

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

**Факультет прикладної математики та інформатики**

(повне найменування назва факультету)

**дискретного аналізу та інтелектуальних систем**

(повна назва кафедри)

## **Дипломна робота**

**РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ЕКО-ПЛАТФОРМИ  
"ЧИСТА УКРАЇНА"**

Виконала: студентка групи ПМІ-43с  
спеціальності

122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Магденко В. Ю.

(підпис)

(прізвище та ініціали)

Керівник

Квасниця Г.А.

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

2023

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

Факультет Прикладної математики та інформатики

Кафедра Дискретного аналізу та інтелектуальних систем

Спеціальність 122 «Комп'ютерні науки»

(шифр і назва)

«ЗАТВЕРДЖУЮ»

Завідувач кафедри \_\_\_\_\_

"31 "серпня" 2022 року

**З А В Д А Н Н Я**

**НА ДИПЛОМНУ У РОБОТУ СТУДЕНТУ**

Магденко Валерії Юрїївни

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка ВЕБ-застосунку для екологічної платформи "Чиста Україна"

керівник роботи \_\_\_\_\_ ,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені Вченою радою факультету від "13" вересня 2022 року № 15

2. Строк подання студентом роботи 13.05.2023р.

3. Вихідні дані до роботи \_\_\_\_\_

4. Зміст дипломної роботи (перелік питань, які потрібно розробити) \_\_\_\_\_

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)



## Реферат

Дана дипломна робота складається з вступу, переліку умовних позначень, п'ятьох розділів, висновків та списку використаних джерел. Обсяг роботи: 46 сторінок, 21 рисуноків. Список використаних джерел складається з 19 найменувань.

Мета роботи - розробка еко-платформи "Чиста Україна", яка має на меті привернути увагу користувачів до екологічної ситуації в Україні та сприяти її покращенню.

У процесі аналізу існуючих ресурсів були виявлені недоліки та обмеження, що спонукали до створення власного веб-застосунку з чітко сформульованими ідеями та вимогами. Головним завданням диплому було створення ефективної платформи, яка могла б залучити громадськість до активної участі у вирішенні екологічних проблем. Впровадження платформи може мати значний вплив на свідомість громадян і спонукати їх до прийняття активних кроків у напрямку сталого розвитку та збереження навколишнього середовища. "Чиста Україна" може стати важливим інструментом для поширення свідомості про екологічні проблеми, залучення громадськості до спільних дій та сприяння створенню екологічно свідомого суспільства в Україні.

У результаті роботи було успішно реалізовано еко-платформу "Чиста Україна", яку можна знайти у вільному доступі. Застосунок реалізований з використанням двох основних технологій: Python та React.

## Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	6
ВСТУП .....	7
<b>1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....</b>	<b>9</b>
<b>1.1 Актуальність .....</b>	<b>9</b>
<b>1.2 Аналіз аналогів і подібних застосунків .....</b>	<b>10</b>
<b>1.3 Постановка задачі .....</b>	<b>12</b>
<b>2. ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ОБРАНІ ТЕХНОЛОГІЇ .....</b>	<b>14</b>
<b>2.1 Клієнт-серверна архітектура.....</b>	<b>14</b>
<b>2.2 Технології для серверної частини .....</b>	<b>14</b>
<b>2.3 Технології для клієнтської частини .....</b>	<b>16</b>
<b>2.4 База даних та технології для роботи з нею.....</b>	<b>19</b>
<b>3. ПРОГРАМНА РЕАЛІЗАЦІЯ .....</b>	<b>21</b>
<b>3.1 Структура бази даних .....</b>	<b>21</b>
<b>3.2 Структура проекту серверної частини.....</b>	<b>21</b>
<b>3.3 Структура проекту клієнтської частини .....</b>	<b>24</b>
<b>3.4 Аналізатори синтаксису коду.....</b>	<b>25</b>
<b>4. РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>27</b>
<b>4.1 Розгортання клієнтської частини .....</b>	<b>28</b>
<b>4.2 Розгортання серверної частини.....</b>	<b>28</b>
<b>5. ФУНКЦІОНАЛЬНІСТЬ ВЕБ-ЗАСТОСУНКУ .....</b>	<b>31</b>
<b>5.1 Логін/Реєстрація.....</b>	<b>31</b>
<b>5.2 Нотатки .....</b>	<b>32</b>
<b>5.3 Профіль .....</b>	<b>34</b>
<b>5.4 Додавання/редагування нотатки.....</b>	<b>35</b>
<b>5.5 Нотатка.....</b>	<b>36</b>
<b>5.6 Процес виконання нотатки .....</b>	<b>39</b>
<b>5.7 Сповіщення .....</b>	<b>40</b>
<b>5.8 Події.....</b>	<b>41</b>
<b>5.9 Детальна інформації про подію та голосування .....</b>	<b>41</b>
<b>5.10 Можливість запросити друга .....</b>	<b>43</b>
ВИСНОВКИ .....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	45

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

**HTTP** — це прикладний протокол передачі даних у мережі.

**WSGI** (Web Server Gateway Interface) - це специфікація, що визначає стандартний інтерфейс між веб-сервером та веб-додатком, написаним на мові програмування Python.

**ASGI** (Asynchronous Server Gateway Interface) - це специфікація, що визначає інтерфейс між веб-сервером та додатком, написаним з використанням асинхронної програмної моделі.

**API** – сукупність усіх методів серверу.

**Фреймворк** – платформа для розробки API.

## ВСТУП

Екологічна ситуація в Україні є досить непростою. Наше природне середовище постійно стикається зі значними екологічними проблемами, які мають серйозні наслідки для здоров'я людей і природних екосистем.

Однією з найважливіших причин негативного стану довкілля є високе промислове навантаження. Викиди підприємств на кожний 1 квадратний кілометр площі України становлять приблизно 6,5 тон. Це надзвичайно велика кількість забруднюючих речовин, які потрапляють у повітря, воду і ґрунт. Крім того, кількість викидів на одну людину в Україні перевищує 80 кілограмів щороку, що свідчить про серйозну проблему забруднення навколишнього середовища. [1]

Іншою проблемою є недостатня утилізація відходів. Тільки 5-10% сміття піддається переробці, тоді як решта відходів накопичується на полігонах, стаючи джерелом екологічної небезпеки. Це ставить під загрозу якість ґрунту, водних ресурсів і здоров'я людей. [2]

Для вирішення цих проблем необхідна посильна участь кожного громадянина та державних органів. Важливим кроком в цьому напрямку є підвищення рівня інформованості населення про екологічні проблеми. На жаль, існує недостатньо веб-сайтів та застосунків, які б могли ефективно інформувати та мотивувати населення до активної участі у поліпшенні екологічної ситуації.

Саме з цією метою було розроблено еко-платформу "Чиста Україна". Ця платформа буде доступна для користування всім бажаючим і допоможе підвищити рівень інформованості населення про екологічні проблеми в Україні. Вона забезпечить користувачам доступ до актуальної і достовірної інформації про стан довкілля, екологічні проекти та способи особистого внеску у покращення екологічної ситуації.

Метою "Чистої України" є сприяння формуванню екологічно свідомого суспільства та мобілізація населення для активної участі у збереженні навколишнього середовища. Шляхом популяризації знань, поширення свідомої екологічної поведінки та надання зручних інструментів для взаємодії та співпраці,

ми можемо досягти позитивних змін і зробити нашу Україну чистішою та здоровішою для майбутніх поколінь.



## 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Актуальність

Серед усіх екологічних проблем України, проблема сміття є «видимою», адже з нею люди стикаються кожен день. За результатами досліджень люди в Україні скаржаться на нерегулярний вивіз сміття, засмічення міських вулиць і зелених зон, відсутність інфраструктури для сортування і нестачу сміттєспалювальних заводів.

В Україні є актуальною проблема засмічення парків, лісів, узбіч доріг, узбережжя водойм. На узбіччя доріг та в лісах нерегулярно викидається значна кількість сміття. Навколишні посадки і ліси обросли несанкціонованими звалищами. Подорожуючи автомобілем від заходу до сходу, з півночі на південь, можна побачити не тільки неймовірну красу нашої України, але і забруднені узбіччя при дорогах та неприбрані зони відпочинку.

Місцева влада бореться зі сміттям протягом багатьох років. Звичайно, те що проводиться на місцевому рівні і спрямоване на вирішення проблеми відходів, має важливе значення, проте покладатися виключно на владу не варто – люди повинні самі робити все для того, щоб правильно позбавлятися від відходів і залишати після себе чисті території.

Завжди приємно споглядати доглянуті вулиці, прибрані узбіччя, порядок біля кожного будинку, чистий ліс. Щоб подорожі містами України приносили тільки захват і приємні враження, а наша країна була однією з найчистіших країн Європи.

Тому, для того щоб це стало системою – кожен житель повинен постаратися на благо свого рідного міста, країни і себе зокрема. А для цього потрібно зовсім небагато – не залишати за собою сміття та прибирати по можливості інші території. Кожен з нас може долучитись до покращення екологічного стану нашої Батьківщини. Світ змінюється, світ дорослішає і люди нарешті почали розуміти справжню ціну кожній краплині чистої води, кожній пухнастій сніжинці, кожному сонячному променю. Тому найменше, що може зробити кожен із нас – любити й оберігати природу навколо себе, тримати цей світ у чистоті. Адже все починається з малого!

Саме для того, щоб тримати нашу Україну в чистоті і розроблено веб-застосунок еко-платформа «Чиста Україна» з базовим функціоналом, який буде простим у користуванні і дуже корисним для нашої країни. Цей застосунок не лише сприятиме зниженню обсягу сміття на нашій території, але й підвищить рівень усвідомленості та актуальності екологічного стану нашої країни серед населення.

## **1.2 Аналіз аналогів і подібних застосунків**

У процесі дослідження було виявлено декілька проектів, що займаються екологічними ініціативами і мають певні схожості з моєю ідеєю. У світі існують деякі проекти та додатки, які спрямовані на публікацію брудних місць, організацію прибирань та набирання балів за участь у цих заходах. Ось кілька прикладів:

- Litterati. Цей додаток дозволяє користувачам фотографувати та позначати сміттєві забруднення за допомогою геолокації. Відмічені фотографії додаються до бази даних, де вони використовуються для аналізу сміттевого забруднення та планування прибирань. [3]
- TrashOut: TrashOut є глобальною спільнотою та мобільним додатком, які допомагають користувачам знаходити та документувати забруднені місця, особливо ті, що пов'язані зі звалищами та незаконним сміттєзвалищами. Він також надає можливість стежити за прогресом прибирання та спілкуватися з іншими користувачами. [4]

Оцінюючи поточний стан розробок в екологічній сфері в Україні, було встановлено, що немає аналогічних проектів, які повністю відповідали би запропонованій ідеї. Проте, варто зазначити декілька українських проектів, що займаються екологічними ініціативами. Наприклад, проект "ЕкоСистема" є державною онлайн-платформою, яка надає актуальну інформацію про стан довкілля в Україні [2]. Також, "Екоплатформа" об'єднує провідних українських експертів, екологічних аудиторів та вчених з метою спільної роботи в галузі промислової екології та екологічної безпеки.

Значним внеском є волонтерська організація "Екодія", де волонтери активно діляться своїми планами, фотозвітами та інформацією про екологічну ситуацію в

Україні. Крім того, існує волонтерська програма "Будуємо Україну Разом", яка об'єднує українську молодь у мережу активних громадян через спільну працю, подорожі та неформальну освіту з метою поліпшення соціального життя в Україні.

Ці проекти та ініціативи показують активність громадян України у сфері екології та волонтерства, що підтверджує потенціал та зацікавленість суспільства в покращенні екологічного стану країни.

Незважаючи на існування деяких подібних проектів, важливо зазначити їхні недоліки відносно запропонованого застосунку:

- Обмежений функціонал: Деякі існуючі додатки є обмежені у функціоналі, бо пропонують лише збір інформації про забруднені місця. Запропонована платформа буде надавати додаткові можливості, такі як участь у прибиранні і бали за це.
- Актуальність і важливість інформації про певне забруднення. Існуючі аналоги тільки містять інформацію про це, а в запропонованому застосунку можна ставити вподобання на важливу нотатку і таким чином піднімати її популярність.

Крім того, моя платформа буде відрізнятися в таких аспектах:

- Геолокаційний пошук: Дана платформа надаватиме можливість шукати нотатки/події поблизу. Це дозволить зручно відстежувати та визначати пріоритети для прибирань.
- Залучення громадськості та соціальну спільноту: Дана платформа може стати місцем розвитку для волонтерських рухів та організацій, адже функціонал публікування подій чудово підходить для оголошень про толоку, громадські роботи та інше.
- Статистика зацікавленості в події завдяки можливості голосувати за участь.
- Залучення до прибирання інших: надсилання інформації про екологічну подію у декілька простих кроків допоможе залучити все більше людей до актуальності цієї проблеми.

Здорова екологія стає все більш важливою темою, і розвиток подібного

застосунок в Україні є дуже важливим з точки зору підвищення екологічної обізнаності та залучення громадськості до вирішення екологічних проблем. Це може сприяти формуванню активної екологічної спільноти, організації прибирань, а також збору цінної інформації про стан довкілля в Україні.

### **1.3 Постановка задачі**

Еко-платформа “Чиста Україна” – це веб-застосунок, де користувач може опублікувати нотатку, тобто інформацію про якесь забруднене місце України, а інший користувач зможе на неї відреагувати, тобто прибрати. За кожне прибирання користувач отримує бали в залежності від рівня забрудненості. Процес прибирання розподіляється на три етапи: "в процесі", "на перевірці" та "завершено". На етапі перевірки власник нотатки повинен підтвердити прибирання території, і тільки після цього вона отримує статус "завершено".

Даний застосунок є зручний для користувачів, зокрема це дуже зручно для людей, які часто подорожують автомобілем Україною. Погодьтеся, дуже часто по дорозі трапляються брудні узбіччя або зони відпочинку з купою сміття. Не завжди у подорожуючих є час і можливість прибрати дану територію. Але тепер можна сміливо сфотографувати дане місце і опублікувати нотатку з геолокацією в еко-платформі. Це дуже зручно, адже люди, які знаходяться поряд, можуть побачити її і прибрати дану територію.

Також, користувач отримує бали за кожну виконану ним нотатку. Бали нараховуються тільки тоді, коли власник нотатки підтвердив її виконання. Ці бали характеризують досвід користувача у системі. І головна ідея цих балів полягала в тому, що, можливо, якщо б в майбутньому цей застосунок був би на державному рівні, або підтримувався провідними українськими компаніями, користувачі могли б обмінювати свої бали на певні бонуси. Наприклад, фішки на паливо від мережі заправок “ОККО”, квиток на сеанс від мережі кінотеатрів «Планета Кіно» та інше.

У даному веб-застосунку реалізований наступний функціонал:

- 1) Реєстрація користувача.

- 2) Зручний інтерфейс для додавання, редагування та видалення нотаток.
- 3) Можливість переглядати всі нотатки з пошуком по області та адресі.
- 4) Можливість переглядати тільки створені нотатки авторизованого користувача.
- 5) Зручний інтерфейс для перегляду нотаток, які виконані авторизованим користувачем.
- 6) Пошук адреси по Google Map з автозаповненням.
- 7) Можливість виконувати нотатку.
- 8) Інтерфейс сповіщень, які повідомляють які нотатки авторизованого користувача зараз на етапі перевірки і потребують його перегляду.
- 9) Можливість ставити вподобання на нотатку.
- 10) Можливість публікувати подію.
- 11) Пошук події або нотатки в радіусі 5, 10 і 20 км від вашого місця знаходження.
- 12) Функція голосування за участь в події.
- 13) Можливість запросити друга до події через розсилку повідомлення на електронну пошту.

Загалом веб-застосунок має бути оформлений у простому стилі, щоб бути зрозумілим для людей різного віку та не відволікати користувачів від пошуку забруднених територій та їх прибирання. Інтерфейс застосунку повинен бути оформлений українською мовою.

## 2. ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ОБРАНІ ТЕХНОЛОГІЇ

Правильно обрані технології – це ключ до успіху будь-якого програмного забезпечення. У розробці свого веб-застосунку я намагалась використовувати сучасні і прогресивні технології. Описані нижче засоби полегшують роботу над проектом, покращують його продуктивність та слідуєть найкращим практикам написання коду.

### 2.1 Клієнт-серверна архітектура

Для розробки веб-застосунку була обрана дворівнева клієнт-серверна архітектура. Дворівнева архітектура складається з таких компонентів:

- сервер, який відповідає за отримання запитів від клієнта та надсилання відповідей, використовуючи свої власні ресурси;
- клієнт, який відповідає за представлення користувацького інтерфейсу;

На стороні клієнта формується запит, який надсилається на сервер. Після успішного опрацювання запиту сервером, він формує відповідь та повертає її клієнту. Цей процес комунікації відбувається за допомогою протоколу передачі даних HTTP. Зазвичай, передача даних здійснюється через TCP/IP з'єднання.

### 2.2 Технології для серверної частини

Веб-застосунок розроблений на мові програмування Python з використання фреймворку FastAPI.

Python - це мова програмування високого рівня, інтерпретована та інтерактивна, яка була створена Гвідо Ван Россумом у 1991 році. На сьогоднішній день Python є найпопулярнішою багатофункціональною мовою програмування. Вона широко використовується такими гігантами технологічного світу, як Google, Amazon, Facebook, Instagram, Dropbox, Uber і багатьма іншими.

Python пропонує високорівневі вбудовані структури даних, а також має динамічний набір тексту та динамічне зв'язування. Синтаксис Python простий і легкий для освоєння, що підкреслює його читабельність і зменшує витрати на обслуговування програм. Python підтримує модулі та пакети, що сприяє

модульності програми та повторному використанню коду. Інтерпретатор Python та велика кількість доступних бібліотек надаються безкоштовно у вихідній або скомпільованій формі для всіх основних платформ і можуть бути вільно поширені.

FastAPI є відносно новим веб-фреймворком, який призначений для створення швидких HTTP API-серверів. Він поєднує в собі можливості двох інших інструментів - асинхронного веб-фреймворку Starlette і бібліотеки Pydantic для валідації даних та серіалізації. [5]

Основні особливості FastAPI такі:

- Висока швидкість роботи, що робить його одним з найшвидших фреймворків Python.
- Висока продуктивність завдяки автоматичному заповненню, яке допомагає зменшити зусилля та час на розробку і налагодження програм.
- Компактний код, який мінімізує дублювання.
- Повна сумісність з відкритими стандартами API, такими як OpenAPI (раніше відомий як Swagger) і JSON Schema.
- Підтримка асинхронного програмування та можливість працювати зі стандартами WSGI і ASGI.

В процесі розробки було вирішено використовувати FastAPI, адже саме цей фреймворк найкраще підійде для створення високопродуктивної та простої програми з найкращими практиками за замовчуванням.

Для запуску FastAPI в даному випадку використовується Uvicorn - швидкий ASGI-сервер, який забезпечує зручну конфігурацію і запуск сервісу. [5, 6]

Для зберігання зображень у хмарному середовищі використовується Firebase Storage. Цей сервіс надає надійну функціональність завантаження та вивантаження файлів для додатків. Google Cloud Storage використовується для зберігання файлів різних типів, включаючи відео, аудіо та інші. Всі дані, що зберігаються у хмарному сховищі, піддані високому рівню безпеки.

Один із ключових факторів, що вплинув на вибір Firebase Storage, - його легкість використання. Інтерфейс інтуїтивно зрозумілий, що дозволяє швидко

налаштувати та використовувати сервіс. Крім того, безкоштовна версія Firebase Storage надає до 10 ГБ пам'яті, що є значним обсягом порівняно з аналогічними рішеннями, наприклад, Heroku Storage.

### **2.3 Технології для клієнтської частини**

Клієнт розроблений на мові програмування JavaScript з використанням бібліотеки React.js.

JavaScript (JS) - це динамічна, об'єктно-орієнтована прототипна мова програмування, яка часто використовується для написання скриптів, що запускаються у браузері. JavaScript надає можливість зручно взаємодіяти з клієнтом на веб-сторінці, асинхронно обмінюватися даними з сервером, керувати браузером та змінювати зовнішній вигляд і структуру веб-сторінки. Ця мова була створена у 1995 році і з тих пір було розроблено багато практик використання, швидких бібліотек та фреймворків, а також розширилося використання JavaScript за межами браузерів. На сьогоднішній день JavaScript є однією з найпопулярніших мов програмування.

React.js - це відкрита JavaScript бібліотека, яка використовується для побудови користувацьких інтерфейсів динамічних веб-, мобільних та десктопних додатків. Вона має відкритий вихідний код і була створена компанією Facebook, яка активно розвиває її разом із спільнотою. Метою React.js є надання швидкості, простоти та масштабованості. Цю бібліотеку можна використовувати в будь-якому веб-сайті на JavaScript для повного створення інтерфейсу або для окремих його частин. Вона має великий набір функціональності та допоміжних пакетів, які можна легко встановити. Тому вибір використання React.js для клієнтської частини мого застосунку є очевидним. [7]

Також, для покращення продуктивності клієнтської сторони, використовується і Redux. Redux - це популярна бібліотека керування станом, яка широко використовується в React. Вона дозволяє ефективно управляти станом додатку, зберігаючи його в одному централізованому місці і забезпечуючи передачу стану між компонентами. [8]



Redux базується на концепції одностороннього потоку даних. Це означає, що стан додатку зберігається в одному об'єкті, який називається "store". Компоненти можуть взаємодіяти зі ним, відправляючи "actions" - об'єкти, що описують зміни стану. Далі, Redux використовує "reducers" - чисті функції, які обробляють ці actions і оновлюють стан додатку.

З React Redux, офіційною бібліотекою, яка інтегрує Redux з React, використання Redux стає ще зручнішим. React Redux надає "контейнери" - спеціальні компоненти, які забезпечують зв'язок між Redux store та компонентами React. Контейнери відповідають за підписку на зміни стану і передачу необхідних частин стану до компонентів через властивості.

Завдяки використанню Redux з React, стає простіше керувати складним станом додатку, спрощується передача даних та стану між компонентами, а також полегшується відлагодження і тестування коду. Використання Redux сприяє підтримці однозначності і передбачуваності стану, що робить розробку більш прозорою і масштабованою. [8]

Для аутентифікації використовується Auth0 — це безпечна й універсальна служба, яка забезпечує функціональність для аутентифікації та авторизації. Вона працює на основі токенів і використовує різних постачальників ідентифікаційних даних, таких як Facebook, Google, GitHub, LinkedIn. [9]

Головні переваги використання Auth0:

- Безпека даних користувачів під час аутентифікації є надзвичайно важливою, але часто становить складність для багатьох проектів. Однак, Auth0 вдалося успішно вирішити цю проблему, дозволяючи користувачам насолоджуватись високим рівнем безпеки під час процесу аутентифікації. Auth0 забезпечує безпеку за допомогою протоколу аутентифікації OAuth2.0, що дозволяє програмам отримувати доступ до ресурсів користувача без необхідності передавати їм логіни та паролі. Цей протокол також дозволяє надавати обмежені права доступу.
- Параметри інтерфейсу: Auth0 надає вбудований інтерфейс

користувача, а також можливість створити власний інтерфейс.

- Аналітика Auth0: Auth0 пропонує ефективні інструменти для відстеження користувачів на веб-сайті або в програмі. Завдяки інтеграції з Auth0 Analytics, можна відстежувати різні події, такі як кількість нових та існуючих користувачів, активність входу в програму, постачальники ідентифікаційних даних, які використовуються для входу та інші. Auth0 надає ці дані за допомогою простих графіків і пропонує функцію фільтрації звітів для отримання більш точної інформації.

Вся ця інформація про активність користувачів є надзвичайно цінною, і будь-яка компанія залюбки використовує її для збільшення прибутковості свого застосування. Таким чином, Auth0 не пропонує лише сучасну аутентифікацію як сервіс, але також надає інструменти для стратегічного планування масштабування проекту в майбутньому. На сьогоднішній день, Auth0 є однією з провідних платформ для реалізації аутентифікації в сучасних програмах. [9]

У клієнтській частині свого веб-застосунку використовується пакет Auth0.js— це клієнтська бібліотека для Auth0.

Google Maps є одним з найбільших географічних агрегаторів на сучасному ринку. Крім цього, Google надає широкий спектр класів, функцій, процедур і структур, доступних для використання у зовнішніх програмах. За допомогою Google API, який є частиною Google Cloud, можна взаємодіяти з базою даних та отримувати детальні відповіді на запити. [10]

Для роботи з локацією застосунок інтегрується з Google Maps Platform. Згідно з відомостями компанії, Google Maps Platform містить дані про практично 99% географічних областей світу. Крім того, вони активно обробляють щодня близько 25 мільйонів оновлень існуючої інформації, які надходять з різноманітних джерел, налічуючи їх понад 1000. Використовуються 3 API з Google Maps Platform, а саме:

- 1) Maps JavaScript API. Даний продукт надає можливість по запиту отримувати візуальну картографічну інформацію. Цей інтерфейс

дозволяє користувачам додавати маркери, лінії, кольори та зображення до наявних карт або створювати власні карти. Крім того, користувачам надається можливість масштабування, стискання, обертання та нахилу карт для більш детального вивчення.

- 2) Places API. Це продукт, який дозволяє користувачам отримувати дані про місцезнаходження та адреси. Places API надає функціональність для пошуку місць, що містяться в певній області – в межах карти, для прикладу. Також підтримує функцію автозаповнення, що спрощує процес пошуку.
- 3) Geocoding API. Це інструмент, який виконує перетворення адреси на географічні координати та навпаки. Цей продукт забезпечує простий спосіб отримання доступу до геокодування та реверсного геокодування через HTTP-запити.

Для доступу до цих API було створено акаунт в Google Cloud Platform, та отримано безоплатний секретний ключ доступу. Для даних і функцій з Google Maps Platform в середовищі React використовуються пакети "@react-google-maps/api" та "use-places-autocomplete". [10]

## **2.4 База даних та технології для роботи з нею**

Для зберігання даних обрана база даних PostgreSQL.

PostgreSQL є однією з найбільш професійних відкритих реляційних баз даних. Вона є надійною, стабільною та масштабованою системою, яка існує протягом тривалого періоду понад три десятиліття. PostgreSQL має високу репутацію завдяки своїй перевірній архітектурі, надійності, забезпеченню цілісності даних, широкому спектру функцій, постійному розвитку та можливостям розширення. Ця система відома своєю високою безпекою та можливістю задовольняти потреби навіть найбільш вимогливих проєктів.

Для взаємодії з базами даних у середовищі Python використовується бібліотека SQLAlchemy. Ця бібліотека надає набір інструментів для роботи з SQL і включає в себе ORM (об'єктно-реляційне відображення), що дозволяє розробникам

маніпулювати базою даних за допомогою Python, не прямо використовуючи SQL-запити. SQLAlchemy дозволяє описувати структуру бази даних та встановлювати способи взаємодії з нею з використанням зручного і гнучкого синтаксису Python.

Також для ефективного керування версіями бази даних використовується Alembic - це інструмент для міграції бази даних, що використовується разом з SQLAlchemy. Міграція бази даних дозволяє керувати інкрементними змінами схеми реляційної бази даних.

У процесі розробки додатків поширені випадки, коли необхідно змінювати структуру таблиць. Тут на допомогу приходить Alembic. Він дозволяє зручно здійснювати зміни схеми бази даних під час розвитку додатків. Крім того, Alembic відстежує зміни в самій базі даних, що дозволяє безпроблемно переходити між різними версіями. Якщо не використовувати Alembic, то всі зміни доведеться відстежувати вручну. [11]

### 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

#### 3.1 Структура бази даних

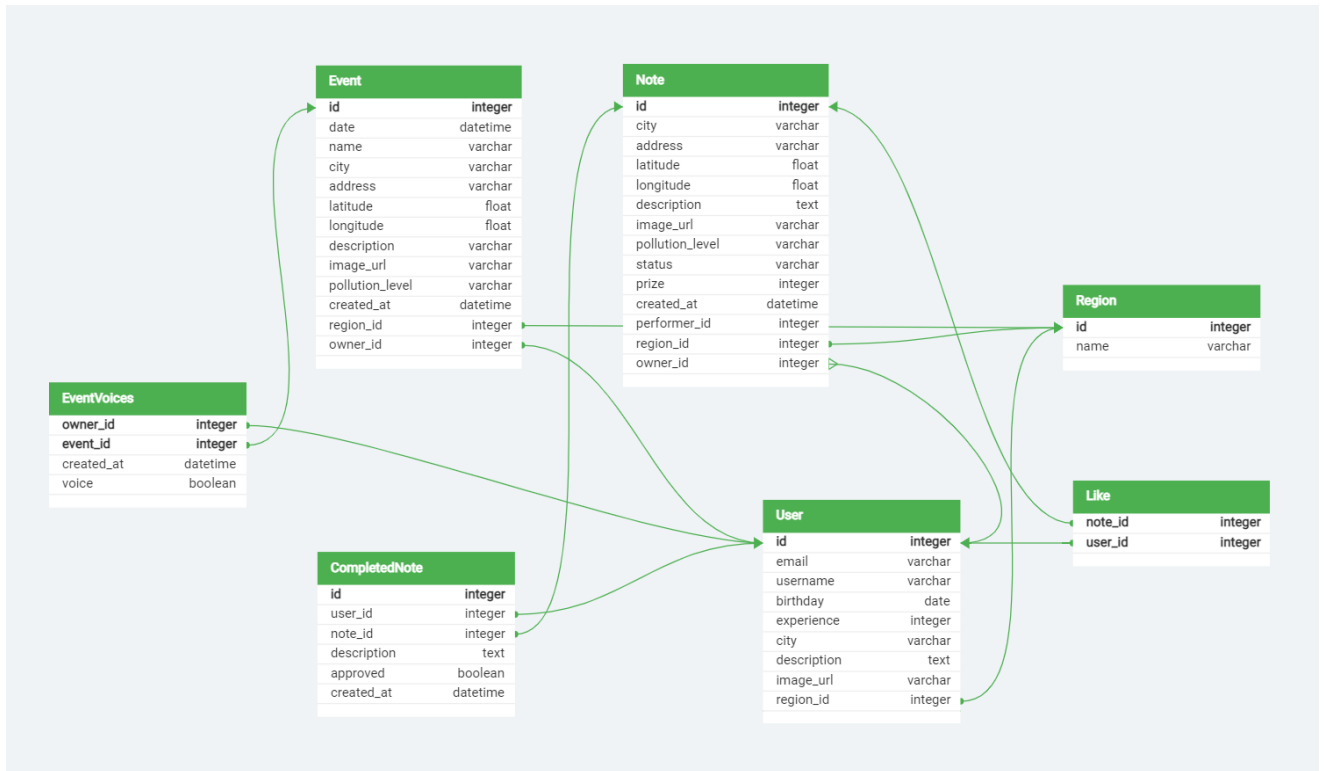


Рисунок 3.1.1 – Схема бази даних

У базі даних є 7 таблиць: Користувач, Нотатка, Виконана нотатка, Регіон, Вподобання, Подія, Голоси за подію. Переважаючі зв'язки: один до багатьох, один до одного.

#### 3.2 Структура проекту серверної частини.

Отже, коренева папка складається з:

- 1) alembic – це папка для роботи з бібліотекою Alembic. Вона містить в собі папку versions – власне папка для зберігання міграцій бази даних. Також там є конфігураційний файл для з'єднання і налаштування бази даних (env.py) та файл з шаблоном для майбутніх міграцій (script.py.mako).
- 2) app – папка для роботи з фреймворком FastAPI. В ній і зосереджена головна логіка проекту. Складається з:
  - a) routers – папка, яка зберігає всі маршрутизатори веб-застосунку.
  - b) services – папка, яка зберігає функціональну частину проекту.

Вона поділена на підпапки, в кожній з яких реалізований базовий клас для наслідування та власне клас з реалізацією методів, що імплементують функціонал певної сутності. Реалізовані сервіси для користувачів, нотаток, використаних нотаток, регіонів, подій та завантаження фотографій

- c) `utils` – папка, що містить допоміжні корисні функції – верифікація JWT токена, генерування рандомного ключа для фотографії та інше.
  - d) `templates` – папка, що містить html-шаблон для повідомлення на електронну пошту.
  - e) Файл для конфігурації (`config.py`), файл моделей бази даних (`models.py`), файл для сталих значень і констант (`constants.py`), файл для винятків програми (`exceptions.py`) та інші.
- 3) `test_data` – це папка, яка містить тестові дані для проекту у JSON-форматі.
- 4) `venv` – віртуальне середовище Python 3.9.
- 5) `main.py` – файл для запуску FastAPI з використання Uvicorn.
- 6) `requirements.txt` - файл залежностей програми, для автоматичного встановлення пакетів Python за допомогою утиліти `pip`.
- 7) Допоміжні файли для налаштування github репозиторію, аналізатору синтаксису коду, секретний файл для налаштування Firebase Storage, файл для заповнення бази даних тестовою інформацією.

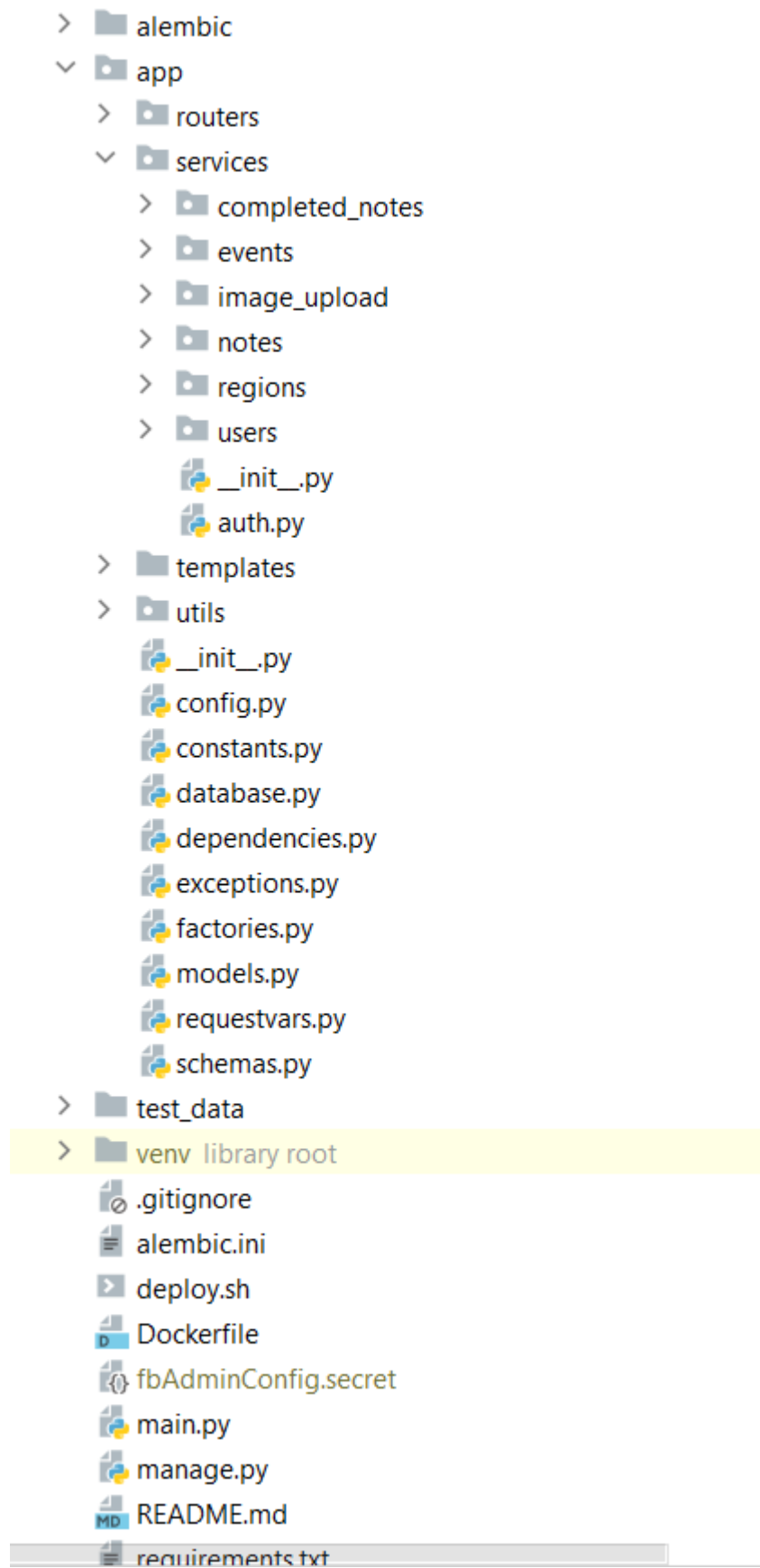


Рисунок 3.2.1 – Структура серверної частини

### 3.3 Структура проекту клієнтської частини

Отже, коренева папка складається з:

- 1) `node_modules` – містить всі зовнішні бібліотеки JavaScript, використовувани додатком.
- 2) `public` - містить базові файли HTML, JSON та зображення.
- 3) `src` - міститься головний код React JavaScript для проекту.

Складається з:

- a) `components` – папка з певними компонентами веб-сторінки.
  - b) `images` – папка для фотографій.
  - c) `layouts` - папка, файли якої відповідають за навігацію та головне меню проекту.
  - d) `pages` – папка з головними сторінками проекту. Містить JavaScript файл для кожної сторінки і відповідний йому CSS файл для підключення каскадної таблиці стилів.
  - e) `redux` – папка, що використовується для організації логіки керування станом додатка з використанням бібліотеки Redux. В цій папці знаходяться файли, які визначають основні елементи архітектури Redux - `store` і `reducer`.
  - f) `routes` – папка, що містить файл з усіма визначеними маршрутами проекту, файл для перевірки дозволів користувачу переглядати сторінку.
  - g) `styles` – папка з CSS файлами, що містять спільні налаштування стилів.
  - h) `toolkits` – папка з файлами певних загальних інструментів UI проекту –кнопки, сповіщення, карта та інше.
  - i) `utils` – папка, що містить допоміжні корисні функції.
  - j) `index.js` – файл, що виконує розгортання React-застосунку .
- 4) Додаткові файли для налаштування GitHub репозиторію, аналізатори коду, файли про додаткові метадані проекту та інсталювані пакети.



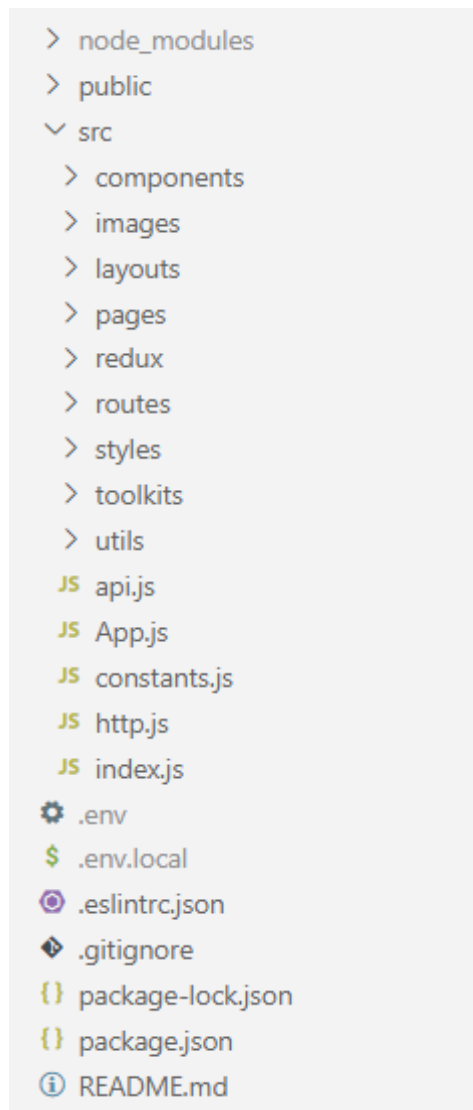


Рисунок 3.3.1 – Структура клієнтської частини

### 3.4 Аналізатори синтаксису коду

Для дотримання найкращих практик, одного стилю і стандарту написання коду використовуються аналізатори синтаксису коду – Linter. Linter (або lint) — це інструменти для дослідження програмного коду з метою виявлення помилок програмування, вад, порушень стилю, а також сумнівних чи підозрілих виразів.

Для серверної частини це аналізатор Flake8. Flake8 – це популярний linter для Python. Правила перевірки коду задаються у корневих файлах setup.cfg або .flake8

Вміст setup.cfg файлу для моєї серверної частини:

[flake8]

```
exclude = .git, env/, .env/, venv/, .venv/, */migrations/*, */versions/*,
*/settings/*, settings.py
ignore = E123,E128,E402,W503,E731,W601,E999,Q003
max-line-length = 120 inline-quotes = single
```

Для клієнтської частини використовується аналізатор ESLint. ESLint вбудований у більшість текстових редакторів, але також можна написати власні правила, які працюють разом із вбудованими правилами ESLint. Правила для аналізу і перевірки коду пишуть у файлі `.eslintrc.json`.

Вміст `.eslintrc.json` файлу для моєї клієнтської частини:

```
{
  "env": { "browser":
    true, "es2021": true
  },
  "extends": [ "eslint:recommended",
    "plugin:react/recommended"
  ],
  "parserOptions": {
    "ecmaFeatures": {
      "jsx": true
    },
    "sourceType": "module"
  },
  "plugins": [
    "react"
  ],
  "rules": {
    "indent": [
      "error",
      4
    ],
    "linebreak-style": 0,
    "quotes": [
      "error",
      "double"
    ],
    "semi": [
      "error",
      "never"
    ],
    "require-jsdoc": 0, "react/prop-
    types": "off"
  }
}
```

#### 4. РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розгортання програмного забезпечення (Software Deployment) включає в себе процес розміщення програмного коду та всіх його залежностей на цільовій інфраструктурі, щоб програма або додаток могли бути виконані та доступні для користувачів. Цей процес може включати такі кроки, як підготовка середовища, налаштування серверів, установка необхідного програмного забезпечення, конфігурування параметрів додатка та перевірка правильності розгортання.

AWS (Amazon Web Services) є одним з провідних провайдерів хмарних обчислень, що надає широкий спектр послуг для розгортання та управління веб-додатками та інфраструктурою. AWS пропонує веб-хмарні сервіси, такі як обчислення, зберігання даних, бази даних, мережеві рішення, інструменти розробки та багато іншого. [15]

AWS має декілька переваг, які роблять його привабливим для використання:

- Гнучкість і масштабованість: AWS надає широкий спектр послуг та ресурсів, які дозволяють масштабувати інфраструктуру залежно від змінних потреб. Можна змінювати розмір серверів, налаштовувати потужність обчислювальних ресурсів та гнучко керувати інфраструктурою, щоб відповідати зростаючим вимогам бізнесу.
- Широкий спектр послуг: AWS надає велику кількість хмарних послуг, що охоплюють обчислення, зберігання даних, бази даних, мережі, аналітику, машинне навчання, ШІ та багато іншого. Це дозволяє використовувати одного провайдера для багатьох потреб інфраструктури, що спрощує управління та знижує складність.
- Надійність і доступність: AWS має розподілену мережу центрів обробки даних, що забезпечує високу доступність і надійність додатків. AWS також пропонує послуги резервного копіювання, механізми реплікації даних та стабільність, які допомагають забезпечити безперебійну роботу систем.
- Безпека: AWS приділяє велику увагу безпеці своїх послуг і надає набір інструментів та служб для захисту даних та додатків. Можна

використовувати ідентифікацію та управління доступом, шифрування інші інструменти для забезпечення високого рівня безпеки. [15, 16]

#### **4.1 Розгортання клієнтської частини**

Для розгортання клієнта було обрано AWS S3. AWS Amazon S3 (Simple Storage Service) - це високомасштабована послуга зберігання об'єктів у хмарі, яка пропонує безпечно та масштабоване зберігання даних в Інтернеті. AWS S3 є одним з найпопулярніших сервісів хмарного сховища та широко використовується розробниками, підприємствами та організаціями для зберігання, резервного копіювання та обробки даних. [17]

Першим кроком розгортання є створення нового бакету (bucket — це сутність для організації зберігання в сховищі) в консолі керування AWS, який буде використовуватись для зберігання React-додатку, та налаштування прав доступу до нього та веб-сайту S3.

Далі потрібно зберегти React-додаток у локальному середовищі. Тобто, це має бути зібраний варіант застосунку, який включає всі необхідні статичні файли.

І останній етап - власне розгортання - перенесення файлів в бакет S3. Для цього використовується інструмент командного рядка AWS (наприклад, AWS CLI) або консоль S3, щоб скопіювати файли додатку в створений бакет S3.

#### **4.2 Розгортання серверної частини**

Серверна частина також розгорталась на AWS з використанням Docker. Docker є відкритим проектом, який надає можливість упакувати, доставляти та запускати програмне забезпечення у контейнерах. Контейнер - це виконавче середовище, що містить усі необхідні компоненти для виконання програми, включаючи код, залежності, бібліотеки та конфігураційні файли. [13]

Основні компоненти Docker включають:

- Docker Engine: Це рушій Docker, який виконує контейнери та надає основні функції Docker, такі як управління контейнерами, мережами та зберіганням.

- Docker Image: Це стандартизований пакет, який містить всі компоненти, необхідні для виконання програми, включаючи код, залежності, бібліотеки, конфігураційні файли тощо. Використовуючи Docker Image, можна легко розгорнути та розповсюджувати свою програму на різних середовищах.
- Docker Container: Це ізольоване виконавче середовище, яке базується на Docker Image. Контейнери запускаються з Docker Image і мають власне файлове середовище, мережеві налаштування та ізольований процес. Вони дозволяють запускати додатки у відокремленому та безпечному середовищі, що полегшує розгортання та управління додатками.
- Docker Hub: Це публічний реєстр Docker Image, де ви можете знайти готові Docker Images, створені іншими користувачами. Ви також можете завантажувати свої Docker Images на Docker Hub та ділитись ними з іншими користувачами. [13, 14]

Окрім Docker, у розгортанні використовувалось також багато AWS компонентів, а саме:

- Elastic Compute Cloud (EC2) - це веб-сервіс, що надає вам можливість запускати віртуальні сервери в хмарі AWS.
- Amazon Elastic Container Registry (ECR) - це керований сервіс реєстрації контейнерів, що надає приватний реєстр для зберігання, керування та розгортання Docker-образів.
- Amazon Elastic Container Service (ECS) - це керований сервіс контейнерів, що дозволяє запускати та керувати контейнерами Docker на AWS.
- Load Balancers (балансувальники навантаження) - використовуються для розподілу трафіку між різними екземплярами EC2 або іншими ресурсами, що обробляють запити.
- Security Groups - це віртуальні механізми, які дозволяють налаштовувати правила доступу до ресурсів. Вони визначають, які IP-

адреси або діапазони IP можуть отримати доступ до ресурсів, які порти відкриті для з'єднань, а також які протоколи дозволені або заборонені.

- Target Groups - це групи екземплярів або контейнерів, до яких можна направляти трафік за допомогою балансувальника навантаження.
- Relational Database Service (RDS) - це само-керований сервіс баз даних в AWS. Він надає можливість швидко розгортати, масштабувати та керувати реляційними базами даних. RDS надає високу доступність, резервне копіювання та моніторинг. [15]

Отже, розгортання проекту на AWS з використанням Docker включає такі кроки:

- a) Створення Dockerfile - це текстовий файл, що містить інструкції для побудови Docker-образу.
- b) Будівництво Docker-образу - Docker виконає інструкції з Dockerfile та створить образ з усіма необхідними компонентами вашого проекту.
- c) Реєстрація Docker-образу в Amazon ECR.
- d) Налаштування RDS для зберігання та керування PostgreSQL.
- e) Налаштування AWS Elastic Container Service (ECS). Це включає в себе створення ECS кластеру, задання конфігурацій для Task Definition (включаючи вказівку Docker-образу з ECR), налаштування балансувальника навантаження та правил доступу та масштабування.
- f) Розгортання проекту на ECS - запуск проекту на кластері. ECS буде автоматично запускати контейнери з використанням налаштувань Task Definition і балансувати навантаження між ними.

Така архітектура дозволяє створити розподілену, масштабовану та безпечну інфраструктуру, що готова до обробки великого обсягу трафіку та забезпечення стійкості до відмов.

## 5. ФУНКЦІОНАЛЬНІСТЬ ВЕБ-ЗАСТОСУНКУ

### 5.1 Логін/Реєстрація

Сторінка для входу повідомляє користувача про неправильно введений пароль або електронну пошту. Для цього використовується Alert (сповіщення) з бібліотеки @material-ui/lab.

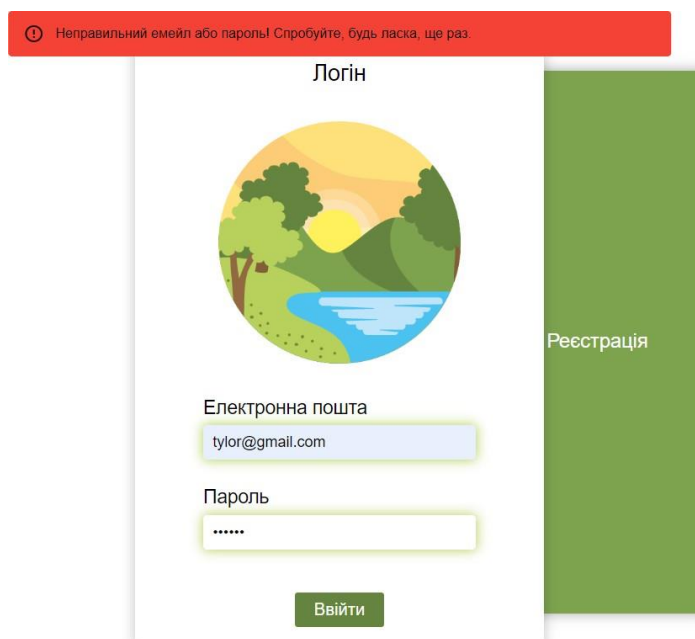


Рисунок 5.1.1 – Сторінка для входу в систему з сповіщенням про неправильно введений пароль або електронну пошту

Сторінка реєстрації також містить валідацію даних, а саме:

- Обов'язкові поля для вводу електронної пошти, ім'я користувача, дати народження, пароль та підтвердження паролю.
- Дата народження не повинна бути пізніше сьогоднішньої дати.
- Паролі повинні збігатись.
- Пароль повинен бути не менше 8 символів і містити букви нижнього регістру (a-z), букви верхнього регістру(A-Z), цифри (0-9), спеціальні символи (!@#\$\$%^&\*').

Регістрація

Ім'я користувача

Ім'я користувача повинне бути заповненим!

Електронна пошта

Електронна пошта повинна бути заповнена!

Дата народження

Дата народження повинна бути заповнена!

Область

Автономна Республіка Крим

Місто

місто

Опис профілю

опис

Пароль

.....

Підтвердження паролю

....

Паролі не збігаються!

Зареєструватись

Рисунок 5.1.2 – Сторінка для реєстрації з усіма правилами валідації  
 Після успішного входу в систему користувача перенаправляє на сторінку з усіма нотатками, а для навігації доступне меню.

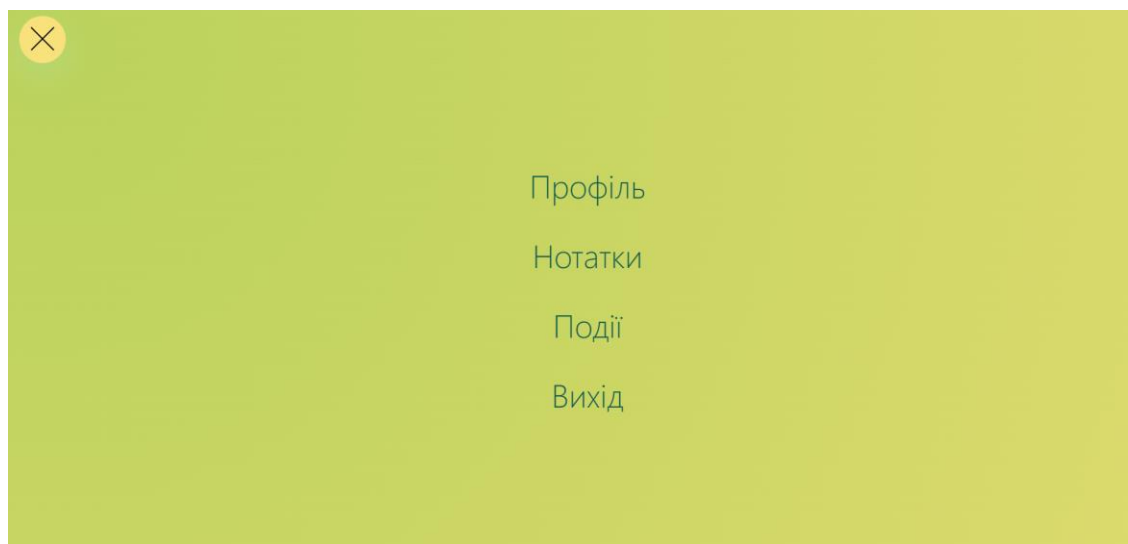


Рисунок 5.1.3 – Головне меню

## 5.2 Нотатки

Дана сторінка містить всі нотатки, які є посортовані по даті додавання. Також реалізований пошук нотаток по області та адресі. Фотографія збільшу свій розмір при наведенні на неї курсором, для зручності користувачу.



Однією з головних функцій цієї сторінки є пошук нотаток в околі 5, 10 і 20 км. Для пошуку нотаток в околі заданого радіусу використовується принцип географічного обчислення відстані між двома точками на поверхні Землі. Для цього використовується бібліотека `geopy.distance`. [19]

Принцип, за яким працює `geopy.distance`, базується на формулі гаверсінуса, яка дозволяє обчислити відстань між двома точками на сфері, використовуючи їх географічні координати (широта і довгота). Формула гаверсінуса враховує кривизну Землі і дає більш точний результат, ніж просте обчислення евклідової відстані. [18]

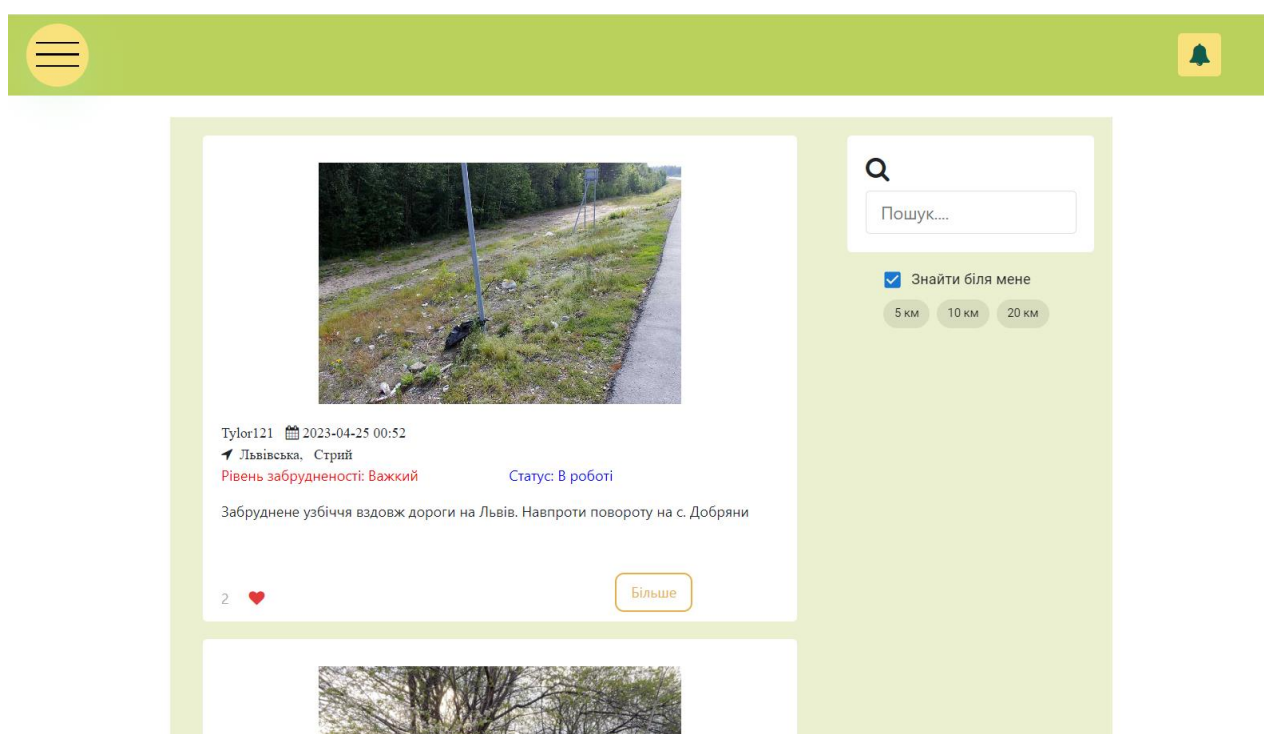


Рисунок 5.2.1 – Сторінка зі списком всіх нотаток

Також на даній сторінці реалізований функціонал Вподобань. Тобто можна або поставити вподобання на нотатку, або забрати його.

З даної сторінки можна перейти на сторінку з детальною інформацією про певну нотатку.

## 5.3 Профіль

На сторінці профілю користувача можна переглянути деталі профілю, нотатки створені користувачем, та нотатки виконані користувачем.

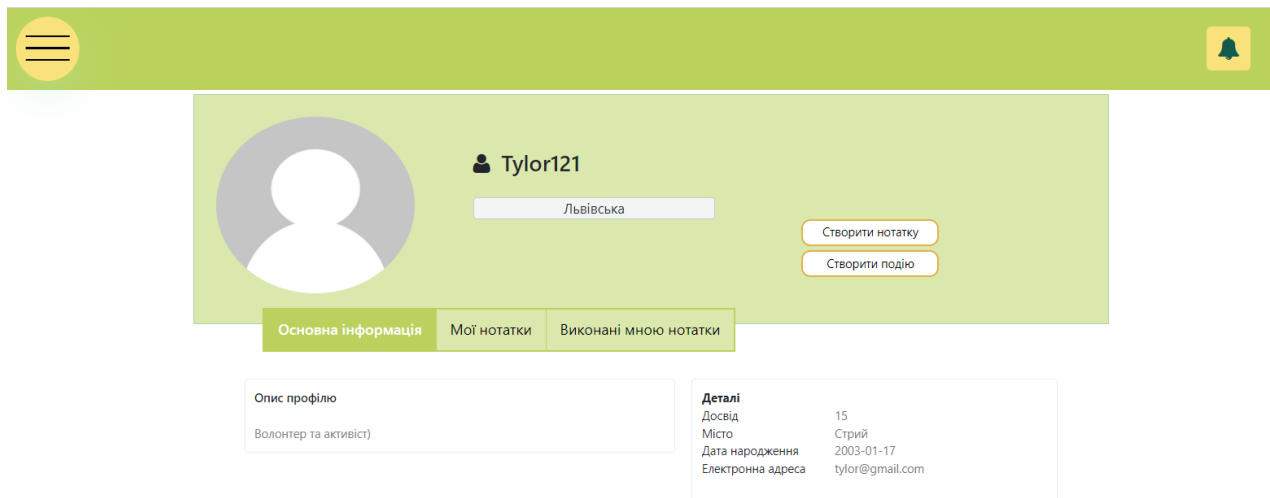


Рисунок 5.3.1 – Сторінка для перегляду деталей профілю

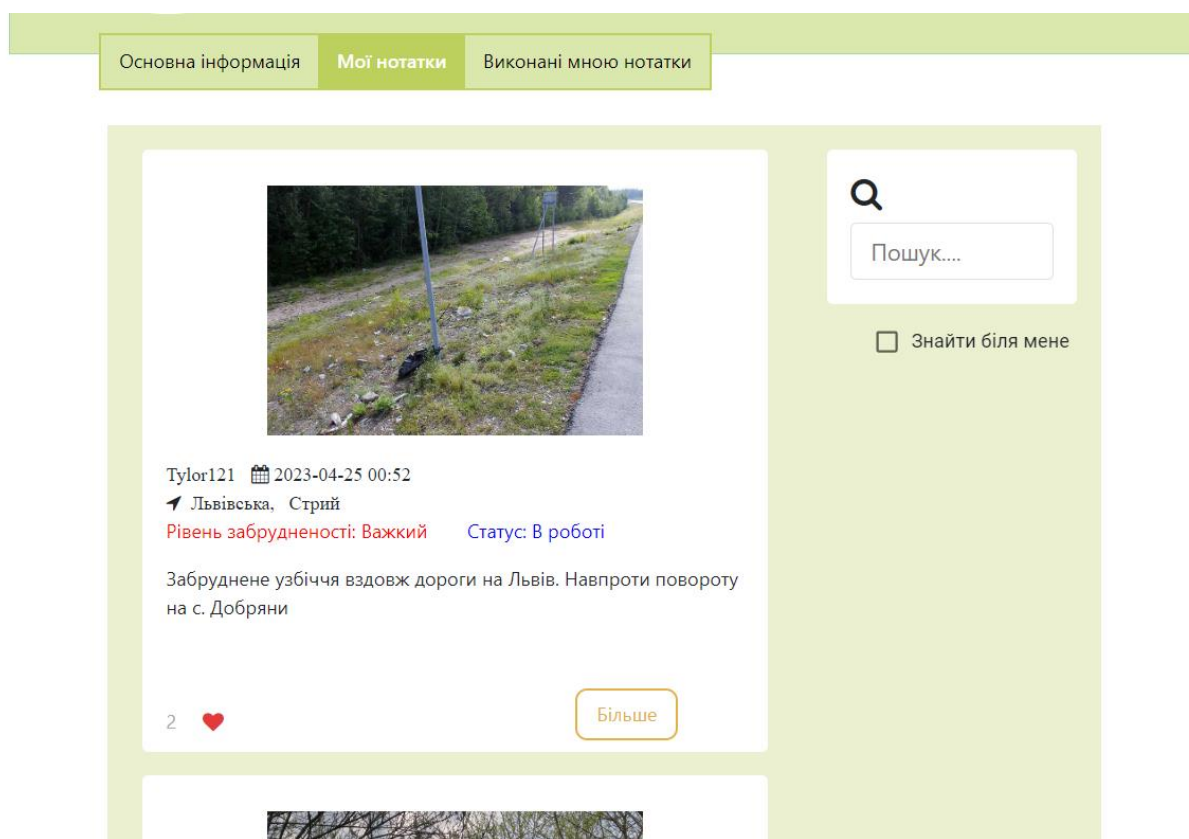


Рисунок 5.3.2 – Сторінка для перегляду нотаток користувача

Нотатки, виконані користувачем, зображені у вигляді таблиці, що є зручно для бачення загальної картини і поточного стану нотаток. Кожен з рядочків таблиці є клікабельний, тобто можна перейти на сторінку детальної інформації певної нотатки. Таблиця містить ті нотатки, в яких користувач безпосередньо бере участь, тобто ті, які на етапі виконання або на перевірці.

Область	Дата створення	Рівень забрудненості	Статус
Івано-Франківська область	2022-05-23 01:30	Середній	Виконано
Львівська область	2022-05-22 23:15	Високий	В процесі

Рисунок 5.3.3 – Таблиця для перегляду виконаних користувачем нотаток

#### 5.4 Додавання/редагування нотатки

### Додати інформацію

+

Місто

Адреса

Опис

Рівень забрудненості

Легкий

Область

Київська

Пошук по локації

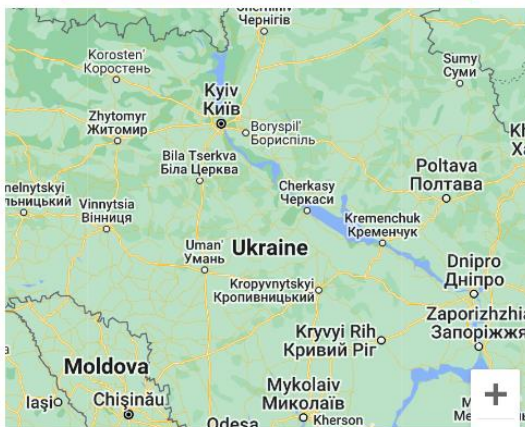


Рисунок 5.4.1 – Форма для створення та редагування нотатки

При додаванні або редагуванні нотатки користувач може додати до нотатки геолокацію. Для цього реалізований зручний пошук з автозаповненням (пропонує 5 варіантів) по будь-якій точці на карті - Google Maps Platform зібрали дані про 99% куточків світу.

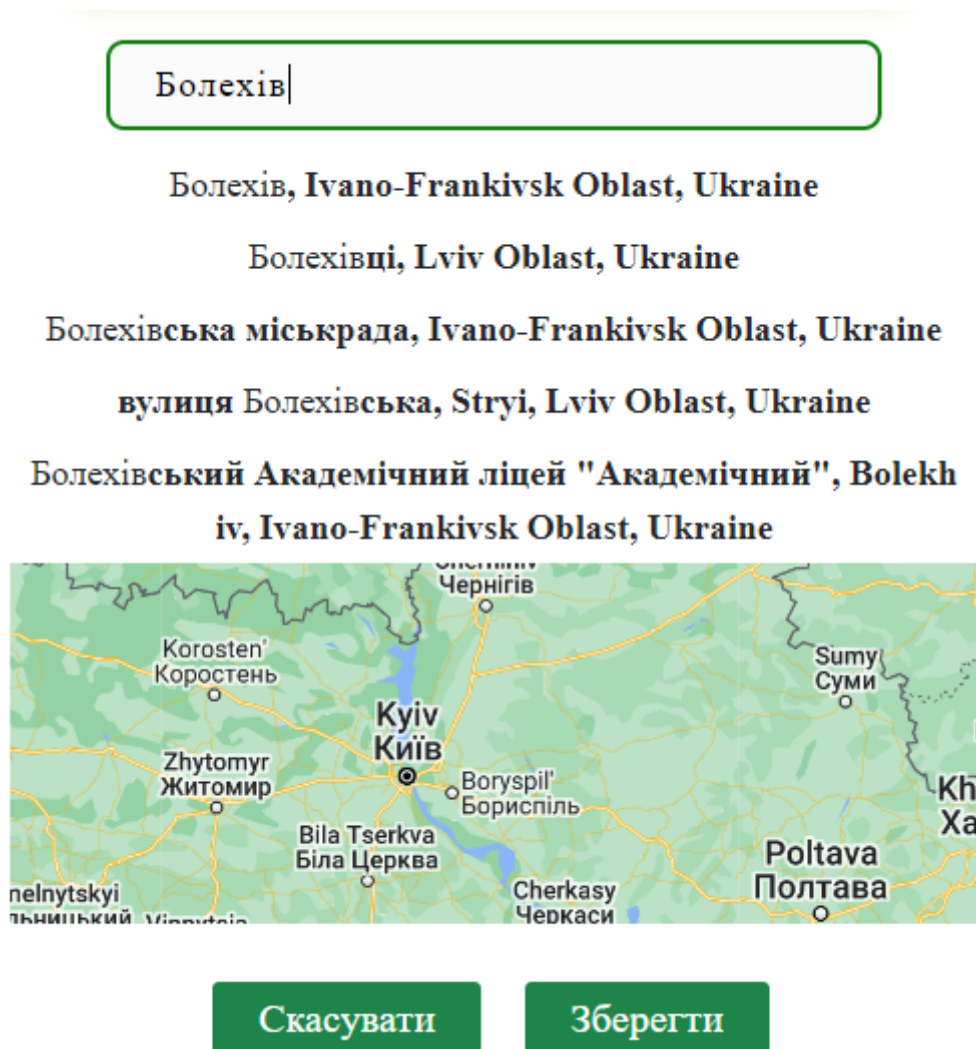


Рисунок 5.4.2 – Демонстрація пошуку по карті

## 5.5 Нотатка

На даній сторінці можна переглянути всю детальну інформацію по даній нотатці. Є можливість уподобати нотатку. Для власника нотатки є можливість відредагувати або видалити її.

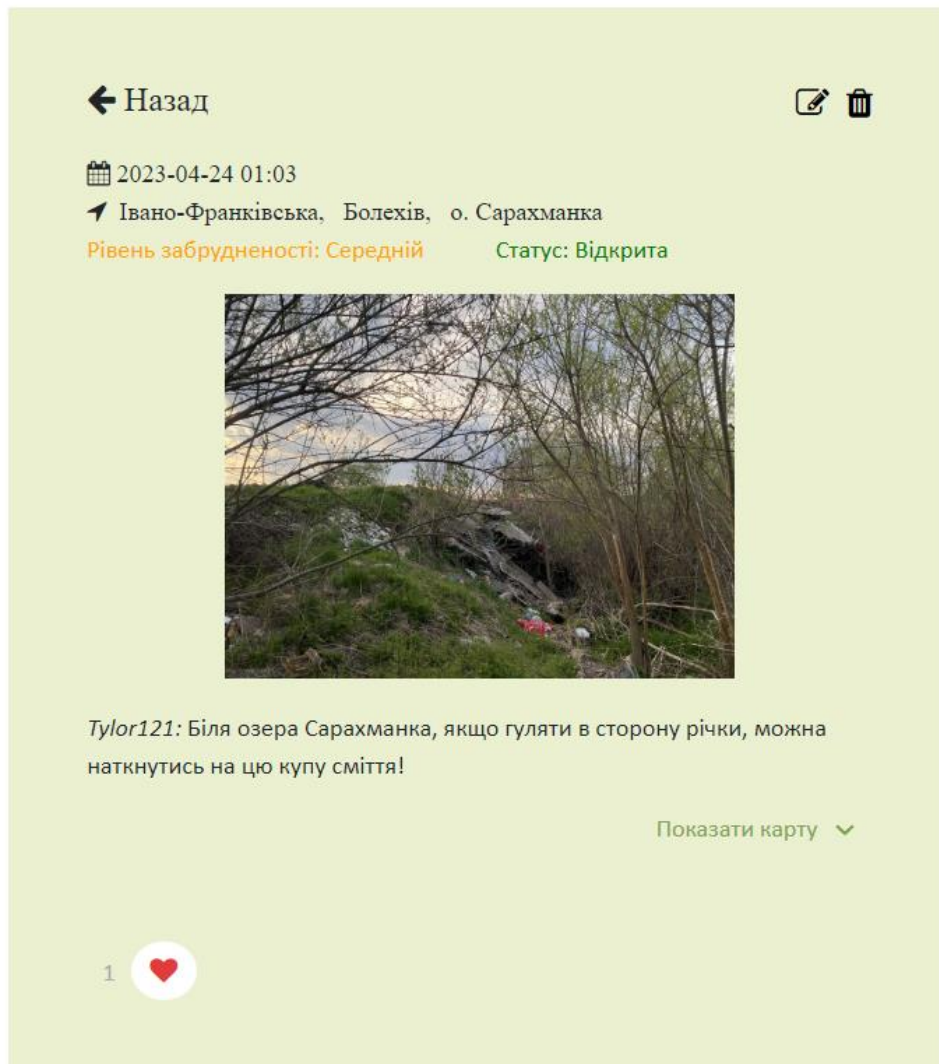


Рисунок 5.5.1 – Детальна інформація про нотатку, яка належить авторизованому користувачу


Якщо нотатка не належить авторизованому користувачу, то він не зможе її видалити чи відредагувати. Проте є можливість виконати дану нотатку.

← Назад

📅 2023-04-24 01:03

📍 Івано-Франківська, Болеків, о. Сарахманка


Рівень забрудненості: Середній      Статус: Відкрита




Tylor121: Біля озера Сарахманка, якщо гуляти в сторону річки, можна наткнутись на цю купу сміття!

Приховати карту ^

Пошук по локації



1 

Призначити на себе

Рисунок 5.5.2 – Детальна інформація про нотатку, яка не належить авторизованому користувачу

## 5.6 Процес виконання нотатки

Процес прибирання проходить в 3 етапи:

1. Користувач призначає нотатку собі.
2. Користувач надсилає результат прибирання – фотографію та короткий опис.
3. Власник нотатки підтверджує або не підтверджує прибирання.

Тільки після підтвердження прибирання, користувачу зараховуються балив залежності від рівня забрудненості:

- Легкий – 10 балів;
- Середній – 15 балів;
- Важкий – 20 балів;

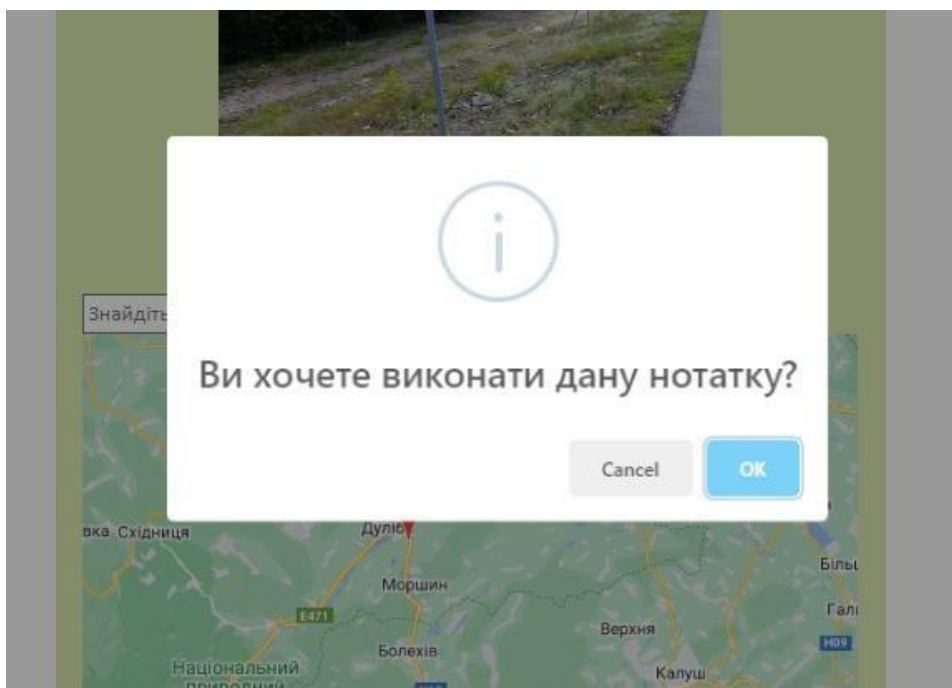


Рисунок 5.6.1 – Підтвердження призначення нотатки

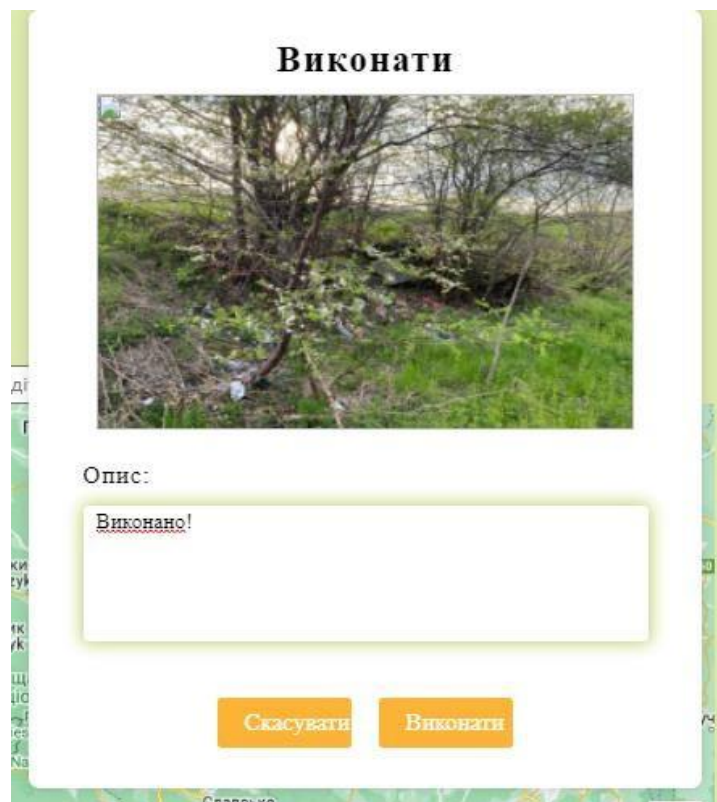


Рисунок 5.6.2 – Форма для результатів виконання нотатки

## 5.7 Сповіщення

Користувач завжди має доступ до сповіщень у “шапці” сайту. Саме у вікні сповіщень користувач дізнається, яка з його нотаток зараз на етапі перевірки.

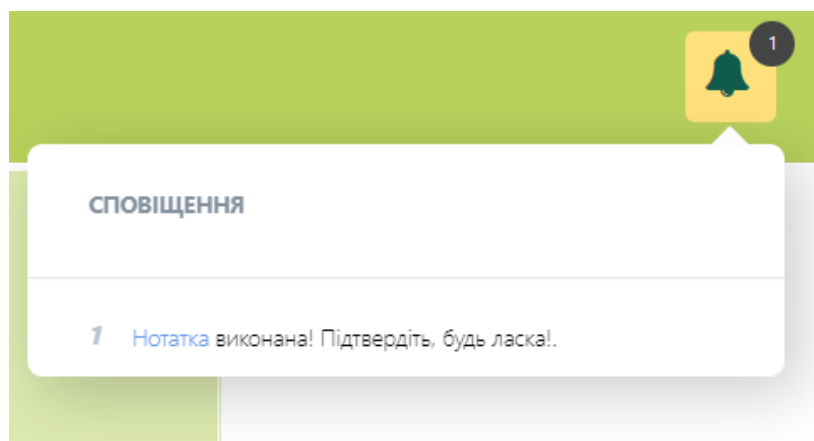


Рисунок 5.7.1 – Вікно сповіщень



## 5.8 Події

Дана сторінка містить події, які є посортовані по даті проведення, але події, які відбувались в минулому не відображаються. Також реалізований пошук подій по області, адресі та в околі 5, 10 і 20 км від розташування користувача. Фотографія збільшує свій розмір при наведенні на неї курсором, для зручності користувачу.

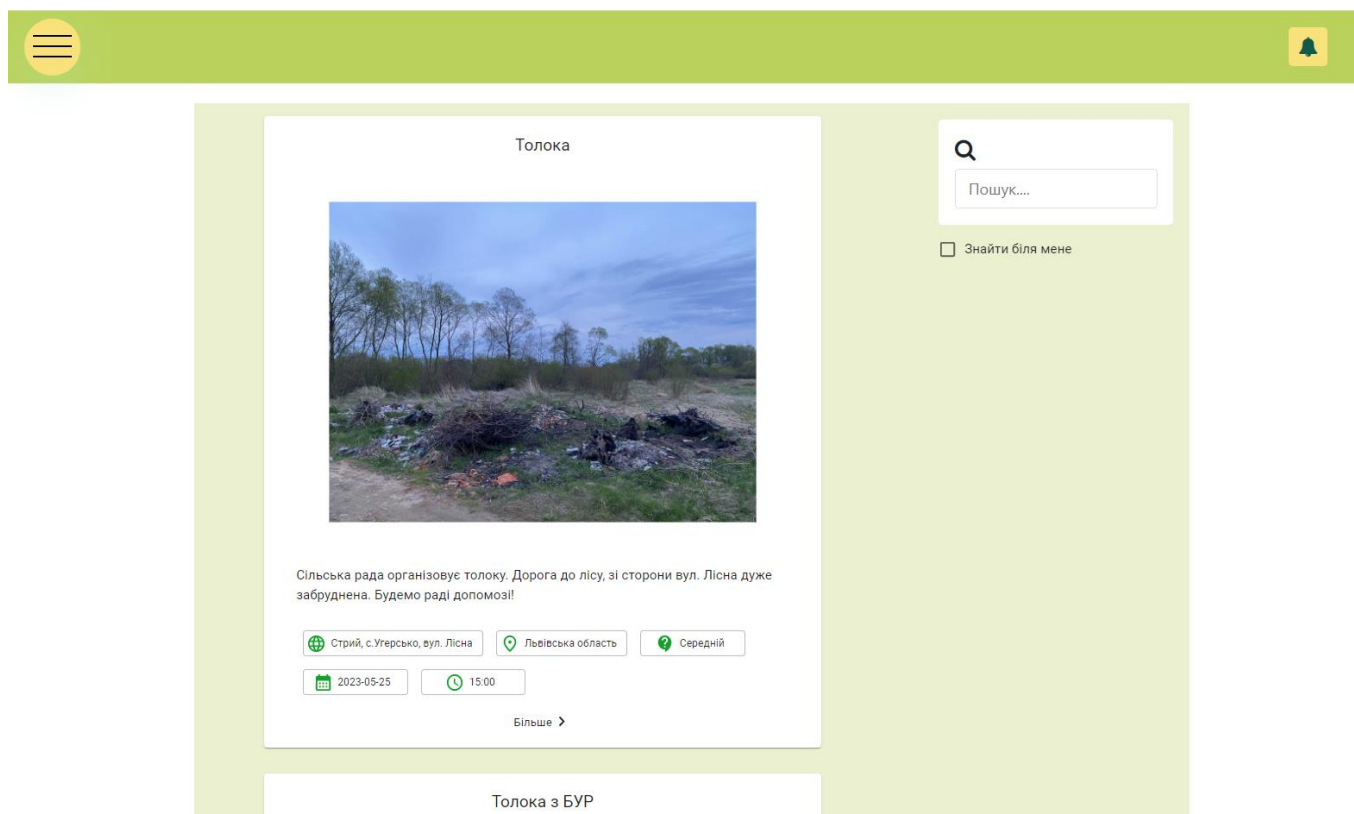


Рисунок 5.8.1 – Сторінка зі списком всіх подій

З даної сторінки можна перейти на сторінку з детальною інформацією про певну подію і також доступний функціонал вподобань.

## 5.9 Детальна інформації про подію та голосування

На даній сторінці є можливість проголосувати за участь в події. Ця статистика є дуже важливою, як і для організаторів події, так і для потенційних учасників.

[Запросити друга](#)

## Толока з БУР

Автор: Jada


Опис: 27 травня відбудеться всеукраїнська толока з БУР у Болехові!

Запрошуємо всіх охочих доєднатися до масштабної акції прибирання, адже участь кожного - це внесок у краще майбутнє та сьогодення нашої батьківщини.

ГО "Будуємо Україну Разом" уже третій рік поспіль організує всеукраїнську толоку, приурочену до Європейського дня парків. Цьогоріч акція відбудеться 27 травня у 63 регіонах(містах), з 10:00 до 14:00.

Організатори забезпечують волонтерів усім необхідним: сміттєвими пакетами, рукавицями, аптечкою та антисептиками.

ТИ + БУР = Чиста Україна  
ТИ + БУР = Чистий Болехів



Болехів, Прибережна територія річки Сукіль

Івано-Франківська область
 Середній

2023-05-27
 10:00

Показати карту ▾

### Чи плануєте Ви взяти участь в даній події?

50%
Так

50%
Ні ✓

2 votes

Рисунок 5.9.1 – Детальний перегляд події

## 5.10 Можливість запросити друга

На сторінці для детального перегляду події є можливість запросити друга до участі. Цей функціонал включає в себе надсилання повідомлення на електронну пошту з інформацією про дану подію.

Розсилка електронних листів відбувається через сервер smtp.gmail.com, який надає послугу SMTP (Simple Mail Transfer Protocol) для відправлення електронних листів через Gmail. Використання цього сервера має такі переваги, як надійність, безпека, зручність та відмінна функціональність.

Запросити друга до події

Електронна адреса

Скасувати    Надіслати

Рисунок 5.10.1 – Діалогове вікно для запрошення

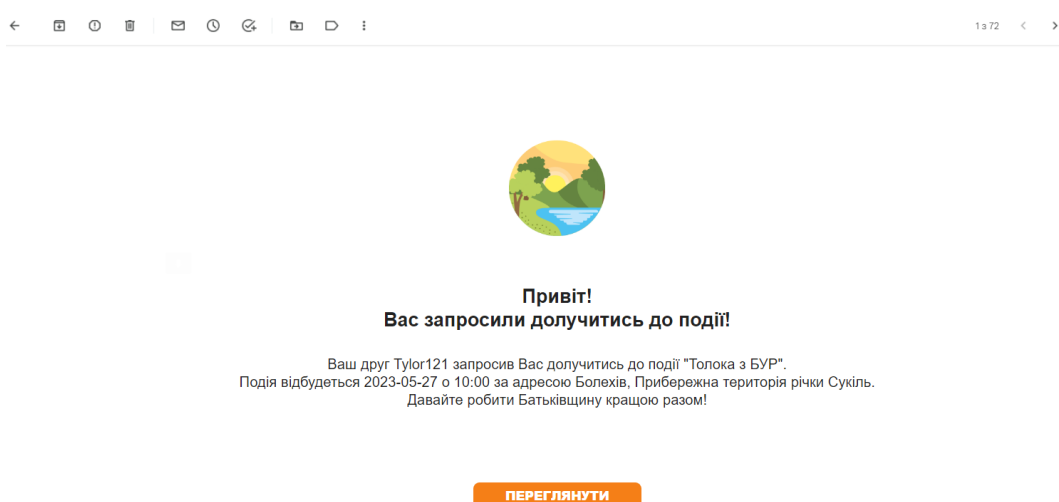


Рисунок 5.10.2 – Вміст надісланого електронного листа

## ВИСНОВКИ

Під час написання дипломної роботи було докладено великих зусиль для реалізації всіх поставлених задач і створення функціонального веб-застосунку. Процес розробки передбачав не лише технічну реалізацію, але й проведення пошуку, моніторингу та аналізу наявних веб-ресурсів з екологічною спрямованістю для українського населення.

Під час цього аналізу було усвідомлено недоліки та обмеження існуючих ресурсів, а також сформульовано чіткі ідеї та вимоги до власного веб-застосунку. Враховуючи ці фактори, було зосереджено зусилля на розробці еко-платформи "Чиста Україна", яка мала потенціал привернути увагу користувачів до екологічної ситуації в їхній країні та сприяти її покращенню.

Технічна складова проекту включала використання сучасних технологій розробки, які забезпечували надійність, безпеку та швидкість роботи програми. Це дозволить не лише використовувати платформу у навчальних цілях, але також забезпечить її потенціал для комерційного використання, залучаючи більше користувачів до активної участі у вирішенні екологічних проблем.

За результатами практичної частини роботи було успішно реалізовано еко-платформу "Чиста Україна", розгорнуту версію якої можна отримати за посиланням <http://eco-platfrom.s3-website-us-east-1.amazonaws.com>. Даний застосунок має потенціал зацікавити людей у екологічній ситуації їхньої країни та вплинути на її покращення. Впровадження платформи може мати значний вплив на свідомість громадян і спонукати їх до прийняття активних кроків у напрямку сталого розвитку та збереження навколишнього середовища.

У майбутньому, еко-платформа "Чиста Україна" може стати важливим інструментом для поширення свідомості про екологічні проблеми, залучення громадськості до спільних дій і сприяння створенню екологічно свідомого суспільства в Україні. Завдяки внеску кожного громадянина та колективного зусилля, ми можемо забезпечити чисте та здорове майбутнє для наступних поколінь.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сталий розвиток для України. [Електронний ресурс]. – Режим доступу: <https://sd4ua.org/integratsiya/>
2. Екосистема. [Електронний ресурс]. – Режим доступу: <https://eco.gov.ua/>
3. Litterati. [Електронний ресурс]. – Режим доступу: <https://www.litterati.org/>
4. TrashOut. [Електронний ресурс]. – Режим доступу: <https://www.trashout.ngo/>
5. FastAPI Documentation. Tiangolo. [Електронний ресурс]. - Режим доступу: <https://fastapi.tiangolo.com>.
6. Abdulazeez, Abdulazeez Adeshina. Building Web APIs with FastAPI and Python: A fast-paced guide to building high-performance, robust web APIs with very little boilerplate code. Kindle Edition, August 2022.
7. React Documentation. React. [Електронний ресурс]. - Режим доступу: <https://reactjs.org/>
8. Alex Banks and Eve Porcello. Learning React: Functional Web Development with React and Redux. May 2017.
9. Auth0 Documentation. Auth0. [Електронний ресурс]. - Режим доступу: <https://auth0.com/docs/>.
10. Google Maps Platform Documentation. Google. [Електронний ресурс]. - Режим доступу: <https://developers.google.com/maps/documentation>.
11. Alembic Documentation. [Електронний ресурс]. – Режим доступу: <https://alembic.sqlalchemy.org/en/latest/tutorial.html>
12. Leonard Richardson and Sam Ruby. RESTful Web APIs: Services for a Changing World. O'Reilly Media, 2013. [Електронний ресурс]. - Режим доступу: <https://archive.org/details/RESTfulWebServices>.
13. Docker Documentation. [Електронний ресурс]. - Режим доступу: <https://docs.docker.com/>
14. Jeff Nickoloff. Docker in Action. Manning Publications, 2016. [Електронний ресурс]. - Режим доступу: <https://pepa.holla.cz/wp-content/uploads/2016/10/Docker-in-Action.pdf>
15. Amazon Web Services. AWS Documentation. [Електронний ресурс]. - Режим

доступу: <https://docs.aws.amazon.com/>.

16. Andreas Wittig and Michael Wittig. Amazon Web Services in Action. Manning Publications, 2015. [Электронный ресурс]. - Режим доступа: <https://s3-ap-southeast-1.amazonaws.com/tv-prod/documents%2Fnull-Amazon+Web+Services+in+Action.pdf>
17. Morgan Perry. The 5 best AWS Deployment Options to Consider in 2022. February 12, 2022. [Электронный ресурс]. - Режим доступа: <https://www.qovery.com/blog/the-5-best-aws-deployment-options-to-consider-in-2022>
18. Movable Type Scripts. Great circle distance between 2 points. [Электронный ресурс]. - Режим доступа: <http://www.movable-type.co.uk/scripts/gis-faq-5.1.html>
19. GeoPy Documentation. [Электронный ресурс]. - Режим доступа: <https://geopy.readthedocs.io/>