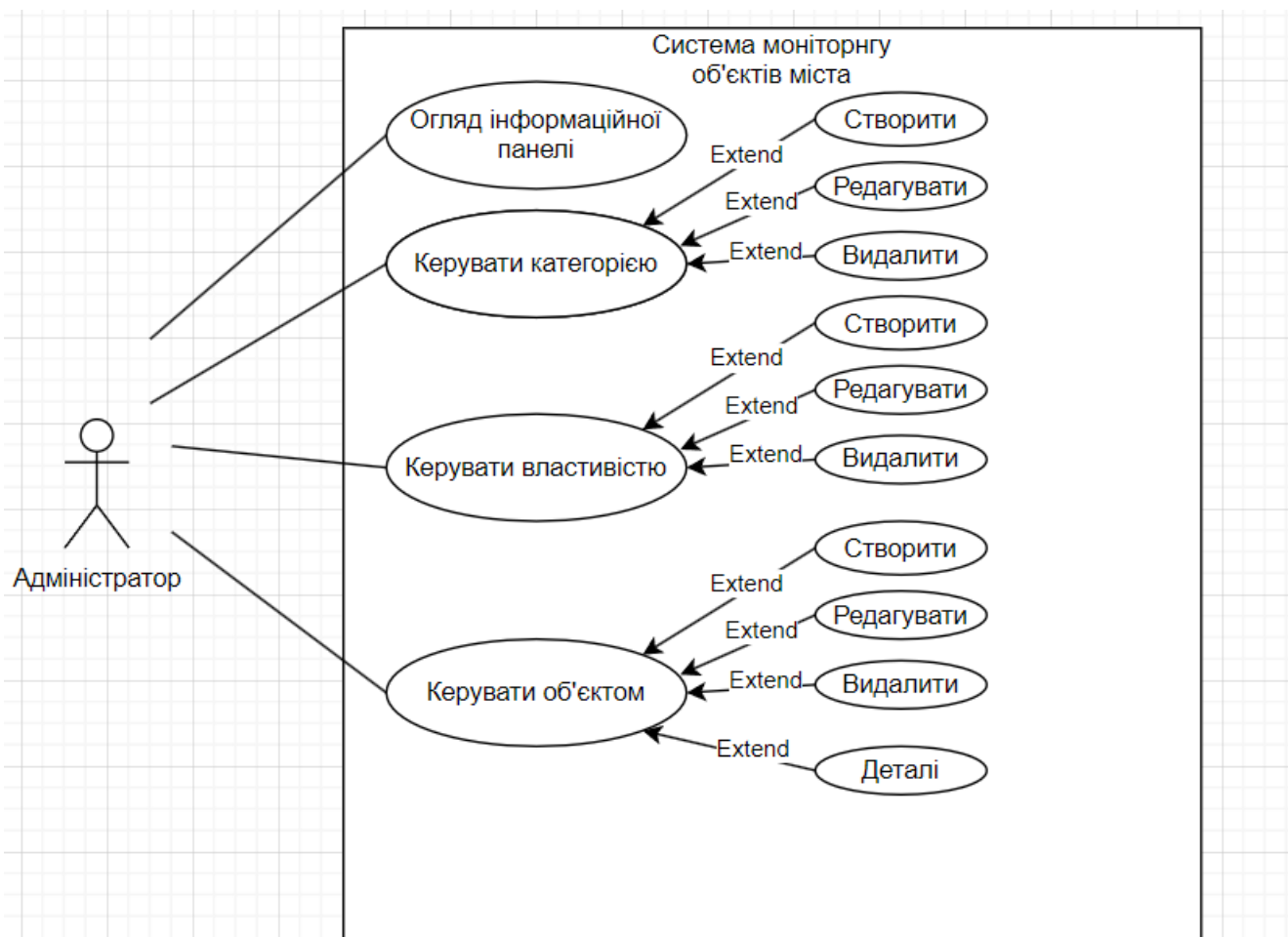


Рис 2.2 Діаграма прецедентів для ролі Адміністратор

## 2.2 Вибір інструментів та підходів для реалізації програмного забезпечення

Для розробки даної веб-аплікації буде використано наступні компоненти та технології:

- Bitbucket [10] – система контролю версій, яка дозволяє зберігати та керувати версіями програмного коду
- Salesforce CRM [13] – платформа управління відносинами з клієнтами, яка використовується для зберігання бази даних
- Salesforce Metadata Api [11] – інтерфейс програмування застосунків (API), який надає можливості для керування метаданими в Salesforce.
- Salesforce Digital Experience [14] [5] – платформа, що дозволяє розгорнути веб-додатки та створювати користувацький інтерфейс на основі Salesforce

- LWC [4] – фреймворком розробки, який дозволяє писати веб-компоненти для платформи Salesforce.
- Apex [1] – мова програмування, яка використовується для розробки серверної частини на платформі Salesforce

База даних буде використовуватись з Salesforce [13]. Також буде використано Salesforce Metadata Api [11] для створення таблиць та додавання властивостей, на додачу планується реалізація Metadata сервісу з методами, необхідними для виконання цих дій. Для створення сайту буде використано Salesforce Digital Experience [14], а для написання компонент та теми – LWC [4].

## **2.3 Специфікація вимог**

### **2.3.1 Середовище функціонування**

Оскільки, темою моєї дипломної роботи є розробка веб-аплікації для моніторингу об'єктів міста, то ця система буде представлена у вигляді сайту. Якщо це сайт, то середовище функціонування є доволі обширним. А саме, для запуску додатку потрібен лише будь-який веб-браузер, які можуть працювати на наступних операційних системах – IOS, Linux, Windows, Android, MacOS. Для використання продукту, у користувача повинен бути доступ до інтернету.

### **2.3.2 Вимоги зовнішніх інтерфейсів**

#### **2.3.2.1 Користувацькі інтерфейси**

Моя система матиме різні графічні інтерфейси, в залежності від ролі користувача, що цю систему використовуватиме. Коли користувач з роллю «гість» заходить у систему, то бачить лише 4 сторінки в меню, а саме – інформаційна панель, моніторинг, підтримка та логін. Також для гостя не буде відображатись кнопка для створення об'єктів міста та не будуть відображатись кнопки у випадаючому меню для редагування та видалення об'єкта міста. Адміністратор, бачитиме додатково сторінки “Категорії” та “Властивості”, використовуючи які він може керувати категоріями та їхніми властивостями. Також для адміністратора

відобразатиметься кнопка для створення об'єкта міста, та появляться нові кнопки у випадаючому меню для кожного об'єкта міста – “Видалити” та “Редагувати”.

### **2.3.2.2 Програмні інтерфейси.**

Для того, щоб сайт коректно відображався, слід використовувати наступні веб-браузери:

- Microsoft Edge версії 101.0.1270.39
- Google Chrome версії 101.0.4951.64
- Safari версії 14.1.2
- Opera 86.04363.50

### **2.3.2.1 Нефункціональні вимоги**

#### **Вимоги до продуктивності:**

- Час відповіді має бути короткий, не більше 2-3 секунд
- Система має витримувати до 500 користувачів одночасно
- Можливість додавання та редагування одночасно до 50 записів про об'єкт міста або категорію

#### **Вимоги до інтерфейсу**

Інтерфейс даної системи має бути максимально простим так зрозумілим і такий, щоб не виснажував читача, оскільки це моніторингова дані, вони мають бути добре читабельними та з відповідними шрифтами.

#### **Вимоги до безпеки**

Це насправді один з найважливіших моментів, оскільки це система з кількома ролями. І, насамперед, авторизація повинна відбуватись згідно з роллю, яку має користувач. Важливо, щоб гість не мав можливості додавати, редагувати, видалити категорію та об'єкт міста, а зареєстрований користувач не мав можливості видалити ні категорії ні об'єкту міста. Окрім цього паролі користувачів хешуються і не зберігаються в тому вигляді, в якому їх вводять користувач. Окрім цього, передбачена валідація паролю для користувача, а саме: довжина паролю не менше

6 символів, пароль повинен мати принаймні один нижній регістр, принаймні один верхній регістр , принаймні одну цифру та принаймні один спеціальний символ.

#### **2.4 Підсумок розділу**

У даному розділі було сформовано основну задачу, яку має вирішувати моя веб-аплікація для моніторингу об'єктів міста. Також було обрано інструменти, технології та підходи для реалізації проекту. А саме: для контролю версій буде використано Bitbucket, для бази даних буде використано Salesforce CRM, для керування метаданими - Salesforce Metadata Api , для створення сайту -Digital Experience. А для розробки компонент, тем буде використано фреймворк LWC та мову програмування Apex для серверної частини.

## РОЗДІЛ 3. ПРОЕКТУВАННЯ ВЕБ АПЛІКАЦІЇ ДЛЯ МОНІТОРНГУ ОБ'ЄКТІВ МІСТА

### 3.1 Проектування архітектури

Для зручності розробки системи на Salesforce з використанням Lightning Web Components (LWC) [4] та Apex [7][8][2] контролерів, я використовував архітектуру Salesforce-орієнтовану архітектуру з трьома рівнями [6]. Головна перевага такої архітектури полягає в тому, що оскільки кожен рівень працює над виконанням власних завдань, кожен рівень може розробляти інший розробник або і ціла команда. Також при зміні, наприклад системи керування базами даних, варто лише змінити код на рівня доступу до даних. Тобто при зміні якогось рівня, інші рівні будуть у повному захисті від неправильного редагування коду. Трирівнева архітектура складається з таких рівнів:

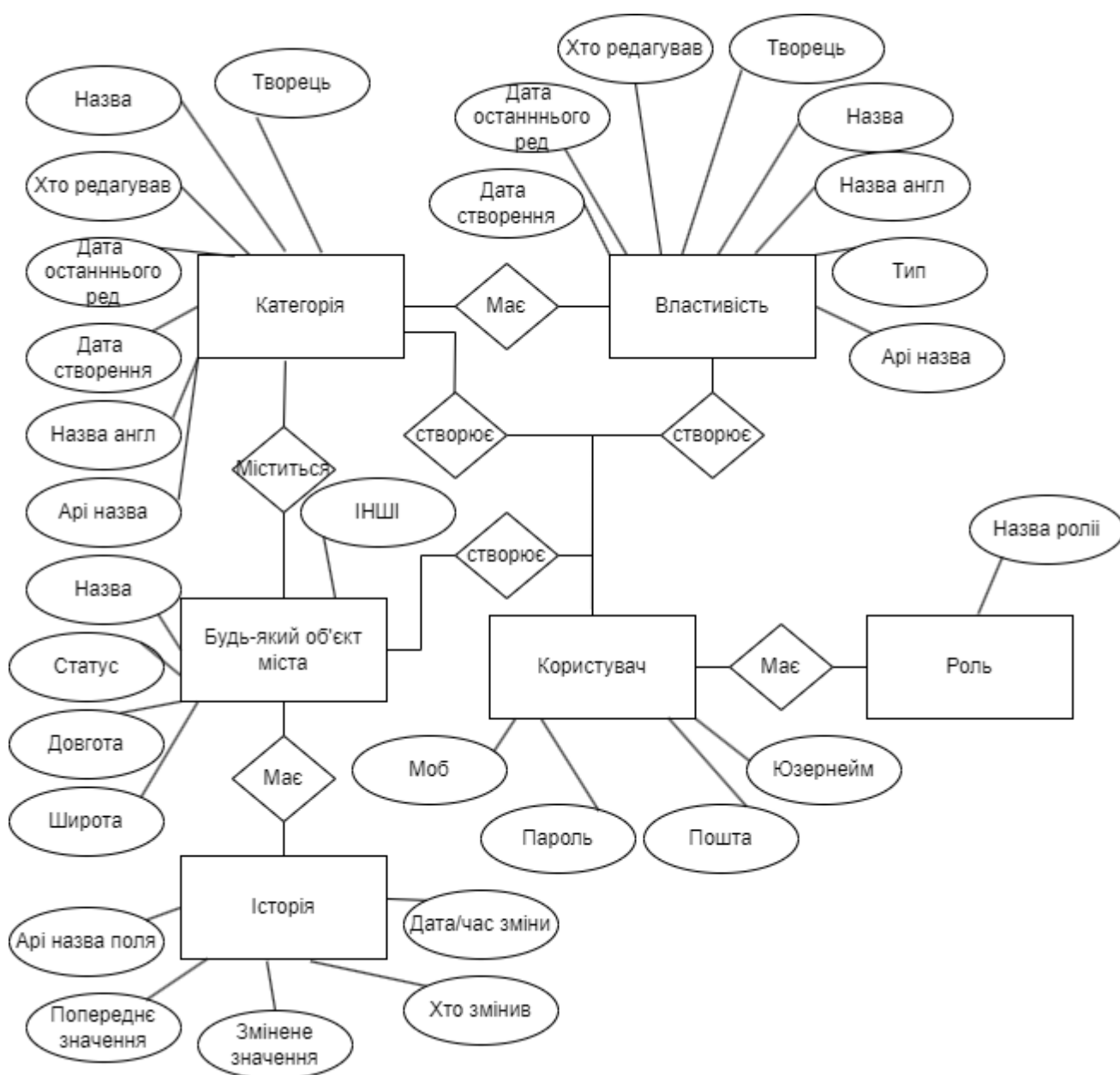
- Презентаційний рівень – рівень комунікації програми та інтерфейс користувача. Головне, що має тут виконуватись це процеси відображення інформації користувачеві Використання Lightning Web Components (LWC) дозволяє створювати компоненти з клієнтською логікою, які забезпечують взаємодію з користувачем та відображення інформації. Кожен LWC компонент відповідає за свою функціональність та може бути розроблений окремою командою.
- Рівень логіки – також відомий як середній рівень. На цьому рівні відбувається процес обробки інформації, зібраної на презентаційному рівні. Цей рівень містить основні сервісні методи, які потрібні для використання на презентаційному рівні. Apex [1] контролери є класами, написаними на мові Apex, які містять методи та логіку для обробки запитів та взаємодії з базою даних. Кожен контролер відповідає за конкретну функціональність або сценарій та може бути розроблений окремою командою.
- Рівень доступу до даних – це рівень, який містить інформацію про сутності, самі сутності, та доступ до даних в базі даних. Здебільшого, це включає Metadata Service, який є сервісом, що містить методи для керування

метаданими в Salesforce, такими як створення таблиць та додавання властивостей. Для взаємодії з Metadata Service використовується Salesforce Metadata API [11], який надає програмний інтерфейс для керування метаданими.

## 3.2 Проектування бази даних

### 3.2.1 Логічна модель бази даних

Для веб-аплікації для моніторингу об'єктів міста було створено логічну модель бази даних, яка зображена на *рис.3.2*.



*рис 3.2*

На цій моделі зображено сутності, які я буду використовувати у моїй аплікації. Оскільки , адміністратор сам створює категорії, то в моїй аплікації автоматично створюються таблиці для цих категорій.

### **3.2.2 Опис сутностей**

Для правильного функціонування моєї системи було створено такі сутності: «Користувач» , «Роль», «Категорія» , «Властивість» , решта таблиць створюються динамічно при створенні певної категорії, це «Об'єкт міста» та «Історія».

Сутність «Користувач» описує зареєстрованого користувача у веб аплікації та складається з таких атрибутів:

- Пароль- пароль користувача
- Юзернейм
- Пошта
- Мобільний телефон

Сутність «Роль» описує ролі для зареєстрованого користувача та складається з таких атрибутів:

- Назва ролі

Сутність «Категорія» описує які саме є категорії міста та складається з таких атрибутів:

- Творець- користувач, який додав категорію
- Назва
- Хто редагував – користувач, що останній вніс зміни
- Дата останнього редагування
- Дата створення
- Назва англійською – використовується для створення таблиці в базі даних
- Арі назва- генерується на основі назви таблиці

Сутність «Властивості» описує якими саме властивостями наділена певна категорія об'єктів міста та складається з таких атрибутів:

- Творець- користувач, який додав категорію
- Назва
- Хто редагував – користувач, що сотанній вніс зміни
- Дата останнього редагування
- Дата створення
- Назва англійською – використовується для створення таблиці в базі даних
- Арі назва- генерується на основі назви властивості
- Тип- тип властивості, наприклад текст, посилання, емейл, номер телефону

Сутність «Об'єкт міста» фактично не існує, а створюється динамічно і описує об'єкт міста та має такі обов'язкові атрибути:

- Статус- статус об'єкта міста, ми можемо вибрати статус лише з наступних значень- в стані ремонту/активний/на етапі будівництва/ не активний / тимчасово зачинений
- Широта- координата
- Довгота – координата
- Назва

Для кожного об'єкта міста автоматично створюється сутність «Історія» яка містить історію змін для даного об'єкта міста та має такі атрибути:

- Дата та час зміни
- Початкове значення- значення до моменту зміни
- Кінцеве значення- значення , на яке було змінено
- Арі назва поля



### 3.2.3 Опис зв'язків між таблицями

База даних матиме наступні зв'язки між сутностями:

- Між сутністю «Користувач» та сутністю «Роль» встановлено зв'язок один-до-одного
- Між сутністю «Користувач» та сутністю «Категорія» встановлено зв'язок один-до-багатьох
- Між сутністю «Користувач» та сутністю «Об'єкт міста» встановлено зв'язок один-до-багатьох
- Між сутністю «Користувач» та сутністю «Властивість» встановлено зв'язок один-до-багатьох
- Між сутністю «Категорія» та сутністю «Властивість» встановлено зв'язок один-до-багатьох
- Між сутністю «Категорія» та сутністю «Об'єкт міста» встановлено зв'язок один-до-багатьох
- Між сутністю «Об'єкт міста» та сутністю «Історія» встановлено зв'язок один-до-багатьох

### 3.3 Підсумок розділу

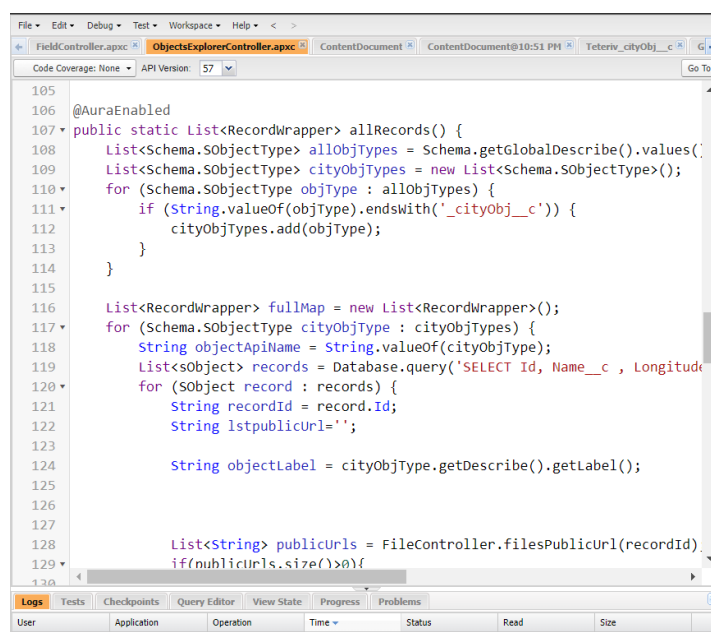
Під час проектування системи було вирішено використати трирівневу архітектуру. Вибір припав саме на неї насамперед, що це дуже популярний підхід, та, окрім цього дуже дієвий та ефективний, оскільки кожен рівень відповідає за своє. Окрім цього, був здійснений процес проектування бази даних. Для рисування діаграм було використано інструмент diagrams.

## РОЗДІЛ 4. РОЗРОБКА ТА ІМПЛЕМЕНТАЦІЯ ВЕБ АПЛІКАЦІЇ ДЛЯ МОНІТОРИНГУ ОБ'ЄКТІВ МІСТА

### 4.1 Імплементация веб аплікації для моніторингу об'єктів міста

Було вирішено використати та зареєструвати безкоштовну організацію розробника, у межах якої відбувався процес розробки веб аплікації для моніторингу об'єктів міста. Не дивлячись на те, що вона безкоштовна, ми маємо широкий спектр можливостей. Таких, як створення сайтів, компонент, 20 мегабайт пам'яті для файлів. Для написання коду було використано наступні середовища:

- Salesforce Developer Console (*рис 4.1*)
- Visual Studio Code



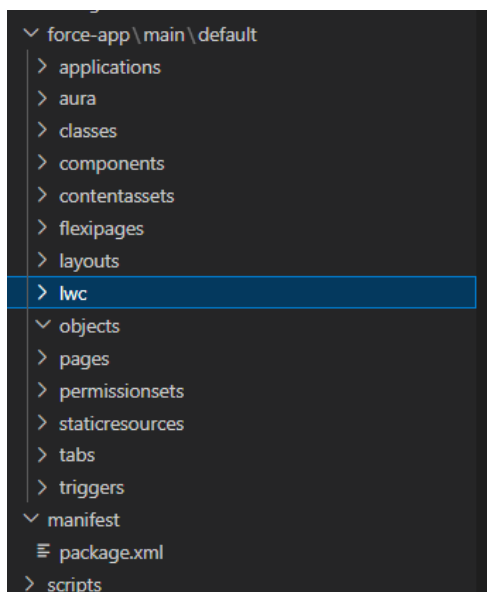
```

105
106 @AuraEnabled
107 public static List<RecordWrapper> allRecords() {
108     List<Schema.SObjectType> allObjTypes = Schema.getGlobalDescribe().values();
109     List<Schema.SObjectType> cityObjTypes = new List<Schema.SObjectType>();
110     for (Schema.SObjectType objType : allObjTypes) {
111         if (String.valueOf(objType).endsWith('_cityObj_c')) {
112             cityObjTypes.add(objType);
113         }
114     }
115
116     List<RecordWrapper> fullMap = new List<RecordWrapper>();
117     for (Schema.SObjectType cityObjType : cityObjTypes) {
118         String objectApiName = String.valueOf(cityObjType);
119         List<Object> records = Database.query('SELECT Id, Name__c, Longitude');
120         for (Object record : records) {
121             String recordId = record.Id;
122             String lstpublicUrl='';
123
124             String objectLabel = cityObjType.getDescribe().getLabel();
125
126
127
128             List<String> publicUrls = FileController.filesPublicUrl(recordId);
129             if(publicUrls.size()>0){

```

*рис 4.1 Salesforce Developer Console*

Salesforce Developer Console - вбудований інструмент на платформі Salesforce, який дозволяє писати, налагоджувати та тестувати код прямо у вашому браузері. Даний інструмент я використовував для написання Apex класів, оскільки саме даний інструмент краще описує помилки в коді. Для написання LWC (Lightning Web Component) компонент було використано Visual Studio Code за підтримки SFDX (Salesforce Developer Experience [5] [14]) команд для отримання та зберігання даних на організацію. Структуру проєкту зображена на *рис 4.2*



*рис 4.2 Структуру проекту Salesforce*

Система розроблялась за використанням архітектурного шаблону MVC. MVC складається з трьох логічних компонентів, а саме:

- Model – модель, що відповідає моделі бази даних
- View – вигляд, тобто те, що користувач бачитиме перед очима. Вигляд включатиме в себе об'єкти інтерфейсу користувача. У випадку salesforce – це html сторінка LWC компоненти.
- Controller – налаштовує взаємодію між моделлю та виглядом. У випадку salesforce – це два контролери (Apex контролер та LWC контролер)

Процес передачі даних відбувається наступним чином- користувач робить певні дії, чим самим викликає методи LWC контролера, який, в свою чергу має імпортовані методи з Apex контролера. А вже саме у Apex контролері відбувається взаємодія з базою даних . Схема MVC для salesforce зображена на *рис 4.3*. Також, в рамках мого проекту було розроблено користувацький інтерфейс. При вході на сайт , користувач має можливість увійти. На *рис. 4.4* зображено форму для авторизації існуючого. Реєстрація користувача відбувається безпосередньо з використанням можливостей Salesforce. Після реєстрації адміністратором

користувача , на пошту, вказану у полі пошта, приходить лист для підтвердження пошти та встановлення паролю.

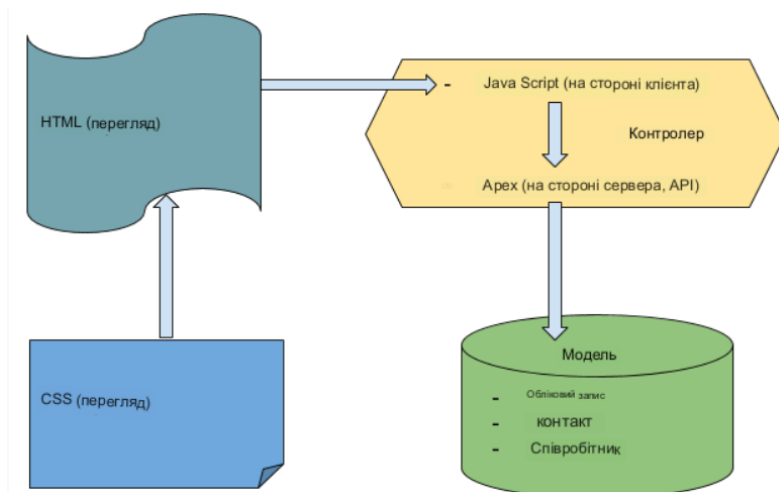


рис 4.3 Схеми MVC Salesforce

рис 4.4 Форм авторизації

Також, на випадок , якщо користувач забув свій пароль, у системі є можливість зміни паролю. Для того , щоб змінити пароль, необхідно натиснути *Забули пароль?* . Після чого, користувачеві система запропонує ввести логін (рис 4.5) , система сама віднайдет пошту для цього користувача , на яку потім відправиться лист з повідомлення з інструкціями для зміни паролю. Коли користувач натисне на посилання для зміни паролю, йому запропонують ввести новий пароль (рис 4.6). Після того, як користувач змінити пароль, пароль оновиться в базі даних та користувач може здійснювати вхід у систему, використовуючи новий пароль.

### Забули пароль?

рис 4.5

**CITYWATCH**  
Без урахування city та його light.

## Змінити пароль

Введіть новий пароль для  
svatoslavkondera@gmail.com.site . Обов'язково має  
містити принаймні:

- ✓ 8 символів
- ✓ 1 буква
- ✓ 1 число

\* Новий пароль

..... добре

\* Підтвердьте новий пароль

..... Співпадають

**Змінити пароль**

Востаннє пароль змінено 06.06.2023 о 14:02.

*рис 4.6*

Зайшовши на сайт , будь який користувач матиме змогу бачити інформаційну панель, зображену на *рис 4.7*. На даній панелі, з використанням сторонньої бібліотеки ChartJS для побудови графіків, відображені графіки , а саме:

- Стовпчаста діаграма для відображення категорій та кількість об'єктів у них
- Кругова діаграма що відображає статуси об'єктів міста, та кількість об'єктів міста з певним статусом у висотках так у одиницях
- Лічильники для об'єктів міста за кожним зі статусів та зміна протягом 24 годин

Дана інформаційна панель дає чітко зрозуміти користувачу, яких об'єктів і з якими статусами побільшало , що може бути корисним саме для моніторингових департаментів. Кольори для графіку категорій та кількість об'єктів генеруються випадковим чином.

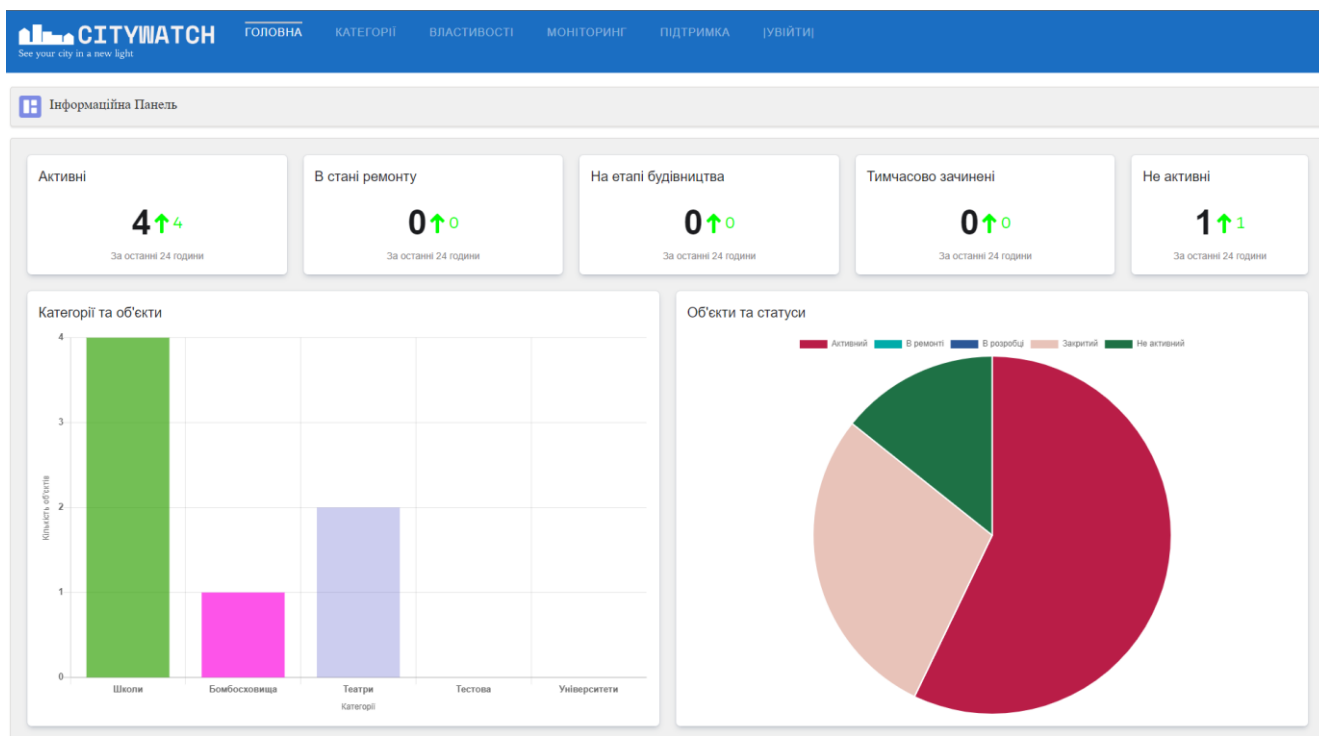


рис 4.7

Наступна сторінка (Категорії) *рис 4.8* доступна лише для адміністраторів сайту, тут є реалізований менеджер категорій, який являє собою динамічну таблицю для створення, редагування та видалення категорій. Адміністратору слід лише ввести назву категорії українською та англійською мовами (за останньою назвою створюється таблиця в базі даних). При натисканні “Додати категорію”, додається рядок до таблиці. Щоб редагувати достатньо курсор мишки поставити в поле з назвою. Для видалення- кнопку видалити біля рядка та натиснути підтвердити.

The screenshot shows the 'Менеджер категорій' (Category Manager) interface. It features a table with columns for category name, English name, creation date, owner, and an action button. A '+ Додати категорію' button is located at the top right.

Назва категорії	Назва Англійською	Дата створення	Власник	Дія
Бомбосховища	Shelters	2023-06-04T20:38:17.000Z	Sviatoslav, Kondera	🗑️
Театри	Theaters	2023-06-04T20:38:19.000Z	Sviatoslav, Kondera	🗑️
Тестова	Testkat	2023-06-04T22:45:11.000Z	Sviatoslav, Kondera	🗑️
Університети	Univers	2023-06-04T22:52:17.000Z	Sviatoslav, Kondera	🗑️
Школи	School	2023-06-04T11:12:21.000Z	Sviatoslav, Kondera	🗑️

Buttons: [Скасувати](#) [Зберегти](#)

рис 4.8

Сторінка “Властивості” *рис. 4.9* теж доступна лише адміністратору сайту та має схожу будову до попередньої. Тут адміністратор має можливість додати властивості до категорії. Після створення категорії, автоматично створюються 4 властивості- довжина, широта, статус та назва. Ці поля навіть адміністратор не має можливості редагувати та видаляти. Для відображення властивості слід обрати категорію з випадаючого списку, після чого ми побачимо всі властивості для даної категорії. Для створення нової властивості, слід натиснути кнопку “Додати властивість”, ввести назву поля, тип поля (Number/Text/ Email/Url/ Text/Checkbox ) та ввести Арі назву поля англійською мовою. Для редагування достатньо просто навести курсор та змінити необхідні дані. Для видалення – натиснути кнопку видалити.

The screenshot shows the 'Налаштування властивостей' (Properties Settings) page. At the top, there is a navigation bar with 'ГОЛОВНА', 'КАТЕГОРІЇ', 'ВЛАСТИВОСТІ', 'МОНІТОРИНГ', 'ПІДТРИМКА', and 'ВІЙТИ'. Below the navigation bar, there is a dropdown menu for 'Університети'. The main content area is a table with the following columns: 'Назва поля', 'API-назва поля', 'Тип поля', 'Дата створення', 'Власник', and 'Дія'. The table contains five rows of data:

Назва поля	API-назва поля	Тип поля	Дата створення	Власник	Дія
Довжина	Longitude__c	Number	2023-06-04T22:52:17.000Z	Sviatoslav, Kondera	🗑️
Широта	Latitude__c	Number	2023-06-04T22:52:17.000Z	Sviatoslav, Kondera	🗑️
Назва	Name__c	Text	2023-06-04T22:52:17.000Z	Sviatoslav, Kondera	🗑️
Статус	Status__c	Picklist	2023-06-04T22:52:17.000Z	Sviatoslav, Kondera	🗑️
Рік Заснування	Year__c	Number	2023-06-04T22:53:15.000Z	Sviatoslav, Kondera	🗑️

At the bottom right of the table, there are buttons for 'Скасувати' and 'Зберегти'.

*рис 4.9*

Головною сторінкою, яка буде доступна всім, буде сторінка, призначена для моніторингу об’єктів міста (*рис 4.10*). Перейшовши на неї користувач бачить екран розділений на 3 колонки. Перша колонка- фільтри, які ми можемо використати для вибору об’єктів міста за певними критеріями, а саме: за ключем пошуку (назвою), за відстанню до об’єкта міста (якщо користувач надав дозвіл на використання власної геопозиції) та обрати, які категорії показувати, а які ні. У правій частині екрану користувач бачить гугл [3] мапу з маркерами для об’єктів

міста, які теж піддаються фільтрації. В самому центрі екрану розташувався список об'єктів міста.

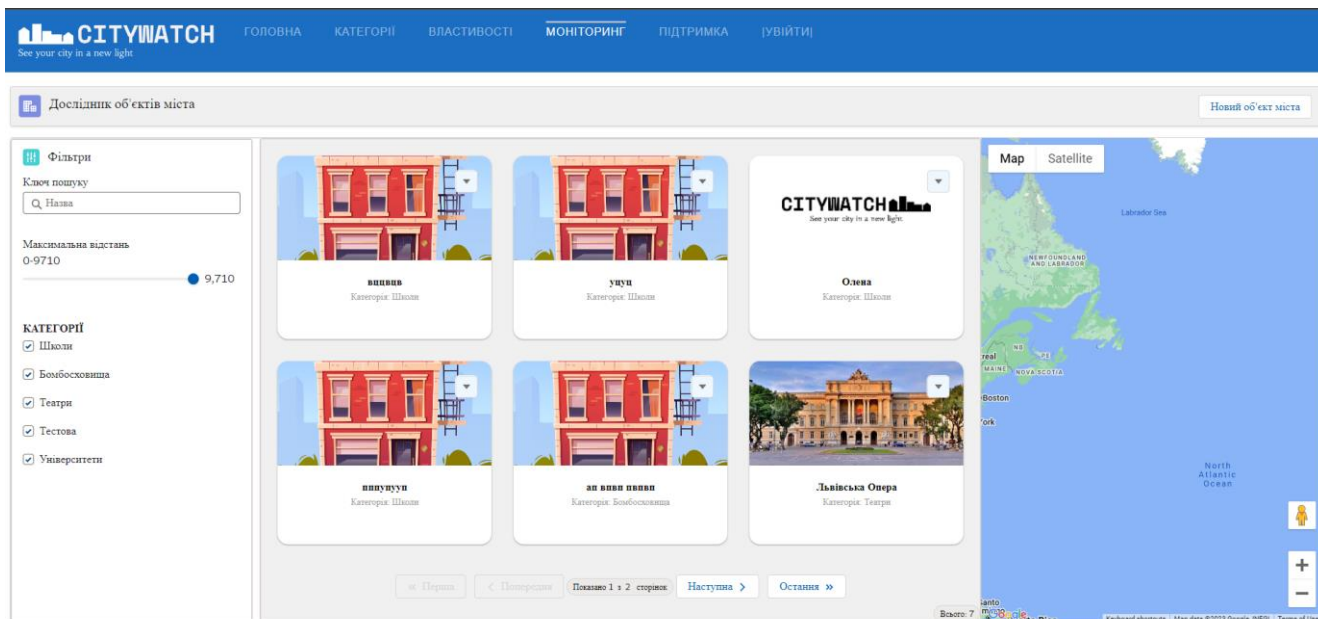


рис 4.10

Кнопка для додавання нового об'єкта міста розташована у правому верхньому куті. Вона відображається лише для адміністраторів системи. Натиснувши на неї, адміністратор повинен спершу обрати категорію (рис 4.11), наступним кроком слід ввести дані для об'єкта міста та обрати локацію (рис 4.12), використовуючи гугл карту. Останнім етапом створення об'єкта міста є додавання фотографій. Для зручного користування даним завантажувачем файлів, було розроблено методи для попереднього відображення даних зображень (рис 4.13).

 The screenshot shows a form titled "Створити об'єкт міста". The form has a white background and a thin border. At the top, it says "Створити об'єкт міста". Below this, there is a section labeled "Виберіть категорію:" followed by five radio button options: Школи, Бомбосховища, Театри, Тестова, and Університети. At the bottom right of the form, there are two buttons: "Скасувати" (light blue) and "Далі" (dark blue).

рис 4.11



### Створити об'єкт міста

Назва:  Статус:

Ректор:

Виберіть місце розташування

рис 4.12

### Створити об'єкт міста

Зображення

систему рівнянь:

$$y_i[q_0] - \Phi_i(x_2, x_3, x_1) = 0; i = \overline{1, m}$$

показники  $x_2, x_3$  при незмінних  $y_i$

$$B^* = \{B_i^*, i = \overline{1, m}\}, B_i^* = \{y_i | b_i^- \leq y_i \leq b_i^+\}$$

$$D_1^* = \{x_1 | x_1 = \langle x_{1j}, j = \overline{1, n_1} \rangle, d_{1j}^- \leq x_{1j} \leq d_{1j}^+\}$$

$$D_2^* = \{x_2 | x_2 = \langle x_{2j}, j = \overline{1, n_2} \rangle, x_{2j}^- \leq x_{2j} \leq x_{2j}^+\}$$

$$D_3^* = \{x_3 | x_3 = \langle x_{3j}, j = \overline{1, n_3} \rangle, x_{3j}^- \leq x_{3j} \leq x_{3j}^+\}$$

виконання умов  $D_2^* \subseteq D_2^+, D_3^* \subseteq D_3^+$

6vr (1).png.png

Ivan-Franko-National-...

рис 4.13

Для видалення фотографії, слід натиснути на неї, після чого відкриється повноцінний перегляд файлу з можливістю видалення (рис 4.13).

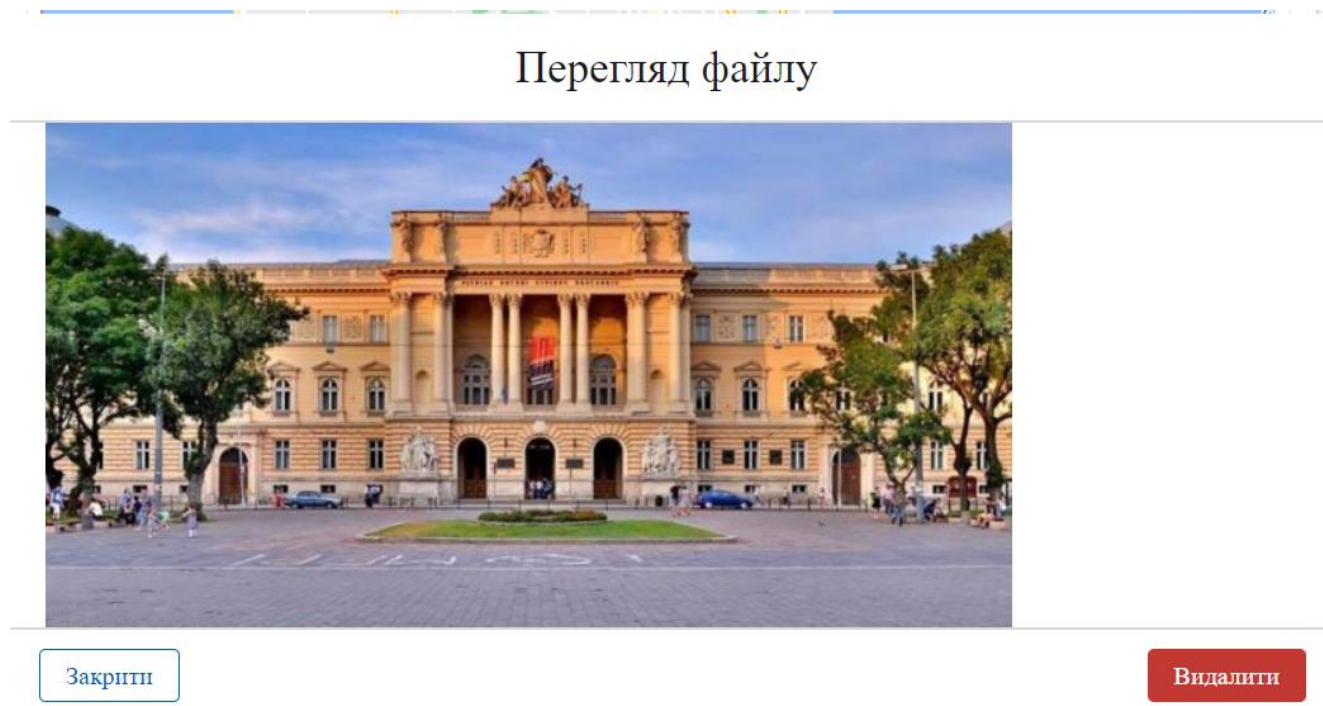


рис 4.14

На кожному об'єкті міста розташована кнопка, яка відкриває меню (дії, які можна виконати з певним об'єктом міста). Для адміністратора є наступні можливості – редагувати, видалити та оглянути деталі (рис 4.15). Для гостя системи доступна лише можливість оглянути деталі.

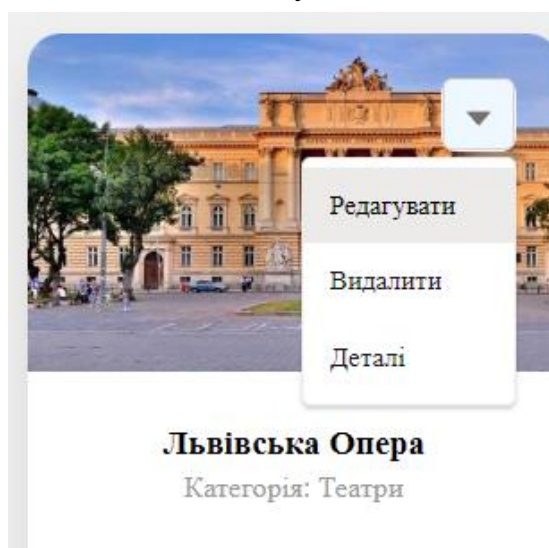


рис 4.15

При натисканні на видалення, об'єкт міста видалається. Для перевірки на успіх операцій, була розроблена система відображення повідомлень про успіх (зеленого кольору) та про помилки (червоного), які відображаються після дій адміністратора з об'єктами (рис 4.16). При натиску на кнопку для редагування – адміністратор має можливість змінити всі значення для властивостей, обрати нову локацію та видалити або додати фотографії (рис 4.17).



рис 4.16

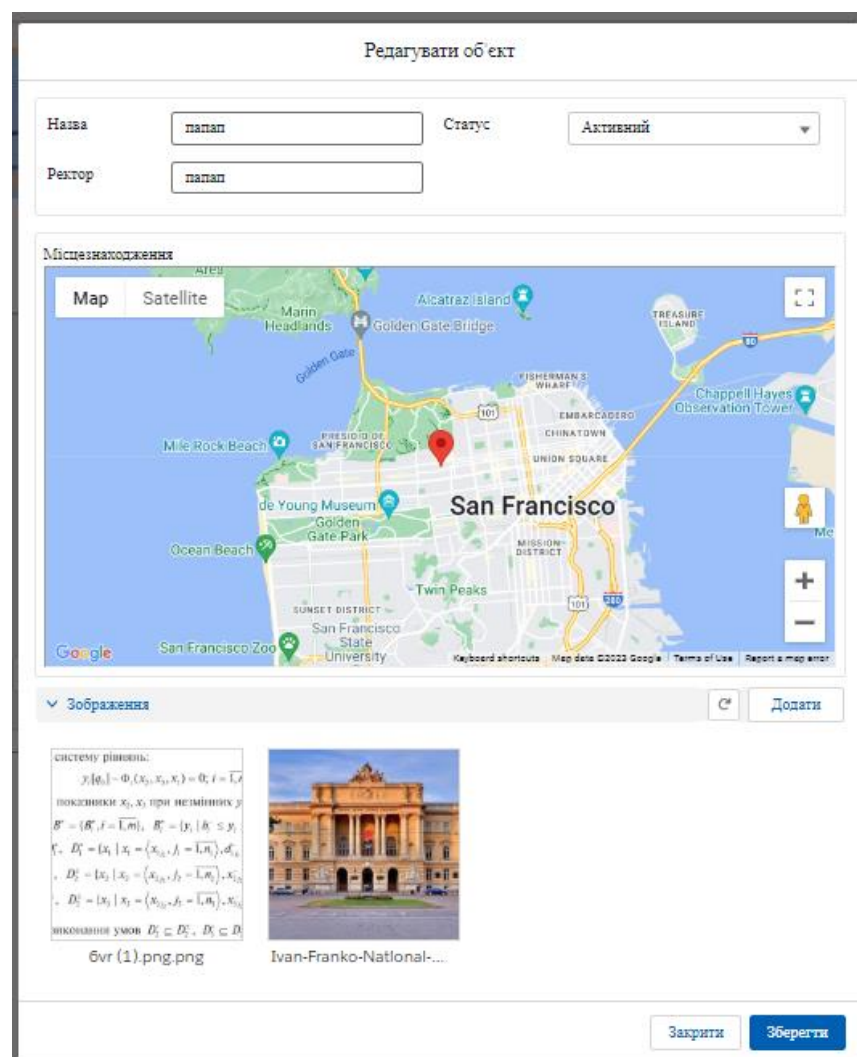
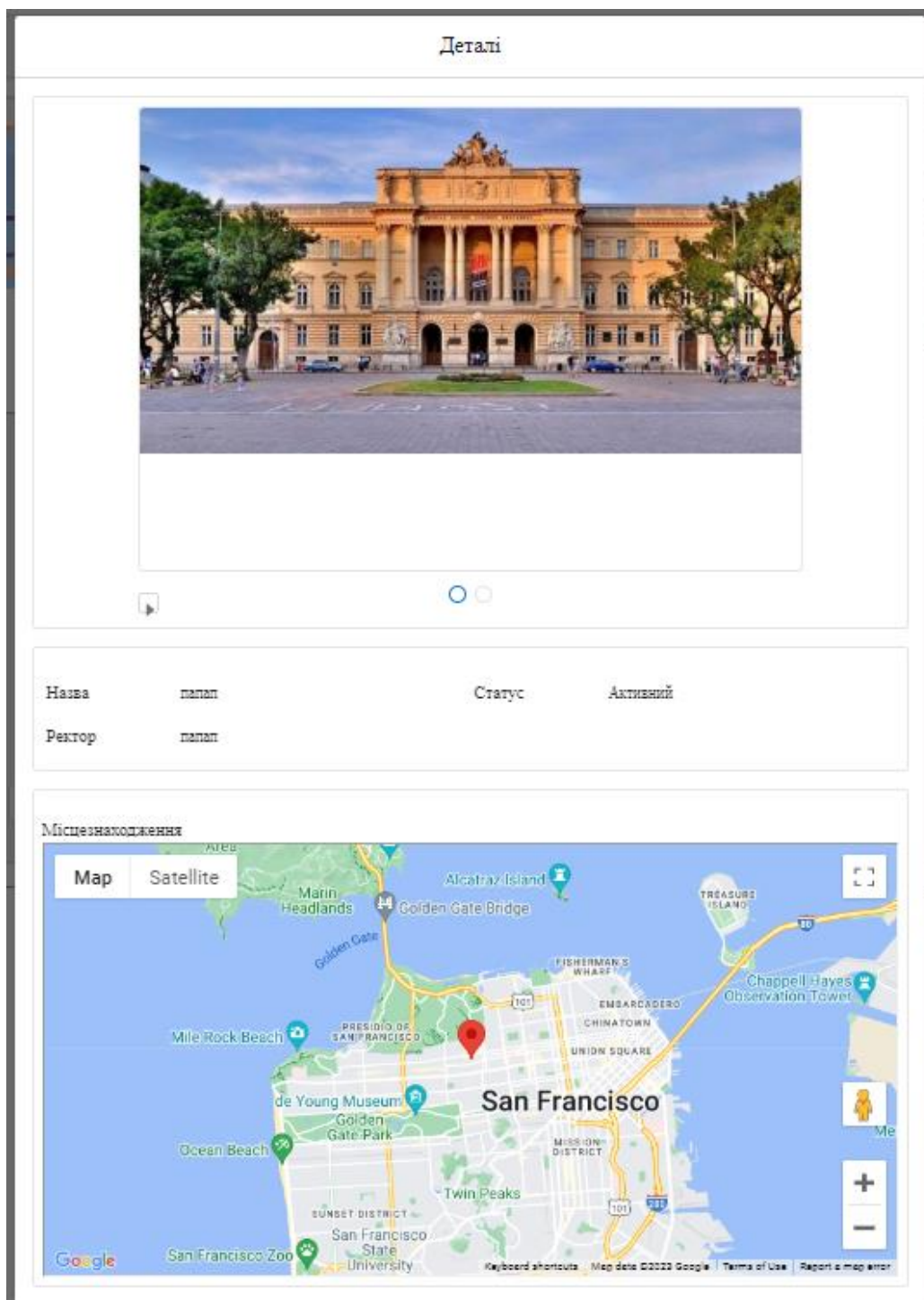


рис 4.17

Головна можливість для гостей та адміністраторів – це оглядати деталі для об'єкта міста. Користувачеві відображається “карусель” з фотографій, які перегортаються кожні кілька секунд з можливістю натиснути кнопку пауза та

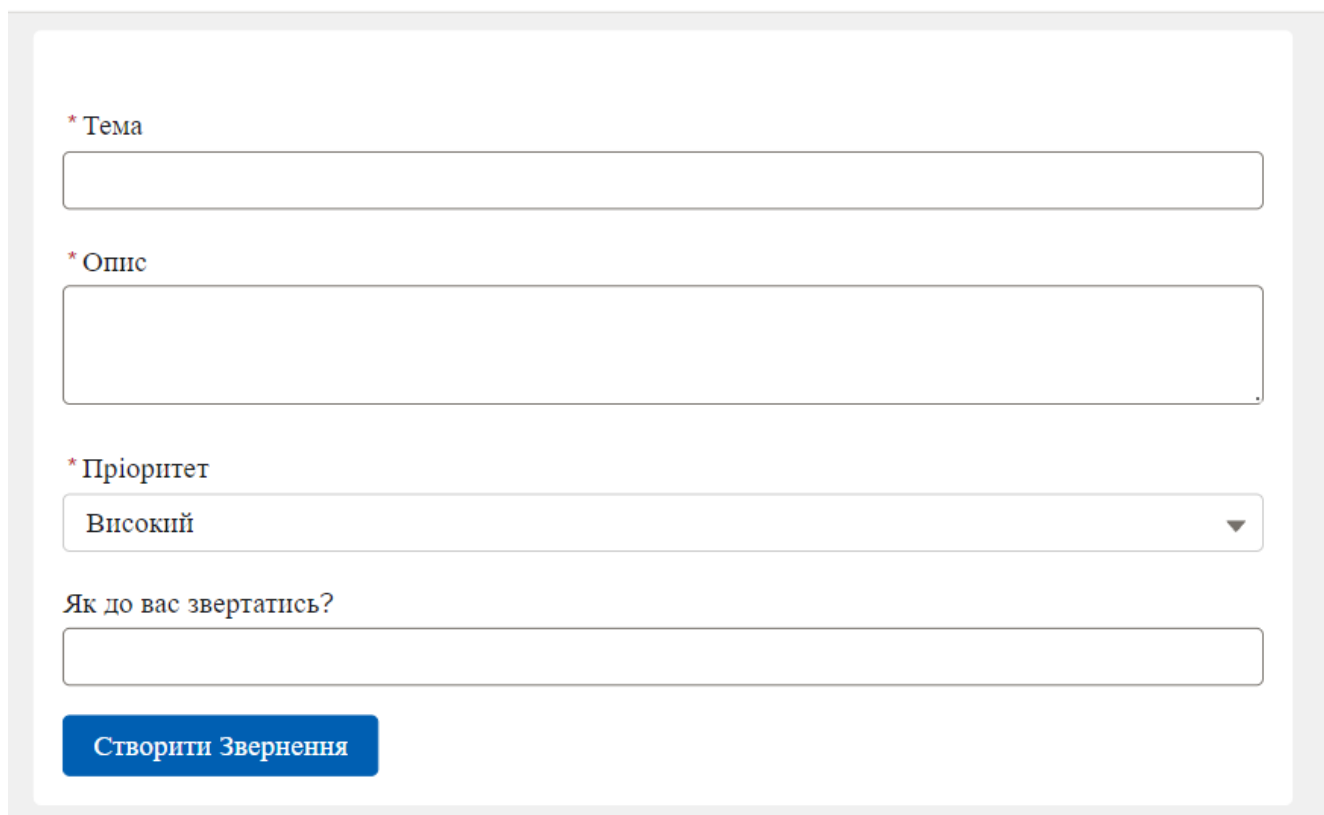
зупинити автоматичне перегортання фотографій. Інтерфейс користувача для перегляду деталей зображень на *рис 4.18* . Він складається з трьох частин-галереї, властивостей та гугл карти з маркером.



*рис 4.18*

Ще одною можливістю для користувача , у разі виникнення питань чи проблем з системою, є можливість створити звернення , заповнивши форму. Дане звернення приходить листом на пошту адміністратору системи. Для цього

користувачу слід заповнити тему, опис, пріоритет та я користувач хочу, щоб до нього звертались. Форма заповнення зображена на *рис. 4.19*.



The image shows a web form for creating a request. It contains the following elements:

- A text input field labeled "\* Тема" (Topic).
- A larger text input field labeled "\* Опис" (Description).
- A dropdown menu labeled "\* Пріоритет" (Priority) with the value "Високий" (High) selected.
- A text input field labeled "Як до вас звертатись?" (How to contact you?).
- A blue button labeled "Створити Звернення" (Create Request).

*рис 4.19*

## 4.2 Підсумок розділу

В процесі розробки було використано систему salesforce як фундамент та LWC (Lightning Web Component) з використанням архітектурного шаблону MVC. Також було розроблено користувацькі інтерфейс згідно з вимог. На додачу, веб-аплікацію для моніторингу об'єктів міста було протестовано вручну і перевірено чи отриманий функціонал відповідає вимогам користувача.

## ВИСНОВКИ

Моніторинг об'єктів міста доволі цікава і популярна тема, якою цікавляться чимало людей у світі, але в нас, в Україні, люди цьому не приділяють увагу. Проблема, насамперед, є в тому, що моніторингові департаменти не хочуть щось міняти, осучаснювати свою систему для моніторингу, використовують застарілі методи представлення моніторингового матеріалу. Це в свою чергу не відповідає вимогам кінцевого користувача, що, відповідно, його відштовхує. Задля забезпечення комфортності доступу до моніторингової інформації було розроблено веб-аплікацію для моніторингу об'єктів міста.

Застосування трірівневої архітектури сприяє системному та ефективному розподілу роботи між командами розробників. Використання Salesforce, зокрема Lightning Web Components (LWC), Apex контролерів, Metadata Service та Metadata API, дає можливість створювати функціональний та зручний інтерфейс для користувачів.

Розроблений веб-додаток дозволяє створювати різні категорії об'єктів міста з різноманітними типами полів, а також додавати категорію бомбосховищ (найактуальніша проблема сьогодення). Завдяки фільтрації, користувач може легко знаходити об'єкти за певними критеріями та встановлювати радіус для відображення об'єктів у певній відстані від користувача. Крім того, додаток надає можливість ввести значення полів, додати локацію та фотографії для кожного об'єкта. Детальний перегляд, редагування та видалення об'єктів також доступні користувачам. Важливим елементом додатка є інформаційна панель з графіками та зміною статусів за останні 24 години.

Незважаючи на те, що подібні додатки наразі не є широко поширеними в Україні, це відкриває можливості для привернення більшої кількості користувачів шляхом надання якісного та зручного продукту. Розробка веб-додатків для моніторингу об'єктів у містах має потенціал для покращення систем моніторингу та забезпечення більшої ефективності в управлінні міськими об'єктами.

## СПИСОК ЛІТЕРАТУРИ

1. Apex Developer Guide (Режим доступу: [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_intro\\_what\\_is\\_apex.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_intro_what_is_apex.htm))
2. Future method in salesforce (Режим доступу: <https://www.apexhours.com/future-method-in-salesforce/>)
3. Google Maps Api code examples (Режим доступу: <https://developers.google.com/maps/documentation/javascript/examples> )
4. Introducing Lightning Web Components (Режим доступу: <https://developer.salesforce.com/docs/component-library/documentation/en/lwc>).
5. LWR Sites for Experience Cloud (Режим доступу: <https://www.learnentityframeworkcore.com/> ).
6. Multitier architecture (Режим доступу: [https://en.wikipedia.org/wiki/Multitier\\_architecture](https://en.wikipedia.org/wiki/Multitier_architecture)).
7. Queueable Apex Salesforce (Режим доступу: <https://www.sfdcpoint.com/salesforce/queueable-apex-salesforce/>)
8. Visualforce Developer Guide (Режим доступу: [https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages\\_intro\\_what\\_is\\_it.htm](https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages_intro_what_is_it.htm) )
9. Web system for city monitoring «City monitor» (Режим доступу: <https://www.centreforcities.org/city-monitor/>).
10. What is Bitbucket (Режим доступу: <https://confluence.atlassian.com/confeval/development-tools-evaluator-resources/bitbucket/bitbucket-what-is-bitbucket>).
11. Understanding MetadataApi (Режим доступу: [https://developer.salesforce.com/docs/atlas.en-us.api\\_meta.meta/api\\_meta/meta\\_intro.htm](https://developer.salesforce.com/docs/atlas.en-us.api_meta.meta/api_meta/meta_intro.htm)).
12. ДЕПАРТАМЕНТ ЕКОЛОГІЇ ТА ПРИРОДНИХ РЕСУРСІВ ЛЬВІВСЬКОЇ ОБЛАСНОЇ ДЕРЖАВНОЇ АДМІНІСТРАЦІЇ (Режим доступу: <https://deply.gov.ua/>).

13.Salesforce CRM (Режим доступа: <https://www.salesforce.com/crm/>).

14.Build customer journeys on a powerful digital experience platform (Режим доступа: <https://www.salesforce.com/products/experience-cloud/overview/>).