

Національний технічний університет України

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

“До захисту допущено”

Завідувач кафедри ЦТЕ

_____ Наталія АУШЕВА

“ ____ ” _____ 2024 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою

“Цифрові технології в енергетиці”

зі спеціальності 122 “Комп’ютерні науки”

на тему: **Веб-система для оцінювання шкоди довкіллю внаслідок
повномасштабної війни**

Виконав:

студент IV курсу, групи ТМ-01

_____ ЛУЗАНЧУК Олексій Сергійович

(прізвище, ім’я, по батькові)

_____ (підпис)

Керівник:

професор каф. цифрових технологій в енергетиці

_____ професор, д.т.н., Володимир СЛІПЧЕНКО

(посада, вчене звання, науковий ступінь, ім’я, ПРІЗВИЩЕ)

_____ (підпис)

Рецензент: *доцент каф. інженерії програмного забезпечення в енергетиці,*

_____ доцент, к.т.н., Олександр ГАГАРІН

(посада, вчене звання, науковий ступінь, ім’я, ПРІЗВИЩЕ)

_____ (підпис)

Н.контроль:

_____ доцент Андрій ДОНЕЦЬ

(посада, ім’я, ПРІЗВИЩЕ)

_____ (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти – перший (бакалаврський)

спеціальність 122 “Комп’ютерні науки”

Освітньо-професійна програма “Цифрові технології в енергетиці”

ЗАТВЕРДЖУЮ

Завідувач кафедри ЦТЕ

Наталія АУШЕВА

«__» _____ 2024 р.

ЗАВДАННЯ
на дипломну роботу студенту

ЛУЗАНЧУКУ Олексію Сергійовичу

(прізвище, ім’я, по батькові)

1. Тема роботи Веб-система для оцінювання шкоди довкіллю
внаслідок повномасштабної війни

керівник роботи

Сліпченко Володимир Георгійович, д.т.н., професор

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

затверджені наказом по університету від «__» _____ 202__р.
№ _____

2. Термін подання роботи студентом **7 червня 2024 року**

3. Вихідні дані до роботи мова програмування JavaScript, мова розмітки HTML, мова стилів CSS, середовище розробки Microsoft Visual Studio Code

4. Перелік питань, які потрібно розробити _____

1) Опрацювати літературу за темою роботи, опрацювати наявні закони про завдання шкоди природі

2) проаналізувати коефіцієнти, значення які будуть приймати участь у формулі

3) обґрунтувати вибір формули, засобів, технологій та алгоритмів для реалізації програмного забезпечення

4) налаштувати архітектурну частину програмного забезпечення

5) створити програмну частину додатку

6) надати інструкцію користувача

5. Орієнтовний перелік графічного (ілюстративного) матеріалу рисунки, що зображують приклади роботи з системою, презентація, діаграми, таблиці

6. Дата видачі завдання «15» вересня 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Затвердження теми роботи		
2.	Аналіз та вивчення задачі	17-20.04.24	
3.	Розробка алгоритму програмної системи	21-25.04.24	
4.	Написання коду для взаємодії елементів системи	25.04-02.05.24	
5.	Візуальна частина програми	03-07.05.24	
6.	Оформлення пояснювальної записки	08-12.05.24	
7.	Захист програмного продукту	13-17.05.23	
8.	Передзахист	03-06.06.23	
9.	Подання готової роботи на кафедру	07.06.2024	
10.	Захист	17-21.06.2024	

Студент

(підпис)

Олексій ЛУЗАНЧУК

(ім'я, ПРІЗВИЩЕ)

Керівник роботи

(підпис)

Володимир СЛІПЧЕНКО

(ім'я, ПРІЗВИЩЕ)

АНОТАЦІЯ

Дипломна робота виконана на 87 сторінках, містить 33 ілюстрації, 1 таблицю, 1 додаток, 12 джерел в переліку посилань.

Мета роботи – створення веб-системи для оцінки шкоди довкіллю внаслідок повномасштабної війни.

Методи та засоби: мова програмування JavaScript та серверне налаштування NodeJS, пакетний менеджер webpack, мова розмітки HTML, мова стилів CSS, середовище розробки Visual Studio Code.

Результат – веб-система для оцінки шкоди довкіллю внаслідок повномасштабної війни.

Ключові слова: САЙТ, ЕКОЛОГІЯ, УКРАЇНА, ШКОДА, ДОВКІЛЛЯ, JavaScript.

ABSTRACT

The thesis is 87 pages long, contains 33 illustrations, 1 table, 1 appendix, and 12 references.

The purpose of the work – to create a web-based system for assessing environmental damage caused by a full-scale war.

Methods and tools: JavaScript programming language and NodeJS server-side add-on, webpack package manager, HTML markup language, CSS style language, Visual Studio Code development environment.

Result – web-based system for assessing the environmental damage caused by a full-scale war.

Keywords: WEBSITE, ECOLOGY, UKRAINE, DAMAGE, ENVIRONMENT, JavaScript.

ЗМІСТ

ВСТУП.....	7
1 ПОСТАНОВКА ЗАДАЧІ ПОБУДОВИ ВЕБ-СИСТЕМИ ДЛЯ ОЦІНКИ ШКОДИ ДОВКІЛЛЮ ВНАСЛІДОК ПОВНОМАСШТАБНОЇ ВІЙН.....	9
1.1 Призначення веб-системи для оцінки шкоди довкіллю внаслідок повномасштабної війни.....	9
1.2 Задачі веб-системи для оцінки шкоди довкіллю внаслідок повномасштабної війни	10
2 ОПИС ІСНУЮЧИХ ДОДАТКІВ ДЛЯ ТРЕКІНГУ ШКОДИ ЗАВДАНОЇ ВНАСЛІДОК ПОВНОМАСШТАБНОЇ ВІЙНИ.	11
3 ЗАСОБИ РОЗРОБКИ ВЕБ-СИСТЕМИ ДЛЯ ВІДСЛІДКОВУННЯ ШКОДИ .	16
3.1 VS Code	16
3.2 JavaScript	17
3.3 HTML та CSS	18
3.4 GitHub	19
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ВЕБ-СИСТЕМИ ДЛЯ ОЦІНКИ ШКОДИ ВНАСЛІДОК ПОВНОМАСШТАБНОЇ ВІЙНИ	21
4.1 Діаграма прецедентів	21
4.2 Архітектура програмного забезпечення	23
4.3 Опис бази даних на основі LocalStorage	27
5 РОБОТА КОРИСТУВАЧА З ВЕБ-ДОДАТКОМ “TrackImp”	32
5.1 Системні вимоги та початок роботи	32
5.2 Функціональні можливості	33
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТОК А	54

ВСТУП

Складна ситуація, яка сталася з нашою країною – війна. Війна зачепила усі сфери життя українського і не тільки народів. Зокрема, сильно постраждала природа: понівечені ліси, забруднені водойми, тварини, які залишилися без дому, викидів у повітря. Все це, звісно, потребує відновлення, виплати репарацій та іншого, але як випадки забруднення відслідковувати?

Актуальність теми відслідковування збитків, завданих природі внаслідок повномасштабної війни, є надзвичайно важливою з кількох ключових причин. Війна має катастрофічний вплив не лише на людське життя, але й на навколишнє середовище, що призводить до довготривалих і часто непоправних наслідків для екосистем. По-перше, війна спричиняє масштабні руйнування природних територій. Ліси, річки, озера та інші природні об'єкти зазнають значних пошкоджень внаслідок бойових дій, використання важкої техніки та вибухів. Це не лише знищує флору і фауну, але й порушує природний баланс екосистем, що може мати довготривалі наслідки для біорізноманіття. По-друге, забруднення навколишнього середовища є ще однією серйозною проблемою. Викиди токсичних речовин, нафтопродуктів, важких металів та інших шкідливих речовин забруднюють ґрунт, воду та повітря. Це призводить до деградації природних ресурсів і становить загрозу для здоров'я людини та тварин. Відслідковування таких випадків забруднення є критично важливим для подальшого відновлення та оздоровлення навколишнього середовища. По-третє, відслідковування збитків допомагає оцінити економічні втрати та визначити масштаби необхідних відновлювальних робіт. Війна вимагає значних фінансових витрат, і точна оцінка екологічних збитків дозволяє планувати витрати на відновлення більш ефективно. Крім того, це допомагає залучити міжнародну підтримку та фінансування для відновлювальних проектів. Врешті, моніторинг збитків є важливим для розробки та впровадження ефективних політик і стратегій щодо захисту навколишнього середовища у воєнний час. Дані про екологічні збитки можуть бути використані

для вдосконалення законодавства та нормативних актів, що регулюють військові дії з точки зору екологічної безпеки.

Зрештою, відслідковування екологічних збитків сприяє підвищенню обізнаності громадськості щодо важливості захисту навколишнього середовища під час війни. Це стимулює громадян та організації до активної участі у відновлювальних заходах і сприяє формуванню відповідального ставлення до природи.

Мета дипломної роботи є створення веб-системи, яка допоможе ефективно визначати та оцінювати шкоду довкіллю після війни, що в подальшому може бути використано для підтримання стійкості природного середовища та швидкого відновлення екосистем.

Зміст диплому включає кілька розділів, кожен з яких присвячений окремому аспекту розробки мобільного додатку. У першому розділі, "Постановка задачі", визначено призначення програмного забезпечення та основні задачі, які воно повинно вирішувати. У другому розділі "Опис існуючих додатків для трекінгу шкоди завданої внаслідок повномасштабної війни" проведено аналіз конкурентів та порівняння функціоналу розробленого додатку з конкурентами. Третій розділ "Засоби розробки веб-системи для відслідковування шкоди" включає в себе опис технічної частини: середовища розробки, мови, БД. Наступний розділ "Опис програмної реалізації веб-системи для оцінки шкоди внаслідок повномасштабної війни" є описом архітектурної частини додатку, в розділі також описана взаємодія з БД, та показана діаграма прецедентів. Фінальним розділом є "Робота користувача з веб-додатком 'TrackImp'" який слугує інструкцією для коректного користування розробленої програми.

1 ПОСТАНОВКА ЗАДАЧІ ПОБУДОВИ ВЕБ-СИСТЕМИ ДЛЯ ОЦІНКИ ШКОДИ ДОВКІЛЛЮ ВНАСЛІДОК ПОВНОМАСТШТАБНОЇ ВІЙНИ

У цьому розділі описується постановка задачі, актуальність проблеми, основні вимоги до системи, цільова аудиторія, інструменти які використовувалися при створенні даної системи, результати та їх можливе подальше застосування.

1.1 Призначення веб-системи для оцінки шкоди довкіллю внаслідок повномасштабної війни

Веб-система для оцінки шкоди довкіллю, завданої повномасштабною війною в Україні, призначена для ефективного моніторингу, аналізу та візуалізації екологічних наслідків військових дій. Система використовує спеціальні формули для розрахунку шкоди на землі, в повітрі та воді на різних територіях України.

Важливою функцією веб-системи є донесення до звичайного користувача важливості моніторингу даних про екологічний стан. Система показує, як природа страждає від війни, використовуючи інтуїтивно зрозумілі візуалізації, фото. Це сприяє підвищенню обізнаності громадськості про екологічні проблеми та необхідність їх вирішення [5].

Основні функції веб-системи включають:

- Моніторинг та збір даних: Збір і регулярне оновлення інформації про екологічний стан.
- Звітність: Надання інтуїтивно зрозумілих та детальних звітів про шкоду довкіллю.

Потенційні користувачі даного мобільного додатку є звичайні люди, які просто “скролять” інтернет новини навіть не підозрюючи, як природа може потерпати від війни, урядові органи, екологічні організації та науковці яким це

може допомогти у розробці планів відновлення та заходів для зменшення негативного впливу на довкілля, а також підвищує обізнаність населення про екологічні наслідки війни.

1.2 Задачі веб-системи для оцінки шкоди довкіллю внаслідок повномасштабної війни

Веб-система може виконувати широкий спектр задач, наприклад: обізнаність користувачів про шкоду довкіллю шляхом перегляду оцінки шкоди, моніторинг стану довкілля, аналіз та оцінка шкоди за різними даними та з різними екологічними ділянками, звітність та документообіг, прийняття рішень, політична підтримка та інші. Застосунок може використовуватися як потужний інструмент у політиці, оскільки зараз у всіх на слуху екологія та її похідні.

Також не менш важливим завданням є можливість працювати з телефону та навіть без інтернету. Розроблений застосунок можна розгорнути в локальній мережі, де експерти зможуть кваліфіковано занести випадок забруднення до бази, а потім під'єднати випадок до усіх інших, коли з'явиться доступ до інтернету.

Система використовує спеціальні формули та алгоритми для оцінки шкоди на землі, в повітрі та воді, а також аналізує дані з різних екологічних ділянок для визначення найбільш постраждалих районів.

Формуються детальні звіти про екологічний стан, забезпечуючи точні дані для розробки планів відновлення та заходів щодо зменшення негативного впливу на довкілля. Веб-система також надає аналітичні дані, допомагаючи визначити пріоритетні напрямки для екологічних заходів і програм відновлення. Важливою функцією є також політична підтримка, адже система може бути використана як інструмент для формування екологічної політики та залучення міжнародної допомоги.

2 ОПИС ІСНУЮЧИХ ДОДАТКІВ ДЛЯ ТРЕКІНГУ ШКОДИ ЗАВДАНОЇ ВНАСЛІДОК ПОВНОМАСШТАБНОЇ ВІЙНИ

Створюючи будь-який додаток, завжди потрібно розуміти та аналізувати своїх конкурентів, їхні продукти, сильні та слабкі сторони. Оскільки дана веб-система виконує функції обрахунку шкоди, завданої природі внаслідок повномасштабної війни, та побудована у вигляді веб-сайту, було знайдено та досліджено схожі за призначенням додатки: "ЕкоЗагроза" [9], "Завдання Штабу" [10] та безліч інформаційних сайтів, які несуть тільки інформацію і не виконують ніякої корисної дії, крім донесення інформації.

Відразу видно недоліки використання вищезгаданих застосунків для подання інформації. Вони потребують обов'язкової реєстрації та завантаження мобільного застосунку, що може бути не таким зручним, як подання даних через сайт. Щодо інформаційного наповнення звичайних сайтів, воно не несе корисної інформації та якихось розрахунків. На рисунках 2.1, 2.2 та 2.3 зображено зовнішній вигляд інтерфейсу конкурентних програм.

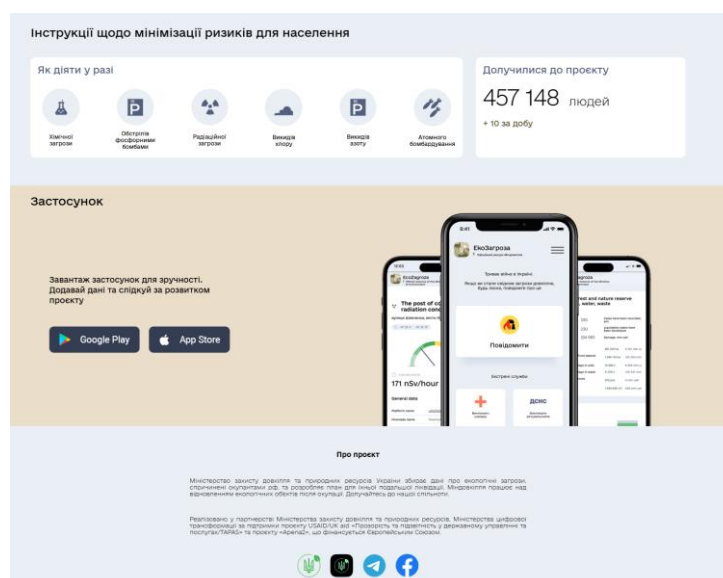


Рисунок 2.1 – Мобільні додатки “ЕкоЗагроза”

Як передати інформацію?

Військова агресія Російської Федерації завдає шкоду не тільки життю та здоров'ю людей, інфраструктурі України, а і навколишньому природному середовищу нашої держави.

Якщо Ви стали свідком екозлочину просимо повідомляти Оперативний штаб будь-яким зручним для Вас способом:

Телефон гарячої лінії

[+380.\(96\).756-83-66](tel:+380967568366) (Telegram, WhatsApp, Viber)

Чатбот у Telegram

[@EcoShkodaBot](https://t.me/EcoShkodaBot)

Email

eco@shtab.gov.ua

Мобільний застосунок



Заповнити онлайн форму



Рисунок 2.2 – Мобільні додатки Завдання Штабу

На рисунку 2.3 представлено звичайний ресурс новин яких в інтернеті налічується більше мільйона.

ТЕХНОЛОГІЯ

ХІД ВІЙНИ #НОВОБИНА #ENGLISH СТАТІ ФРАГМЕНТИ ВИБІР РЕДАКЦІЇ

Якої шкоди завдала війна довкіллю України – The Economist

ФРАГМЕНТИ e

Отримуйте найкращі статті на е-малі (раз на два тижні)

Підписатися

Подивитись свіжий випуск

Стежити

Facebook Instagram Twitter YouTube

2023-05-17 10:20

Британський тижневик The Economist пише про масштаби шкоди, завданої довкіллю України через повномасштабне вторгнення Росії. Наводимо переклад тексту.

До російського вторгнення люди з доколишніх сіл з'їжджалися на береги Оскільського водосховища, щоб скупатися, послухати спів птахів і половити рибу. Сьогодні значна частина цієї території, яка простягається приблизно на сотню кілометрів на південь від Куп'янська, змінена до невпізнання.

Зруйновані села, визволені українцями під час контрнаступу чотири місяці тому, обвуглені дерева і залишки згорілих російських танків вистилають східні береги водосховища. Минулої весни, щоб запобігти переправі російських військ через річку Донець нижче за течією від Осколу, українські війська підірвали шлюзи водосховища. Цей крок допоміг сповільнити російський наступ.

Але це також прискрало Оскіл на загибель. З водосховища вилилося понад 350 млн кубометрів води. Рівень води різко впав, оголивши значні ділянки дна водосховища. Птахи, які колись гніздилися в цій місцевості, зникли. Вважається, що загинуло близько 2 млн особин риби. Місцеві колодязі пересохли.

Ліси, спустошені пожежами, спричиненими безперервними обстрілами, або вирубані для будівництва окопів; ґрунтові води та ґрунти, забруднені важкими металами та токсичними хімічними речовинами від боєприпасів,

Російське вторгнення з окупованих територій. Скільки Росія зробила на українській землі

Маніпулятор

Що там? (10 фото)

shoduk

Погода: +24°

Рисунок 2.3 – Мобільні додатки Завдання Штабу

Переваги створеного застосунку, в подальшому "TrackImp". Однією з головних переваг "TrackImp" є його простота та зручність у використанні. Додаток не потребує обов'язкової реєстрації або завантаження на мобільний пристрій. Користувачі можуть отримати доступ до всіх функцій додатку безпосередньо через веб-браузер, що значно спрощує процес використання.

Інтерфейс додатку інтуїтивно зрозумілий і не перевантажений зайвою інформацією, що дозволяє користувачам швидко знаходити необхідні дані та здійснювати потрібні дії. "TrackImp" розроблений з урахуванням потреб експертів у галузі екології. Додаток надає інструменти для детального аналізу та оцінки шкоди довкіллю, включаючи можливість введення та обробки великих обсягів даних. Це робить його ідеальним вибором для науковців, дослідників та державних службовців, які займаються питаннями екологічної безпеки.

Однією з найважливіших функцій "TrackImp" є можливість масового введення даних за допомогою зразка Excel. Адміністратори системи можуть завантажувати до бази майже сто записів за раз, що значно полегшує процес внесення інформації та забезпечує ефективний моніторинг. Це особливо корисно у випадках, коли потрібно швидко обробити великі обсяги даних, наприклад, після екологічної катастрофи або під час регулярного моніторингу. Ще однією важливою перевагою "TrackImp" є можливість роботи в офлайн-режимі. Це дозволяє експертам використовувати додаток навіть у тих місцях, де відсутнє інтернет-з'єднання. У локальній мережі користувачі можуть кваліфіковано заносити випадки забруднення до бази, а потім синхронізувати ці дані з основною базою, коли з'явиться доступ до інтернету. Така функціональність є критично важливою в умовах екологічних катастроф або в регіонах з нестабільним інтернет-з'єднанням. Порівняння схожих веб-систем за функціоналом наведені у таблиці 2.1.

Веб-система "TrackImp" значно перевершує своїх конкурентів завдяки ряду унікальних функцій та переваг. Вона забезпечує масове введення даних, легкість у використанні, орієнтованість на експертів, автоматичний збір даних з різних джерел, підтримку роботи в офлайн-режимі, інтуїтивно зрозумілий інтерфейс, детальний аналіз та оцінку шкоди довкіллю, формування звітів, підтримку

прийняття рішень та політичну підтримку, а також можливість кастомізації. Ці переваги роблять "TrackImp" ідеальним інструментом для моніторингу та оцінки екологічної шкоди, забезпечуючи ефективність та зручність у використанні, яких бракує конкурентним додаткам.

Таблиця 2.1 – Порівняння схожих додатків

Функціональність	TrackImp	ЕкоЗагроза	Завдання Штабу	Інформаційні сайти
Масове введення даних за допомогою Excel	Так	Ні	Ні	Ні
Легкість використання через веб-браузер	Так	Ні, лише мобільний додаток	Ні, лише мобільний додаток	Так
Орієнтованість на експертів	Так	Ні	Ні	Ні
Підтримка роботи в офлайн-режимі	Так	Ні	Ні	Ні
Інтуїтивно зрозумілий інтерфейс	Так	Так	Перевантажений інформацією	Так, але обмежена функціональність
Формування звітів та документообіг	Так	Обмежений	Обмежений	Ні
Можливість кастомізації	Так	Так	Так	Ні

Завдяки наведеній таблиці можна наглядно подивитися на переваги створеної програми.

3 ЗАСОБИ РОЗРОБКИ ВЕБ-СИСТЕМИ ДЛЯ ВІДСЛІДКОВУННЯ ШКОДИ

Розроблена веб-система включає в себе такі методи розробки, як JavaScript, CSS, HTML, Git, Visual Studio. "TrackImp" може розгортатися у локальній мережі та працювати без доступу до Інтернету. Також програма використовує Excel — програму пакету Microsoft Office, яка є доступною на багатьох пристроях та має велику сумісність. Веб-додаток завантажений в онлайн застосунок контролю версій — GitHub, і є open-source проектом.

3.1 VS Code

Під час розробки веб-системи "TrackImp" було використано Visual Studio Code (VS Code) як основне середовище розробки. Це потужний і гнучкий інструмент, який забезпечив ефективну та зручну роботу над проектом завдяки своїм численним функціям та перевагам.

Visual Studio Code доступний для операційних систем Windows, macOS та Linux. Це дозволяє розробникам працювати на будь-якій зручній платформі, що підвищує гнучкість та зручність команди розробників. VS Code є легким редактором коду, який швидко запускається та працює, навіть на комп'ютерах з обмеженими ресурсами[4]. Це дозволяє розробникам швидко починати роботу і не витратити час на очікування завантаження середовища. Visual Studio Code підтримує велику кількість розширень та плагінів, які можна легко встановити з вбудованого магазину розширень. Це дозволяє адаптувати середовище розробки під конкретні потреби проекту. Наприклад, під час розробки "TrackImp" було використано розширення для підтримки JavaScript, CSS, HTML, Git та інші.

Вбудована підтримка Git дозволяє легко виконувати операції з контролю версій безпосередньо з VS Code. Це включає в себе коміти, пуші, пулі, створення

гілок та інші операції. Така інтеграція спрощує процес спільної розробки та управління кодом.

Visual Studio Code має вбудовані можливості для налагодження коду. Це включає в себе налаштування точок зупинки, перегляд змінних у реальному часі та аналіз виконання коду, що значно спрощує процес відлагодження і виявлення помилок.

3.2 JavaScript

JavaScript був обраний як основна мова програмування для розробки веб-системи "TrackImp". Завдяки своїй гнучкості та широким можливостям, JavaScript дозволив створити ефективну та зручну веб-систему для оцінки шкоди довкіллю.

Переваги використання JavaScript

JavaScript працює у всіх сучасних веб-браузерах, що робить його універсальним інструментом для розробки веб-додатків. Це дозволяє "TrackImp" бути доступним на різних платформах і пристроях без необхідності додаткових налаштувань.

JavaScript дозволяє створювати динамічні та інтерактивні веб-додатки. Це дуже важливо для "TrackImp", який включає функціонал для відображення даних в реальному часі, взаємодії з користувачами та візуалізації результатів оцінки шкоди довкіллю.

Існує багато бібліотек та фреймворків, які розширюють можливості JavaScript. Під час розробки "TrackImp" використовувалися популярні бібліотеки, такі як React для побудови користувацького інтерфейсу, а також D3.js для візуалізації даних. Це значно спрощує процес розробки та забезпечує високу якість кінцевого продукту[6]. У розробці додатку використовувалася бібліотека "Notiflix"[7].

Використання JavaScript у "TrackImp"

JavaScript відіграє ключову роль у взаємодії з DOM (Document Object Model). Це дозволяє динамічно змінювати структуру HTML-сторінки без її перезавантаження, що є важливим для створення інтерактивних користувацьких інтерфейсів. Наприклад, коли користувач вводить дані про новий випадок екологічної шкоди, JavaScript обробляє ці дані та оновлює відповідні елементи DOM, щоб відобразити нову інформацію в режимі реального часу.

Для обчислення формул, необхідних для оцінки шкоди довкіллю, JavaScript використовує вбудовані математичні функції. Наприклад, розрахунок площі забрудненої території, обсягів викидів у повітря або водні ресурси, заснований на введених даних. Ці обчислення виконуються безпосередньо у браузері користувача, що зменшує навантаження на сервер і підвищує швидкість роботи програми.

JavaScript також використовується для операцій з локальним сховищем (Local Storage). Це дозволяє зберігати дані на стороні клієнта, що є корисним для офлайн-режиму. Наприклад, якщо користувач вводить дані про екологічний інцидент, але не має доступу до Інтернету, ці дані можуть бути збережені у локальному сховищі. Після відновлення з'єднання, ці дані можуть бути автоматично передані на сервер.

Операції з базою даних, побудованою на основі локального сховища, також реалізовані за допомогою JavaScript. Для цього використовуються індексовані бази даних (IndexedDB), які дозволяють зберігати великі обсяги структурованих даних на стороні клієнта. Це забезпечує високу продуктивність і надійність зберігання даних. Наприклад, дані про всі випадки екологічної шкоди можуть бути збережені в IndexedDB, що дозволяє швидко здійснювати пошук і обробку цих даних без необхідності постійного звернення до сервера.

Таким чином, JavaScript став ключовою технологією для розробки "TrackImp", забезпечуючи його функціональність, інтерактивність та ефективність у зборі та обробці даних про екологічні збитки.

3.3 HTML та CSS

HTML (HyperText Markup Language) був використаний для структурування вмісту веб-системи "TrackImp". Він забезпечує основну розмітку сторінок, визначаючи елементи, такі як заголовки, параграфи, форми та таблиці, що відображають дані про екологічні збитки. HTML дозволяє створювати семантично правильні веб-сторінки, що робить їх зрозумілими як для користувачів, так і для пошукових систем.

CSS (Cascading Style Sheets) використовується для стилізації HTML-елементів і надання їм привабливого вигляду. У "TrackImp" CSS забезпечує дизайн і розташування елементів на сторінці, включаючи кольори, шрифти, відступи та макети. Завдяки CSS, веб-додаток виглядає сучасно і зручно для користувачів, забезпечуючи приємний досвід взаємодії з додатком.

HTML: Використовується для створення основної структури веб-сторінок, включаючи форми для введення даних, таблиці для відображення результатів оцінки шкоди, кнопки для взаємодії користувачів та інші елементи інтерфейсу.

CSS: Використовується для стилізації цих елементів, включаючи налаштування кольорових схем, шрифтів, розмірів елементів та їхнього розташування на сторінці. CSS також забезпечує адаптивний дизайн, що дозволяє "TrackImp" коректно відображатися на різних пристроях, включаючи мобільні телефони та планшети.

Ці технології разом забезпечують функціональність і естетичний вигляд веб-системи "TrackImp", роблячи її зручною і привабливою для користувачів.

3.4 GitHub

GitHub для збереження версій

Розробка "TrackImp" включає використання GitHub, популярної платформи для контролю версій і спільної розробки. Це забезпечує кілька важливих переваг:

- **Контроль версій:** Всі зміни у коді зберігаються як окремі коміти, що дозволяє відстежувати історію змін, повертатися до попередніх версій коду та аналізувати, хто і коли вносив зміни.
- **Спільна робота:** GitHub дозволяє кільком розробникам одночасно працювати над проєктом, використовуючи гілки для розробки нових функцій і злиття цих гілок у головну версію коду після перевірки.
- **Безпека та надійність:** Код зберігається у хмарі, що захищає його від втрат у випадку апаратних збоїв на локальних машинах.

GitHub Pages — це сервіс, що дозволяє розміщувати статичні веб-сайти безпосередньо з репозиторію GitHub. Для "TrackImp" це означає, що користувачі можуть легко отримати доступ до веб-додатку через Інтернет. Основні переваги деплою на GitHub Pages:

- **Простота налаштування:** Публікація веб-додатку на GitHub Pages є простою та швидкою. Достатньо налаштувати відповідний репозиторій і активувати GitHub Pages у налаштуваннях.
- **Доступність:** Після деплою веб-додаток стає доступним за унікальною URL-адресою, що дозволяє користувачам з усього світу легко отримати до нього доступ.
- **Безкоштовність:** GitHub Pages пропонує безкоштовний хостинг для статичних веб-сайтів, що є значною перевагою для багатьох проєктів.
- **Доступ до веб-додатку:** Відвідувати сайт, використовуючи надану URL-адресу, і взаємодіяти з усіма функціями додатку.
- **Оновлення та покращення:** Завдяки збереженню коду на GitHub, розробники можуть регулярно оновлювати додаток, виправляти помилки і додавати нові функції, зберігаючи при цьому історію змін.
- **Спільна робота:** Інші розробники можуть вносити свої пропозиції або робити внески до проєкту через pull requests, що сприяє покращенню якості та функціональності веб-додатку.

Таким чином, використання GitHub для збереження версій та деплою "TrackImp" на GitHub Pages забезпечує надійну, доступну та зручну платформу для розробки, публікації та подальшого розвитку веб-системи.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ВЕБ-СИСТЕМИ ДЛЯ ОЦІНКИ ШКОДИ ВНАСЛІДОК ПОВНОМАСТАБНОЇ ВІЙНИ

Цей розділ описує файлову структуру застосунку взаємодію з БД, демонструє взаємодію функцій між собою.

4.1 Діаграма прецедентів

UML (Unified Modeling Language) — це уніфікована мова моделювання, яка використовується розробниками програмного забезпечення для візуалізації процесів та роботи систем. Вона не є мовою програмування, а радше набором правил та стандартів для створення діаграм. Ці діаграми допомагають розробникам і інженерам спілкуватися на одній мові, без необхідності заглиблюватися у фактичний код продукту. Створення діаграм за допомогою UML є ефективним способом швидко передати складні ідеї чи структури.

Одним із типів діаграм UML є діаграма варіантів використання (також відома як діаграма прецедентів або use case), яка відображає типи ролей і їхню взаємодію із системою. Хоча ця діаграма не показує порядок виконання кроків, вона ілюструє функціональні вимоги (що система може робити) з точки зору користувача. Діаграми варіантів використання можуть бути представлені текстом або графічно.

Перед розробкою застосунку корисно створити діаграму прецедентів, щоб краще зрозуміти, як реалізувати функціональні можливості додатку, які класи створювати та як проектувати графічний інтерфейс користувача.

При проектуванні додатку в середовищі “Visual Paradigm” було створено діаграми для звичайного користувача та експертів: молодшого та старшого.

Діаграми прецедентів даного веб-застосунку зображено на рисунках 4.1. та 4.2.

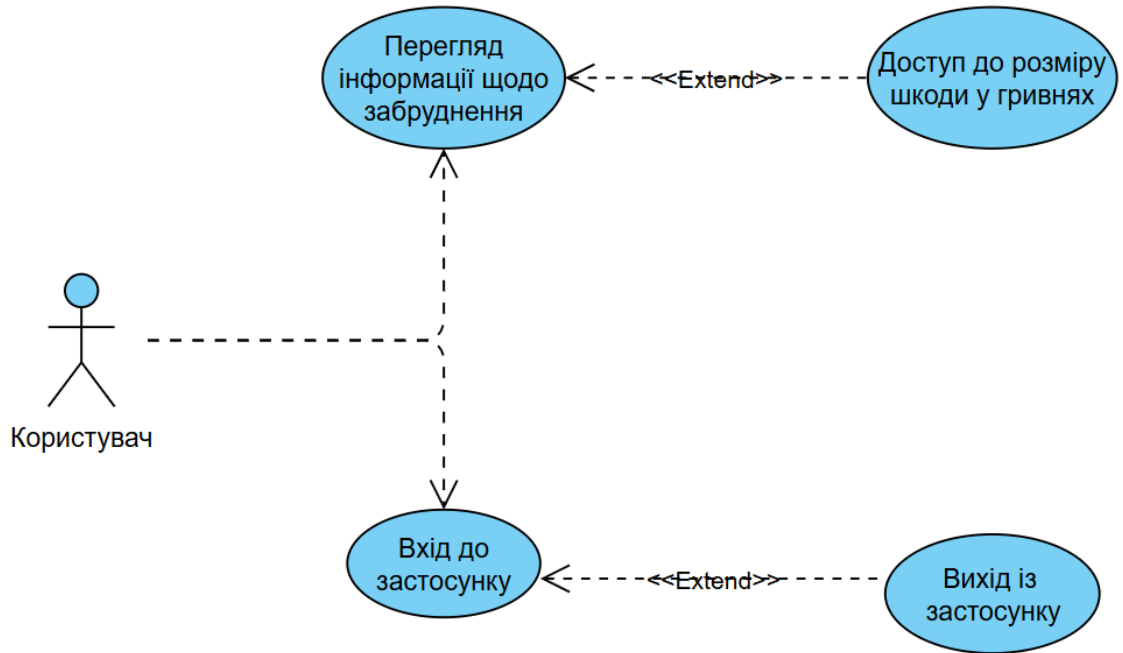


Рисунок 4.1– Діаграма прецедентів користувача з системою (проста)

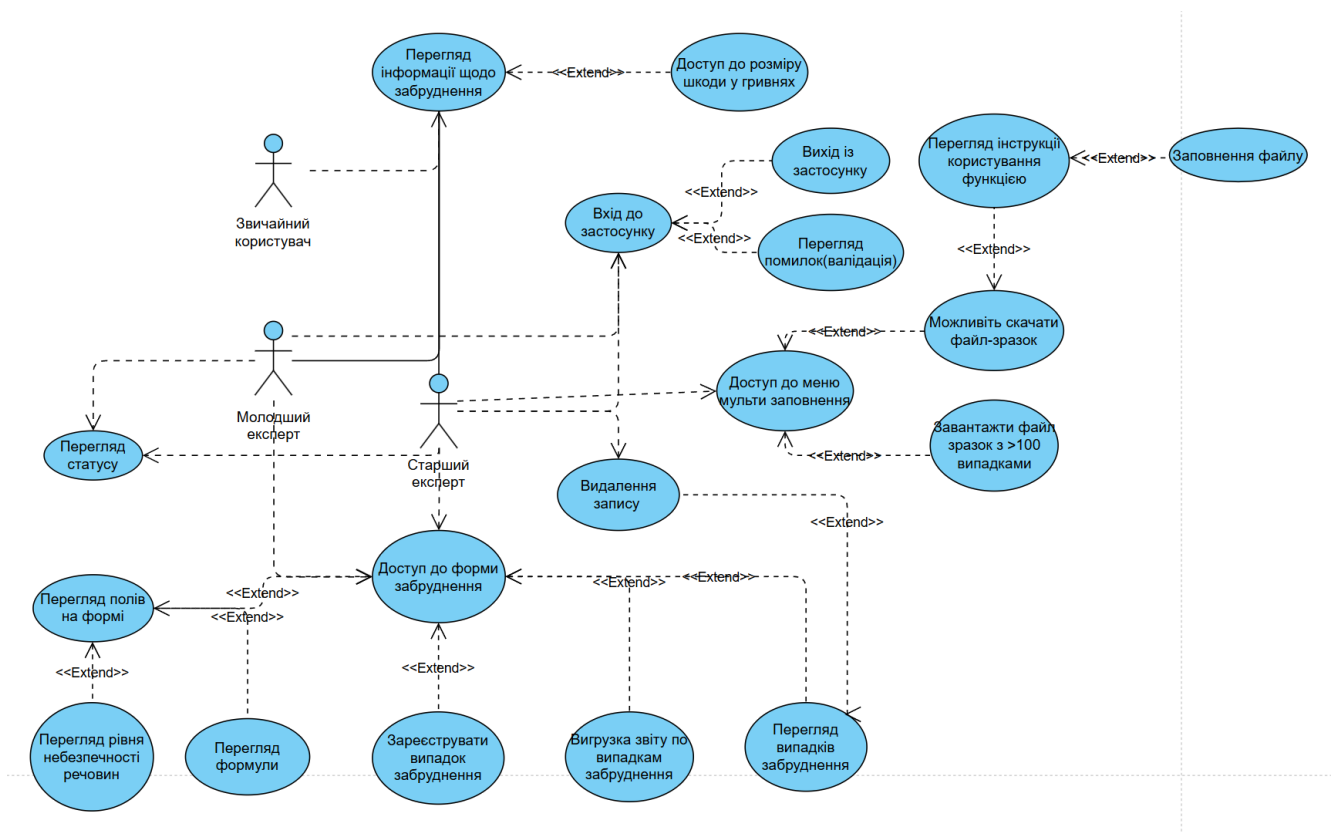


Рисунок 4.2 – Діаграма прецедентів користувача з системою (поглиблена)

На наданій діаграмі зображена UML-діаграма, яка описує процес обробки заяв про забруднення, показуючи взаємодію різних учасників системи для виконання цього завдання. Учасниками є звичайний користувач, який подає заявку про забруднення, молодший експерт, що обробляє заявку на початковому етапі, та старший експерт, який проводить детальну обробку заявки.

Процес розпочинається з того, що користувач подає заявку про забруднення. Молодший експерт отримує цю заявку, переглядає її та може виконати додаткові дії, такі як отримання детальної інформації, завантаження зразка, перегляд інструкції або вихід з системи. Після цього молодший експерт перевіряє статус заявки і може видалити її, якщо вона недійсна, або виконати інші перевірки. Потім заявка направляється старшому експерту.

Старший експерт отримує заявку, переглядає її та може виконати аналогічні дії, що й молодший експерт. Після перевірки статусу заявки старший експерт може її видалити, виконати додаткові перевірки або зареєструвати випадок. У разі реєстрації випадку старший експерт може завантажити звіт про випадки забруднення.

Додаткові можливості системи включають отримання інформації про розмір шкоди у гривнях, доступ до меню мультизаповнення, перегляд інструкції користувача, перегляд помилок валідації та завантаження файлу-зразка з 100 випадками. UML-діаграма послідовності чітко описує процес обробки заяв про забруднення, відображаючи дії та взаємодію різних учасників системи. Цей опис є лаконічним, але зберігає всю необхідну інформацію.

4.2 Архітектура програмного забезпечення

Архітектура програмного забезпечення (ПЗ) - це високорівневий план або дизайн системи програмного забезпечення. Вона визначає основні структурні компоненти системи, їхні взаємозв'язки та засоби взаємодії, щоб задовольнити

вимоги функціональності, які пред'являються до програми. Файлову структуру розробленого додатку можна переглянути на рисунку 4.3.

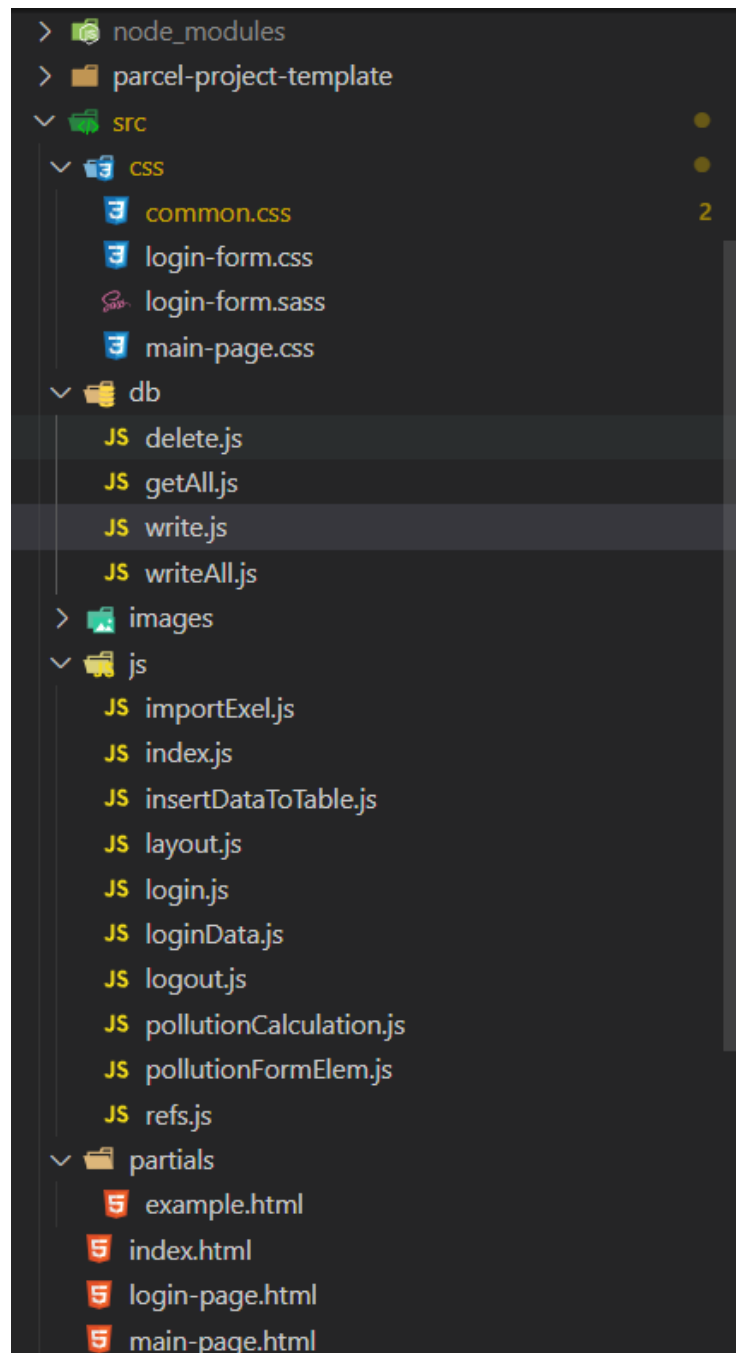


Рисунок 4.3 – Файлова структура додатку

Деякі ключові папки та файли, які можна побачити:

`node_modules`: папка містить залежності JavaScript, які використовуються в проекті.

`parcel-project-template`: папка містить шаблон проекту, який використовується для створення веб-додатку.

`images`: папка містить зображення, які використовуються в веб-додатку.

`db`: папка містить дані, які використовуються веб-додатком.

`css`: папка містить файли CSS, які використовуються для стилізації веб-додатку.

`partials`: папка містить HTML-фрагменти, які використовуються для створення сторінок веб-додатку.

`js`: папка містить файли JavaScript, які використовуються для реалізації функціональності веб-додатку.

Веб-додаток має архітектуру MVC (Model-View-Controller).

Model (Модель): Модель знаходиться в папках `db` та `js`. Ці папки містять дані, які використовуються веб-додатком, а також файли JavaScript, які використовуються для доступу до цих даних та їх маніпулювання.

View (Вид): Вид знаходиться в папках `css` та `partials`. Ці папки містять файли CSS, які використовуються для стилізації веб-додатку, а також HTML-фрагменти, які використовуються для створення сторінок веб-додатку.

Controller (Контролер): Контролер знаходиться в папці `js`. Ця папка містить файли JavaScript, які використовуються для обробки взаємодії користувача з веб-додатком та оновлення моделі та виду відповідно.

Архітектура MVC має ряд переваг, які роблять її популярним вибором для розробки веб-додатків. Ці переваги включають модульність, повторюваність, здатність до простого тестування. MVC розбиває веб-додаток на три чітко визначені частини, що робить його більш модульним та легким для обслуговування MVC сприяє повторному використанню коду, що може заощадити час та зусилля розробників. MVC полегшує тестування веб-додатків, оскільки кожен компонент (модель, вид та контролер) можна тестувати окремо[8].

Архітектурну схему середнього рівня можна побачити на рисунку 4.4.

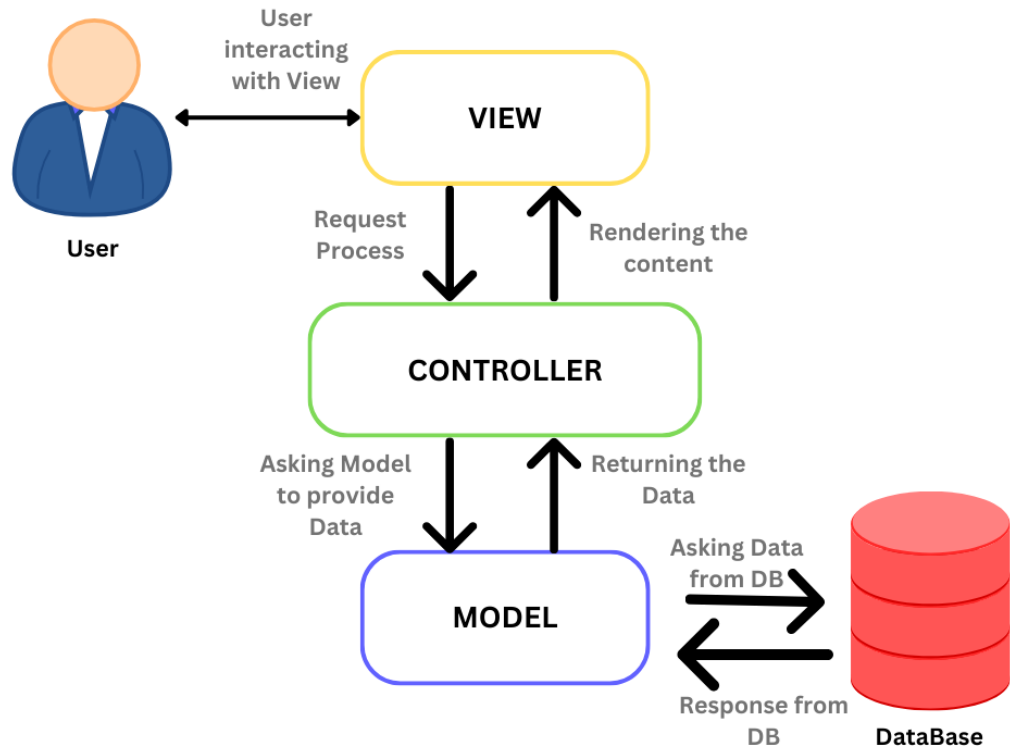


Рисунок 4.4 – MVC архітектура

На рисунку 4.4 зображено архітектуру MVC (Model-View-Controller), яка часто використовується в розробці веб-додатків. Ось основні компоненти та їх взаємодія:

User (Користувач): користувач взаємодіє з інтерфейсом користувача, який представлений "View".

View (Відображення): відповідає за рендеринг контенту та відображення інформації користувачу. отримує запити від користувача та передає їх контролеру.

Controller (Контролер): отримує запити від View, обробляє їх та запитує необхідні дані у моделі. Після отримання даних від моделі повертає їх до View для відображення.

Model (Модель): відповідає за бізнес-логіку та взаємодію з базою даних. Отримує запити від контролера, звертається до бази даних, обробляє отримані дані та повертає їх до контролера.

DataBase (База даних): зберігає дані, необхідні для роботи додатка. Взаємодіє з моделлю для збереження та отримання даних.

Цей рисунок ілюструє типову схему роботи архітектури MVC, яка допомагає розділити логіку додатка на три окремі компоненти для покращення структури та підтримки коду.

4.3 Опис бази даних на основі LocalStorage

При розробці системи було вирішено не використовувати наявні готові рішення для збереження інформації, а зробити щось власне, заточене під конкретні задачі проєкту. Зокрема, важливими критеріями були можливість працювати без інтернету, ненавантаженість на систему та простота у використанні. Наша база даних також оптимізована під роботу у браузері, що унеможлиблює потребу у розгортанні та потужному адмініструванні даних. Функція експорту даних до Excel дозволяє зручно обробляти та аналізувати інформацію. У подальшому можна розглянути можливість інтеграції з повноцінною базою даних для подальшого розвитку проєкту, проте на поточному етапі функціонал даної бази даних вже задовольняє потреби проєкту більш ніж достатньо.

```

▼ 0: {ID: "uhUY2NisfW8D10qS2YhSr", pollutionType: "Вода", pollutionRegion: "Автономна республіка Крим",...}
  ID: "uhUY2NisfW8D10qS2YhSr"
  pollutionArea: 50
  pollutionDated: "11.05.2024"
  pollutionExpense: 43.733
  pollutionRegion: "Автономна республіка Крим"
  pollutionSubstance: "Надзвичайно небезпечні"
  pollutionTerritory: "Природні території та об'єкти природно-заповідного фонду"
  pollutionType: "Вода"

```

Рисунок 4.5 – Вигляд одного з записів у БД

На рисунку видно, що об'єкт запису до бази даних є звичайним JavaScript об'єктом - структурою даних. Однак, насправді все, що потрапляє до LocalStorage,

перетворюється у формат JSON за допомогою спеціального методу об'єкту JSON. Така конвертація дозволяє зберігати дані у форматі, який зручний для зберігання та обробки в LocalStorage, забезпечуючи при цьому зручний доступ до них.

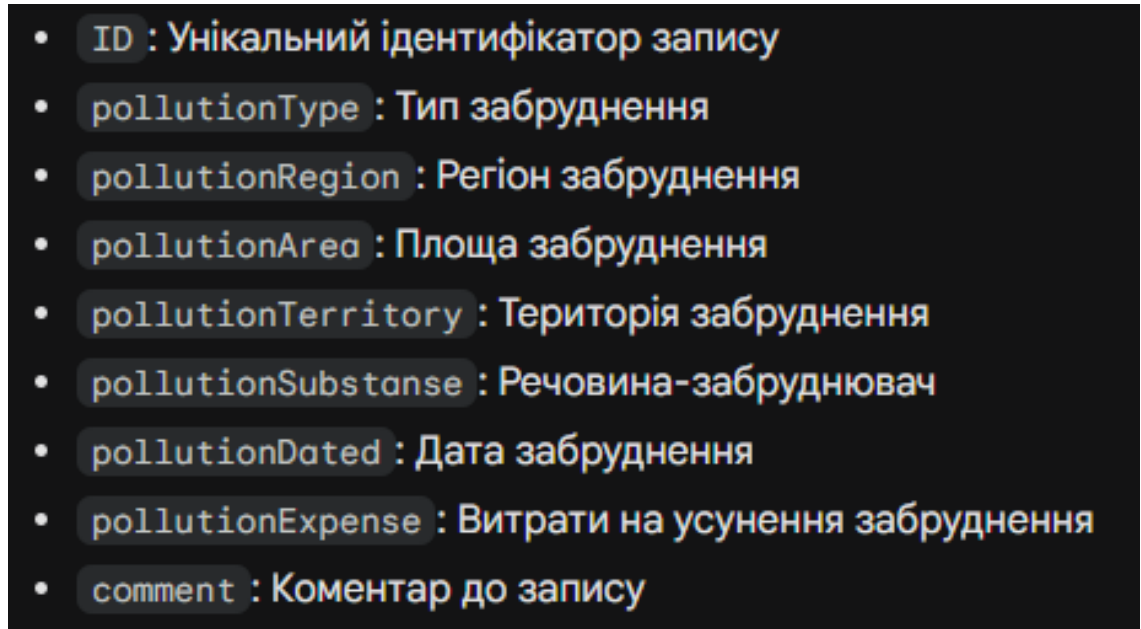


Рисунок 4.6 – Структура записів у БД

Дивлячись на рисунок, можна легко і просто зрозуміти які змінні у БД відповідають значенням. Це, до речі, також описано у формулі, яку можна побачити натиснувши на підказку. На рисунку 4.6. зображено файлова структура написаної БД.

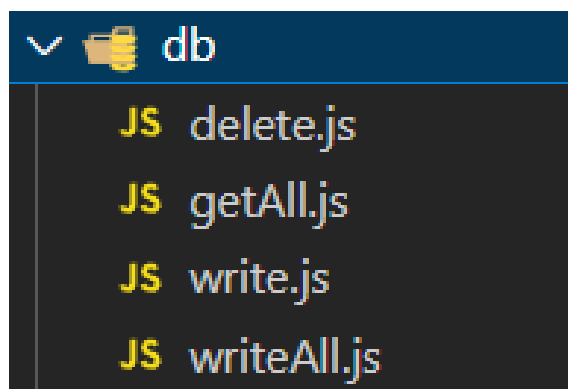


Рисунок 4.7 – Файл з функціями взаємодії БД

Файли `delete.js`, `write.js` та `writeAll.js` реалізують функціональність для маніпуляції з даними, що зберігаються у локальному сховищі браузера (`localStorage`). Ці файли забезпечують операції додавання, видалення та оновлення записів. В свою чергу файл `getAll.js` виконує запит до бази та “витягує” всі записи у вигляді масива, з яким потім зручно працювати.

Функція видалення запису (`delete.js`)

У файлі `delete.js` реалізована функція `deleteRecord(id)`, яка приймає `id` запису для видалення. Вона використовує функцію `getAll()` для отримання всіх записів, фільтрує їх, виключаючи той, що має вказаний `id`, і викликає функцію `addAll(filtered)` для оновлення локального сховища новим списком записів. Якщо `id` не переданий, виводиться повідомлення про помилку. Також функція використовує конструкцію `try...catch` для обробки помилок, що дозволяє більш зрозуміло діагностувати та повідомляти про проблеми, які можуть виникнути під час видалення запису.

Функція додавання запису (`write.js`)

У файлі `write.js` реалізована функція `add(obj)`, яка приймає об'єкт `obj` для додавання до сховища. Функція отримує існуючі записи з `localStorage`. Якщо записів немає, створює новий масив з одним об'єктом і зберігає його в `localStorage`. Якщо записи є, додає новий об'єкт до масиву, потім зберігає оновлений масив в `localStorage` і оновлює сторінку після додавання запису. У випадку помилки виводиться повідомлення про помилку. Використання конструкції `try...catch` також забезпечує детальне інформування про помилки, які можуть виникнути під час додавання даних.

Функція оновлення всіх записів (`writeAll.js`)

У файлі `writeAll.js` реалізована функція `addAll(arr)`, яка приймає масив `arr` для запису в `localStorage`. Функція перевіряє наявність даних у `localStorage`. Якщо даних немає, виводиться повідомлення про помилку. Функція видаляє поточний запис у `localStorage` і записує новий масив. У випадку помилки виводиться повідомлення

про помилку. Завдяки конструкції `try...catch`, всі помилки обробляються та детально описуються, що допомагає швидко зрозуміти та виправити проблему.

Функція отримання всіх записів (`getAll.js`)

Отримання даних здійснюється за допомогою функції `getAll()`, яка імпортована в `delete.js` і використовується для отримання всіх записів з `localStorage`. Ця функція збирає всі дані з `localStorage` та повертає їх у вигляді масиву, що спрощує подальшу роботу з ними.

Основні операції CRUD

- Файли забезпечують базову функціональність CRUD (Create, Read, Update, Delete) для роботи з даними у локальному сховищі браузера. Основні операції включають:

- Додавання нових записів через функцію `add(obj)`, яка перевіряє наявність існуючих даних, додає новий об'єкт до масиву і зберігає оновлений масив назад в `localStorage`.

- Видалення записів за їх ідентифікатором через функцію `deleteRecord(id)`, яка отримує всі записи, фільтрує їх для видалення вказаного запису і зберігає оновлений список.

- Оновлення всього масиву записів через функцію `addAll(arr)`, яка повністю перезаписує існуючі дані в `localStorage` новим масивом.

Обробка помилок

Кожна функція використовує конструкцію `try...catch` для обробки помилок. Це дозволяє забезпечити більш зрозуміле та зручне розуміння проблем, які можуть виникнути при виконанні функцій. Повідомлення про помилки описують конкретні проблеми та надаються українською мовою, що покращує взаємодію користувача з системою та спрощує діагностику та виправлення помилок.

5 РОБОТА КОРИСТУВАЧА З ВЕБ-ДОДАТКОМ “TrackImp”

У цьому розділі надано детальний опис процесу використання веб-додатку "TrackImp" з боку користувача. В цьому розділі будуть описані основні функціональні можливості додатку, послідовність дій користувача при взаємодії з різними елементами інтерфейсу, а також наведені приклади використання для кожної з функцій. Також буде надано інформацію про можливі проблеми та шляхи їх вирішення, а також рекомендації щодо оптимального використання додатку.

5.1 Системні вимоги та початок роботи

Для перевірки розробленого додатку було використано сервіс Sauce Labs, ось що пишуть про перевірку на цьому сервісі авторитетні джерела [11]“Якщо ваш додаток пройшов перевірку на сайті Sauce Labs, це можна розглядати як позитивний сигнал щодо якості вашого програмного забезпечення. Sauce Labs відомий своєю експертизою в автоматизованому тестуванні та надає високоякісні інструменти для тестування веб-додатків. Отже, якщо ваш додаток успішно пройшов їх перевірку, це може свідчити про те, що ваше програмне забезпечення пройшло певний рівень якості та надійності. Така перевірка може підвищити довіру користувачів до вашого додатка та підкреслити вашу відданість якості продукту”. На рисунку 5.1 можна побачити результат тестування додатку у сервісі Sauce Labs.

Firefox on Mac - cypress/e2e/examples/traversal.cy.js	12	126	30s
Firefox on Mac - cypress/e2e/examples/assertions.cy.js	12	126	31s
Firefox on Mac - cypress/e2e/examples/actions.cy.js	12	126	31s
Chrome on Windows - cypress/e2e/examples/traversal.cy.js	11	125	26s
Chrome on Windows - cypress/e2e/examples/assertions.cy.js	11	125	27s
Chrome on Windows - cypress/e2e/examples/actions.cy.js	11	125	28s

Рисунок 5.1 – Перевірка Sauce Labs

Окремо можна сказати, що при написанні застосунку навмисно було використано не застарілі методи та функції. Це свідчить про високу сумісність додатка з сучасними стандартами програмування та забезпечує його ефективність та стабільність. На рисунку 5.2 можна побачити результат тестування розмітки додатку в сервісі Validator.w3.org[12].

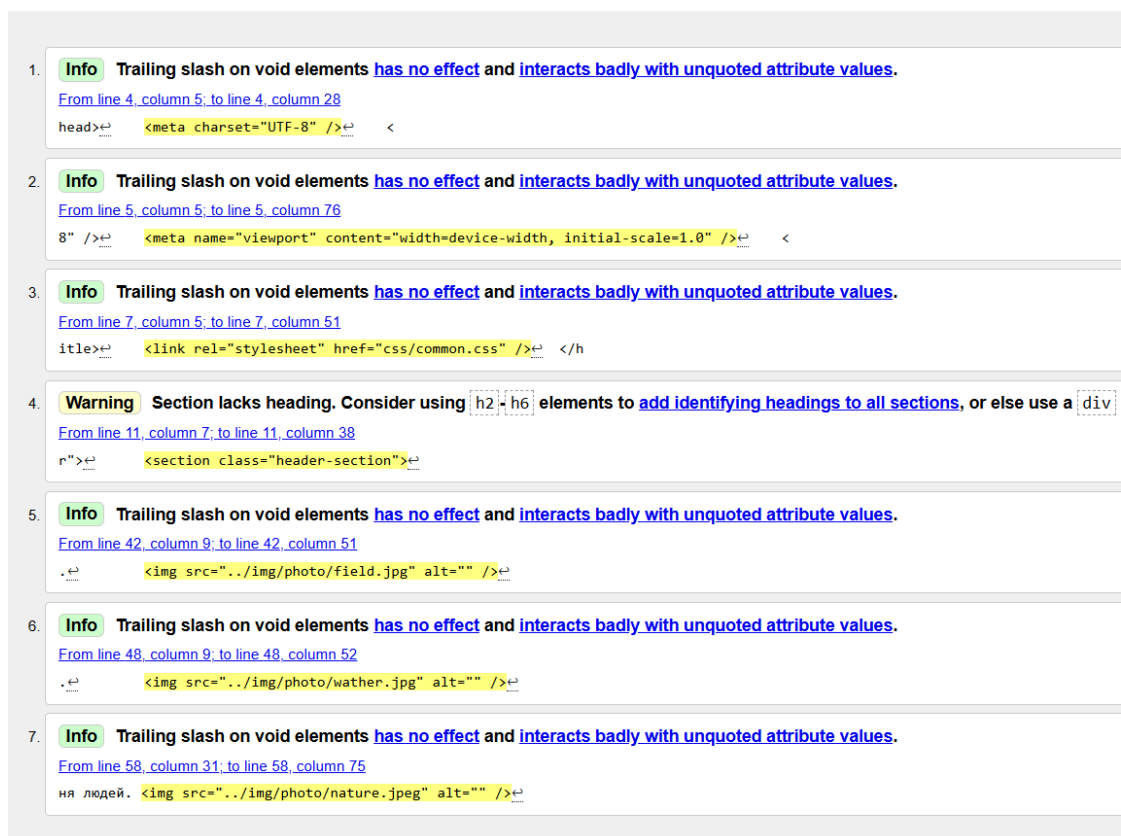


Рисунок 5.2 – Результат тестування розмітки

З рисунка можна побачити, що розмітка написана правильно і мета дані програми будуть успішно використовуватися у рекламі, та просуванні сайту, тому що, правильно написана розмітка є більш пріоритетною у використанні СММ роботів та подальшому відображенні сайту в топі пошуку.

5.2 Функціональні можливості

Після того як користувач зайдє на сайт він побачить головну сторінку “Info” (рис. 5.3).

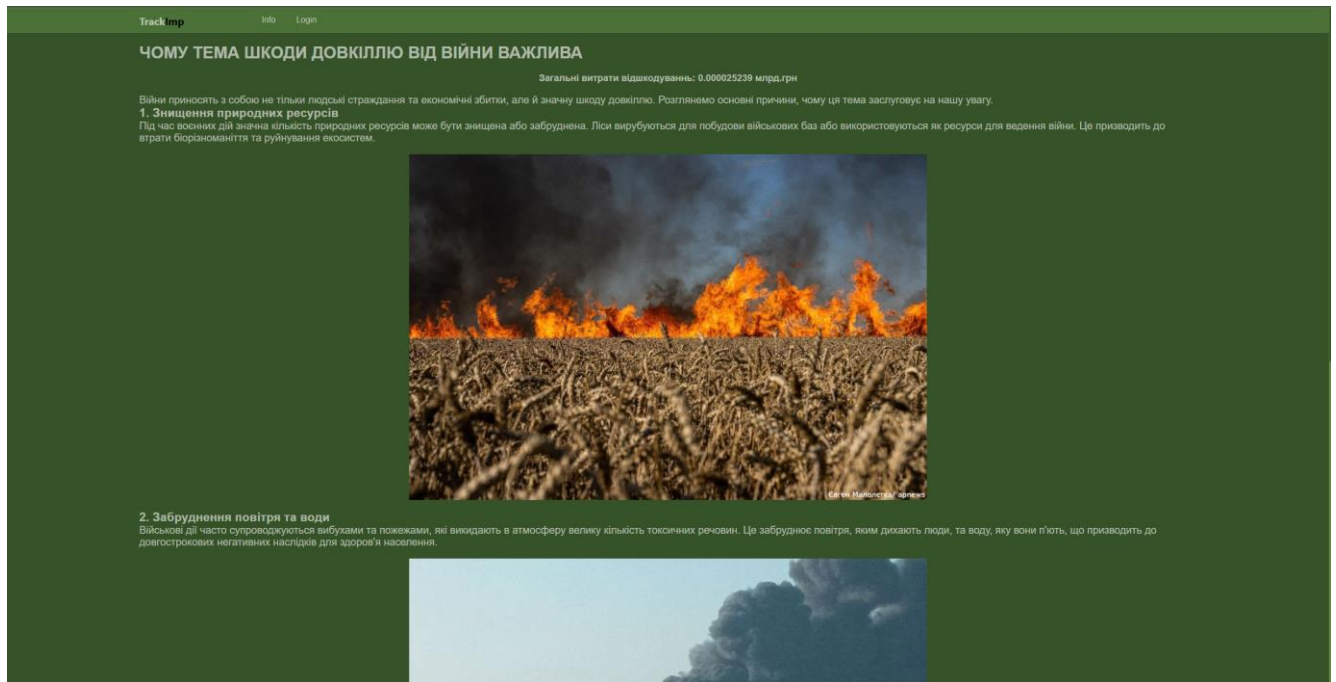


Рисунок 5.3 – Вигляд початкового екрану “Info”

На початковій сторінці, яка називається "Info", звичайний користувач може побачити інформацію щодо забруднення, спричиненого повномасштабною війною, а також загальні витрати на відновлення, які динамічно додаються в залежності від того, як експерти додають випадки забруднення.

Далі користувач має можливість натиснути на кнопку Login, після натискання він побачить форму логіну в застосунок, зайти можуть тільки авторизовані користувачі. На рисунку 5.4. показано форму логіну користувача.

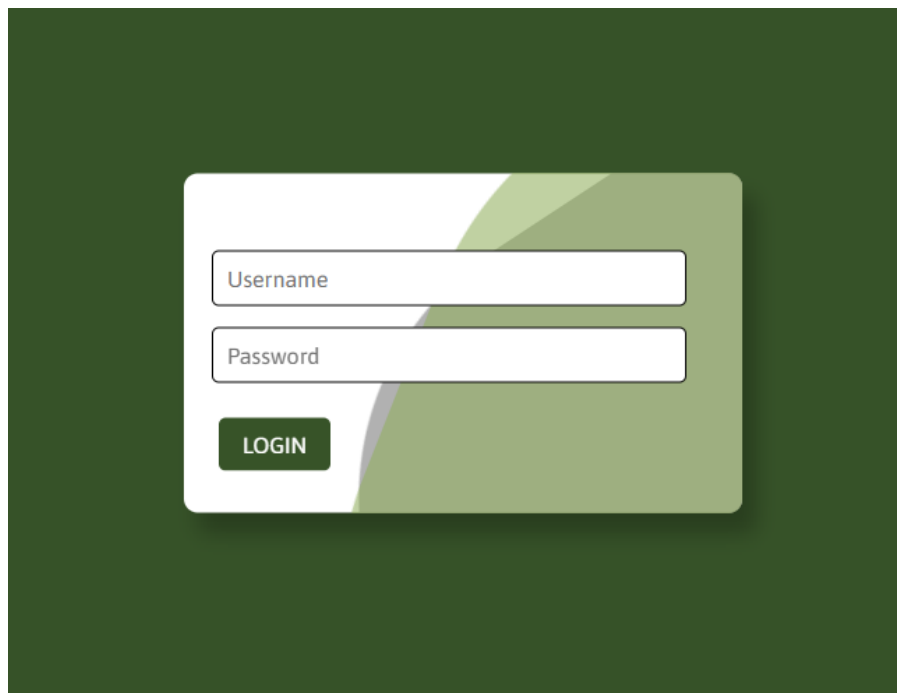


Рисунок 5.4 – Форма логіну

Форма логіну користувача, окремо важливо зазначити що форма має приємну анімацію та валідацію введених даних (рис 5.5 та рис 5.6):

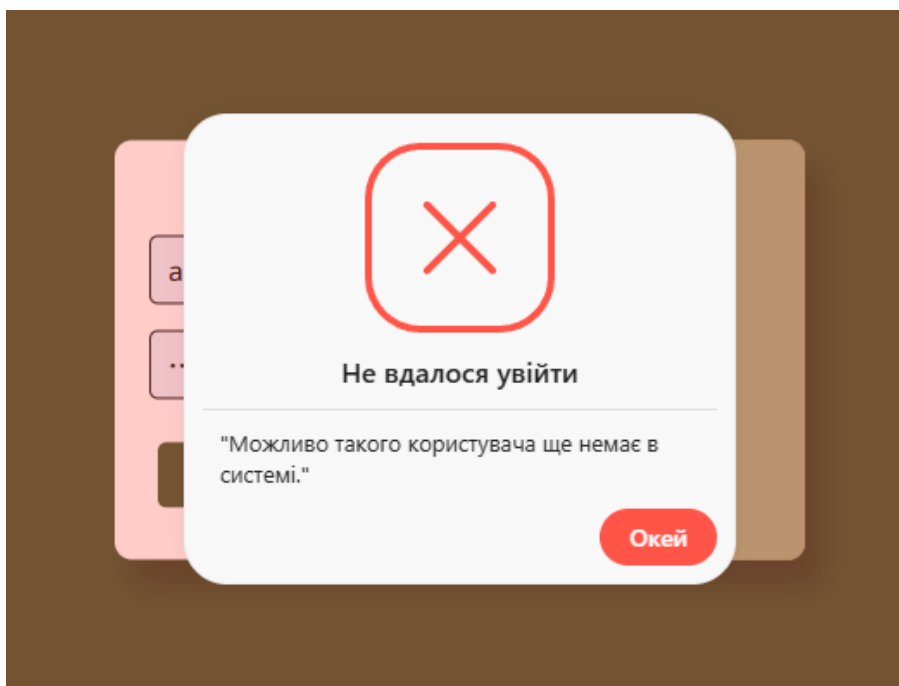


Рисунок 5.5 – Вікно помилки коли користувач увів неправильні дані



Рисунок 5.6 – Вікно помилки коли користувач не заповнив поля

Після введення коректних даних для входу у застосунок, експерт потрапляє на головний дашборд, де він може проводити операції рисунок 5.7.

The dashboard features a header with 'Track Imp', 'Info', 'LogOut', and 'Сторінка Адміністратора'. Below the header, there are sections for file uploads and a main form titled 'Форма для розрахунку шкоди' (Form for damage calculation). The form includes dropdown menus for 'Екологічна ділянка' (Ecological area), 'Область України' (Region of Ukraine), 'Спеціальні території' (Special territories), and 'Забруднююча речовина' (Polluting substance), along with text input fields for 'Площа забрудненої Квм' (Contaminated area in sqm) and 'Дата' (Date), and a 'Розрахувати дані' (Calculate data) button.

Below the form is a table titled 'Історія запитів' (Request history) with the following data:

Екологічна ділянка	Область України	Площа забруднення Квм	Спеціальні території	Забруднююча речовина	Дата	Розмір Шкоди(тис.грн)	
Вода	Автономна республіка Крим	50	Природні території та об'єкти природно-заповідного фонду	Надзвичайно небезпечні	11.05.2024	43.733	Видалити
Вода	Автономна республіка Крим	50	Природні території та об'єкти природно-заповідного фонду	Надзвичайно небезпечні	11.05.2024	43.733	Видалити
Повітря	Закарпатська	100	Округ санітарної (гігієно-санітарної) охорони у лікувально-оздоровчих місцевостях і на території курортів	Дуже небезпечні	15.03.2023	5.18	Видалити
Повітря	Житомирська	100	Охоронні зони навколо об'єктів культурної спадщини	Дуже небезпечні	16.03.2023	5.38	Видалити
Земля	Донецька	150	Охоронні зони навколо об'єктів культурної спадщини	Надзвичайно небезпечні	17.03.2023	5.022	Видалити
Повітря	Вінницька область	342	Охоронні зони навколо гідрометеорологічних станцій	Дуже небезпечні	2024-05-20	25.239	Видалити

Рисунок 5.7 – Дашборд адміна

На дашборді ми бачимо увесь наявний функціонал, який присутній у веб-застосунку. Розберемо його поетапно: у верхній секції, де розташований хедер, ми бачимо логотип та кнопку "Інфо", а також кнопку виходу з додатку – "логаут". Також помітно, що ми зайшли у додаток під записом адміністратора, тобто старшого експерта. Далі йде те саме меню для імпорту з Excel, яке складається з

кнопки "Завантажити файл" та "Обрати файл для вставки". Головна частина додатку – це форма для розрахунку шкоди. Після неї розташована історія запитів з наявними кейсами забруднення, де кожен з кейсів має кнопку "Видалити", а також кнопку експорту даних(рис 5.8).

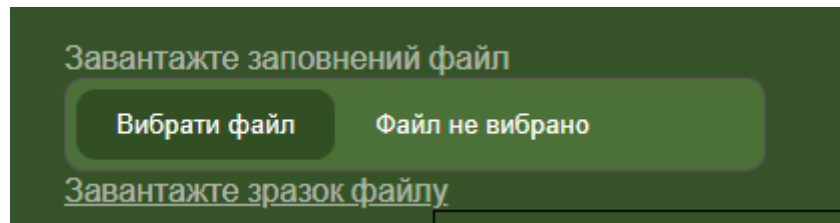


Рисунок 5.8 – Екран вибору

Функція Excel імпорту це найголовніша частина роботи після самої форми, дивлячись на рисунок ми можемо бачити поле в яке користувач заносить файл, який стилізовано під додаток, та посилання на завантаження файлу зображене на рисунку 5.9.

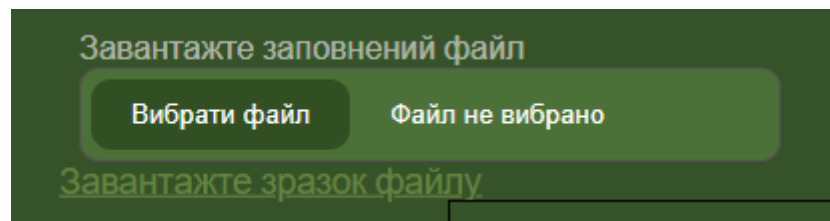


Рисунок 5.9 – Наведення на посилання

При наведенні мишки на посилання ми бачимо, що курсор змінює свій стиль на вказівний палець, а посилання збільшується у розмірах на 20% та змінює свій колір. Все це зроблено для зручності користувача. Всі кнопки та клікабельні елементи у роботі покращені вищеописаним функціоналом, тому далі в роботі не буде акцентуватися на цьому.

Нижче на рисунках 5.10, 5.11 показано вміст файлу який завантажується до директорії ПК шляхом натискання "Завантажте зразок файлу".

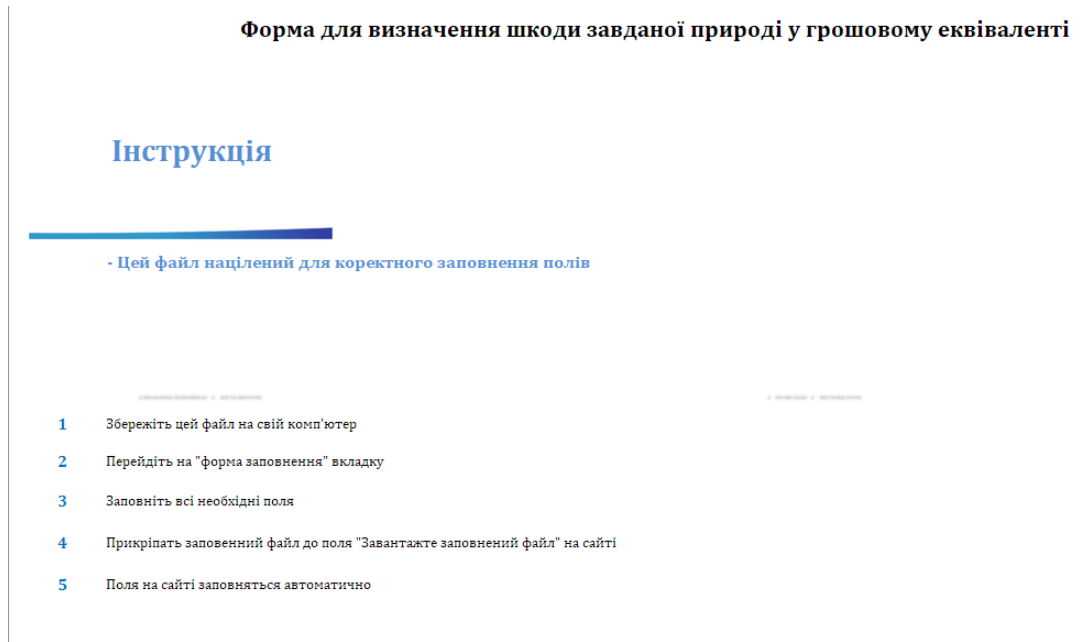


Рисунок 5.10 – Інструкція користувача (користування Excel імпортом)

Обов'язкові поля для заповнення

Обов'язкові поля для заповнення (оберіть значення з випадаючого списку)

Поля для заповнення - заповніть якщо потрібно

ФОРМА

Поля для заповнення постраждалої території					
Вид забруднення	Область	Площа забруднення Кв.м	Вид земельної ділянки	Вид забруднюючої речовини	Дата забруднення (формат ДД.ММ.РР)
Вода	іономна республіка Кр	50	Природні території та об'єкти природно-заповідного фонду	Надзвичайно небезпечні	11.05.2024
Повітря	Закарпатська	100	Охоронні зони навколо гідрометеорологічних станцій	Помірно небезпечні	20.05.2024

Рисунок 5.11 – Форми заповнення випадків забруднення(Excel)

Як можна побачити на рисунку 5.11 заповнено тільки два випадки забруднення, але якщо користувач захоче заповнити більше, то в розроблену функціоналі є така можливість 1<к-сть випадків <= 100, якщо ж експерт захоже заповнити більше випадків, то він може відправити 100 випадків, а потім знову відкрити імпорт. Для зручності користування Excel файлом було розроблено систему випадуючих значень(рис 5.12).

Площа кв.м	Вид земельної ділянки	Вид забруднюючої речовини	Дата забруднення (формат ДД.ММ.РР)
	Охоронні зони навколо гідрометеорологічних станцій	Надзвичайно небезпечні	5/11/2024
<div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> Природні території та об'єкти природно-заповідного фонду Штучно створені об'єкти природно-заповідного фонду Охоронні зони навколо цінних природних об'єктів Охоронні зони навколо об'єктів культурної спадщини Охоронні зони навколо гідрометеорологічних станцій Особливо цінні землі сільськогосподарського призначення Округ санітарної (гігієнічно-санітарної) охорони у лікувально-оздоровчих місцевостях і на території курортів Прибережні захисні смуги уздовж річок, навколо водойм та на островах, заплави малих річок Прибережні захисні смуги уздовж морів, морських заток і лиманів та на островах, Пляжні зони уздовж морів, морських заток і лиманів Водоохоронні зони уздовж річок, морів, навколо озер, водосховищ та інших водойм Охоронні зони наземних, надземних і підземних трубопроводів </div>			

Рисунок 5.12 – Випадки забруднення, які були заповнені через Excel імпорт

Поля вид забруднення, область, вид земельної ділянки, вид забруднюючої речовини повторюють свій зміст з випадваючих списків на самій формі, яка розташовується на сайті.

Після того як експерт заповнив усі випадки, він повертається на сайт, та натискає на кнопку “Вибрати файл”, обирає файл зі своєї файлової директорії і бачить, що його випадки додалися до таблиці рисунок 5.13.

Історія запитів							
Екологічна ділянка	Область України	Площа забруднення кв.м	Спеціальні території	Забруднююча речовина	Дата	Розмір Шкоди(тис.грн)	
Повітря	Житомирська	100	Охоронні зони навколо об'єктів культурної спадщини	Дуже небезпечні	16.03.2023	5.38	Видалити
Земля	Донецька	150	Охоронні зони навколо об'єктів культурної спадщини	Надзвичайно небезпечні	17.03.2023	5.022	Видалити
Повітря	Вінницька область	342	Охоронні зони навколо гідрометеорологічних станцій	Дуже небезпечні	2024-05-20	25.239	Видалити
Вода	Автономна республіка Крим	50	Природні території та об'єкти природно-заповідного фонду	Надзвичайно небезпечні	11.05.2024	43.733	Видалити
Повітря	Закарпатська	100	Охоронні зони навколо гідрометеорологічних станцій	Помірно небезпечні	20.05.2024	4.316	Видалити
Вода	Донецька область	4214	Особливо цінні землі сільськогосподарського призначення	Дуже небезпечні	2024-06-29	881.779	Видалити
Земля	Тернопільська область	12	Захисні, охоронні та інші зони з особливими умовами користування навколо військових та інших оборонних об'єктів	Інші	2024-01-17	0.07	Видалити

Рисунок 5.13 – Випадки забруднення, які були заповнені через Excel імпорт

На рисунку 5.14 зображена форма для розрахунку шкоди та її поля.

Форма для розрахунку шкоди

Екологічна ділянка

Оберіть природне середовище

Область України

Оберіть область

Площа забруднення Кв.м

Спеціальні території

Оберіть територію

Забруднююча речовина

Наведіть для підказки

Дата

дд . мм . рррр

Розрахувати дані

Рисунок 5.14 – Форми заповнення випадку забруднення (сайт)

Дивлячись на рисунок можна побачити поля, які беруть участь у формулі. Для зручності користування формною, та власного розуміння, які змінні використовуються у формулі, було створено допоміжну кнопку “Як користуватися формною”, при натисканні на кнопку ми бачимо допоміжне меню рисунок 5.15.

Форма	Формула розрахунку
Вид забруднення <input type="text" value="Земля"/>	$РШ = ВЗ * НГОЗ * ПЗ * СТ * ЗР$ РШ - розмір шкоди, грн
Область України <input type="text" value="Дніпропетровська область"/>	ВЗ - вид забруднення(екосистема, яка постраждала)
Площа забруднення Кв.м <input type="text" value="1000"/>	НГОЗ - нормова оцінка ділянки, яка зазнала забруднення(область)
Спеціальні території <input type="text" value="Охоронні зони навколо гідрометеорологічних стг"/>	ПЗ - площа забруднення кв.м
Рівень небезпеки <input type="text" value="Помірно небезпечні"/>	СТ - підвид територіх, яка зазнала забруднення
Дата <input type="text" value="07 . 05 . 2024"/>	ЗР(РН) - рівень небезбечності забруднюючої речовини
<input type="button" value="Розрахувати дані"/>	

Рисунок 5.15 – Інструкція до форми, та власне формула

Майже кожне поле форми це випадаючий список, користувач бачить тільки власне значення, а не фактичне значення коефіцієнтів, так як це експерти, але для наглядності в роботі ми покажемо їх значення на рисунках 5.16 (Територія з особливим режимом використання земель), та 5.17 (Адміністративно-територіальна одиниця), 5.18 (КОЕФІЦІЄНТИ небезпечності забруднюючих речовин). З рисунку видно формулу $РШ=ВЗНГОЗПЗСТЗР$, де ВЗ - вид забруднення (екосистема, яка постраждала), НГОЗ - нормова оцінка ділянки, яка зазнала забруднення (область), ПЗ - площа забруднення кв.м, СТ - підвид території, яка зазнала забруднення, ЗР (РН) - рівень небезпечності забруднюючої речовини.

Формула також є універсальною у використанні, оскільки може бути адаптована для розрахунку шкоди як для водних ресурсів, так і для повітряних. Це робить її дуже корисною в різних екологічних ситуаціях. Простота впровадження та використання формули дозволяє застосовувати її навіть тим користувачам, які не є фахівцями в галузі екології. Кожна змінна має конкретне значення, що робить процес введення даних і отримання результатів простим і зрозумілим.

Адаптивність і гнучкість формули є ще однією перевагою. Додаткові змінні дозволяють враховувати різні фактори та деталі, що підвищує точність розрахунків. У порівнянні з іншими існуючими формулами, які можуть бути більш спеціалізованими або складними, ця формула надає простий і ефективний спосіб оцінки екологічної шкоди. Інші методики можуть вимагати глибокого знання екологічних процесів або спеціалізованих даних, тоді як ця формула дозволяє швидко і ефективно провести оцінку, використовуючи доступні дані.

Загалом, формула є зручним інструментом для оцінки екологічної шкоди, оскільки поєднує в собі простоту, універсальність і точність. Вона дозволяє швидко реагувати на екологічні інциденти, проводити необхідні розрахунки та приймати обґрунтовані рішення щодо подальших дій.

Майже кожне поле форми це випадуючий список, користувач бачить тільки власне значення, а не фактичне значення коефіцієнтів, так як це експерти, але для наглядності в роботі ми покажемо їх значення на рисунках 5.16 (Територія з особливим режимом використання земель), 5.17 (Адміністративно-територіальна одиниця), та 5.18 (КОЕФІЦІЄНТИ небезпечності забруднюючих речовин).

Загалом, використання наступних рисунків та відповідних коефіцієнтів дозволяє підвищити точність і надійність розрахунків екологічної шкоди, роблячи процес оцінки більш прозорим і зрозумілим для всіх зацікавлених сторін.

Територія з особливим режимом використання земель	Значення коефіцієнта
Природні території та об'єкти природно-заповідного фонду	10
Штучно створені об'єкти природно-заповідного фонду	5
Охоронні зони навколо цінних природних об'єктів	3,5
Охоронні зони навколо об'єктів культурної спадщини	4
Охоронні зони навколо гідрометеорологічних станцій	2
Особливо цінні землі сільськогосподарського призначення	2,5
Округ санітарної (гірничо-санітарної) охорони у лікувально-оздоровчих місцевостях і на території курортів	4
Прибережні захисні смуги уздовж річок, навколо водойм та на островах, заплави малих річок	6
Прибережні захисні смуги уздовж морів, морських заток і лиманів та на островах,	5
Пляжні зони уздовж морів, морських заток і лиманів	6
Водоохоронні зони уздовж річок, морів, навколо озер, водосховищ та інших водойм	4
Охоронні зони наземних, надземних і підземних трубопроводів	2,5
Охоронні зони уздовж повітряних і підземних кабельних ліній зв'язку, а також навколо випромінювальних споруд телерадіостанцій та радіорелейних ліній	1,5
Охоронні зони уздовж повітряних і підземних кабельних ліній електропередачі	1,5
Захисні, охоронні та інші зони з особливими умовами користування навколо військових та інших оборонних об'єктів	1,5
Зони відчуження та безумовного (обов'язкового) відселення, що зазнали радіоактивного забруднення внаслідок Чорнобильської катастрофи	2
Зона санітарної охорони навколо об'єктів, у яких є підземні та відкриті джерела водопостачання, водозабірні та водоочисні споруди, водоводи, об'єкти оздоровчого призначення	6
Санітарно-захисні зони навколо об'єктів, які є джерелом виділення шкідливих речовин, запахів, підвищеного рівня шуму, вібрації, ультразвукових і електромагнітних хвиль, електронних полів, іонізуючих випромінювань	1,5
Прикордонна смуга уздовж державного кордону України	3
Сільськогосподарські угіддя, включені в установленому порядку до складу екомережі	3
Землі, зарезервовані для заповідання	4
Інші території з особливим режимом використання земель	4

Рисунок 5.16 – Коефіцієнти,
що застосовуються для врахування природоохоронної цінності, наявності
обмежень (обтяжень), які зумовлюють особливий режим використання земельних
ділянок*[1]

Автономна Республіка Крим	74585
Області:	
Вінницька	63018
Волинська	55480
Дніпропетровська	64830
Донецька	66854
Житомирська	42717
Закарпатська	40740
Запорізька	76797
Івано-Франківська	41549
Київська	58005
Кіровоградська	52279
Луганська	55756
Львівська	56267
Миколаївська	64591
Одеська	58732
Полтавська	74010
Рівненська	72865
Сумська	66914
Тернопільська	67838
Харківська	60553
Херсонська	64605
Хмельницька	86603
Черкаська	98207
Чернівецька	70138
Чернігівська	45458
м. Київ	118825
м. Севастополь та адміністративно-територіальні одиниці, які входять до його складу	240639

Рисунок 5.17 – Приблизна оцінка гектару землі у областях України[2]

Варто зазначити, що величина визначена у гектарах, для використання цих даних було проведено, перерахування у метри квадратні, та помножено на спеціальних коефіцієнт, який застосовується під час воєнного часу – 300 одиниць.

Група небезпечності	Ступінь небезпеки	Перелік забруднюючих речовин (показників вимірювань), що відповідають групі небезпечності*	Кн	
I	Надзвичайно небезпечні (ГДК/ОДК < 0,2 мг/кг)	Бенз-а-пірен Кадмій** Миш'як Нафта Нафтопродукти*** Ртуть	Селен Свинець Стирол Фенол Фтор Цинк	4,0
II	Дуже небезпечні (ГДК/ОДК 0,2-0,5 мг/кг)	Бензол Бор Кобальт Ксилоли Мідь Молибден	Нікель Сірководень Сурма Толуол Хром	3,0
III	Помірно небезпечні (ГДК/ОДК > 0,5 мг/кг)	Аніонні поверхневоактивні речовини (АПАР) Ацетальдегід Барій Ванадій Вольфрам Марганець Нітрати	Стронцій Сульфати Формальдегід	2,5
IV	Інші (рівні ГДК/ОДК не встановлені)	Амоній Хлориди		1,5

Рисунок 5.18 – Коефіцієнти небезпечності забруднюючих речовин[3]

* Перелік забруднюючих речовин (показників вимірювань), що відповідають групі небезпечності, не є вичерпним. Якщо забруднююча речовина відсутня у переліку, групу її небезпечності визначають за величиною ГДК або ОДК.

** Більшість назв забруднюючих речовин (показників вимірювань) подані за назвами хімічних елементів.

*** Терміни нафта і нафтопродукти подані згідно з ДСТУ 3437-96 "Нафтопродукти. Терміни та визначення".

В нашій роботі наведений такий перелік речовин він з'являється при наведенні курсора на поле "Забруднююча речовина" рисунок 5.19.

Водоохоронні зони уздовж річок, морів, навколо

Забруднююча речовина

Надзвичайно небезпечні

Дата

дд . мм . рррр

Розрахувати дані

Надзвичайно небезпечні речовини:

- Ртуть (Hg)
- Органічні ртутні сполуки
- Плутоній (Pu)
- Амонійна перхлорат
- Сарин (GB)
- Сірчана кислота (H2SO4)
- Фосген (CG)
- Арсенік (As)
- Ціаніди
- Радіоактивні ізотопи

Дуже небезпечні речовини:

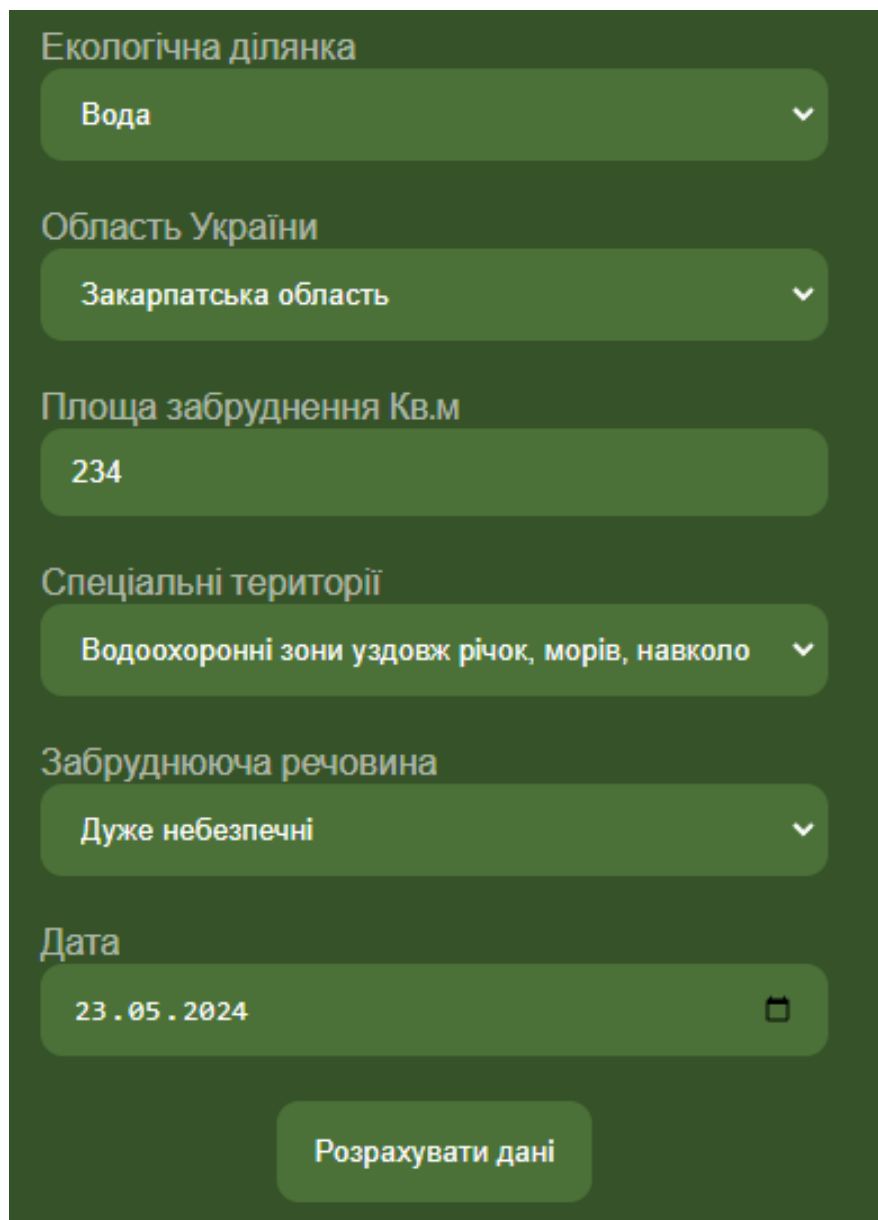
- Дихлорвінілбензол (DCB)
- Селян (HD)
- Нітрогліцерин
- Монооксид вуглецю (CO)
- Диоксини
- Хлор (Cl2)
- Фосфор (P)
- Мустангаз (HN3)
- Ацетонитрил
- Фосфорний трихлорид (PCl3)

Помірно небезпечні речовини:

- Толуол
- Бензол
- Сірка діоксид (SO2)
- Амоніак (NH3)
- Формальдегід
- Бензпірен
- Оксиди азоту (NOx)
- Сірчані діоксиди (SOx)
- Хлористий водень (HCl)
- Фториди

Рисунок 5.19 – Перелік небезпечних речовин посорттованих по групах небезпечності

На рисунку 5.20 зображено зовнішній вигляд форми, її поля. Варто зауважити що форма, при перегляді здається малою та некомплексною, але це не так насправді формула та її розрахунок займають велику частину програмного коду.



Екологічна ділянка

Вода

Область України

Закарпатська область

Площа забруднення Кв.м

234

Спеціальні території

Водоохоронні зони уздовж річок, морів, навколо

Забруднююча речовина

Дуже небезпечні

Дата

23 . 05 . 2024

Розрахувати дані

Рисунок 5.20 – Заповнена форма

Після заповнення форми та натискання кнопки “Розрахувати дані” з’являється випадаюче вікно з розрахованою сумою рисунок 5.21.

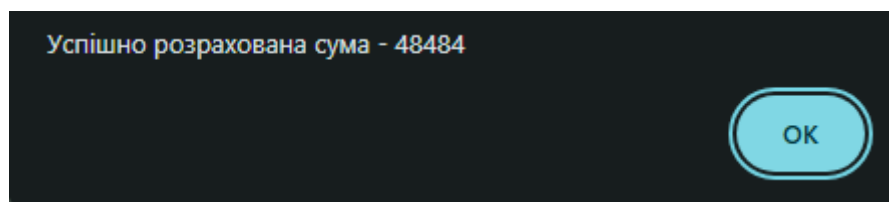


Рисунок 5.21 – Випадаюче вікно з розрахованою шкодою

Коли буде натиснута будь-яка клавіша або кнопка "ОК", запис автоматично буде додано до Історії запитів. Сам функціонал історії записів включає в себе історію усіх створених користувачем записів, функціонал кнопки видалення, та функціонал експорту усіх записів. Далі будуть розписані можливості та функціонал таблиці "Історія запитів", дані додаються до цієї таблиці коли експерт, або молодший експерт заповнюють форму та натискають кнопку "Розрахувати дані", та коли експерт користується функціоналом Excel імпорту, також у експерта (адміна) є можливість видалити окремий запис з БД. Кнопка "export data" є доступною як для експерта так і для молодшого експерта. Файл Excel завантажується на ПК оператора та має такий вигляд рисунок 5.22.

ID	pollutionType	pollutionRegion	pollutionArea	pollutionTerritory	pollutionSubstance	pollutionDated	pollutionExpense
3VCTDUQjgWmkAS4QSMho1	Повітря	Закарпатська	100	Округ санітарної (гірично-санітарної) охорони у лікувально-оздоровчих місцевостях і на території курортів	Дуже небезпечні	15.03.2023	Тра.18
tdTEbbTgut_rN_SJ-Kr7X	Повітря	Житомирська	100	Охоронні зони навколо об'єктів культурної спадщини	Дуже небезпечні	16.03.2023	Тра.38
uTwmvLUeEjntnR6GDu0k1	Земля	Донецька	150	Охоронні зони навколо спадщини	Надзвичайно небезпечні	17.03.2023	5.022
L0C_zn7q2fS2pFxmmlx-M	Повітря	Вінницька область	342	Охоронні зони навколо гідрометеорологічних станцій	Дуже небезпечні	20.05.2024	25.239
sfk9a5-cnXzZaM13qbUzy	Вода	Автономна республіка Крим	50	Природні території та об'єкти природно-заповідного фонду	Надзвичайно небезпечні	11.05.2024	43.733
I9-8Dxabrc7_Oy8rOGJgs	Повітря	Закарпатська	100	Охоронні зони навколо гідрометеорологічних станцій	Помірно небезпечні	20.05.2024	4.316
eJррOK1yfpo_pTRQKF2K	Вода	Закарпатська область	234	Водоохоронні зони уздовж річок, морів, навколо озер, водосховищ та інших водойм	Дуже небезпечні	23.05.2024	48.484

Рисунок 5.22 – Вміст файлу експортованої БД у Excel

Вище описаний увесь функціонал програми, експерти також можуть вийти з додатку і програма перепитає чи впевнені вони що хочуть вийти меню показане на рисунку 5.23.



Рисунок 5.23 – Спливаюче вікно при виході

У розробленій програмі також наявна градація експертів вище був описаний функціонал старшому експерту меню молодшого експерта показане на рисунку 5.24.

Екологічна ділянка	Область України	Площа забруднення Кв.м	Спеціальні території	Забруднювач речовина	Дата	Розмір Шкоди(тис.грн)
Повітря	Закарпатська	100	Округ санітарної (прічно-санітарної) охорони у лікувально-одорожніх місцевостях і на території курортів	Дуже небезпечна	15.03.2023	5.18
Повітря	Житомирська	100	Охоронні зони навколо об'єктів культурної спадщини	Дуже небезпечна	16.03.2023	5.38
Земля	Донецька	150	Охоронні зони навколо об'єктів культурної спадщини	Надзвичайно небезпечна	17.03.2023	5.022
Повітря	Вінницька область	342	Охоронні зони навколо гідрометеорологічних станцій	Дуже небезпечна	2024-05-20	25.239
Вода	Автономна республіка Крим	50	Природні території та об'єкти природно-заповідного фонду	Надзвичайно небезпечна	11.05.2024	43.733
Повітря	Закарпатська	100	Охоронні зони навколо гідрометеорологічних станцій	Помірно небезпечна	20.05.2024	4.316
Вода	Закарпатська область	234	Водозахронні зони уздовж річок, морів, навколо озер, водосховищ та інших водойм	Дуже небезпечна	2024-05-23	48.484

Рисунок 5.24 – Дашборд молодшого експерта

Молодшому експерту недоступна функція Excel імпорту, функція видалення записів із БД.

Отже, що розроблена веб-система виконує поставлені задачі та зручна у користуванні. Простота та легкість є великим плюсом розробленого додатку.

ВИСНОВКИ

В процесі написання дипломної роботи була розроблена веб-система для оцінки шкоди довкіллю внаслідок повномасштабної війни. В рамках дослідження було проведено аналіз схожих існуючих додатків, вивчено відповідну літературу та нормативно-правову базу України, що дозволило спроектувати та розробити веб-систему під назвою “TrackImp”.

При створенні додатку були використані такі інструменти, як середовище розробки VsCode, мова програмування JavaScript та середовище NodeJS для взаємодії з сервером, тестування та завантаження бібліотек. Для фронтенду використовувалася мова розмітки HTML та стилізація на базі CSS. Також було побудовано власну базу даних на основі JS LocalStorage. Код проекту був розміщений у репозиторії на GitHub.

Також, було опрацьовано літературу, курси в тому числі курс на платформі “Prometheus”[5], чинні закони України[1],[2],[3]. Виведено формулу, яка здатна прорахувати шкоду завдану різним природним середовищам її перевага у тому, що вона не перевантажена багатьма змінними, та зрозуміла користувачу-експерту навіть без додаткових консультацій. В дипломній записці надана інструкція користувача, для коректного використання веб-додатку.

Було спроектовано та розроблено архітектурну та програмну частини веб-системи. Додаток був наповнений необхідним контентом і включив усі функціональні модулі, що забезпечують повноцінну роботу. Після цього проводилося ретельне тестування та налагодження, щоб гарантувати стабільну та ефективну роботу програми. Додаток був протестований на практиці для оцінки його реальної ефективності та корисності.

Війна – це завжди трагедія, і я щиро бажаю, щоб у світі не було приводів для воєнних конфліктів, а також для знищення природи. Проте, така є природа людини. Навіть після закінчення війни залишаться ті, хто викидає папірці повз урну,

забруднює ліси та різними способами шкодить навколишньому середовищу. Покарання за такі дії повинне бути більш суворим і явним. Система, розроблена в додатку "TrackImp", може бути розширена та адаптована для вирішення цих питань, допомагаючи контролювати та запобігати екологічним правопорушенням.

Завдяки веб-системі "TrackImp" користувачі можуть оцінювати масштаби екологічної шкоди, спричиненої війною, та активно сприяти відновленню навколишнього середовища. Цей додаток є важливим кроком у напрямку екологічної свідомості та відповідальності, підкреслюючи необхідність дбайливого ставлення до природи задля збереження нашого спільного майбутнього.

У фіналі хочеться продемонструвати у яких напрямках додаток може використовуватися, нижче перераховані далеко не всі варіанти, це показує актуальність теми дипломної роботи, також тренд людства, та зацікавленість у збереженні природи. При розробці цього додатку було продумано безліч варіантів як він може використовуватися у сучасному суспільстві, ось кілька прикладів:

Природні катастрофи: додаток може використовуватися для оцінки екологічної шкоди після природних катастроф, таких як землетруси, урагани, повені або лісові пожежі. За допомогою "TrackImp" можна швидко збирати дані про постраждалі території, визначати рівень забруднення водних і повітряних ресурсів, а також оцінювати масштаби збитків.

Промислові аварії: додаток може стати незамінним інструментом у випадку промислових аварій, таких як витіки хімічних речовин або нафти. "TrackImp" дозволяє ефективно оцінювати рівень забруднення, площу ураженої території та рівень небезпеки забруднюючих речовин, що допоможе вжити необхідних заходів для ліквідації наслідків аварії.

Моніторинг якості повітря та води у містах: "TrackImp" може використовуватися муніципальними органами для моніторингу якості повітря та води у містах. За допомогою додатку можна збирати дані про рівень забруднення,

визначати найбільш проблемні ділянки та приймати рішення щодо поліпшення екологічної ситуації в містах.

Оцінка впливу будівельних робіт: при здійсненні масштабних будівельних робіт, таких як будівництво доріг, мостів або промислових об'єктів, "TrackImp" може використовуватися для оцінки впливу на навколишнє середовище. Додаток дозволить визначати площу ураженої території, рівень забруднення та допоможе вживати заходів для мінімізації негативного впливу на екосистему.

Екологічні дослідження та наукові проекти: "TrackImp" може бути корисним для вчених та дослідників, які займаються екологічними дослідженнями. Додаток дозволяє збирати та аналізувати дані про стан екосистем, рівень забруднення та інші параметри, що допоможе в проведенні наукових досліджень та розробці стратегій збереження довкілля.

Екологічна освіта та просвіта: додаток може використовуватися у навчальних закладах для проведення екологічних уроків та підвищення екологічної свідомості серед молоді. "TrackImp" дозволяє наочно демонструвати вплив людської діяльності на навколишнє середовище та навчати студентів методам збереження природи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про затвердження Методики визначення розміру шкоди, заподіяної внаслідок самовільного зайняття земельних ділянок, використання земельних ділянок не за цільовим призначенням, псування земель, порушення режиму, нормативів і правил їх використання {Назва Постанови із змінами, внесеними згідно з Постановою КМ № 131 від 16.02.2022}. URL: <https://zakon.rada.gov.ua/laws/show/963-2007-п#n66>.
2. Про затвердження Методики визначення розмірів шкоди, зумовленої забрудненням і засміченням земельних ресурсів через порушення природоохоронного законодавства {Із змінами, внесеними згідно з Наказом Міністерства охорони навколишнього природного середовища № 149 від 04.04.2007 Наказом Міністерства захисту довкілля та природних ресурсів № 241 від 04.11.2020}. URL: <https://zakon.rada.gov.ua/laws/show/z0285-98#Text>
3. Про затвердження Методики нормативної грошової оцінки земель сільськогосподарського призначення {Із змінами, внесеними згідно з Постановами КМ № 637 від 23.08.2017 № 105 від 07.02.2018 № 477 від 05.06.2019}. URL: <https://zakon.rada.gov.ua/laws/show/831-2016-п#n30>.
4. Встановлення та налаштування Visual Studio Code. URL: <https://cases.media/article/vstanovlennya-ta-nalashtuvannya-visual-studio-code>.
5. Оцінка шкоди довкіллю від російської агресії. <https://apps.prometheus.org.ua/>. URL: https://apps.prometheus.org.ua/learning/course/course-v1:ANTS+EDA101+2023_T1/home.
6. JavaScript Learning. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
7. Notiflix. URL: <https://www.npmjs.com/package/notiflix>.
8. Побудова незалежних застосунків з використанням мікрофронтендів. Переваги та недоліки різних рішень. URL: <https://dou.ua/forums/topic/43354/>.
9. ЕкоЗагроза. URL: <https://ecozagroza.gov.ua/>.
10. Завдання Штабу. URL: <https://shtab.gov.ua/>.
11. Automated Testing. URL: <https://saucelabs.com/resources/topic-hub/automated-testing>.
12. Nu Html Checker. URL <https://validator.w3.org/>.

ДОДАТОК А

Веб-система для оцінювання шкоди довкіллю внаслідок повномасштабної війни

Текст програми

УКР.НТУУ”КПІ імені Ігоря Сікорського”

Аркушів 36

Київ 2024

- ImpotExcel,
- index,
- insertDataToTables,
- login
- logout
- pollutionCalculation
- delete
- getALL
- write
- writeALL
- index.html
- login.html

Реалізація логіки взаємодії модулів використаних в програмі

```
import { login } from './login';
import { logout } from './logout';
import { readExel } from './importExel';
import { getPolValue } from './pollutionCalculation';

import { layout } from './layout';
import { insertDataToTable } from './insertDataToTable';
import { refs } from './refs';
import exportFromJSON from 'export-from-json';
import { getAll } from '../db/getAll';

try {
  refs.loginForm.addEventListener('submit', login);
  refs.loginText.addEventListener('click', logout);
} catch (error) {
```

```
console.error('ПОМИЛКА!!', error);
}

try {
  const exportD = document.querySelector('.export-data');
  exportD.addEventListener('click', download);
  function download(params) {
    const fileName = 'exported';
    const data = JSON.parse(localStorage.getItem('pollutions'));
    // const fields = [
    //   'Екологічна ділянка',
    //   'Область',
    //   'Площа забруднення Кв.м',
    //   'Вид земельної ділянки',
    //   'Вид забруднюючої речовини',
    //   'Дата забруднення',
    // ];
    const exportType = 'xls';

    exportFromJSON({ data, fileName, exportType });
  }
} catch (error) {
  console.log('Error in export data button', error);
}

try {
  const pollutionForm = document.querySelector('.pollution-form');

  refs.exelInput.addEventListener('change', readExel);
  pollutionForm.addEventListener('submit', getPolValue);
```

```

} catch (error) {
  console.log('ПОМИЛКА!!', error);
}

window.addEventListener('DOMContentLoaded', () => {
  // запис загальної суми відшкодувань
  try {
    var EXPENSESlocal = localStorage.getItem('expences');
    if (EXPENSESlocal) {
      refs.expensesText.textContent = `Загальні витрати відшкодувань: ${
        Math.floor(EXPENSESlocal) / 1000000000
      } млрд.грн`;
    }
  } catch (error) {
    console.error('ПОМИЛКА!!', error);
  }
  // повторний login користувачів

  try {
    console.log(refs);
    if (JSON.parse(localStorage.getItem('isAdminLogin'))) {
      refs.loginText.textContent = 'LogOut';
      refs.roleLabel.textContent = 'Сторінка Адміністратора';
      refs.loginForm.classList.add('hide-elem');
      refs.hideMainContentContainer.classList.remove('hide-elem');
      refs.hideExelContainer.classList.remove('hide-elem');
      insertDataToTable('admin');
      // refs.hideTableButtonsColumn.classList.remove('hide-elem');

      // refs.mother.innerHTML = "";
    }
  }
}

```

```

// refs.loginText.dataset.login = 'in';
console.log('welcome administrator');
// refs.mother.innerHTML = layout.adminPage;
// insertDataToTable('admin');
} else if (JSON.parse(localStorage.getItem('isExpertLogin'))) {
  localStorage.setItem('isExpertLogin', true);
  refs.loginText.textContent = 'LogOut';
  refs.roleLabel.textContent = 'Сторінка Експерта';
  // refs.mother.innerHTML = "";
  refs.loginForm.classList.add('hide-elem');
  refs.hideMainContentContainer.classList.remove('hide-elem');
  // refs.loginText.dataset.login = 'in';
  console.log('welcome expert');
  // refs.mother.innerHTML = layout.expertPage;
  insertDataToTable('expert');
}
} catch (error) {
  console.error('ПОМИЛКА!!', error);
}
// додаю обробники подій та обробку помилок
});

import { Notify } from 'notiflix';

import { nanoid } from 'nanoid';
import { pollRefs } from './pollutionFormElem';
import { add } from './db/write';

// console.log(lib.queryAll('books'));

```

```
let EXPENSES = 1;

export function getPolValue(evt) {
  evt.preventDefault();
  const formElem = evt.target.elements;
  let formElemArray = [];

  // validation

  for (const item of formElem) {
    // console.log(item.value);
    if (!item.value && item.nodeName !== 'BUTTON') {
      Notify.failure('Заповніть всі поля!');
      return;
    }
    if (item.nodeName === 'SELECT') {
      formElemArray.push(item.selectedOptions[0].label);
    } else if (item.nodeName !== 'BUTTON') {
      formElemArray.push(item.value);
    }

    if (item.name === 'date' || !item.name) {
      continue;
    }
    console.log(item.value);
    EXPENSES *= Number(item.value);
  }

  const typeOfPII = formElemArray[0];
```

```
console.log(formElemArray);
console.log(typeOfPII);

switch (typeOfPII) {
  case 'Вода':
    EXPENSES /= 300;
    break;
  case 'Повітря':
    EXPENSES /= 300;
    break;
  case 'Земля':
    EXPENSES /= 300;
    break;
  default:
    break;
}

// storage.save(ex);
var EXPENSESlocal = localStorage.getItem('expences');
alert(`Успішно розрахована сума - ${Math.floor(EXPENSES)}`);

var transferToJSON = {
  ID: nanoid(),
  pollutionType: formElemArray[0],
  pollutionRegion: formElemArray[1],
  pollutionArea: formElemArray[2],
  pollutionTerritory: formElemArray[3],
  pollutionSubstance: formElemArray[4],
  pollutionDated: formElemArray[5],
  pollutionExpense: Math.floor(EXPENSES) / 1000,
```

```
    comment: formElemArray.pollutionType,
  };

  console.log(transferToJSON);

  add(transferToJSON);

  // db.set('Obj1', transferToMongo);

  // console.log(transferToMongo);

  // console.log(EXPENSESlocal);
  if (EXPENSESlocal) {
    localStorage.setItem('expences', Number(EXPENSESlocal) + Number(EXPENSES));
  } else {
    localStorage.setItem('expences', EXPENSES);
  }
  // alert(`Розмір шкоди у гривнях зiставляє - ${EXPENSES}`);
  // refresh page
  // location.reload(true);
  // localStorage.setItem('expences', EXPENSES);
  //calculation
}

import { insertDataToTable } from './insertDataToTable';
import { layout } from './layout';
import { users } from './loginData';
import { refs } from './refs';
import { Report } from 'notiflix';
import { insertDataToTable } from './insertDataToTable';
```

```
// refs.loginForm.addEventListener('submit', login);

export function login(evt) {
  evt.preventDefault();

  let login = evt.target.elements.login.value;
  let password = evt.target.elements.password.value;

  if (password && login) {
    const loginPased = users.find(
      obj => login === obj.username && password === obj.password
    );

    if (loginPased) {
      const role = loginPased.role;
      console.log(refs.loginText.dataset);

      switch (role) {
        case 'administrator':
          console.log(refs);
          localStorage.setItem('isAdminLogin', true);
          refs.loginText.textContent = 'LogOut';
          refs.roleLabel.textContent = 'Сторінка Адміністратора';
          refs.loginForm.classList.add('hide-elem');
          refs.hideMainContentContainer.classList.remove('hide-elem');
          refs.hideExelContainer.classList.remove('hide-elem');
          insertDataToTable('admin');
          // refs.hideTableButtonsColumn.classList.remove('hide-elem');

          // refs.mother.innerHTML = "";

```

```

// refs.loginText.dataset.login = 'in';
console.log('welcome administrator');
// refs.mother.innerHTML = layout.adminPage;
insertDataToTable('admin');

return;
case 'expert':
    localStorage.setItem('isExpertLogin', true);
    refs.loginText.textContent = 'LogOut';
    refs.roleLabel.textContent = 'Сторінка Експерта';
    // refs.mother.innerHTML = "";
    refs.loginForm.classList.add('hide-elem');
    refs.hideMainContentContainer.classList.remove('hide-elem');
    // refs.loginText.dataset.login = 'in';
    console.log('welcome expert');
    // refs.mother.innerHTML = layout.expertPage;
    insertDataToTable('expert');

    return;
case 'ecologist':
    console.log('welcome ecologist');

    return;
default:
    return;
}
}
Report.init({
    failure: {
        backOverlayColor: 'rgba(255,85,73,0.3)',
    },
},

```

```

});

Report.failure(
  'Не вдалося увійти',
  '"Можливо такого користувача ще немає в системі."<br/><br/>',
  'Окей'
);
return;
}
alert('Заповінь поля!');
}

```

```

import readXlsxFile from 'read-excel-file';
import { nanoid } from 'nanoid';
import { add } from '../db/write';

export function readExel(e) {
  let fromExelArray = [];
  let file = e.target.files[0];
  readXlsxFile(file, { sheet: 'map table' }).then(row => {
    console.log(row);
    for (let index = 1; index < row.length; index++) {
      const elements = row[index];
      let expense = 1;
      console.log(expense);

      switch (elements[0].trim()) {
        case 'Вода':
          expense *= 10 / 300;
          break;

```

```
case 'Повітря':  
    expense *= 5 / 300;  
    break;  
case 'Земля':  
    expense *= 1.5 / 300;  
    break;  
  
default:  
    break;  
}  
console.log(expense);  
  
switch (elements[1].trim()) {  
    case 'Автономна республіка Крим':  
        expense *= 656;  
        break;  
    case 'Вінницька':  
        expense *= 738;  
        break;  
    case 'Волинська':  
        expense *= 483;  
        break;  
    case 'Дніпропетровська':  
        expense *= 748;  
        break;  
    case 'Донецька':  
        expense *= 837;  
        break;  
    case 'Житомирська':  
        expense *= 538;
```

```
break;
case 'Закарпатська':
    expense *= 518;
    break;
case 'Запорізька':
    expense *= 685;
    break;
case 'Івано-Франківська':
    expense *= 577;
    break;
case 'Київська':
    expense *= 704;
    break;
case 'Кіровоградська':
    expense *= 883;
    break;
case 'Луганська':
    expense *= 639;
    break;
case 'Львівська':
    expense *= 482;
    break;
case 'Миколаївська':
    expense *= 737;
    break;
case 'Одеська':
    expense *= 855;
    break;
case 'Полтавська':
    expense *= 794;
```

```
    break;
case 'Рівненська':
    expense *= 521;
    break;
case 'Сумська':
    expense *= 737;
    break;
case 'Тернопільська':
    expense *= 523;
    break;
case 'Харківська':
    expense *= 717;
    break;
case 'Херсонська':
    expense *= 889;
    break;
case 'Хмельницька':
    expense *= 654;
    break;
case 'Черкаська':
    expense *= 499;
    break;
case 'Чернівецька':
    expense *= 467;
    break;
case 'Чернігівська':
    expense *= 667;
    break;
default:
    console.log('Область не знайдена');
```

```
}  
console.log(expense);  
  
expense *= elements[2];  
console.log(expense);  
  
switch (elements[3].trim()) {  
  case 'Природні території та об'єкти природно-заповідного фонду':  
    expense *= 10;  
    break;  
  case 'Пляжні зони уздовж морів, морських заток і лиманів':  
    expense *= 6;  
    break;  
  case 'Водоохоронні зони уздовж річок, морів, навколо озер, водосховищ та  
інших водойм':  
    expense *= 4;  
    break;  
  case 'Охоронні зони наземних, надземних і підземних трубопроводів':  
    expense *= 2.5;  
    break;  
  case 'Охоронні зони уздовж повітряних і підземних кабельних ліній зв'язку, а  
також навколо випромінювальних споруд телерадіостанцій та радіорелейних ліній':  
    expense *= 1.5;  
    break;  
    case 'Охоронні зони уздовж повітряних і підземних кабельних ліній  
електропередачі':  
    expense *= 1.5;  
    break;  
    case 'Захисні, охоронні та інші зони з особливими умовами користування  
навколо військових та інших оборонних об'єктів':
```

expense *= 1.5;

break;

case 'Зони відчуження та безумовного (обов'язкового) відселення, що зазнали радіоактивного забруднення внаслідок Чорнобильської катастрофи':

expense *= 2;

break;

case 'Зона санітарної охорони навколо об'єктів, у яких є підземні та відкриті джерела водопостачання, водозабірні та водоочисні споруди, водоводи, об'єкти оздоровчого призначення':

expense *= 6;

break;

case 'Санітарно-захисні зони навколо об'єктів, які є джерелом виділення шкідливих речовин, запахів, підвищеного рівня шуму, вібрації, ультразвукових і електромагнітних хвиль, електронних полів, іонізуючих випромінювань':

expense *= 1.5;

break;

case 'Прикордонна смуга уздовж державного кордону України':

expense *= 3;

break;

case 'Сільськогосподарські угіддя, включені в установленому порядку до складу екомережі':

expense *= 3;

break;

case 'Землі, зарезервовані для заповідання':

expense *= 4;

break;

case 'Інші території з особливим режимом використання земель':

expense *= 4;

break;

// Додайте інші області тут

```
default:
    expense *= 2;
}

console.log(expense);

switch (elements[4].trim()) {
    case 'Надзвичайно небезпечні':
        expense *= 4;
        break;
    case 'Дуже небезпечні':
        expense *= 3;
        break;
    case 'Помірно небезпечні':
        expense *= 2.5;
        break;
    case 'Інші':
        expense *= 1.5;
        break;
    default:
        expense *= 1.5;
}

console.log(expense);

var date = new Date(elements[5]);
console.log(expense);
console.log(date);

var day = date.getDate();
```

```
var month = date.getMonth() + 1; // Додаємо 1, оскільки місяці у JavaScript  
починаються з 0
```

```
var year = date.getFullYear();
```

```
// Формуємо рядок у відповідному форматі
```

```
var dateString =
```

```
(day < 10 ? '0' : '') +
```

```
day +
```

```
'.' +
```

```
(month < 10 ? '0' : '') +
```

```
month +
```

```
'.' +
```

```
year;
```

```
console.log(dateString); // Виведе "11.05.2024"
```

```
let obj = {
```

```
  ID: nanoid(),
```

```
  pollutionType: elements[0],
```

```
  pollutionRegion: elements[1],
```

```
  pollutionArea: elements[2],
```

```
  pollutionTerritory: elements[3],
```

```
  pollutionSubstance: elements[4],
```

```
  pollutionDated: dateString,
```

```
  pollutionExpense: Math.floor(expense) / 1000,
```

```
};
```

```
add(obj);
```

```
// 0 вода
```

```

// 1 область
//2 площа
//3 важливість теритоїї
//4 субстанція
//5 дата
//6 сомент
}
});
}

import { getAll } from './getAll';
import { addAll } from './writeAll';

export function deleteRecord(id) {
  if (id) {
    let allRecords = getAll();
    console.log(allRecords);
    let filtered = allRecords.filter(({ ID }) => ID !== id);
    addAll(filtered);
    return;
  }
  return console.error('ПОМИЛКА НЕ ПЕРЕДАЛИ АЙДІ ЗАПИСУ');
}

export function getAll(params) {
  try {
    let pollutionsLocal = localStorage.getItem('pollutions');
    if (!pollutionsLocal) {
      return console.log('ПОМИЛКА В localStorage - немає записів');
    }
  }
}

```

```

    let smth = JSON.parse(pollutionsLocal);
    return smth;
  } catch (error) {
    console.log('ПОМИЛКА В localStorage', error);
  }
}

export function add(obj) {
  try {
    let pollutionsLocal = localStorage.getItem('pollutions');
    if (!pollutionsLocal) {
      return localStorage.setItem('pollutions', JSON.stringify([obj]));
    }
    let smth = JSON.parse(pollutionsLocal);
    console.log(smth);
    smth.push(obj);

    localStorage.setItem('pollutions', JSON.stringify(smth));
    location.reload(true);
  } catch (error) {
    console.log('ПОМИЛКА В localStorage', error);
  }
}

export function addAll(arr) {
  try {
    let pollutionsLocal = localStorage.getItem('pollutions');
    if (!pollutionsLocal) {
      return console.error(

```

```
<html lang="ru">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Page title</title>
<link rel="stylesheet" href="css/common.css" />
</head>
<body>
<header class="header">
<section class="header-section">
<!-- <div class="container"> -->
<nav class="navigation">
<div class="logo">
<a href="./index.html">Track<span class="black-logo">Imp</span></a>
</div>
<ul class="nav-flex-menu">
<li class="nav-flex-menu-item">
<a class="nav-page" href="./index.html">Info</a>
</li>
<li id="login-option" class="nav-flex-menu-item">
<a class="nav-page" href="./login-page.html">Login</a>
</li>
</ul>
</nav>
<!-- </div> -->
</section>
</header>
<div class="mother-board">
```

<h1>ЧОМУ ТЕМА ШКОДИ ДОВКІЛЛЮ ВІД ВІЙНИ ВАЖЛИВА</h1>

<p class="expenses"></p>

<div class="fornt-text">

Війни приносять з собою не тільки людські страждання та економічні збитки, але й значну шкоду довкіллю. Розглянемо основні причини, чому ця тема заслуговує на нашу увагу. <h3>1. Знищення природних ресурсів</h3> Під час воєнних дій значна кількість природних ресурсів може бути знищена або забруднена. Ліси вирубуються для побудови військових баз або використовуються як ресурси для ведення війни. Це призводить до втрати біорізноманіття та руйнування екосистем. Військові дії часто супроводжуються вибухами та пожежами, які викидають в атмосферу велику кількість токсичних речовин. Це забруднює повітря, яким дихають люди, та воду, яку вони п'ють, що призводить до довгострокових негативних наслідків для здоров'я населення. Військові дії призводять до масового переселення тварин, руйнування їхнього природного середовища та безпосередньої загибелі від бойових дій. Це має катастрофічні наслідки для популяцій багатьох видів, включаючи ті, що знаходяться під загрозою зникнення. Зображення: Фото дикої природи, постраждалої від війни. 5. Токсичні залишки та відходи Залишки військової техніки, боєприпасів та інших військових матеріалів можуть містити токсичні речовини, які забруднюють ґрунт та воду. Це робить землі непридатними для сільського господарства та небезпечними для проживання людей.


```

</div>
<!-- <div class="mother-board">
<ul>
<div class="go-back-button"><a href="./index.html">Go back</a></div>
<form class="login">
<input id="login" type="text" placeholder="Username" />
<input id="password" type="password" placeholder="Password" />
<button class="login-button">Login</button>
</form>
</ul>
</div> -->
<!-- <div class="mother-board">
<h1 class="role-header">Welcome to admin's page</h1>
<input class="exel-input" type="file" accept=".xls,.xlsx" />
</div> -->
<script src="./js/index.js" type="module"></script>
</body>
</html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>login</title>
<link rel="preconnect" href="https://fonts.googleapis.com" />
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
<link

```

```

href="https://fonts.googleapis.com/css2?family=Asap:ital,wght@0,100..900;1,100..900&di
rel="stylesheet"
/>
<link rel="stylesheet" href="./css/common.css" />
</head>
<body>
<header class="header">
<section class="header-section">
<!-- <div class="container"> -->
<nav class="navigation">
<div class="logo">
<a href="./index.html">Track<span class="black-logo">Imp</span></a>
</div>
<ul class="nav-flex-menu">
<li class="nav-flex-menu-item">
<a class="nav-page" href="./index.html">Info</a>
</li>
<li data-login="out" id="login-option" class="nav-flex-menu-item">
<a class="nav-page login-set" href="./login-page.html">Login</a>
</li>
</ul>
</nav>
<p class="role-label"></p>
<!-- </div> -->
</section>
</header>

```

```

<div class="mother-board">
<form class="login">
<input id="login" type="text" placeholder="Username" />
<input id="password" type="password" placeholder="Password" />
<button class="login-button">Login</button>
</form>

<div class="hide-elem main-content-container">
<div class="hide-elem exel-container">
<label for="exel-input">Завантажте заповнений файл</label>
<div class="exel-flex-container">
<input
id="exel-input"
class="exel-input"
type="file"
accept=".xls,.xlsx"
/>
<a class="template" href="..template.xlsx" download>
Завантажте зразок файлу</a
>
</div>
</div>
<form class="pollution-form">
<a
class="template absolute"
href="https://snipboard.io/pQqklx.jpg"
target="_blank"

```

```
rel="noopener noreferrer"
>Як користуватися формою</a
>
<h3 class="pollution-form-title">Форма для розрахунку шкоди</h3>
<div class="pollution-form-fields-container">
<label for="pollution-type">Екологічна ділянка</label>
<select
class="pollution-form-field"
name="pollution-type"
id="pollution-type"
>
<option value="">Оберіть природне середовище</option>
<option value="10">Вода</option>
<option value="5">Повітря</option>
<option value="1.5">Земля</option>
</select>
</div>
<div class="pollution-form-fields-container">
<label for="region-select">Область України</label>
<select
class="pollution-form-field"
name="region"
id="region-select"
>
<option value="">Оберіть область</option>
<option value="656">Автономна республіка Крим</option>
```

```
<option value="738">Вінницька область</option>
<option value="483">Волинська область</option>
<option value="748">Дніпропетровська область</option>
<option value="837">Донецька область</option>
<option value="538">Житомирська область</option>
<option value="518">Закарпатська область</option>
<option value="685">Запорізька область</option>
<option value="577">Івано-Франківська область</option>
<option value="704">Київська область</option>
<option value="883">Кіровоградська область</option>
<option value="639">Луганська область</option>
<option value="482">Львівська область</option>
<option value="737">Миколаївська область</option>
<option value="855">Одеська область</option>
<option value="794">Полтавська область</option>
<option value="521">Рівненська область</option>
<option value="737">Сумська область</option>
<option value="523">Тернопільська область</option>
<option value="717">Харківська область</option>
<option value="889">Херсонська область</option>
<option value="654">Хмельницька область</option>
<option value="499">Черкаська область</option>
<option value="467">Чернівецька область</option>
<option value="667">Чернігівська область</option>
</select>
</div>
```

```

<div class="pollution-form-fields-container">
<label for="pollution-area">Площа забруднення Кв.м</label>
<input
class="pollution-form-field"
id="pollution-area"
type="number"
name="pollution-area"
/>
</div>
<div class="pollution-form-fields-container">
<label for="special-territories-select">Спеціальні території</label>
<select
class="pollution-form-field"
name="special-territories"
id="special-territories-select"
>
<option value="">Оберіть територію</option>
<option value="10">
Природні території та об'єкти природно-заповідного фонду
</option>
<option value="5">
Штучно створені об'єкти природно-заповідного фонду
</option>
<option value="3.5">
Охоронні зони навколо цінних природних об'єктів
</option>

```

<option value="4">

Охоронні зони навколо об'єктів культурної спадщини

</option>

<option value="2">

Охоронні зони навколо гідрометеорологічних станцій

</option>

<option value="2.5">

Особливо цінні землі сільськогосподарського призначення

</option>

<option value="4">

Округ санітарної (гірничо-санітарної) охорони у

лікувально-оздоровчих місцевостях і на території курортів

</option>

<option value="6">

Прибережні захисні смуги уздовж річок, навколо водойм та на

островах, заплави малих річок

</option>

<option value="5">

Прибережні захисні смуги уздовж морів, морських заток і лиманів

та на островах

</option>

<option value="6">

Пляжні зони уздовж морів, морських заток і лиманів

</option>

<option value="4">

Водоохоронні зони уздовж річок, морів, навколо озер, водосховищ

та інших водойм

</option>

<option value="2.5">

Охоронні зони наземних, надземних і підземних трубопроводів

</option>

<option value="1.5">

Охоронні зони уздовж повітряних і підземних кабельних ліній

зв'язку, а також навколо випромінювальних споруд

телерадіостанцій та радіорелейних ліній

</option>

<option value="1.5">

Охоронні зони уздовж повітряних і підземних кабельних ліній

електропередачі

</option>

<option value="1.5">

Захисні, охоронні та інші зони з особливими умовами користування

навколо військових та інших оборонних об'єктів

</option>

<option value="2">

Зони відчуження та безумовного (обов'язкового) відселення, що

зазнали радіоактивного забруднення внаслідок Чорнобильської

катастрофи

</option>

<option value="6">

Зона санітарної охорони навколо об'єктів, у яких є підземні та

відкриті джерела водопостачання, водозабірні та водоочисні

споруди, водоводи, об'єкти оздоровчого призначення

</option>

<option value="1.5">

Санітарно-захисні зони навколо об'єктів, які є джерелом виділення шкідливих речовин, запахів, підвищеного рівня шуму, вібрації, ультразвукових і електромагнітних хвиль, електронних полів, іонізуючих випромінювань

</option>

<option value="3">

Прикордонна смуга уздовж державного кордону України

</option>

<option value="3">

Сільськогосподарські угіддя, включені в установленому порядку до складу екомережі

</option>

<option value="4">Землі, зарезервовані для заповідання</option>

<option value="4">

Інші території з особливим режимом використання земель

</option>

</select>

</div>

<div class="pollution-form-fields-container">

<script src="https://unpkg.com/@popperjs/core@2/dist/umd/popper.min.js"></script>

<script src="https://unpkg.com/tippy.js@6/dist/tippy-bundle.umd.js"></script>

<div class="hide-elem hide-formted-text">

<pre>

Надзвичайно небезпечні речовини:

Ртуть (Hg)

Органічні ртутні сполуки

Плутоній (Pu)

Амонійна перхлорат

Сарин (GB)

Сірчана кислота (H₂SO₄)

Фосген (CG)

Арсенік (As)

Ціаніди

Радіоактивні ізотопи

Дуже небезпечні речовини:

Дихлорвінілбензол (DCB)

Селян (HD)

Нітрогліцерин

Монооксид вуглецю (CO)

Диоксини

Хлор (Cl₂)

Фосфор (P)

Мустангаз (HN₃)

Ацетонитрил

Фосфорний трихлорид (PCl₃)

Помірно небезпечні речовини:

Толуол

Бензол

Сірка діоксид (SO₂)

Амоніак (NH₃)

Формальдегід

Бензпірен

Оксиди азоту (NO_x)

Сірчані діоксиди (SO_x)

Хлористий водень (HCl)

Фториди</pre>

</div>

<label for="danger-level-select">Забруднююча речовина</label>

<select

title=""

class="pollution-form-field"

name="danger-level"

id="danger-level-select"

>

<option value="">Наведіть для підказки</option>

<option value="4">Надзвичайно небезпечні</option>

<option value="3">Дуже небезпечні</option>

<option value="2.5">Помірно небезпечні</option>

<option value="1.5">Інші</option>

</select>

</div>

<div class="pollution-form-fields-container">

<label for="date-field">Дата</label>

<input

type="date"

```

class="pollution-form-field"
name="date"
id="date-field"
/>
</div>
<button
class="pollution-form-button"
type="submit"
id="calculate-button"
>
Розрахувати дані
</button>
</form>
<h3>Історія запитів</h3>
<div class="table-container">
<table id="pollutionTable" class="table">
<tr>
<th>Екологічна ділянка</th>
<th>Область України</th>
<th>Площа забруднення Кв.м</th>
<th>Спеціальні території</th>
<th>Забруднююча речовина</th>
<th>Дата</th>
<th>Розмір Шкоди(тис.грн)</th>
</tr>
</table>

```

```
<button class="delete-button export-data">export data</button>
</div>
</div>
</div>
<script type="module" src="./js/index.js"></script>
<script>
function deleteRow(index) {
  try {
    const evt = document.getElementsByClassName('delete-button')[index];
    const id = evt.id;
    if (id) {
      let pollutionsLocal = localStorage.getItem('pollutions');
      if (!pollutionsLocal) {
        return console.log('ПОМИЛКА В localStorage - немає записів');
      }
      let smth = JSON.parse(pollutionsLocal);
      console.log(smth);
      let filtered = smth.filter(({ ID }) => ID !== id);
      let result = confirm('Ви упевнені, що хочете видалити запис?');
      if (result) {
        localStorage.setItem('pollutions', JSON.stringify(filtered));
        location.reload(true);
      }
      return;
    }
    return;
  }
}
```

```

return console.error('ПОМИЛКА НЕ ПЕРЕДАЛИ АЙДІ ЗАПИСУ');
} catch (error) {
console.error('ПОМИЛКА у видаленні елемента', error);
}
}

const content = document.querySelector('.hide-formted-text');

tippy('#danger-level-select', {
content: content.innerHTML,
allowHTML: true,
maxWidth: 260,
placement: 'right-start',
});
</script>
</body>
</html>

'ПОМИКАЛІ неможливо записати дані в базу так як вона відсутня'
);
}
localStorage.removeItem('pollutions');
localStorage.setItem('pollutions', JSON.stringify(arr));
} catch (error) {
console.log('ПОМИЛКА В localStorage', error);
}
}

```