

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Реінжиніринг інформаційної системи аналізу збитків від техногенних та природних катастроф»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ.м-01 Шишкін Олексій Васильович

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

_____ «__» грудня 2021 р.

Науковий керівник

(підпис)

к.т.н., доц., Марченко А.В.
(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.
(науковий ступінь, вчене звання, прізвище та ініціали)

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

Студент _____
(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ В. В. Шендрик
«__» _____ 2021 р.

З А В Д А Н Н Я
на кваліфікаційну роботу магістра студентів

Шшикін Олексій Васильович

1 Тема роботи Реінжиніринг інформаційної системи аналізу збитків від техногенних та природніх катастроф

затверджені наказом по університету від «29» жовтня 2021 р. № 0787-VI

2 Термін здачі студентом закінченого проекту «__» грудня 2021 р.

3 Вхідні дані до проекту Методика розрахунку наслідків лісової пожежі, методика оцінки обстановки під час землетрусу, методичні розробки для оцінювання еколого-економічного збитку від викиду НХР

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їй належить розробити) 1) Аналіз предметної області, 2) Проектування інформаційної системи аналізу збитків від техногенних та природніх катастроф, 3) Розробка інформаційної системи аналізу збитків від техногенних та природніх катастроф

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) аналіз предметної області, мета і задачі дипломного проекту, порівняння аналогів, функціональні вимоги до інформаційної системи, моделювання роботи інформаційної системи, декомпозиція першого рівня, декомпозиція підпроцесу аналіз збитків від викиду НХР, діаграма варіантів використання, засоби реалізації, демонстрація додатку, висновки, оприлюднення роботи

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Оформлення планування робіт		
2.	Проведення аналізу предметної області		
3.	Проведення структурно-функціонально моделювання процесів		
4.	Розробка нових модулів системи		
5.	Тестування нових модулів системи		
6.	Здача пояснювальної записки та файлів розробленого проекту		

Студент _____
(підпис)

Шишкін О.В.

Керівник роботи _____
(підпис)

к.т.н., доц. Марченко А.В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Реінжиніринг інформаційної системи аналізу збитків від техногенних та природніх катастроф».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 30 найменувань, додатків. Загальний обсяг роботи – 110 сторінок, у тому числі 53 сторінок основного тексту, 3 сторінки списку використаних джерел, 54 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено реінжинірингу інформаційної системи аналізу збитків від техногенних та природніх катастроф.

В роботі проведено огляд останніх досліджень та публікацій, які стосуються аналізу збитків, аналіз існуючого додатку та визначення напрямку реінжинірингу, моделювання та проектування додатку.

У роботі виконано розширення функціоналу системи, а саме таких його частин:

- модуль розрахунку збитків від катастроф які сталися у минулому;
- модуль моделювання катастроф які можуть відбутися у майбутньому;
- модуль збереження результатів розрахунків;
- модуль роботи з історією розрахунків.

Результатом проведеної роботи є розширена інформаційна система аналізу збитків від техногенних та природніх катастроф.

Ключові слова: катастрофа, інформаційна система, збитки, мапа, реципієнти, аналіз, історія розрахунку, минуле, майбутнє.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень і публікацій.....	8
1.2 Аналіз програмних продуктів - аналогів	10
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ЗБИТКІВ ВІД ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ	14
2.1 Постановка задачі	14
2.2 Моделювання роботи web-додатку в IDEF0	15
2.3 Моделювання варіантів використання	22
2.4 Проектування моделі бази даних	24
3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ЗБИТКІВ ВІД ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ	28
3.1 Програмна реалізація	28
3.1.1 Розробка бази даних	28
3.1.2 Розширення модулю роботи з API погоди	31
3.1.3 Розробка модулю роботи з історією розрахунків	33
3.2 Використання програмного додатку.....	38
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТОК А.....	57
ДОДАТОК Б.....	65
ДОДАТОК В	68

ВСТУП

З кожним роком в світі відбувається все більше і більше катастроф, як техногенного так і природнього характеру. Їх масштаби варіюються від трансграничних до локальних[1], але незалежно від масштабу катастрофи, все одно необхідно проводити розрахунок збитків, який базується на визначенні зони враження, матеріальних та людських збитків. Не менш важливим аспектом, є можливість аналізу катастроф, які виникали та збитки від яких були розраховані. На основі проаналізованих даних, можна буде виконати прогнозування катастроф, які можуть виникнути у майбутньому, та зменшити збитки від них.

Об'єктом дослідження дипломного проектування є процес модернізації існуючої системи розрахунку збитків від катастрофи природнього або техногенного характеру за рахунок впровадження нових модулів розширення функціоналу системи.

Предметом дослідження є інформаційна система(ІС) аналізу збитків від техногенних та природніх катастроф.

Метою кваліфікаційної роботи магістра є розширення функціоналу існуючої інформаційної системи, що буде використовуватися зацікавленими особами для розрахунку збитків від техногенних або природніх катастроф, їх аналізу та прогнозування потенційних збитків від катастроф, які можуть виникнути у майбутньому.

Для досягнення мети кваліфікаційної роботи потрібно вирішити такі задачі:

- реалізувати модуль розрахунку збитків від катастроф, які відбулися чи могли відбутися у минулому часі, та прогнозування збитків від катастроф, що можуть статись у майбутньому;
- реалізувати базу даних для збереження всіх результатів розрахунків, які були виконані інформаційною системою;

– реалізувати модуль доступу до збереженої інформації та її обробки.

Причиною ініціалізації проекту є потреба в модернізації та розширенні функціоналу вже існуючого рішення, яке виконує аналіз збитків.

Результатом проекту буде інформаційна система, що надаватиме змогу потенційним користувачам швидко та точно провести аналіз збитків від техногенних та природніх катастроф, змодельовати катастрофу яка могла виникнути у майбутньому або минулому, а також переглянути повну історію розрахунків збитків від катастроф, які було виконано засобами розробленої ІС.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Розрахунок збитків від катастроф, які виникають в результаті людських дій або природніх катаклізмів, безспірно є однією з найважливіших частин реагування на загрозу, що виникла. Але не менш важливим процесом є аналіз всіх катастроф схожого характеру. Данні, отримані в ході аналізу, є критично важливими, оскільки на їх основі можна виконати прогнозування катастроф які можуть виникнути в майбутньому та за рахунок цього знизити потенційні втрати як людські так і матеріальні.

В ході виконання магістерської кваліфікаційної роботи, будуть розроблені додаткові модулі, які розширять функціонал вже існуючої інформаційної системи для розрахунку збитків від природніх або техногенних катастроф. Природні катастрофи, які представлені в додатку: лісові пожежі та землетруси. Група катастроф техногенного характеру в системі представлена викидами небезпечної хімічної речовини(НХР).

Нові модулі які будуть представлені, дозволять проводити аналіз катастроф, які виникали у минулому, та збитки від яких було розраховано за допомогою інформаційної системи. Вся інформація про виконані розрахунки, буде зберігатися у базі даних, та буде представлена у зручному вигляді на веб-сторінках які відносяться до певного типу катастроф. Окрім аналізу катастроф, розрахунок для яких уже було виконано, новий модуль моделювання катастрофи, дозволить виконувати розрахунок для потенційних катастроф минулого та майбутнього, що в свою чергу буде сприяти прогнозуванню катастроф та зменшення збитків від них.

Для виконання аналізу збитків від перелічених катастроф існують методики, визначення зон враження, та проведення розрахунків збитків, які були нанесено на основі інформації про території, що потрапили в катастрофу.

Для розрахунку збитків від викиду НХР, була використана методика, яка представлена в дисертаційній роботі к.е.н., доцента кафедри управління факультету економіки та менеджменту Сумського державного університету Опанасюк Юлії Анатоліївни (кафедра управління навчально-наукового інституту фінансів, економіки та менеджменту ім. О.Балацького)[3].

В основі даної методики лежать методичні підходи до прямих і не прямих методів оцінки економічного збитку від надзвичайних ситуацій (НС).

До прямих методів відносять такі методи, як метод контрольних районів, метод аналітичних залежностей, комбінований метод або так званий пореципієнтний метод. Кількість методів непрямого розрахунку збитків значно більша і включає наступні методи: метод ринковий, метод соціологічних опитувань, метод готовності платити, метод згоди отримувати компенсацію, гедоністичний метод та нормативний метод.

В існуючій інформаційній системі використовується пореципієнтний метод визначення еколого-економічного збитку від НС, даний метод дуже трудомісткий та передбачає роботу з громіздкими інформаційними базами. Цей метод також буде використовуватися у нових модулях системи.

Розрахунки нових модулів будуть виконуються базуючись на часу виникнення катастрофи, типі речовини, яка була викинута, погодних умовах та місцевості в якій відбувся викид. Алгоритм виконання розрахунку наступний:

1. Визначення часу виникнення катастрофи.
2. Визначення погодних умов які мали місце під час виникнення катастрофи.
3. Визначення глибини прогнозованої зони хімічного забруднення, Гпзхз, км.

4. Визначення ширини прогнозованої зони хімічного забруднення, Шпзхз, км.
5. Визначення площі прогнозованої зони забруднення, Spзхз, км².
6. Визначення площі зони можливого хімічного забруднення Sзмхз, км².
7. Нанесення зони враження на мапу.
8. Враховуючи об'єкти які потрапили до зони враження визначаються збитки які були нанесені катастрофою.

В рамках ІС поводяться розрахунки для лісових пожеж та землетрусів. Для розрахунків збитків. як від представлених катастроф, за основу було взято «Методику визначення розміру і площі зон руйнувань в осередку землетрусу» [3] та «Методику розрахунку параметрів лісових пожеж» [4]. В обох випадках алгоритм розрахунку схожий, а нові модулі дещо його розширюють:

1. Визначення часу виникнення катастрофи.
2. Визначення погодних умов які мали місце під час виникнення катастрофи.
3. Визначити зони враження.
4. Нанесення зони враження на мапу.
5. Враховуючи об'єкти, які потрапили до зони враження визначаються збитки, які були нанесені катастрофою.

1.2 Аналіз програмних продуктів - аналогів

Програмних продуктів-аналогів зі схожим функціоналом не існує. Тому розглядається лише розроблений в ході виконання бакалаврської

роботи web-додаток аналізу збитків від техногенних та природних катастроф (рис 1.1).

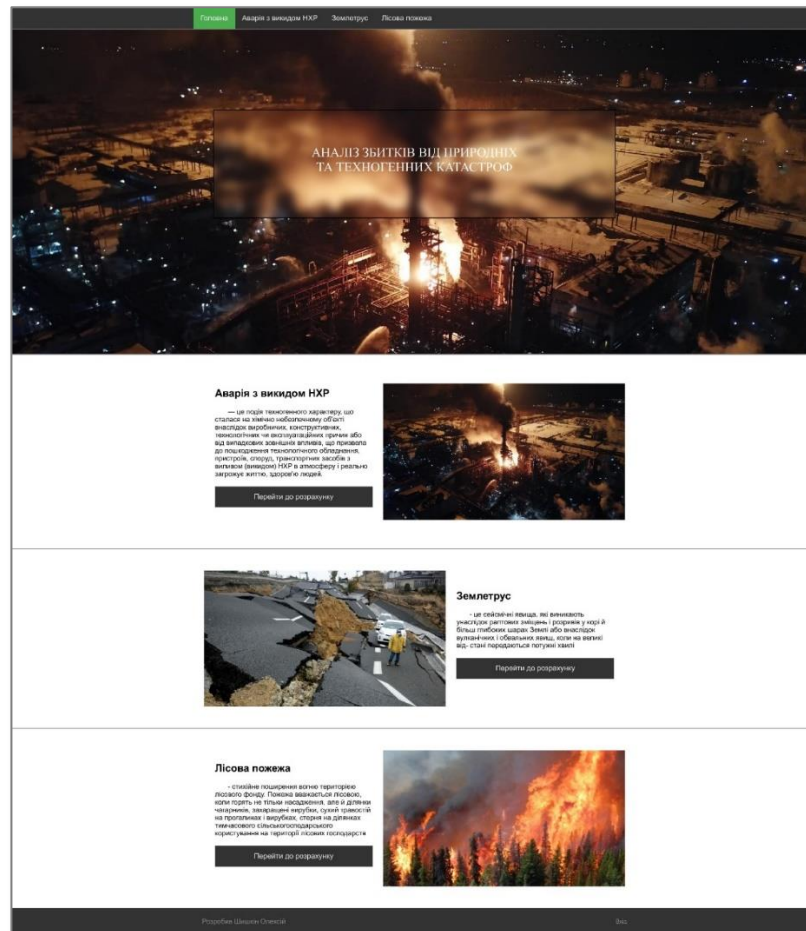


Рисунок 1.2 – Web-додаток аналізу збитків від техногенних та природних катастроф

Провівши аналіз тестового впровадження та функціоналу додатку, було виявлено ряд недоліків:

- відсутність можливості перегляду історії розрахунків;
- відсутність можливості аналізу катастроф у минулому;
- відсутність можливості моделювання катастрофи в майбутньому.

Порівняння об'єкту модернізації з розроблюваною інформаційною системою аналізу збитків від техногенних та природніх катастроф наведено у таблиці 1.1.

Таблиця 1.1 – Аналіз існуючого та розроблюваного додатку для аналізу збитків

Критерій	Web-додаток аналізу збитків від техногенних та природніх катастроф	Розроблюваний додаток
Робота з динамічною мапою	+	+
Аналіз збитків від техногенних катастроф	+	+
Аналіз збитків від природніх катастроф	+	+
Просте розширення списку аварійно-небезпечних зон	+	+
Отримання інформації про погодні умови від API	+	+
Зручний вивід звіту про розрахунки	+	+
Автоматичний підрахунок реципієнтів	+	+

Продовження таблиці 1.1

Критерій	Web-додаток аналізу збитків від техногенних та природніх катастроф	Розроблюваний додаток
Можливості перегляду історії розрахунків	-	+
Можливості аналізу катастроф у минулому	-	+
Можливості моделювання катастрофи в майбутньому	-	+

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ЗБИТКІВ ВІД ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ

2.1 Постановка задачі

Метою виконання даної роботи є розширення функціоналу існуючої інформаційної системи, що буде використовуватися зацікавленими особами, для розрахунку збитків від техногенних або природних катастроф, їх аналізу та прогнозування потенційних збитків від катастроф які можуть виникнути у майбутньому. На даний момент інформаційна система виконує ряд задач:

- оцінка становища при аварії (визначення напрямку руху та розмірів зони розповсюдження хмари зараженого повітря, тривалості дії факторів забруднення);
- оцінка становища під час виникнення катастроф природного характеру;
- графічне відображення результатів аналізу та визначення даних щодо ушкодження природних ресурсів зараженої місцевості;
- розрахунок матеріальних збитків в результаті катастрофи, термінів відновлення господарської території та природних ресурсів на основі методики, представленої в роботах.

В рамках виконання магістерської кваліфікаційної роботи, наступні модулі повинні бути додані:

- збереження історії розрахунків;
- відображення історії розрахунків у зручному форматі;
- аналіз збитків від катастроф у минулому;
- моделювання катастроф у майбутньому;

Для реалізації поставленого завдання були виділені наступні задачі дослідження:

- аналіз функціоналу нових модулів;
- розробка інтерфейсу нових модулів;
- розробка нових модулів розрахунків;
- розробка бази даних;
- відладка та тестування;
- оформлення супроводжуючої документації.

2.2 Моделювання роботи web-додатку в IDEF0

Наступним кроком проектування інформаційної системи, після того, як мета та основні задачі дослідження було сформовано, є розробка контекстної діаграми IDEF0. Даний вибір обумовлений тим, що діаграми в нотації IDEF0 не тільки описують систему в цілому, та її взаємодію з навколишнім середовищем[2], а також допомагають виділити та підкреслити нові модулі системи які будуть додані.

Проаналізувавши існуючі елементи контекстної діаграми, для аналізу збитків від техногенних або природних катастроф, а також елементи які будуть додані в ході реалізації нових модулів, було сформовано наступний перелік даних:

- Вхід: інформація про погодні умови, інформація про катастрофу, інформація про дату та час виникнення катастрофи.
- Вихід: звіт про результати аналізу, запис в базі даних.
- Управління: методика аналізу збитків при викиду НХР, методика аналізу збитків при лісовій пожежі, методика аналізу збитків при землетрусі, документація до інформаційної системи.

– Механізми: користувач, OpenStreetMap API, Dark Sky API, інформаційна система.

На основі цих даних, була розроблена контекстна діаграма, що представлена на рис. 2.1.

Контекстна діаграма надає лише загальне уявлення про роботу системи, тому для більш детального опису логіки та послідовності робіт необхідно виконати декомпозицію. Процес аналізу збитків від техногенних або природніх катастроф, як і раніше можна розбити на 5 підпроцесів. Підпроцес визначення типу катастрофи був замінений на попереднє налаштування ІС, в свою чергу аналіз збитків від викиду НХР, аналіз збитків від лісових пожеж, аналіз збитків від землетрусів, формування звіту по збиткам залишились без змін.

До першого підпроцесу, попереднє налаштування ІС, були сформовані наступні вхідні дані:

- Вхід: інформація про катастрофу.
- Вихід: тип катастрофи або режим роботи “перегляд даних про катастрофу з БД”.
- Управління: документація до інформаційної системи.
- Механізми: користувач, інформаційна система.

До входів другого, третього та четвертого підпроцесів додалась інформація про дату та час виникнення катастрофи. Виходи, управління та механізми, залишилися незмінними.

До п'ятого підпроцесу, формування звіту про збитки, було розширено та сформовані наступні вхідні дані:

- Вхід: результати розрахунків збитків, режим роботи “перегляд даних про катастрофу з БД”.
- Вихід: звіт про результати аналізу, запис в базі даних.
- Управління: документація до інформаційної системи.
- Механізми: інформаційна система.

Діаграма другого рівня представлена на рис.2.2.

Після виконання декомпозиції першого рівня, також потрібно виконати декомпозицію нового підпроцесу, а також оновити існуючі діаграми для підпроцесів аналізу збитків від викиду НХР, аналізу збитків від лісових пожеж та аналізу збитків від землетрусів.

Процес попереднього налаштування ІС, розбивається на 2 підпроцеси: визначення режиму роботи ІС, визначення типу катастрофи.

До першого підпроцесу, визначення режиму роботи ІС, були сформовані наступні вхідні дані:

- Вхід: інформація про катастрофу.
- Вихід: режим роботи “перегляд даних про катастрофу з БД”, режим роботи "розрахунок виникнувшої катастрофи", режим роботи "прогнозування потенційної катастрофи" и.
- Управління: документація до інформаційної системи.
- Механізми: користувач, інформаційна система.

До другого підпроцесу, визначення типу катастрофи, були сформовані наступні вхідні дані:

- Вхід: інформація про катастрофу, розрахунок виникнувшої катастрофи, прогнозування потенційної катастрофи.
- Вихід: тип катастрофи.
- Управління: документація до інформаційної системи.
- Механізми: користувач, інформаційна система.

Декомпозиція процесу визначення режиму роботи ІС представлена на рис. 2.3.

Процеси аналізу збитків від викиду НХР, лісових пожеж та землетрусів, мають однаковий алгоритм дій та зазнають однакових змін, тому їх декомпозицію можна представити на прикладі викиду НХР.

Як і раніше процес аналізу збитків від викиду НХР, розбивається на 3 підпроцеси: визначення розмірів зони враження, визначення територій які потрапили в зону враження, аналіз збитків.

До першого підпроцесу, на вхід додалася інформація про дату та час виникнення катастрофи.

Декомпозиція процесів аналізу збитків представлені на рис. 2.4.

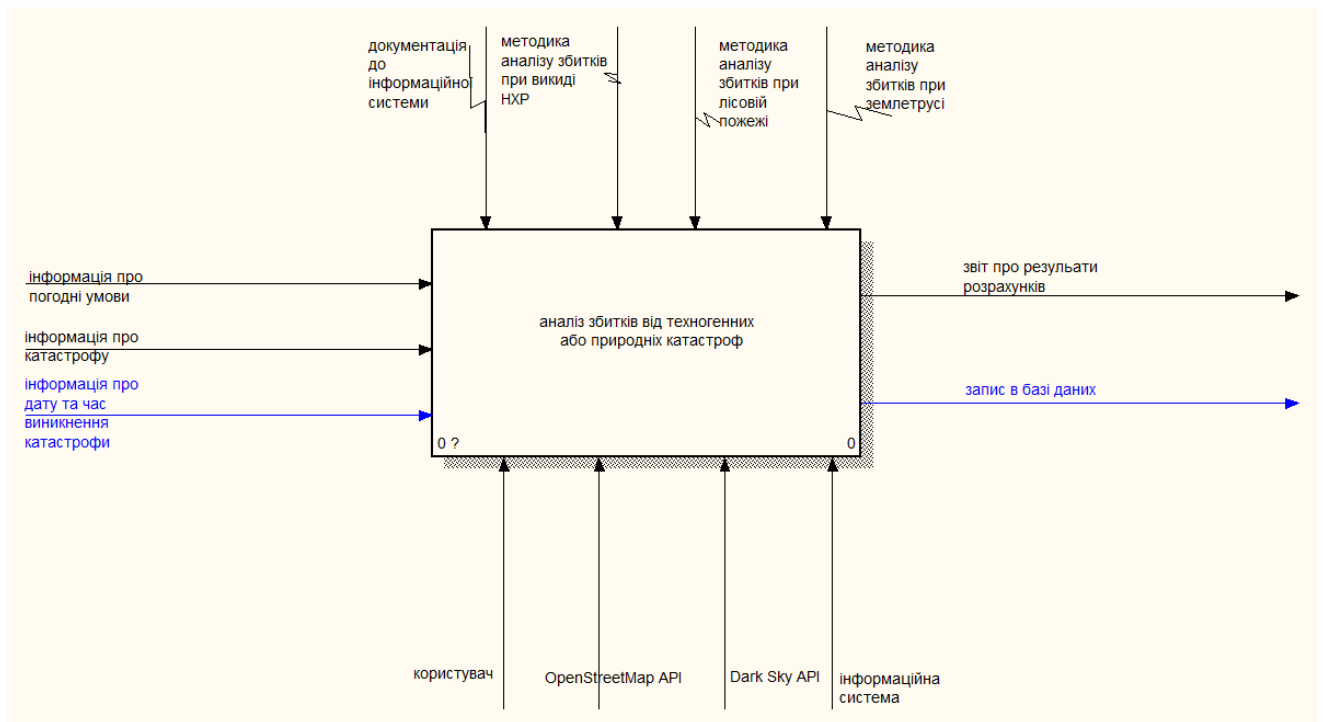


Рисунок 2.1 - Контекстна діаграма

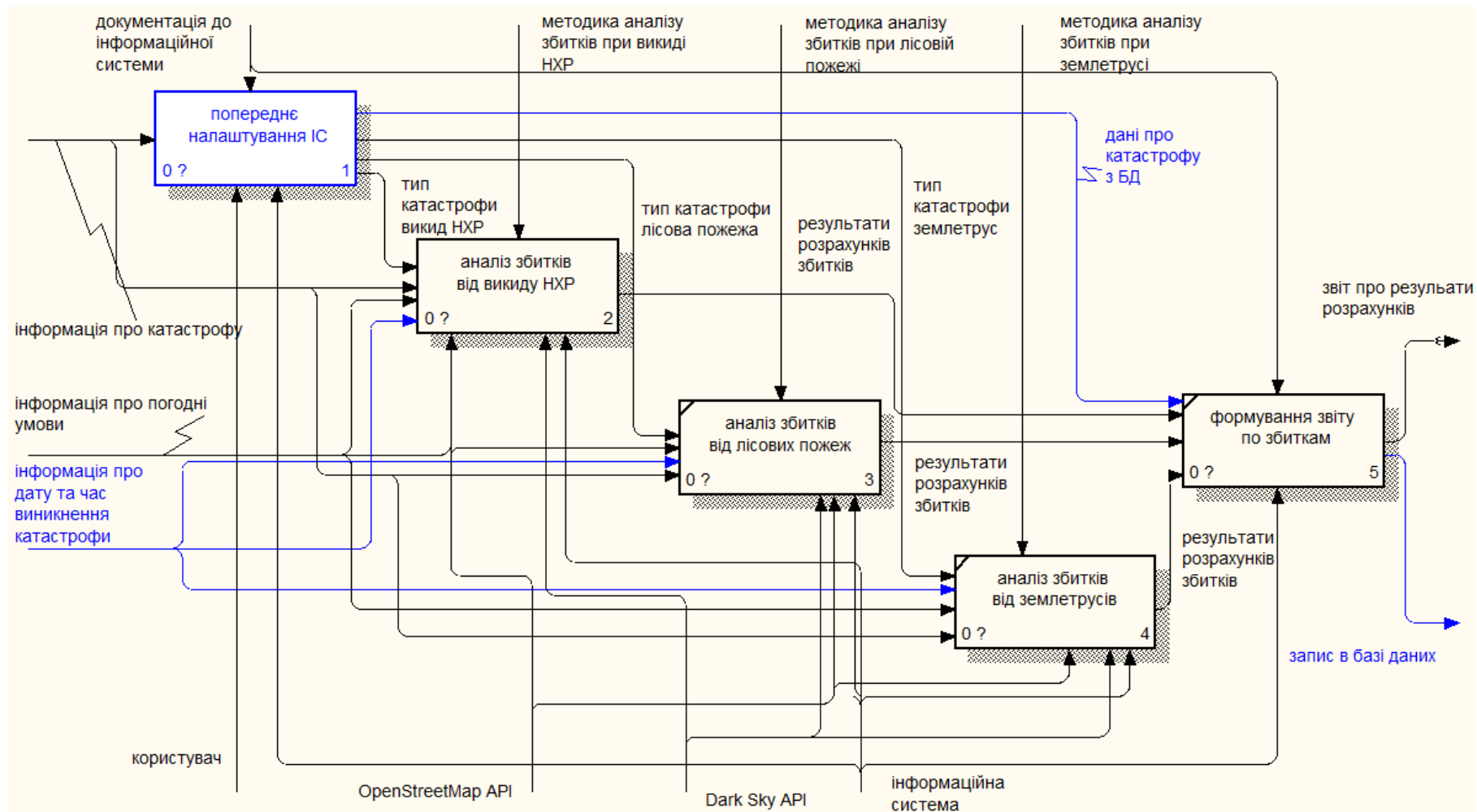


Рисунок 2.2 – Декомпозиція діаграми А-0

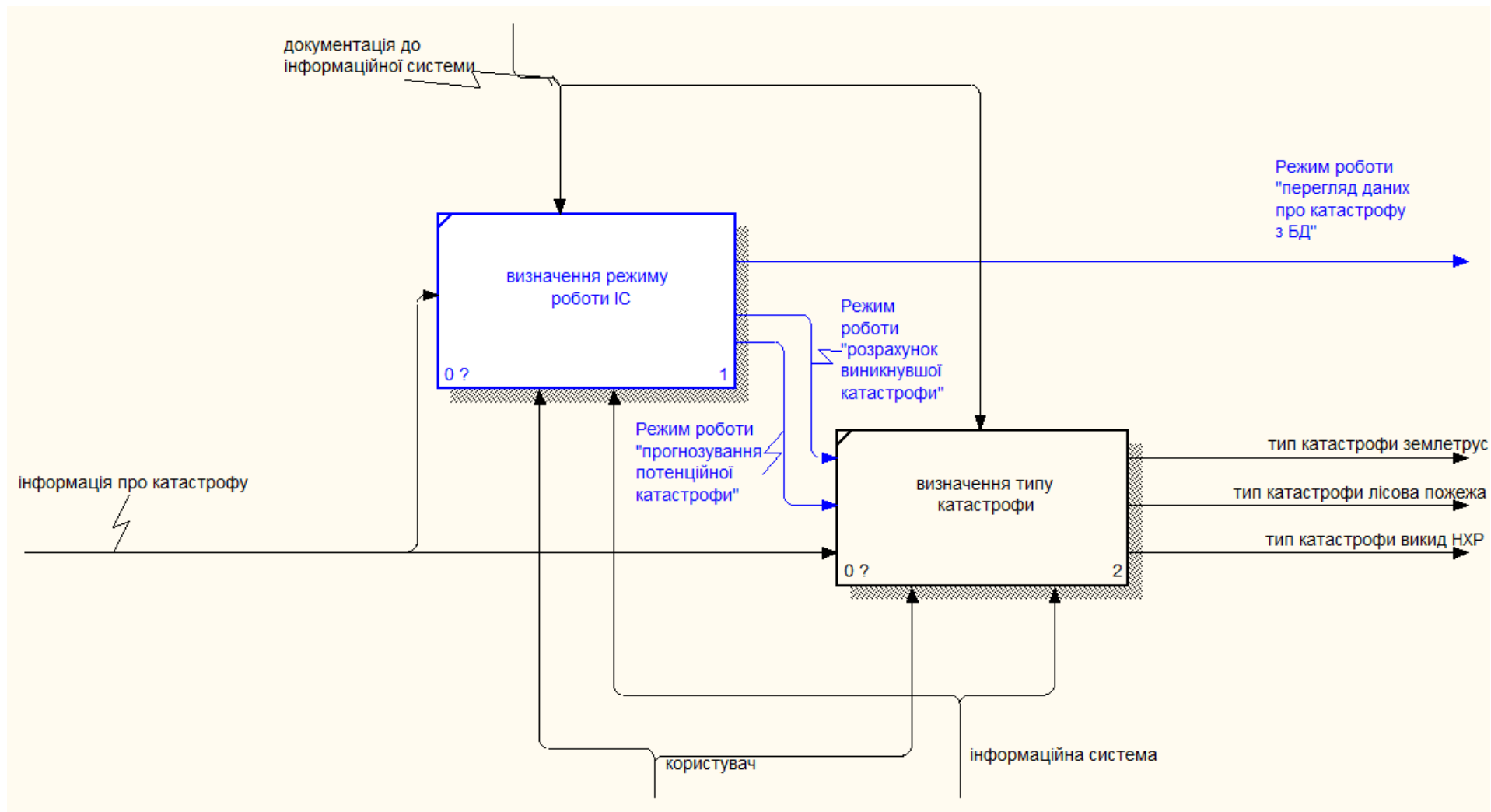


Рисунок 2.3 – Декомпозиція процесу визначення режиму роботи ІС

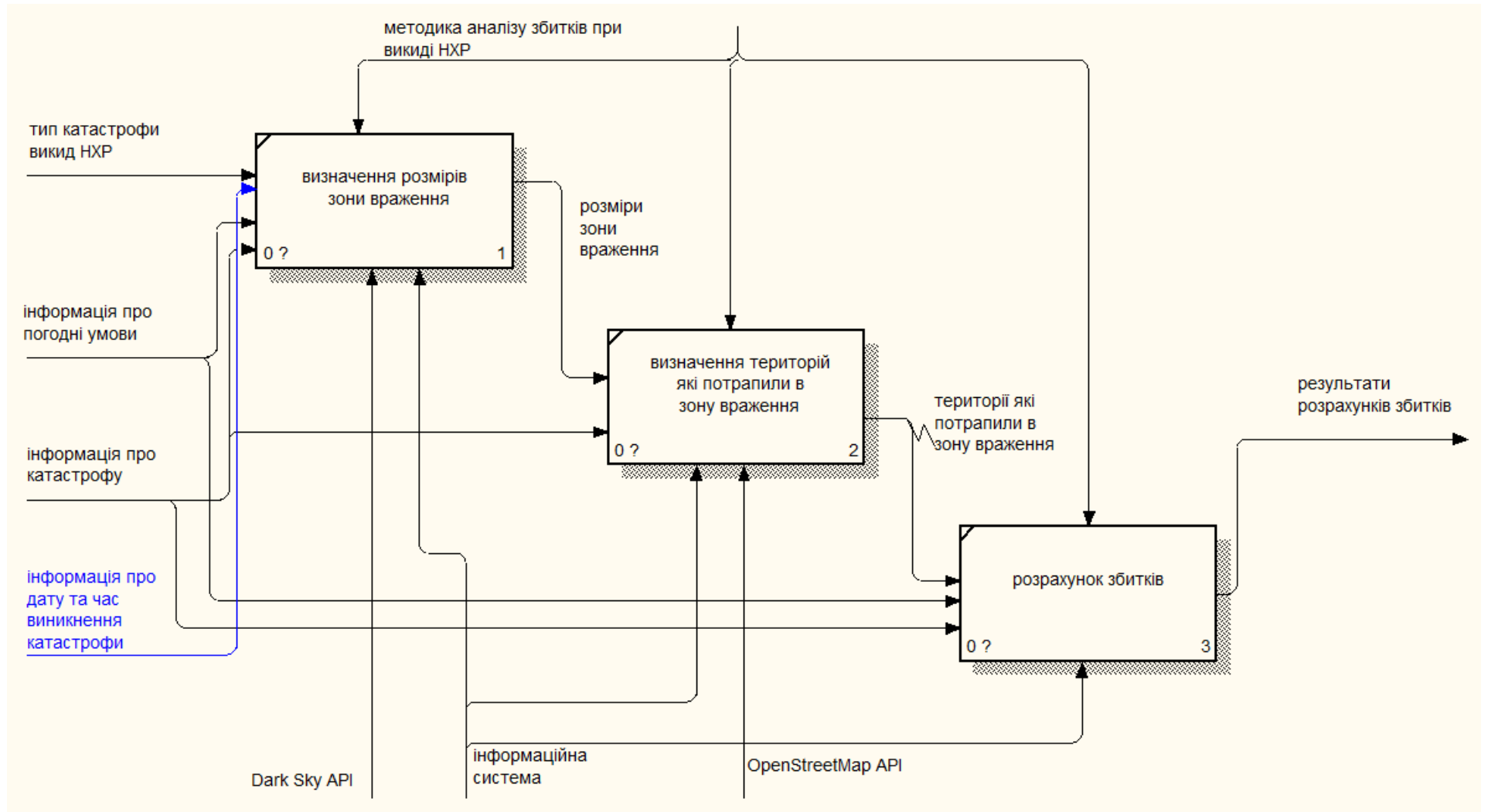


Рисунок 2.4 – Декомпозиція процесу аналізу збитків при викиду НХР

2.3 Моделювання варіантів використання

Наступним кроком після оновлення контекстних діаграм необхідно оновити діаграми варіантів використання (Use Case Diagram). Даний вид діаграм відображає взаємодію акторів та варіантів використання. Оновлення даного типу діаграм необхідно для того, щоб показати інформацію про те, які нові функції виконує система і яким чином кожен актор взаємодіє з ними[4].

В розроблюваній системі можна виділити наступних акторів:

- Відвідувач – має доступ до функціоналу виконання аналізу збитків, прогнозування потенційних катастроф у майбутньому, аналіз катастроф які виникли у минулому та перегляд історії всіх розрахованих катастроф у системі.

- Користувач з правом читання-запису – має доступ до функціоналу виконання аналізу збитків та вносити інформацію про аварійно небезпечні об'єкти.

- Адміністратор – має доступ до розрахунків та повного налаштування додатку.

- MySQL – зберігає інформацію необхідну для розрахунків, а також історію розрахунків які було виконано.

- Dark Sky API – надає інформацію про погодні умови.

- OpenStreetMap – надає функціонал для роботи з мапою.

Після визначення акторів в системі, формується перелік варіантів використання. Варіанти використання:

- Виконання аналізу збитків від викиду НХР.

- Виконання аналізу збитків від лісових пожеж.

- Виконання аналізу збитків від землетрусів.

- Виконання прогнозування потенційних катастроф у майбутньому.

- Виконання розрахунку для катастроф які виникли у минулому.
- Перегляд історії розрахунків.
- Внесення інформації про аварійно небезпечні об'єкти.
- Налаштування web-додатку.
- Авторизація.

На основі сформованих даних про варіанти використання та акторів, було розроблено Use Case діаграму, яка представлена рис. 2.4

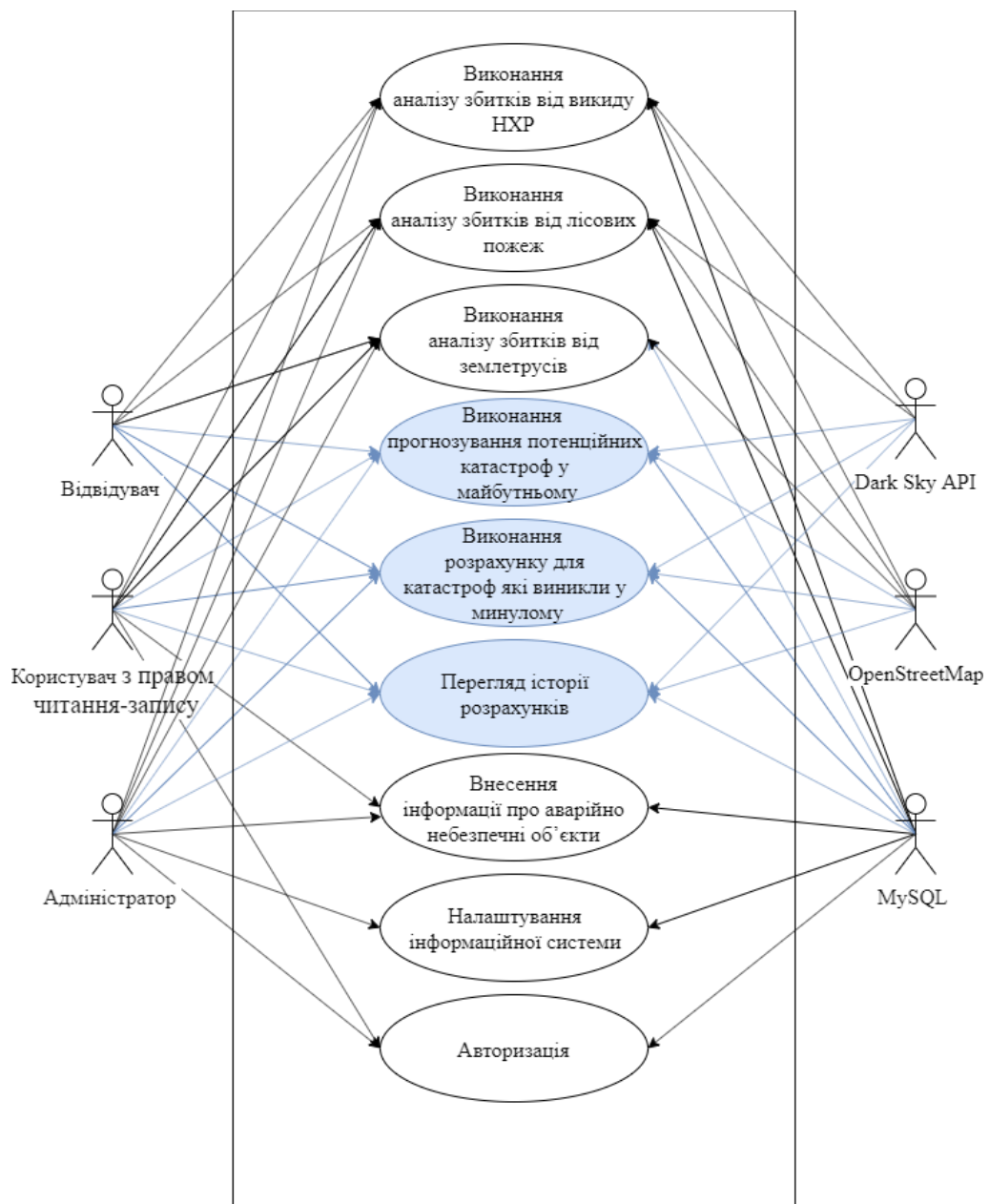


Рисунок 2.4 – Use Case Diagram

2.4 Проектування моделі бази даних

В попередній реалізації системи не біло необхідності логічно пов'язувати дані, тому статичні дані, які необхідні для розрахунків, зберігалися у вигляді інформаційної бази. З розширенням функціоналу системи, з'являється необхідність зберігати дані які повинні бути пов'язані між собою, саме тому інформаційна база яка наразі реалізована в системі, перетвориться в базу даних з декількома новими таблицями.

База даних буде містити 20 таблиць. З яких – 13 статичні таблиці, які зберігають дані необхідні для розрахунків, 7 логічно пов'язані таблиці, які зберігають дані про розрахунки які були виконані системою.

Статичні таблиці:

1. agriculture - мітить інформацію про те, які сільськогосподарські культури переважають у якій області України;
2. earthquakeeffects - мітить інформацію про наслідки землетрусу в залежності від його інтенсивності;
3. earthquake_life_support_system - мітить інформацію про те, яких наслідків землетрус нанесе системам життєзабезпечення;
4. earthquakedeath - мітить інформацію про смерті людей, в залежності від типу будівлі, в якій вони знаходилися під час землетрусу та його інтенсивності;
5. koef_kzm - мітить інформацію про коефіцієнт K_{zm} необхідний для розрахунку глибини зони враження хмари НХР;
6. nhr_cloud - мітить інформацію про коефіцієнти зменшення глибини розповсюдження хмари НХР;
7. pollution_duration - мітить інформацію про тривалість дії НХР визначається;
8. population_loss - мітить інформацію про втрати людей під час викиду НХР, в залежності від забезпеченості людей протигазами;

9. `tabledeptvalue`- містить інформацію про глибину розповсюдження хмари забрудненого повітря у разі аварії на хімічно небезпечних об'єктах;

10. `vertical_stability` - містить інформацію про вертикальну стабільність повітря;

11. `v_transfer` - містить інформацію про швидкість переносу повітряних мас;

12. `wood_damage` - містить інформацію про збиток нанесений деревам пожежею;

13. `wood_damage_char` - містить опис кожного виду враження дерев;

По сформованому переліку статичних сутностей до кожної з них було виділено їх атрибути та розроблено фрагмент моделі бази даних, яку можна побачити на рисунку 2.5.

Логічно пов'язані таблиці:

1. `factory` - містить інформацію про хімічно небезпечні заводи;

2. `users` - містить інформацію про користувачів в системі;

3. `cat_type` – містить інформацію про види катастрофи;

4. `cat_initial_data` – містить інформацію про вхідні дані, які є загальними для всіх видів катастроф;

5. `nhf_initial_data` - містить інформацію про вхідні дані, які є унікальними для викидів НХР;

6. `earth_initial_data` - містить інформацію про вхідні дані, які є унікальними для землетрусів;

7. `fire_initial_data` - містить інформацію про вхідні дані, які є унікальними для лісових пожеж.

Фрагмент моделі бази даних, яка відповідає за логічно пов'язані сутності можна побачити на рисунку 2.6.

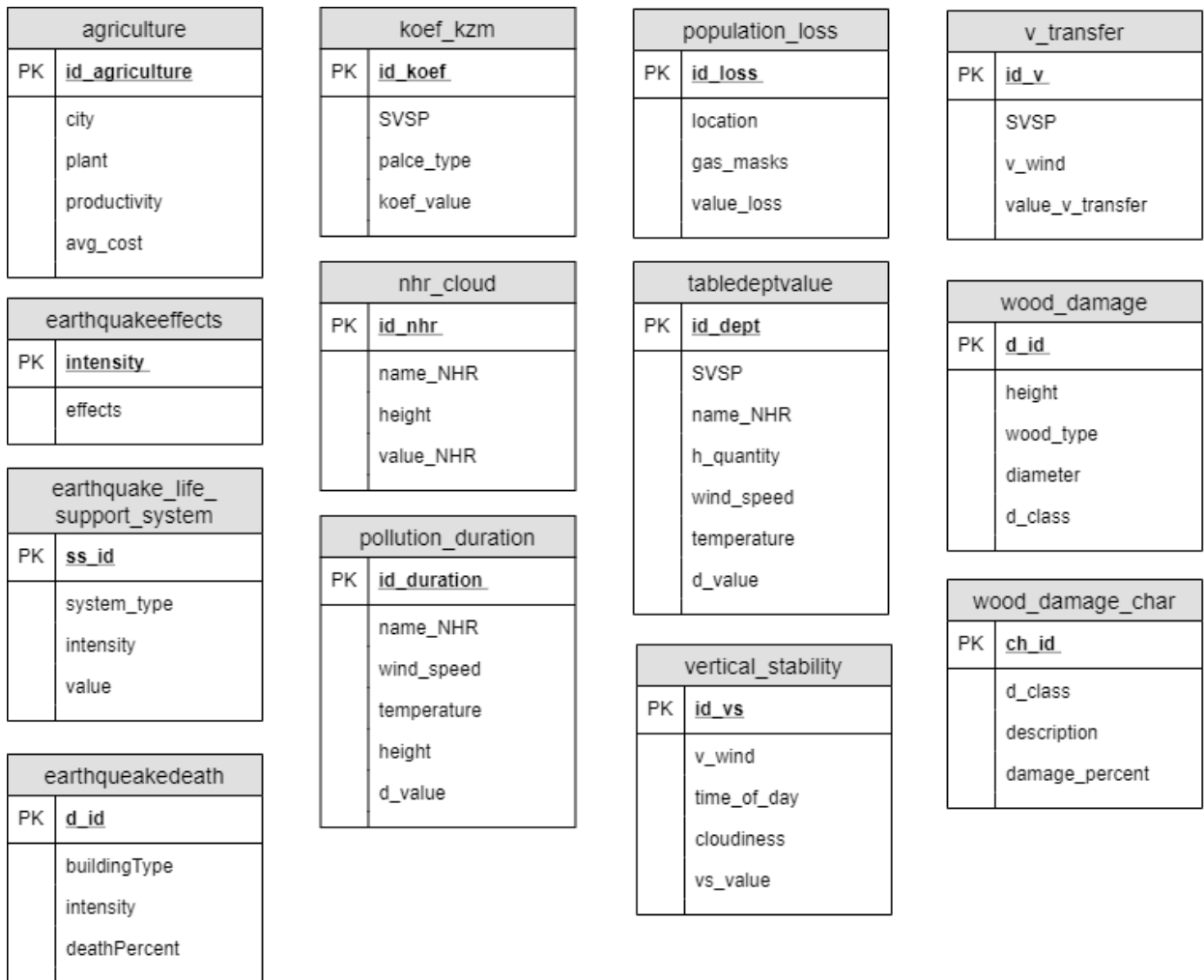


Рисунок 2.5 – Фрагмент моделі бази даних зі статичними сутностями

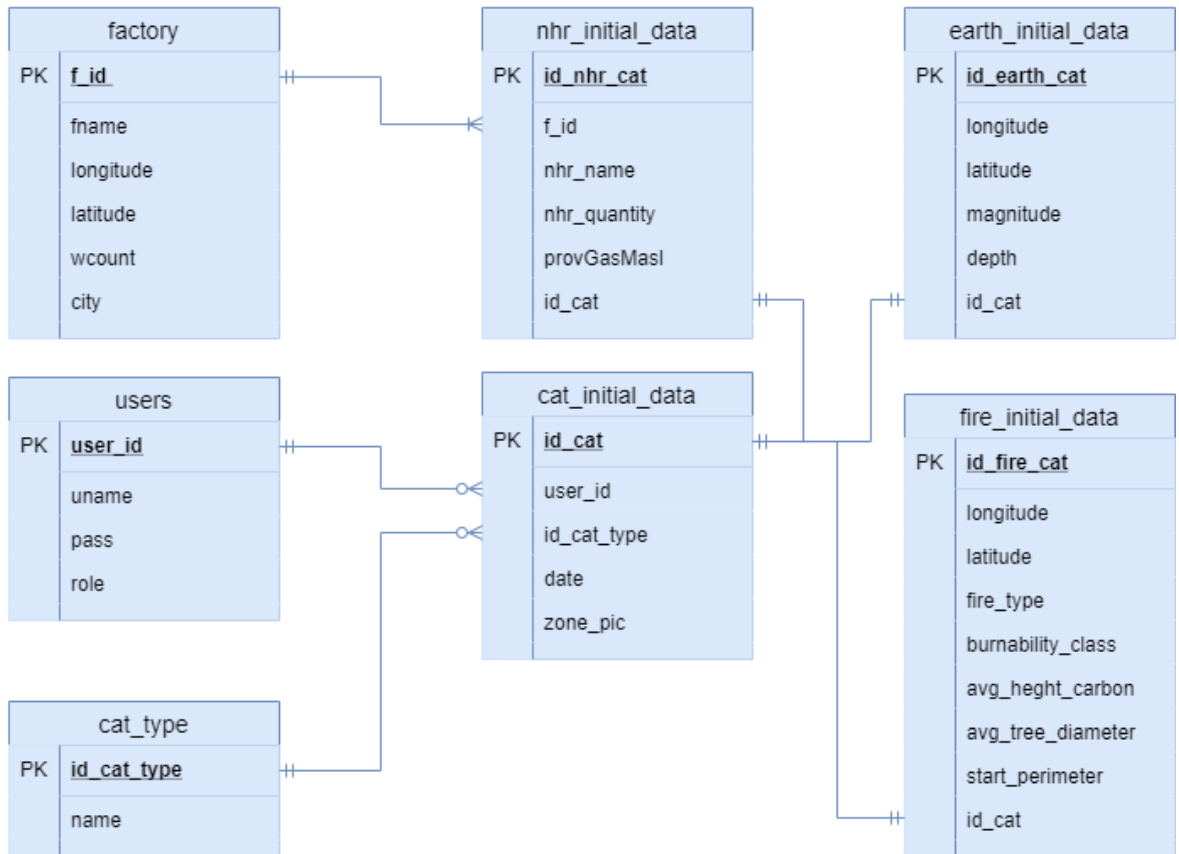


Рисунок 2.6 – Фрагмент моделі бази даних з логічно пов'язаними сутностями

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ЗБИТКІВ ВІД ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ

3.1 Програмна реалізація

Переглянути весь програмний код можна в додатку В або в гітхаб репозиторії за посиланням <https://github.com/alexe4540/diploma>.

3.1.1 Розробка бази даних

База даних додатку була розроблена з використанням СУБД MySQL. База складається з двох частин. Перший фрагмент є набором не пов'язаних між собою таблиць з даними. Кожна таблиця містить поля, тип яких залежить від даних, які в ній зберігаються, та містить статичні дані необхідні для проведення розрахунків. Другий фрагмент є набором логчно пов'язаних таблиць, які необхідні для збереження інформації про катастрофи, аналіз яких було проведено за допомогою розробленої ІС.

База даних «diploma» містить 13 таблиць з статичними даними:

1. agriculture
2. earthquakeeffects
3. earthquake_life_support_system
4. earthquakedeath
5. koef_kzm
6. nhr_cloud
7. pollution_duration
8. population_loss

9. tabledeptvalue
10. vertical_stability
11. v_transfer
12. wood_damage
13. wood_damage_char

База даних містить 7 логічно пов'язаних таблиць:

1. Таблиця «factory» містить інформацію про хімічно небезпечні заводи:
 - f_id – ідентифікатор (int) {PK};
 - fname – назва заводу (varchar);
 - longitude – широта на якій розміщений завод (double);
 - latitude – довгота на якій розміщений завод (double);
 - wcount – кількість працівників на заводі (float);
 - city – місто в якому розташовано завод (varchar);
2. Таблиця «users» містить інформацію про модераторів та адміністраторів в системі:
 - user_id – ідентифікатор (int) {PK};
 - uname – логін користувача (varchar);
 - pass – пароль користувача у вигляді хеш суми (varchar);
 - role – роль користувача (enum(«admin», «moderator»)).
3. Таблиця «cat_type» містить інформацію про вид катастрофи:
 - id_cat_type – ідентифікатор (int) {PK};
 - name – назва катастрофи (varchar).
4. Таблиця «cat_initial_data» містить спільну інформацію для всіх видів катастроф:
 - id_cat – ідентифікатор (int) {PK};
 - user_id – ідентифікатор користувача який виконав розрахунок (int) FK;
 - id_cat_type – ідентифікатор типу катастрофи (int) FK;
 - date – дата виникнення катастрофи (date);

- zone_pic – назва зображення яке відповідає катастрофі (varchar);
5. Таблиця «nhr_initial_data» містить інформацію специфічну для викидів НХР:
- id_nhr_cat – ідентифікатор (int) {PK};
 - f_id– ідентифікатор хімічно небезпечного заводу на якому виникла аварія (int) FK;
 - id_cat – ідентифікатор катастрофи із таблиці cat_initial_data, яка зберігає загальні дані про катастрофу (int) FK;
 - nhr_name – вид викинутої речовини (varchar);
 - nhr_quantity – кількість викинутої речовини (int);
 - provGasMasl – відсоток працівників забезпечених протигазами(int).
6. Таблиця «earth_initial_data» містить інформацію специфічну для землетрусів:
- id_earth_cat – ідентифікатор (int) {PK};
 - id_cat – ідентифікатор катастрофи із таблиці cat_initial_data, яка зберігає загальні дані про катастрофу (int) FK;
 - longitude – широта на якій виник землетрус (double);
 - latitude – довгота на якій виник землетрус (double);
 - magnitude – магнітуда (int);
 - depth – глибина на якій виникли поштовхи(int).
7. Таблиця «fire_initial_data» містить інформацію специфічну для лісових пожеж:
- id_fire_cat – ідентифікатор (int) {PK};
 - id_cat – ідентифікатор катастрофи із таблиці cat_initial_data, яка зберігає загальні дані про катастрофу (int) FK;
 - longitude – широта на якій виникла пожежа(double);
 - latitude – довгота на якій виникла пожежа(double);
 - fire_type– вид пожежі (varchar);
 - burnability_class – клас горимості лісових насаджень (int);

- avg_heght_carbon – середня висота нагару(int);
- avg_tree_diameter - середній діаметр деревостою(int);
- start_perimeter – початковий периметр загорання(int).

Переглянути фізичне відображення створеної інформаційної бази можна на рисунку 3.3.

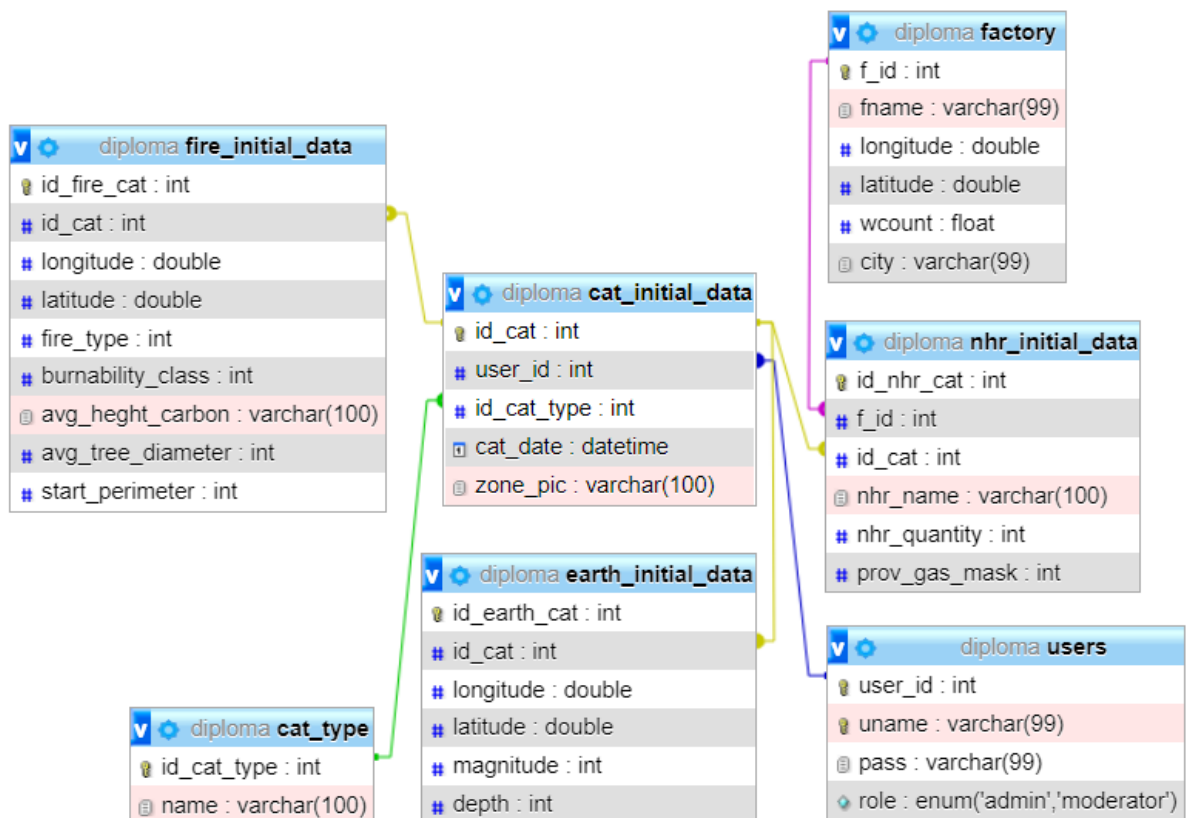


Рисунок 3.1 – Відображення фізичної реалізації нових таблиць бази даних у вікні дизайнера phpMyAdmin

3.1.2 Розширення модулю роботи з API погоди

Для отримання даних про погодні умови на момент виникнення катастрофи використовується API сервісу Dark Sky. За допомогою даного API можна отримати інформацію про погодні умови, вказавши довготу, широту, дату та час для яких нам потрібні дані.

Раніше, модуль роботи з API погоди отримував дані лише на момент розрахунку, вказуючи тільки довготу та широту. З розширенням функціоналу ІС та доданням можливості виконувати розрахунки для катастроф, які виникли у минулому, а також прогнозування катастроф, які можуть виникнути у майбутньому, виникла необхідність розширити існуючий модуль роботи з API, та додати можливість вказати дату та час виникнення катастрофи.

Для отримання інформації від API, клієнтська частина додатку генерує посилання, за допомогою функції `getWetherData(latitude, longitude , date)` (рис 3.2). Функція приймає параметри: `latitude` – широта, `longitude` – довгота, місцевості в якій треба дізнатися погодні умови, `date` – дата та час за які потрібно дізнатися погодні умови. Згенероване посилання передається до функції `getJSONWether(url)`.



```
1  const KEY = "199cb0b508acd58820ca7ed8fea3afb3";
2
3  function getWetherData(latitude, longitude, date){
4      const url=`https://api.darksky.net/forecast/${KEY}/${latitude},${longitude},${date}?
      lang=uk&units=auto`;
5
6      return getJSONWether(url);
7  }
```

Рисунок 3.2 – Функція `getWetherData`

Функція `getJSONWether(url)` (рис. 3.3) приймає на вхід, згенерований `url`, що є посиланням за яким сервер буде звертатися до API. Функція відправляє асинхронний POST запит до API та очікує відповідь від нього.


```
1  async function getJSONWether(url) {  
2      let result;  
3  
4      await $.ajax({  
5          type : "POST", //type of method  
6          url  : "/router/wetherRouter.php", //your page  
7          data : { url : url}, // passing the values  
8          success: function(res){  
9              result = res;  
10             }  
11         });  
12  
13         return result;  
14     }
```

Рисунок 3.3 – Функція getJSONWether

3.1.3 Розробка модулю роботи з історією розрахунків

Одним з головних нововведень системи є можливість збереження та перегляду історії розрахунків. Всі розрахунки які було проведено, відображаються на динамічній мапі з відповідними іконками, які відповідають певному виду катастрофи. В якості мапи, як і раніше використовується Open Street Map, так як окрім відображення самої мапи, надає широкі можливості роботи з нею. Наприклад, на мапу можна додати маркери які позначають певну точку, кастомізувати маркер, а також додати впливаючу підказку яка буде надавати певну інформацію.

Для роботи з мапою використовується JS бібліотека Leaflet, яка надає звичний функціонал переміщення та збільшення мапи, а також виставлення маркерів, на мапі та виконувати певні дії з ними.

Після завантаження сторінки історії, до неї одразу додається блок з мапою, який створюється за допомогою методу `map()` бібліотеки Leaflet. Даний метод виконує встановлення довготи та широти точки, яка буде в центрі доданої мапи, налаштовує приближення, та виконує рендер готової мапи на сторінку. Переглянути код описаного процесу можна на рисунку 3.4.

```
1  const map = L.map('map').setView([49.0384, 31.4513], 6);  
2  L.tileLayer('https://{s}.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png', {  
3      attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">  
    OpenStreetMap</a> contributors'  
4  }).addTo(map);  
5  L.control.scale().addTo(map);
```

Рисунок 3.4 – Код додавання мапи на сторінку

Модуль складається з 2 частин. Перша частина працює безпосередньо з мапою, до неї відносяться функції `fillMap(catList)` (Рис 3.5) та `createLink(id_cat_type, id_cat)` (рис. 3.6). Друга частина модулю виконує роботу вже фільтрами, для налаштування даних які будуть показані на мапі. Вона складається з слухачів, які реагують на зміни в полях фільтрів та кнопки яка відповідає за початок фільтрації даних(рис. 3.7).

Функція `fillMap` один параметер: `catList` – масив даних про катастрофи, які необхідно відобразити на мапу. Дана функція очищує мапу від маркерів, якщо такі присутні, формує список маркерів та кастомізує їх, за рахунок вказання кольору та іконки для маркеру який відповідає тому чи іншому типу катастрофи. За допомогою функції `createLink` генеруються посилання на сторінки з детальною інформацією про катастрофу. Після виконання попередніх налаштувань, за допомогою функції `marker.bindPopup()` згенеровані маркери додаються на мапу.

Функція `createLink()` приймає два параметри - `id_cat_type` та `id_cat`, та повертає згенероване посилання.

```

1  function fillMap(catList) {
2      markersLayer.clearLayers();
3
4      catList.forEach(cat => {
5          let markerColor, icon;
6
7          switch(Number(cat.id_cat_type)) {
8              case 1:
9                  icon = 'flask';
10                 markerColor = 'blue';
11                 break;
12              case 2:
13                 icon = 'mountain';
14                 markerColor = 'green';
15                 break;
16              case 3:
17                 icon = 'fire-alt';
18                 markerColor = 'red';
19                 break;
20              default:
21                 break;
22          }
23
24          const marker = L.marker([cat.latitude, cat.longitude], {icon: L.AwesomeMarkers.icon
25          ({icon, markerColor, prefix: 'fa'}) }).addTo(markersLayer);
26
27          const link = createLink(cat.id_cat_type, cat.id_cat);
28
29          marker.bindPopup(
30              `<b>Вид катастрофи:</b> ${cat.name}<br/>
31              <b>Дата:</b> ${cat.cat_date}<br/>
32              <b>Користувач:</b> ${cat.uname ?? 'Гість'}<br/>
33              <a href="${link}">Переглянути деталі</a>
34          `
35          )
36      })
37  }

```

Рисунок 3.5 – Функція fillMap

```

1  function createLink(id_cat_type, id_cat) {
2      let page = '';
3
4      switch(Number(id_cat_type)) {
5          case 1:
6              page = "nhrCalculator.php";
7              break;
8          case 2:
9              page = "earthquakeCalculator.php";
10             break;
11          case 3:
12             page = "fireCalculator.php";
13             break;
14          default:
15             break;
16      }
17
18      return ` /pages/${page}?id_cat=${id_cat}`;
19  }

```

Рисунок 3.6 – Функція createLink

Слухачі фільтрів оновлюють значення змінних які будуть використовуватися для витягнення даних з бази даних.

Слухач кліку кнопки очікує клік по кнопці «Застосувати фільтр» та викликає запит до бази даних з урахуванням вказаних значень в налаштуваннях фільтрів.

Результати виконання фільтрації даних для різних налаштувань фільтрації можна побачити на рисунках 3.8-3.10.

```

1  applyFilterButton.addEventListener('click', async () => {
2    const catlistJSON = await apiRequest('cat_list', 'dbrouter', {
3      id_cat_type: catType.value,
4      from_date: fromDate.value !== "" ? fromDate.value : null,
5      to_date: toDate.value !== "" ? toDate.value : null,
6    });
7
8    const catList = JSON.parse(catlistJSON);
9
10   fillMap(catList);
11 })

```

Рисунок 3.7 – Слухач кнопки «Застосувати фільтр»

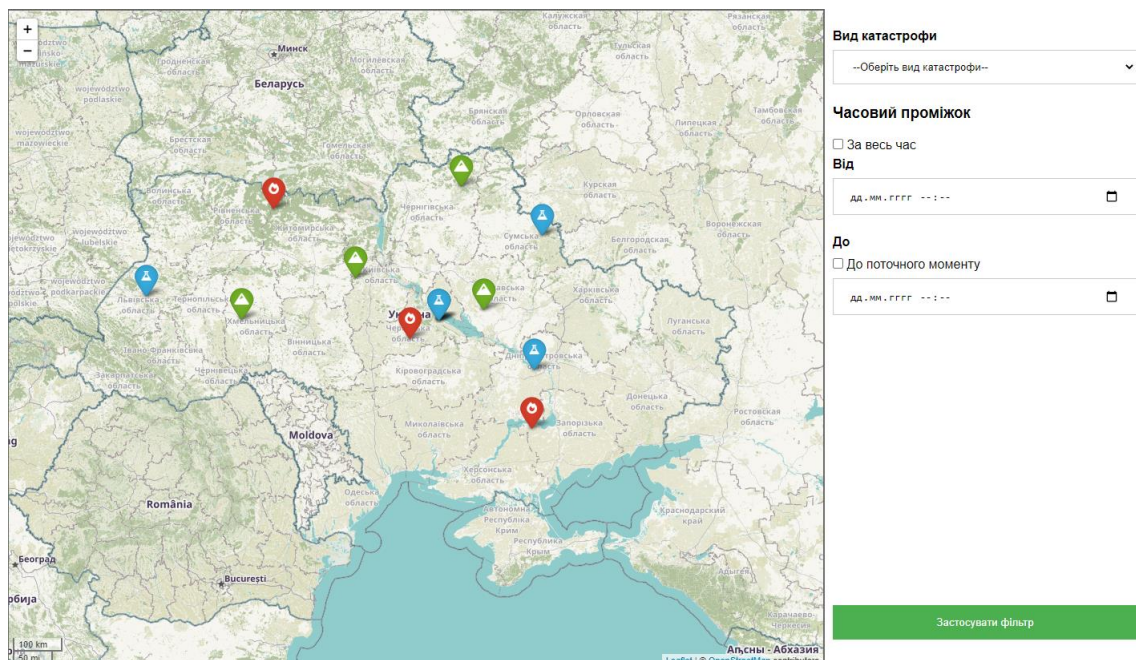


Рисунок 3.8 – Результат без задання фільтрів

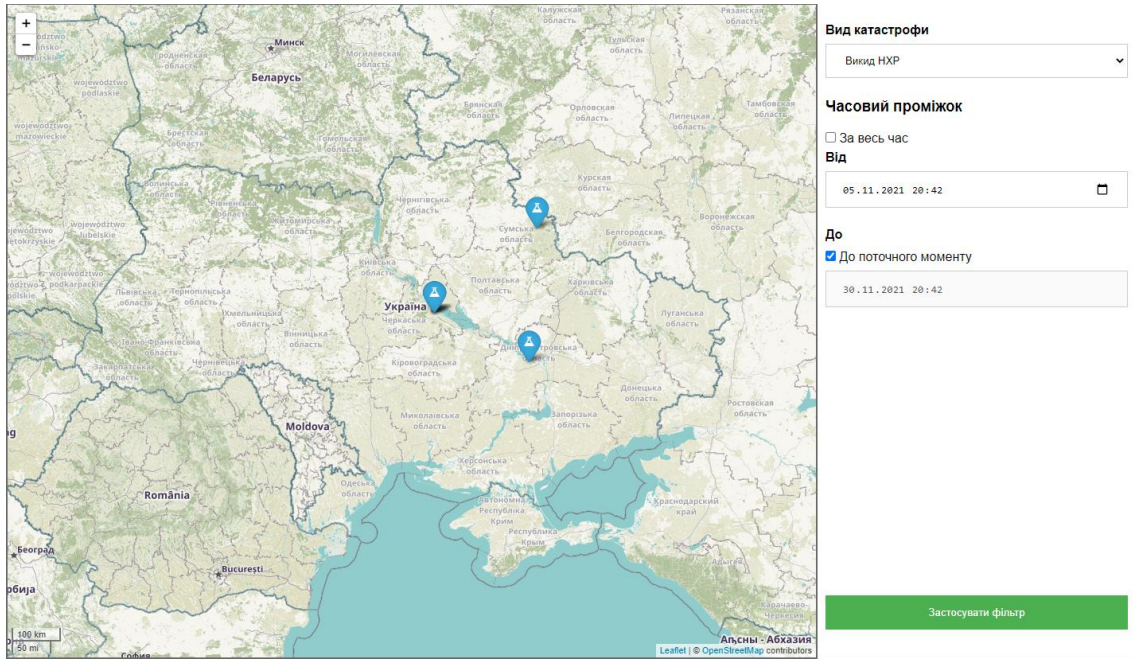


Рисунок 3.9 – Результат виконання фільтрації за типом катастрофи «Викид НХР» та в часовому проміжку від 5.11.21 до 30.11.21

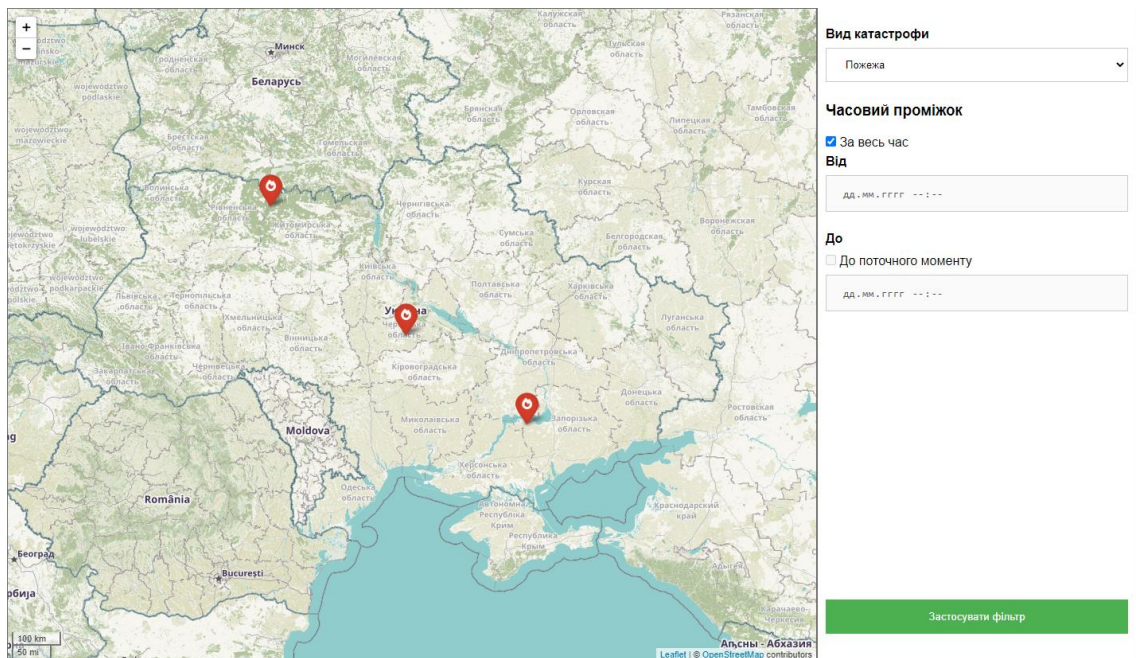


Рисунок 3.10 – Результат виконання фільтрації за типом катастрофи «Пожежа» за весь час роботи системи

3.2 Використання програмного додатку

Перша сторінка, яку бачить користувач, після того як ІС буде відкрита, буде головною сторінкою. Вона майже не зазнала змін, окрім нового пункту меню «Історія розрахунків» в шапці сайту.

Вигляд головної сторінки можна побачити на рисунку 3.11.

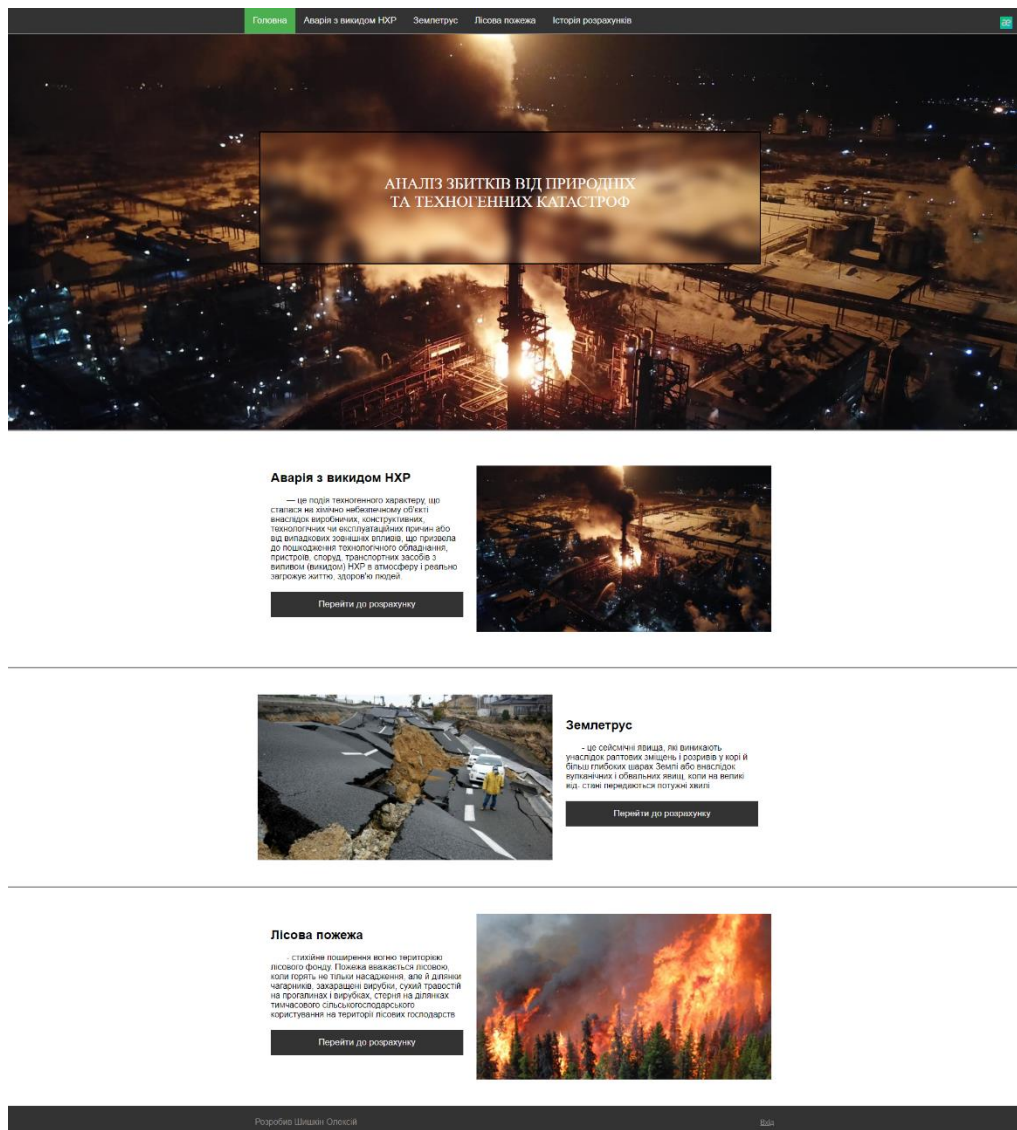


Рисунок 3.11 – Вигляд головної сторінки додатку

Відкривши сторінку аналізу збитків від аварії з викидом НХР (рис. 3.12), користувач бачить оновлену форму для введення початкових даних та мапу.

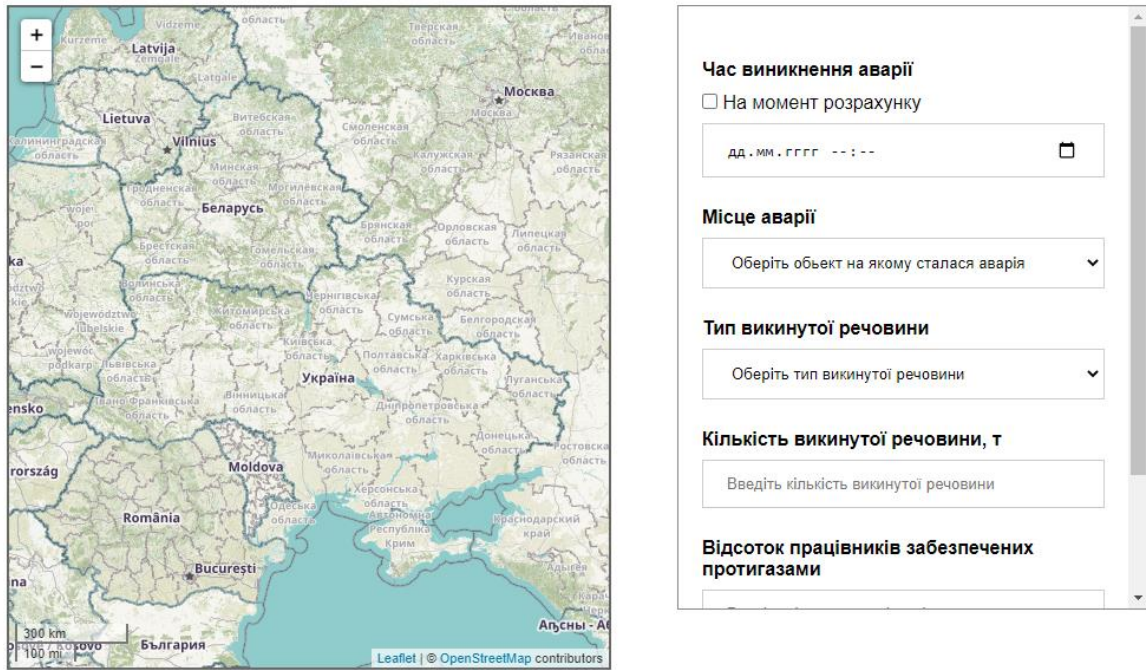


Рисунок 3.12 – Початковий вигляд сторінки аналізу збитків від аварії з викидом НХР

Для того, щоб надати користувачу можливість виконувати розрахунки для катастрофи, яка виникла у минулому, або моделювати катастрофу, яка може відбутися у майбутньому, форма з вхідними даними, була розширена новим полем:

– Час виникнення аварії – дозволяє ввести дату та час виникнення викиду НХР. Для моделювання катастрофи у майбутньому, поле має обмеження. Дата не повинна перевищувати поточну, більше ніж на місяць, так як похибка розрахунків буде занадто великою. Для розрахунків у минулому обмежень немає. Також, присутній чекбокс, який дозволяє автоматично вказати поточну дату та час.

Після введення даних та натискання кнопки «Розрахувати», на сторінку додається розділ з результатами розрахунків, який як і раніше містить мапу з відображеною на ній зоною враження, а також таблиці з результатами аналізу збитків.

Для того, щоб не заповнювати базу даних, непотрібними даними, а також надати користувачеві більшої варіативності, в кінці блоку з результатами розрахунків, присутня кнопка «Зберегти розрахунок». Після успішного збереження розрахунків, користувач бачить повідомлення «Дані успішно збережені».

Вигляд інформаційного блоку представлено на рисунку 3.13. Загальний вигляд оновленої сторінки представлено на рисунку 3.14.

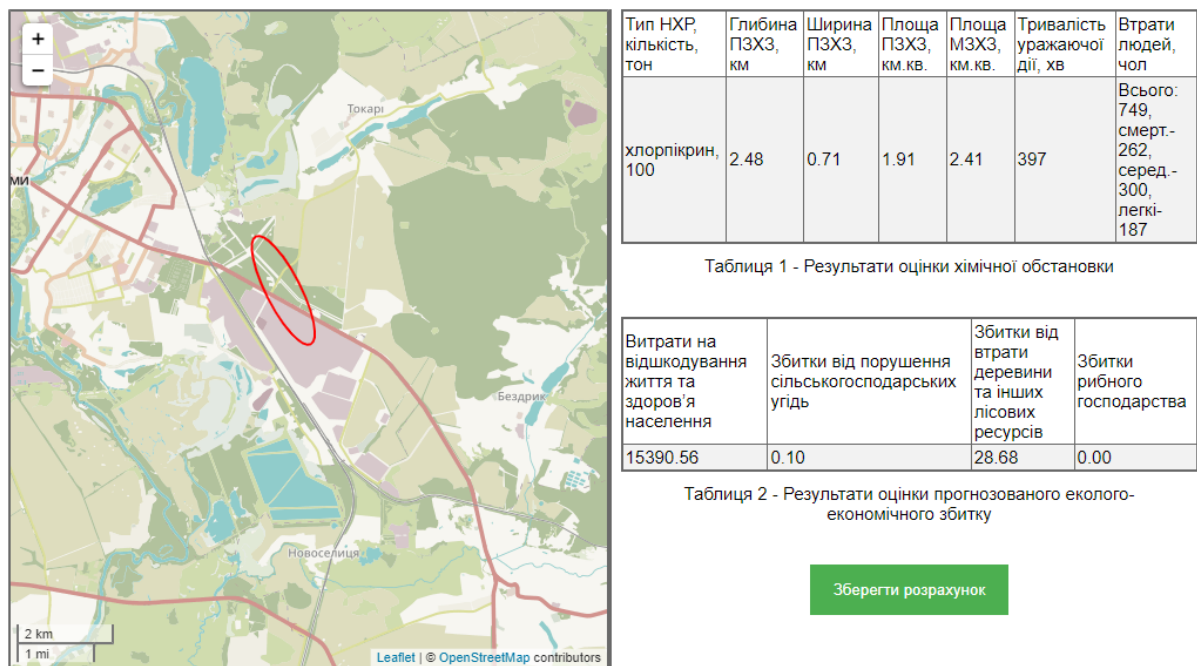
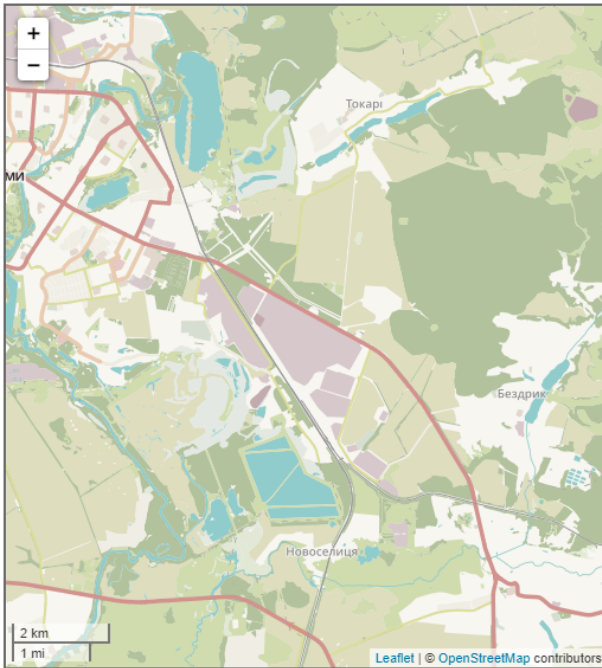


Рисунок 3.13 – Блок з результатами аналізу



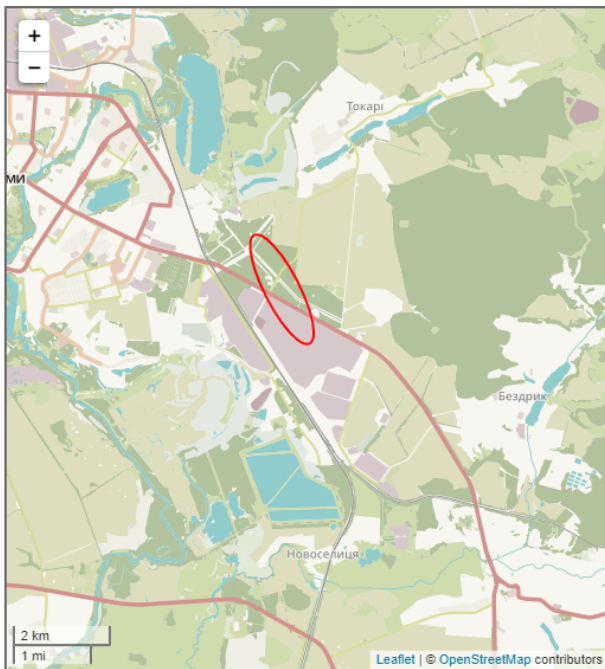
Час виникнення аварії
 На момент розрахунку
 18.07.2020 14:15

Місце аварії
 СумиХімпром

Тип викинутої речовини
 хлорпкрин

Кількість викинутої речовини, т
 100

Відсоток працівників забезпечених протигазами



Тип НХР, кількість, тон	Глибина ПЗХЗ, км	Ширина ПЗХЗ, км	Площа ПЗХЗ, км.кв.	Площа МЗХЗ, км.кв.	Тривалість уражаючої дії, хв	Втрати людей, чол
хлорпкрин, 100	2.48	0.71	1.91	2.41	397	Всього: 749, смерт.-262, серед.-300, легкі-187

Таблиця 1 - Результати оцінки хімічної обстановки

Витрати на відшкодування життя та здоров'я населення	Збитки від порушення сільськогосподарських угідь	Збитки від втрати деревини та інших лісових ресурсів	Збитки рибного господарства
15390.56	0.10	28.68	0.00

Таблиця 2 - Результати оцінки прогнозованого еколого-економічного збитку

Зберегти розрахунок

Рисунок 3.14 – Загальний вигляд оновленої сторінки аналізу збитків від НХР

Відкривши сторінку аналізу збитків від землетрусу (рис. 3.15), користувач бачить оновлену форму для введення початкових даних та мапу.

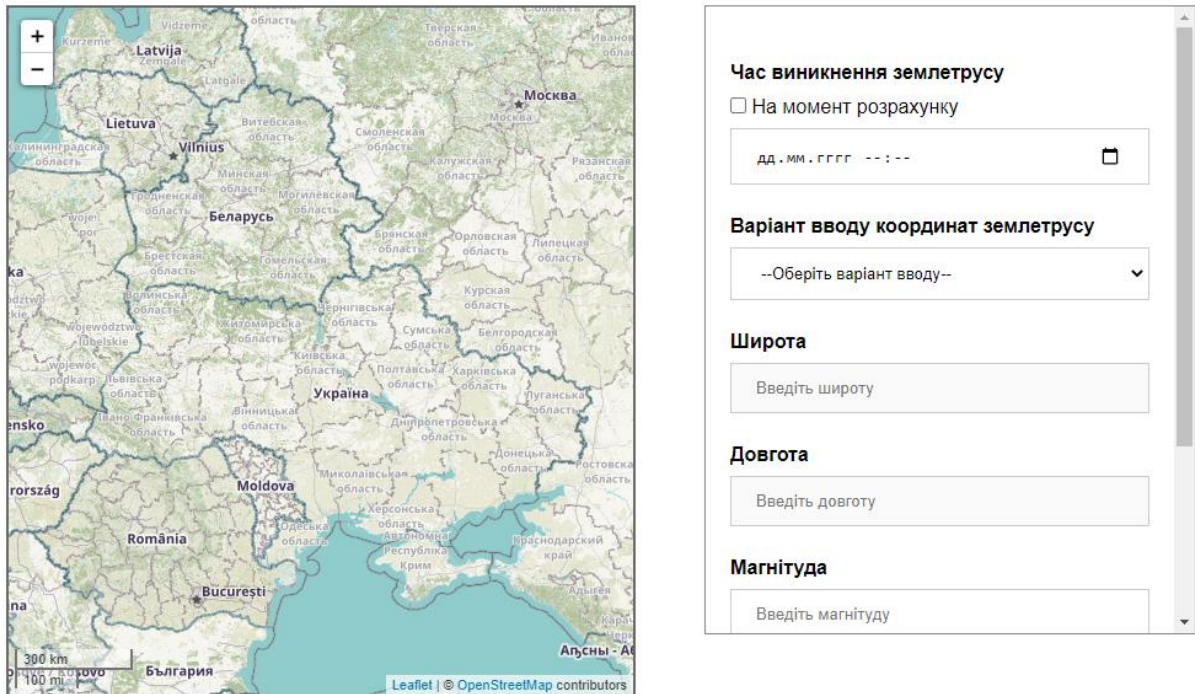
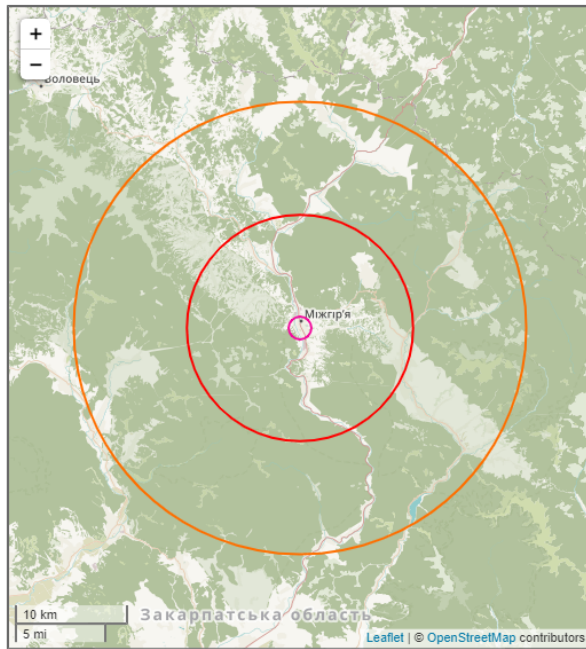


Рисунок 3.15 – Початковий вигляд сторінки аналізу збитків від землетрусу

Як і на сторінці аналізу збитків від викиду НХР, форма введення початкових даних, розширена новим полем для введення дати та часу виникнення землетрусу.

Блок з результатами розрахунку збитків, також був розширений можливістю зберегти результати розрахунків.

Вигляд інформаційного блоку представлено на рисунку 3.16. Загальний вигляд оновленої сторінки представлено на рисунку 3.17.



	Інтенсивність землетрусу	Наслідки землетрусу
В епіцентрі виникнення	8	Руйнування в будівлях: великі тріщини в стінах, падіння карнизів, димових труб. Зсуви і тріщини шириною до декількох сантиметрів на схилах гір.
В радіусі 10км від епіцентру	8	Руйнування в будівлях: великі тріщини в стінах, падіння карнизів, димових труб. Зсуви і тріщини шириною до декількох сантиметрів на схилах гір.
В радіусі 20км від епіцентру	8	Руйнування в будівлях: великі тріщини в стінах, падіння карнизів, димових труб. Зсуви і тріщини шириною до декількох сантиметрів на схилах гір.

Таблиця 1 - Наслідки землетрусу в кожній зоні, в залежності від інтенсивності на поверхні

Тип будівлі	Втрати населення в епіцентрі, %	Втрати населення в радіусі 10км, %	Втрати населення в радіусі 20км, %
Дерев'яні	0	0	0
Цегляні малоповерхові (1-2 поверху)	15	15	15
Цегляні багатоповерхові	3	3	3
Цегляні з неповною каркасною стіною	3	3	3

Таблиця 2 - Втрати населення в кожній зоні, в залежності від інтенсивності на поверхні, %


Система	Стійкість систем в епіцентрі, %	Стійкість систем в радіусі 10км, %	Стійкість систем в радіусі 20км, %
водопостачання	48/53	48/53	48/53
електропостачання	60/75	60/75	60/75
газопостачання	77/85	77/85	77/85
теплопостачання	50/77	50/77	50/77
транспорт	68/85	68/85	68/85
каналізація	82/90	82/90	82/90
зв'язок	82/90	82/90	82/90

Таблиця 3 - Стійкість систем життєзабезпечення, %

Зберегти розрахунок

Рисунок 3.16 – Блок з результатами аналізу

Головна
Аварія з викидом НХР
Землетрус
Лісова пожежа
Історія розрахунків



Маркер на мапі ▼

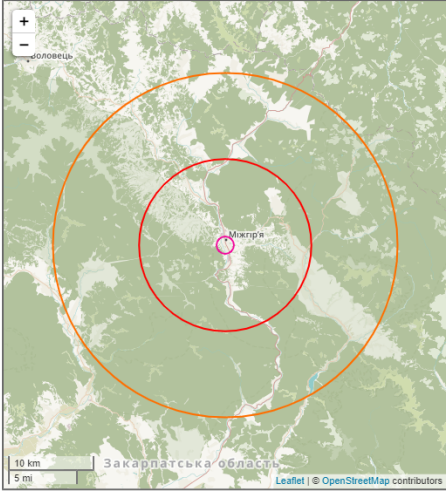
Широта
48.52424193937457

Довгота
23.499755859375004

Магнітуда
8

Глибина гіпоцентру
100

Розрахувати



	Інтенсивність землетрусу	Наслідки землетрусу
В епіцентрі виникнення	8	Руйнування в будівлях: великі тріщини в стінах, падіння карнизів, димових труб. Зсуви і тріщини шириною до декількох сантиметрів на схилах гір.
В радіусі 10км від епіцентру	8	Руйнування в будівлях: великі тріщини в стінах, падіння карнизів, димових труб. Зсуви і тріщини шириною до декількох сантиметрів на схилах гір.
В радіусі 20км від епіцентру	8	Руйнування в будівлях: великі тріщини в стінах, падіння карнизів, димових труб. Зсуви і тріщини шириною до декількох сантиметрів на схилах гір.

Таблиця 1 - Наслідки землетрусу в кожній зоні, в залежності від інтенсивності на поверхні

Тип будівлі	Втрати населення в епіцентрі, %	Втрати населення в радіусі 10км, %	Втрати населення в радіусі 20км, %
Дерев'яні	0	0	0
Цегляні малоповерхові (1-2 поверху)	15	15	15
Цегляні багатоповерхові	3	3	3
Цегляні з неповною каркасною стіною	3	3	3

Таблиця 2 - Втрати населення в кожній зоні, в залежності від інтенсивності на поверхні, %

Система	Стійкість систем в епіцентрі, %	Стійкість систем в радіусі 10км, %	Стійкість систем в радіусі 20км, %
водопостачання	48/53	48/53	48/53
електропостачання	60/75	60/75	60/75
газопостачання	77/85	77/85	77/85
теплопостачання	50/77	50/77	50/77
транспорт	68/85	68/85	68/85
каналізація	82/90	82/90	82/90
зв'язок	82/90	82/90	82/90

Таблиця 3 - Стійкість систем життєзабезпечення, %

Зберегти розрахунок

Розробив Шижкін Олександр
Вхід

Рисунок 3.17 – Загальний вигляд оновленої сторінки аналізу збитків від землетрусу

Відкривши сторінку аналізу збитків від лісової пожежі(рис. 3.18), користувач бачить оновлену форму для введення початкових даних та мапу.

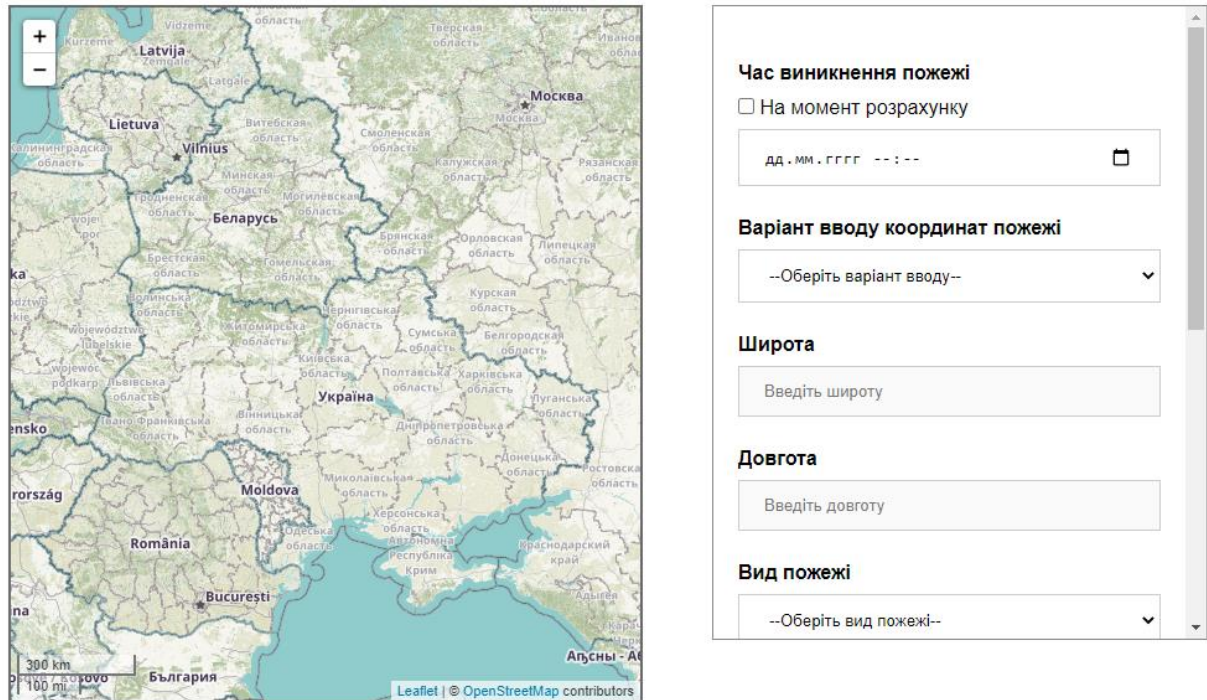
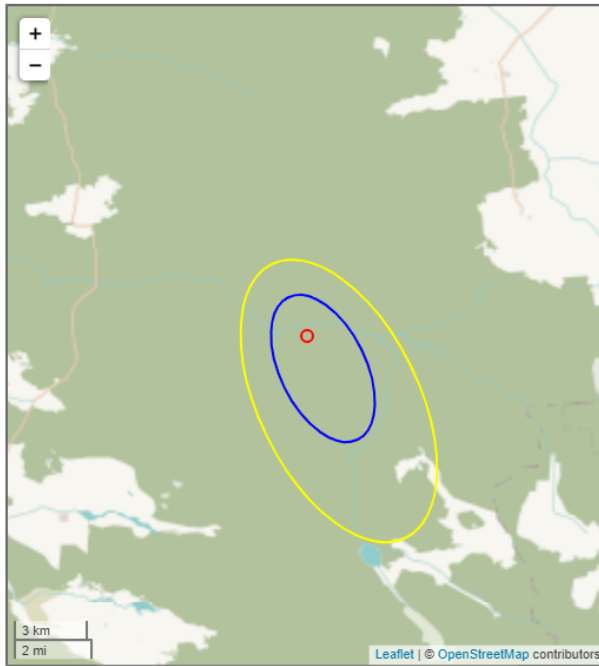


Рисунок 3.18 – Початковий вигляд сторінки аналізу збитків від лісової пожежі

Форма введення початкових даних, розширена можливістю введення дати та часу виникнення пожежі, для надання користувачу можливості виконати розрахунок для пожежі яка виникла в минулому, або змоделювати пожежу яка може виникнути у майбутньому.

Блок відображення результатів розрахунку, розширено можливістю збереження результати розрахунку.

Вигляд інформаційного блоку представлено на рисунку 3.19. Загальний вигляд оновленої сторінки представлено на рисунку 3.20.



Час	Площа лісової пожежі, Га	Периметр лісової пожежі, км
На момент виявлення(червоний)	18	1500
Через добу після виявлення(синій)	0.485	11004
Через дві доби після виявлення(жовтий)	1.683	20508

Таблиця 1 - Розповсюдження пожежі в часових проміжках

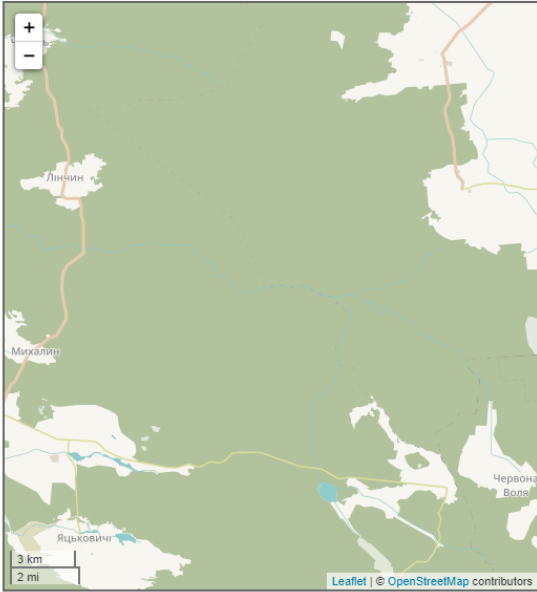
Клас пожежної небезпеки	Швидкість фронту пожежі, м/с	Швидкість флангу пожежі, м/с	Швидкість тилу пожежі, м/с	Відсоток деревини непридатної для реалізації
1	120	26	3	Древостой после пожара заметно изреживается; характеризуется сохранением жизнедеятельности значительного количества деревьев верхнего полога и отмиранием подчиненной части древостоя после низовых пожаров средней силы. Відсоток непридатної деревини = 31 - 70

Таблиця 2 - Загальні відомості про пожежу

Зберегти розрахунок

Рисунок 3.19 – Блок з результатами аналізу

Головна
Аварія з викидом НХР
Землетрус
Лісова пожежа
Історія розрахунків



Час виникнення пожежі

На момент розрахунку

05.12.2021 10:30

Варіант вводу координат пожежі

Маркер на мапі

Широта

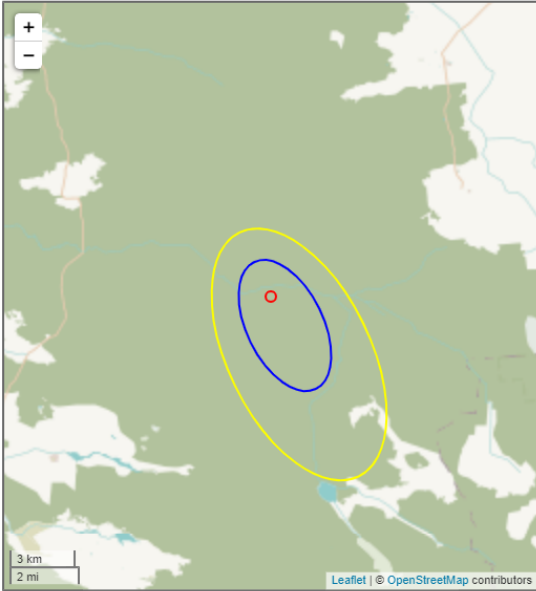
51.056213030485075

Довгота

27.09228515625

Вид пожежі

Верховий стійкий



Час	Площа лісової пожежі, Га	Периметр лісової пожежі, км
На момент виявлення(червоний)	18	1500
Через добу після виявлення(синій)	0.485	11004
Через дві доби після виявлення(жовтий)	1.683	20508

Таблиця 1 - Розповсюдження пожежі в часових проміжках

Клас пожежної небезпеки	Швидкість фронту пожежі, м/с	Швидкість флангу пожежі, м/с	Швидкість тилу пожежі, м/с	Відсоток деревини непридатної для реалізації
1	120	26	3	Древостой после пожара заметно изреживается; характеризуется сохранением жизнедеятельности значительного количества деревьев верхнего полога и отмиранием подчиненной части древостоя после низовых пожаров средней силы. Відсоток непридатної деревини = 31 - 70

Таблиця 2 - Загальні відомості про пожежу

Зберегти розрахунок

Розробив Шишкін Олексій
Вхід

Рисунок 3.20 – Загальний вигляд оновленої сторінки аналізу збитків лісової пожежі

Відкривши нову сторінку з історією розрахунків, користувач бачить перед собою мапу з маркерами та панель з фільтрами. (рис. 3.21).

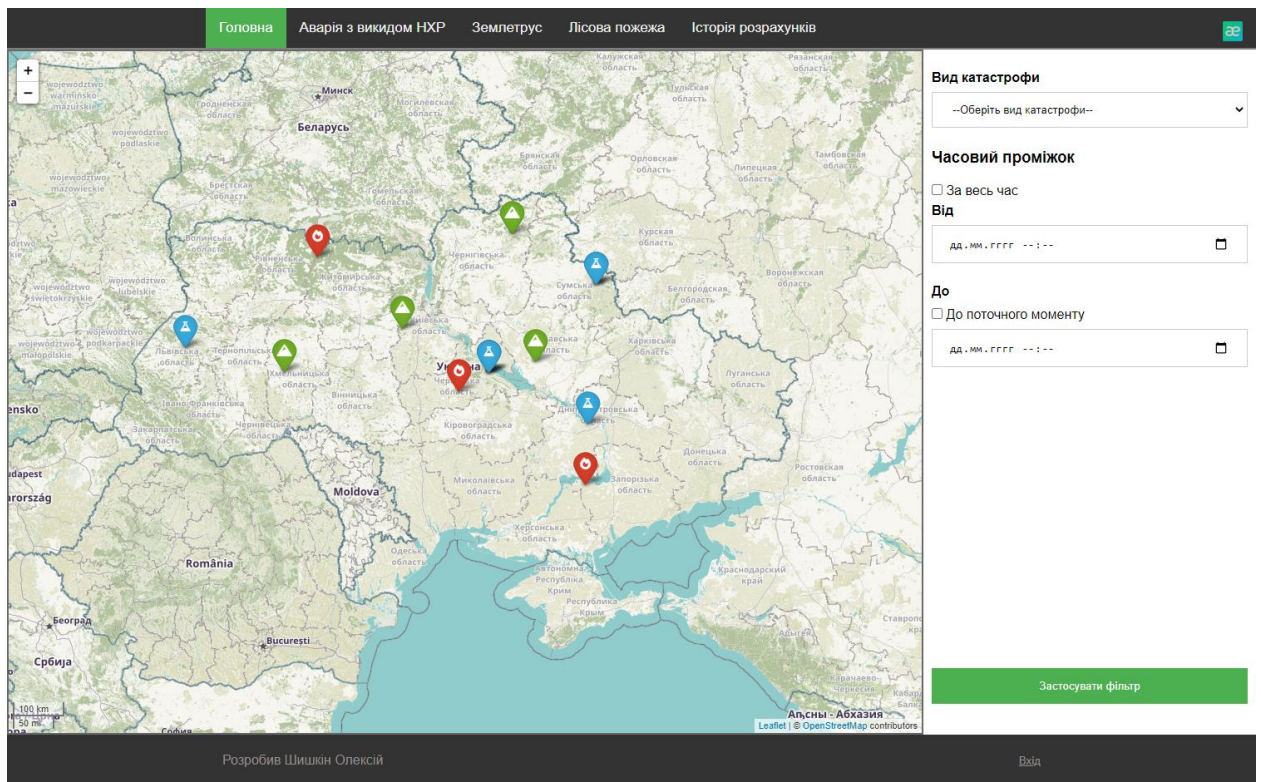


Рисунок 3.21 – Сторінка «Історія розрахунків»

Маркери на мапі поділені на 3 групи, кожна з яких відповідає певному виду катастрофи:

- червоні з іконкою вогню – лісові пожежі;
- зелені з іконкою гори – землетруси;
- блакитні з іконкою мензурки – аварії з викидом НХР.

Кожен маркер на мапі має впливаючу підказку, яка з’являється після натиску на нього (рис. 3.22). Підказка містить в собі коротку інформацію про катастрофу:

- вид катастрофи – вказує до якого виду відноситься катастрофа;
- дата – вказує дату коли катастрофа виникла;

- користувач – користувач який виконав розрахунок, якщо користувач не зареєстрований в системі, то буде відображено «Гість»;
- переглянути деталі – посилання на сторінку з детальною інформацією про розрахунок(рис. 3.23).

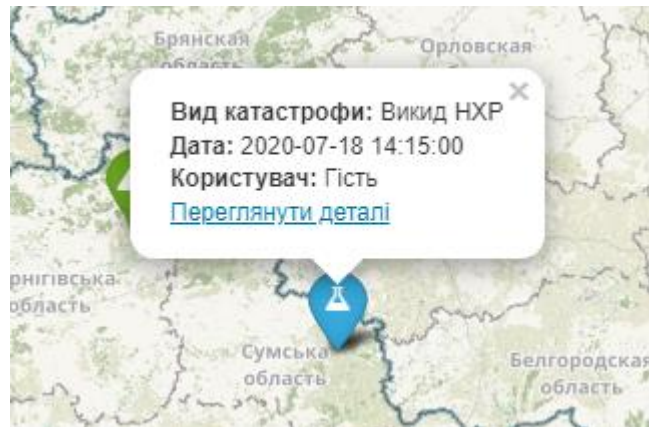


Рисунок 3.22 – Впливаюча підказка з короткою інформацією про катастрофу

Головна Аварія з викидом НХР Землетрус Лісова пожежа Історія розрахунків

Тип НХР	Глибина	Ширина	Площа	Площа	Тривалість	Втрати
кількість, тон	ПЗХЗ, км	ПЗХЗ, км	ПЗХЗ, км кв.	МЗХЗ, км кв.	уражачої дії, хв	людей, чол
хлорпкірин, 100	2.48	0.71	1.91	2.41	397	Всього: 749, смерт- 262, серед- 300, легк- 187

Таблиця 1 - Результати оцінки хімічної обстановки

Витрати на відшкодування життя та здоров'я населення	Збитки від порушення сільськогосподарських угідь	Збитки від втрати деревини та інших лісових ресурсів	Збитки рибного господарства
15390.56	0.10	28.68	0.00

Таблиця 2 - Результати оцінки прогнозованого еколого-економічного збитку

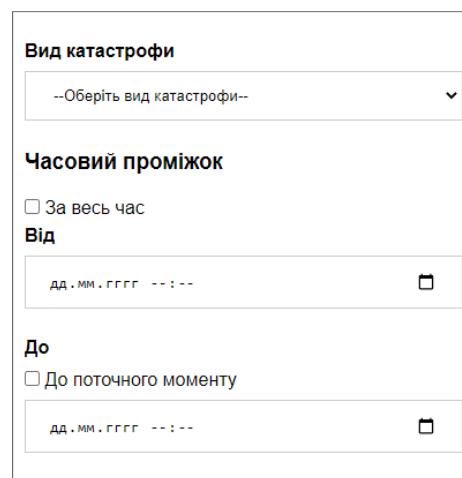
Розробив Шашен Олексій

Вхід

Рисунок 3.23 – Сторінка з детальною інформацією про катастрофу

Панель з фільтрами(рис. 3.24), має наступні поля для фільтрації результатів:

- вид катастрофи – який вид катастрофи потрібно показувати на мапі. Можна обрати як якийсь певний вид катастрофи, так і всі одразу;
- часовий проміжок – дозволяє вказати з якої по яку дату необхідно відобразити історію розрахунки. Можна переглянути історію за весь час, так і вказати певний проміжок часу.



The image shows a filter panel with the following sections:

- Вид катастрофи**: A dropdown menu with the text "--Оберіть вид катастрофи--" and a downward arrow.
- Часовий проміжок**: A section with a checkbox labeled "За весь час".
- Від**: A date input field with the placeholder "дд . мм . гggg -- : --" and a calendar icon.
- До**: A section with a checkbox labeled "До поточного моменту".
- До**: A date input field with the placeholder "дд . мм . гggg -- : --" and a calendar icon.

Рисунок 3.24 – Панель фільтрації

Після налаштування фільтрів користувач натискає кнопку «Застосувати фільтр», після чого на мапі відображаються всі катастрофи, які відповідають заданому фільтру (рис. 2.25).

Головна Аварія з викидом НХР Землетрус Лісова пожежа Історія розрахунків

Вид катастрофи
Землетрус

Часовий проміжок
 За весь час
Від
01.11.2021 10:55
До
 До поточного моменту
05.12.2021 10:55

Застосувати фільтр

Розробив Шишлян Олексій

Вхід

Рисунок 2.25 – Результати фільтрації

ВИСНОВКИ

В ході виконання кваліфікаційної роботи магістра було проведено реінжиніринг інформаційної системи аналізу збитків від техногенних та природніх катастроф.

Актуальність проведення реінжинірингу обумовлюється стабільним поступовим збільшенням кількості катастроф. Лише за 2021 рік в світі виникло понад 340 масштабних катастроф природного характеру, а також більше 120 техногенного. З яких, 4 масштабні катастрофи природного характеру та 2 техногенного в Україні [23]. Кожна катастрофа нанесла значних збитків, які можна було б зменшити в кілька разів, якби була можливість зберігати та аналізувати дані про катастрофи, які мали місце у минулому, або моделювати катастрофи, які тільки можуть відбутися у майбутньому.

Для визначення напрямку реінжинірингу інформаційної системи, який би задовольняв потребам користувачів, було проведено аналіз існуючого додатку, та визначено напрямки розширення функціоналу.

В ході реінжинірингу додатку було додано можливість виконувати розрахунки для катастроф, які виникли у минулому, моделювати катастрофи, які можуть виникнути у майбутньому, а також зберігати результати для виконаних розрахунків для подальшого їх аналізу. Результати роботи по реінжинірингу системи наведені у вигляді схем, зображень, діаграм та таблиць, розміщених вище в документі. Програмний код оновленої ІС наведено в додатку В, також його можна знайти за посиланням на гітхаб репозиторій: <https://github.com/alex4540/diploma>.

Для підвищення якості розробки було розроблено діаграму Ганта, оцінено ризики та розроблено календарний план робіт. Переглянути результати виконаної роботи можна у представленому документі та у додатках А.

Для підвищення якості нових модулів системі було проведено тестування. Усі виявлені дефекти було виправлено.

Область використання – центри екологічної інспекції та потенційно небезпечні промислові підприємства.

Результати роботи були апробовані на всеукраїнському конкурсі студентських наукових робіт з «Інформаційних систем і технологій» в місті Хмельницький та на всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2021» в місті Хмельницький.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Основы безопасной жизнедеятельности [Электронный ресурс]. – Режим доступа: <http://www.diagram.com.ua/info/obzhd/obzhd60.shtml> – 17.04.2020р. – Класификация катастроф. ОБЖД.
- 2 Методология IDEF0 [Электронный ресурс]. – Режим доступа: <https://itteach.ru/bpwin/metodologiya-idef0> – 17.04.2020р. – Методология IDEF0.
- 3 Ткачук А.І. Цивільний захист / А.І. Ткачук, О.В. Пуляк – Кропивницький, 2017 – 144с.
- 4 UML Use Case Diagrams [Электронный ресурс]. – Режим доступа: <https://www.uml-diagrams.org/use-case-diagrams.html> – 18.04.2020р. – Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors).
- 5 Responsibility Assignment Matrix [Электронный ресурс]. – Режим доступа: <http://acqnotes.com/acqnote/careerfields/responsibility-assignment-matrix1> – 19.04.2020р. – Responsibility Assignment Matrix (RAM) - AcqNotes.
- 6 Ю. А. Опанасюк, Еколого-економічна оцінка техногенних катастроф [Текст]: дисертація на здобуття наукового ступеня канд. економічних наук / — Суми : СумДУ, 2015. — 242 с.
- 7 Методика розрахунку параметрів лісових пожеж [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/metodika-rascheta-parametrov-lesnyh-pozharov-kak-dinamicheskikh-protsesov-na-poverhnosti-zemli-s-ispolzovaniem-dannyh-kosmicheskogo/viewer> – 18.04.2020р. – Методика расчета параметров лесных пожаров как динамических процессов на поверхности земли.
- 8 Microsoft Developer Network [Электронный ресурс] – Режим доступа до ресурсу: <https://msdn.microsoft.com>.

- 9 Stack Overflow [Электронный ресурс] – Режим доступа до ресурсу: <http://stackoverflow.com/>.
- 10 Habrahabr [Электронный ресурс] – Режим доступа до ресурсу: <https://habrahabr.ru/>.
- 11 Github [Электронный ресурс] – Режим доступа до ресурсу: <https://github.com/>.
- 12 Офіційний сайт «Портал знань» [Электронный ресурс] – Режим доступа : <http://www.znannya.org/>.
- 13 Шевчук А., Касьянов А. Patterns via C#. Приемы объектно-ориентированного проектирования: ITVDN / А. Шевчук, Д. Охрименко, 2015 – 288с.
- 14 Орлов, С.А. Технология разработки программ / С. А. Орлов – СПб.: Питер, 2004.
- 15 Федорчук, Д. А. Разработка WEB приложений на PHP и MySQL / Д.А. Федорчук - СПб. : Корона-принт, 2003.
- 16 Овицкий, Н.Ф. PHP как инструмент для работы с MySQL базой данных / Л.З Овицкий - М. : Вильямс, 2008.
- 17 Прохоров, В.В. Разработка WEB приложений с нуля / В.В. Прохоров - Мн.: БГУИР, 2009.
- 18 Crockford D. JavaScript: The Good Parts / D. Crockford - O'Reilly Media, Inc, USA, 2008.
- 19 Flanagan D. JavaScript: The Definitive Guide : Activate Your Web Pages / D. Flanagan - O'Reilly Media, Inc, USA, 2011.
- 20 Stefanov S. JavaScript Patterns / S. Stefanov - O'Reilly Media, Inc, USA, 2010.
- 21 Nixon R. Learning PHP, MySQL & JavaScript 5e : With jQuery, CSS & HTML5 / R. Nixon - O'Reilly Media, Inc, USA, 2018
- 22 Leaflet [Электронный ресурс] – Режим доступа до ресурсу: <https://leafletjs.com/>.

23 Dark Sky API [Электронный ресурс] – Режим доступа до ресурсу:
<https://darksky.net/dev>

24 Part 1. How to Use the Dark Sky API [Электронный ресурс] – Режим
доступу до ресурсу: [https://github.com/initialstate/wunderground-
sensehat/wiki/Part-1.-How-to-Use-the-Dark-Sky-API](https://github.com/initialstate/wunderground-sensehat/wiki/Part-1.-How-to-Use-the-Dark-Sky-API)

25 Part 3. Hyper Local Weather Dashboard [Электронный ресурс] –
Режим доступа до ресурсу: [https://github.com/initialstate/wunderground-
sensehat/wiki/Part-3.-Hyper-Local-Weather-Dashboard](https://github.com/initialstate/wunderground-sensehat/wiki/Part-3.-Hyper-Local-Weather-Dashboard)

26 Leaflet: Make a web map! [Электронный ресурс] – Режим доступа до
ресурсу: <https://maptimeboston.github.io/leaflet-intro/>

27 EM-DAT Public [Электронный ресурс] – Режим доступа до ресурсу:
<https://public.emdat.be/data>

28 MDN Web Docs [Электронный ресурс] – Режим доступа до ресурсу:
<https://developer.mozilla.org/ru/>

29 W3schools [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.w3schools.com/>

30 Refactoring guru[Электронный ресурс] – Режим доступа до ресурсу:
<https://refactoring.guru/uk/design-patterns>

ДОДАТОК А

ПЛАНУВАННЯ РОБІТ

1 Ідентифікація ідеї проекту

Метою кваліфікаційної роботи магістра є вдосконалення інформаційної системи, що використовується зацікавленими особами для аналізу збитків від техногенних або природніх катастроф.

Дипломний проект призначений для того, щоб надати користувачу зручний спосіб виконувати оцінку збитків від катастроф які виникли у минулому, проводити прогнозування збитків від катастроф які можуть виникнути у майбутньому а також переглядати історію розрахунків для більш детального аналізу.

Інформаційна система повинна бути реалізована у повній мірі та доступний в мережі Інтернет. Інформаційна система повинна складатися із взаємозалежних розділів із чітко розділеними функціями.

1.2 Деталізація мети методом SMART

Конкретна (Specific). Вдосконалити інформаційну систему, для аналізу збитків від катастроф техногенного або природнього характеру.

Вимірювана (Measurable). Результатом виконання проекту, має бути інформаційна система, отримати доступ до якої, можна ввівши адресу сайту до адресного рядку, в свою чергу, інформаційна система мусить бути

розміщений на хостингу, пов'язані з ним бази даних(БД), форми та модулі мають бути вірно налаштовані та поєднані.

Досяжна (Achievable). Проект потребує розробника із знанням HTML, CSS, мови програмування JavaScript, PHP, MySQL та об'єктно-орієнтованого програмування і програмного забезпечення (ПЗ), яке використовується для розроблення даного додатку.

Реалістична (Relevant). Мету реально досягнути, так як розробка інформаційної системи за допомогою сучасних Web-технологій, не вимагає надзвичайно складних дій або велику кількість ресурсів.

Обмежена у часі (Time—framed). Обмеженість в часі зумовлена рішенням замовника, щоб отримати програмне забезпечення як можна швидше.

1.3 Планування змісту структури робіт IT—проекту (WBS)

Структурна декомпозиція робіт (work breakdown structure, WBS) - це ієрархічна структура робіт, побудована з метою логічного розподілу усіх робіт з виконання проекту і подана у графічному вигляді. Це сукупність декількох рівнів, кожний з яких формується в результаті розподілу роботи попереднього рівня на її складові. Елементом найнижчого рівня є група робіт, або так званий робочий пакет (work package).

Виконаємо побудову WBS структури, у якій зазначимо всі виконувані роботи в залежності від головних етапів:

1. Формування технічного завдання – перший етап роботи над проектом, під час якого обумовлюються основні вимоги до майбутнього web-додатку. Технічне завдання включає в себе: визначення предметної області, призначення продукту, визначення засобів розробки, визначення цільової аудиторії, розробка макету інтерфейсу та інше.

2. Вдосконалення програмного продукту – написання нових модулів, що забезпечують функціонування програмного продукту:

- Розробка та наповнення бази даних.
- Розробка модулю розрахунку збитків від катастроф які виникли у минулому.
- Розробка модулю прогнозування катастроф які можуть виникнути у майбутньому.
- Розробка модулю збереження та перегляду історії розрахунків.
- Розробка користувальницького інтерфейсу нових модулів.

3. Тестування – перевірка роботи програмного продукту, виявлення помилок:

- Модульне тестування.
- Інтеграційне тестування.
- Системне тестування.

4. Здача програмного продукту:

- Розробка документації.
- Введення в експлуатацію.

WBS-структура для даного проекту представлена на рис. 1.1

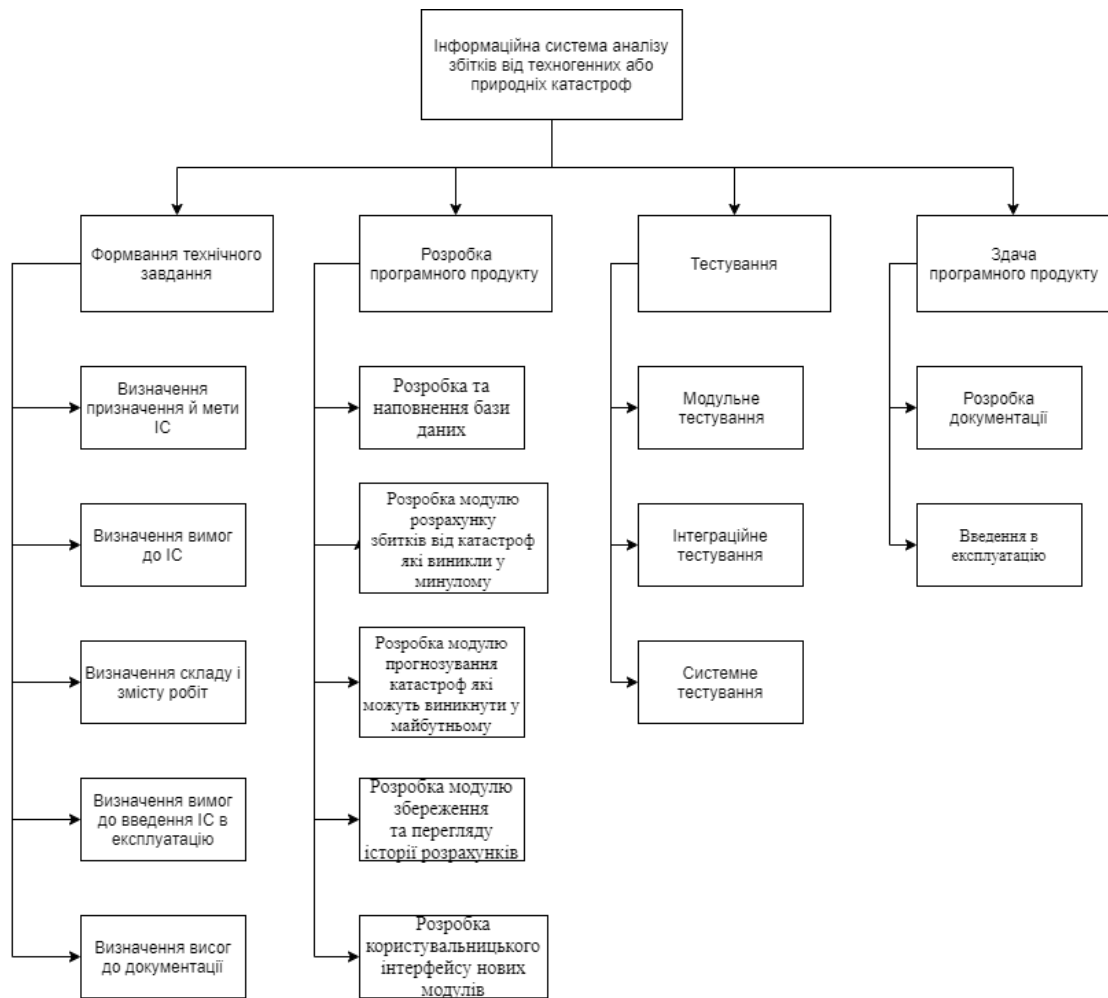


Рисунок 1.1 - WBS-структура web-додатку аналізу збитків від природних або техногенних катастроф

Наступним кроком розробки структури проекту є визначення організаційної структури (OBS) проекту.

Організаційна структура проекту (OBS) – це графічне відображення учасників проекту, які відповідають та які відносяться до певної частини робіт. Визначається з елементарних робіт(робіт найнижчого рівня) кожної з гілок WBS. На верхньому рівні знаходиться керівник та команда управління проектом, на наступних – відповідальні особи.

OBS-структура для даного проекту представлена на рисунку 1.2

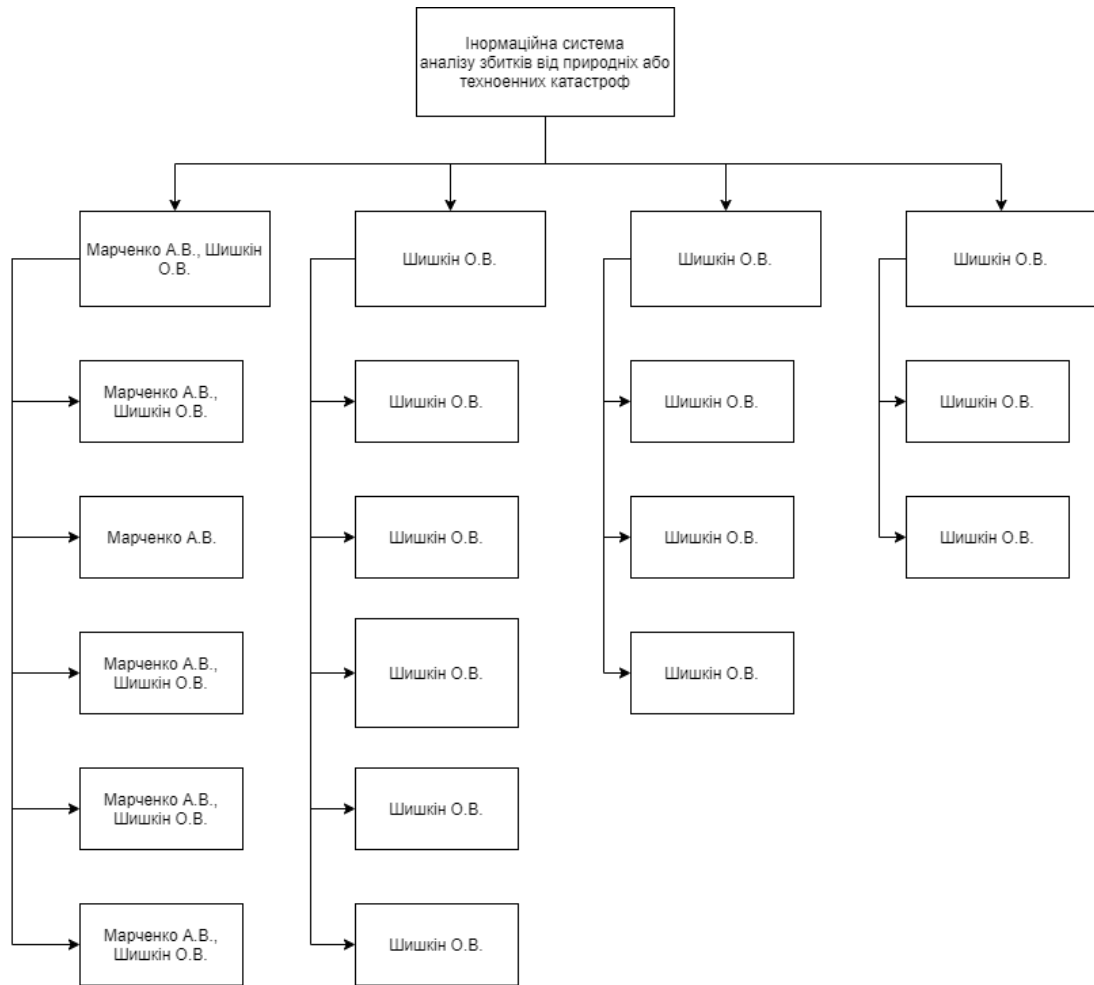


Рисунок 1.2 - Організаційна структура виконавців

1.4 Побудова календарного графіку виконання ІТ—проекту

Діаграма Ганта – лінійна діаграма, яка показує задачі проекту, що представляються часовими відрізками, які характеризуються датами початку та закінчення робіт. Діаграма дозволяє відслідковувати відсоток робіт, виконаних по кожному завданню. Цей графік представлено за допомогою програмного засобу MS Project.

Графік виконання дипломного проекту представлено у вигляді Діаграми Ганта на рисунку 1.3. Список робіт для побудови діаграми Ганта представлено на рисунку 1.4

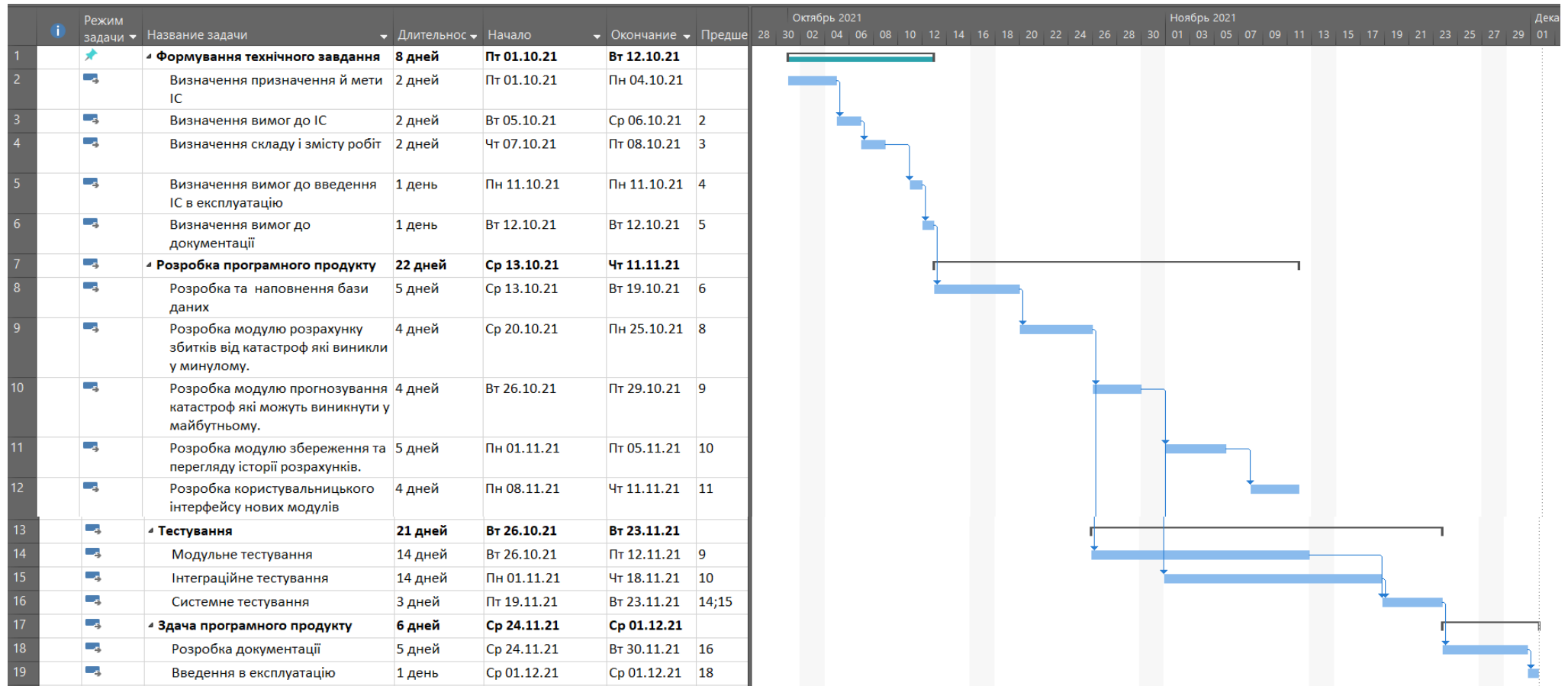


Рисунок 1.3 – Діаграма Ганта

1.5 Планування ризиків проекту

Ризик – це імовірна подія, яка у випадку своєї появи негативно або позитивно вплине на проект.

Процес управління ризиками включає наступні етапи:

1. Ідентифікація.
2. Процес оцінювання ризиків, який включає в себе якісний, кількісний аналіз.
3. Заходи реагування на ризики.
4. Моніторинг заходів і ризиків.

Ризики проекту представлені у таблиці 1.1

Таблиця А.1 – Ризики проекту

№	Ризик	Вплив	Ймовірність	РР	Усунення
1	Неповне ТЗ	4	2	С	Тісний контакт з замовником, що дозволить при виявленні неповноти ТЗ доповнити його
2	Недостатня кваліфікація кадрів	2	2	Н	Відвідування кадрами компанії курсів підвищення кваліфікації
3	Помилки під час планування	3	2	С	Піл час планування адекватно оцінювати можливості
4	Незаплановані завдання які потрібно виконати на пізніх етапах виконання проекту	3	1	Н	Оцінити вплив завдання на хід роботи над проектом, та в разі необхідності домовитися з замовником про збільшення часу на виконання проекту
5	Складні у виконанні або неможливі для виконання завдання	5	2	С	Оновити кадри або домовитися з замовником відмінити завдання

Ризики представити за допомогою RBM матриці (Risk Breakdown Matrix).

Класифікація ризиків за імовірністю виникнення:

- слабоймовірний;
- малоймовірний;
- імовірні;
- досить імовірні;
- майже імовірні.

Класифікація ризиків за величиною втрат:

- мінімальна;
- низька;
- середня;
- висока;
- максимальна.

Виконаємо класифікацію ризиків даного проекту. Для цього складемо таблицю А.2.

Таблиця А.2 – Probability / Impact Matrix

5					
4					
3					
2		2	3	1	5
1			4		
Probability / Impact	1	2	3	4	5

ДОДАТОК Б**АПРОБАЦІЯ**





Міністерство освіти і науки України
Хмельницький національний університет

СЕРТИФІКАТ



Шишкін Олексій Васильович

учасник XIII Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2021»

Голова оргкомітету АПКН-2021

СИНЮК Олег Миколайович
проректор Хмельницького національного
університету з наукової роботи,
доктор технічних наук, професор

*м. Хмельницький
15-16 жовтня 2021*

E-mail: apkt.khnu@gmail.com

ДОДАТОК В

ЛІСТИНГ ПРОГРАМНОГО КОДУ

Index.php

```

<? session_start() ?>

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="css/main.css">
  <link rel="stylesheet" href="css/mainpage.css">
</head>

<body>
  <?php include ('./php/module/header.html'); ?>

  <div class="Site-content">
    <section>
      <div class="">
        <div class="img-bg">
          <div class="box">
            <h1>Аналіз збитків від природних <br>
              та техногенних катастроф</h1>
          </div>
        </div>
      </div>
    </section>
    <section>
      <div class="container">
        <div class="decription-block">
          
          <div class="decription">
            <h1>Аварія з викидом НХР</h1>
            <p> – це подія техногенного характеру, що
              сталася на хімічно небезпечному об'єкті
              внаслідок виробничих, конструктивних,
технологічних чи експлуатаційних причин або від
              випадкових зовнішніх впливів, що призвела до
пошкодження технологічного обладнання,
              пристроїв, споруд, транспортних засобів з виливом
(викидом) НХР в атмосферу і реально
              загрожує життю, здоров'ю людей.</p>
              <a href="/pages/nhrCalculator.php">Перейти до
розрахунку</a>
          </div>
        </div>
      </div>
    </section>
  </div>
</div>

```

```

        </div>
    </section>
    <section>
        <div class="container">
            <div class="decription-block">
                
                <div class="decription">
                    <h1>Землетрус</h1>
                    <p> - це сейсмічні явища, які виникають унаслідок
глибоких шарах Землі
на великі від-
розрахунку</a>
                    або внаслідок вулканічних і обвальних явищ, коли
                    стані передаються потужні хвилі</p>
                    <a href="/pages/earthquakeCalculator.php">Перейти до
                </div>
            </div>
        </div>
    </section>
    <section>
        <div class="container">
            <div class="decription-block">
                
                <div class="decription">
                    <h1>Лісова пожежа</h1>
                    <p>- стихійне поширення вогню території лісового
фонду. Пожежа вважається
ділянки чагарників, захарашені вирубки,
на ділянках тимчасового
лісових господарств</p>
                    лісовою, коли горять не тільки насадження, але й
                    сухий травостій на прогалинах і вирубках, стерня
                    сільськогосподарського користування на території
                    <a href="fireCalculator.php">Перейти до
                </div>
            </div>
        </div>
    </section>
</div>
<? include('./php/module/footer.php'); ?>
</body>

</html>

```

nhrCalculator.php

```

<?php
session_start();
include("../config/conection.php");
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

```



```

        for ($i = 0; $i < count($enum); $i++) {
            ?>
            <option value="<?php echo $enum[$i] ?>">
                <?php echo $enum[$i] ?>
            </option>
            <?php
            }
            ?>
        </select>

        <label for="countSub"><b>Кількість викинутої
речовини, т</b></label>
        <input type="number" placeholder="Введіть кількість
викинутої речовини" name="countSub" required>

        <label for="wcount"><b>Відсоток працівників
забезпечених протигазами</b></label>
        <input type="number" placeholder="Введіть відсоток
працівників" name="wcount" required>

        <button id="calculateButton">Позрахувати</button>
    </div>
</div>
<?php
}
?>
</section>
<section id="resultSection">
    <div class="nhr-container container">
        <div id="mapImg" class="pagePart"></div>
        <div class="pagePart">
            <div id="resultTable" class="vertical"></div>
            <div id="damageTable" class="vertical"></div>
            <?php
            if (!isset($_GET['id_cat'])) {
                ?>
                <button id="saveButton" class="save-button">Зберегти
позрахунок</button>
            <?php
            }
            ?>
        </div>
    </div>
</section>
</div>

<?php include('../php/module/footer.php'); ?>

<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.0/jquery.min.js'></scr
ipt>
    <script
src='https://cdnjs.cloudflare.com/ajax/libs/skycons/1396634940/skycons.min.js
'></script>
    <script src="../../js/helpers/readJSON.js"></script>
    <script src="../../js/calc/getWether.js"></script>
    <script src="../../js/getters/localGetter.js"></script>
    <script src="../../js/getters/serverGetter.js"></script>
    <script src="../../js/helpers/drawOnCanvas.js"></script>
    <script src="../../js/helpers/drawTable.js"></script>
    <script src="../../js/calc/workWithMap.js"></script>
    <script src="../../js/calc/parceCanvas.js"></script>

```

```

    <script src="../../js/lib/html2canvas.min.js"></script>
    <script src="../../js/calc/nhr.js"></script>
    <script src="../../js/module/nhr.js"></script>
</body>

</html>

```

fireCalculatr.php

```

<?php
session_start();
include("../config/conection.php");
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css" />
    <script
src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js"></script>
    <link rel="stylesheet" href="../../css/main.css">
    <title>Document</title>
</head>

<body>
    <?php include('../../php/module/header.html'); ?>
    <div class="Site-content">
        <section>
            <?php
            if (!isset($_GET['id_cat'])) {
                ?>
                <div class="nhr-container container">
                    <div id="map" class="pagePart"></div>
                    <div class="form-container">
                        <div id="msg"></div>
                        <label for="fdate"><b>Час виникнення
пожежі</b></label>
                        <div class="checkbox">

```



```

        <input type="checkbox" id="dateNow"
name="dateNow">
        <label for="dateNow" style="display:
inline">На момент розрахунку</label>
    </div>
    <input name="fdate" type="datetime-local"
required>

    <label for="fname"><b>Варіант вводу
координат пожежі</b></label>
    <select name="inputType" id="inputType">
        <option value="" disabled selected>--
Оберіть варіант вводу--</option>
        <option value="marker">Маркер на
мапі</option>
        <option value="coordinates">Координати
місця виникнення</option>
    </select>

    <label for="latitude"><b>Широта</b></label>
    <input type="number" placeholder="Введіть
широту" name="latitude" required disabled>

    <label
for="longitude"><b>Довгота</b></label>
    <input type="number" placeholder="Введіть
довготу" name="longitude" required disabled>

    <label for="fireType"><b>Вид
пожежі</b></label>
    <select id="fireType" name="fireType">
        <option value="" disabled selected>--
Оберіть вид пожежі--</option>
        <option value="1">Верховий
стійкий</option>
        <option value="2">Верховий
побіжний</option>
        <option value="3">Низовий</option>
    </select>

    <label for="burnabilityClass"><b>Клас
горимості лісових насаджень</b></label>

```

```

                                <select                                id="burnabilityClass"
name="burnabilityClass">
                                <option value="" disabled selected>--
Оберіть клас горимості--</option>
                                <option value="1">Чисті і з домішкою
листяних порід хвойні насадження (крім модринових)</option>
                                <option value="2">Чисті з домішкою
хвойних порід листяні насадження, а також модринові насадження</option>
                                </select>

                                <label        for="avgHeghtCarbon"><b>Середня
висота нагару</b></label>
                                <select                                id='avgHeghtCarbon'
name="avgHeghtCarbon">
                                <option        value=""            disabled
selected>Оберіть середню висоту нагару</option>

                                <?php
                                $enum_params                                =
mysql_fetch_assoc(mysql_query($dbc, "SHOW COLUMNS FROM wood_damage
WHERE Field = 'height'"));
                                preg_match("/^enum\(\('(.*)'\)\)$/",
$enum_params['Type'], $res);
                                $enum = explode("'", "'", $res['1']);

                                for ($i = 0; $i < count($enum); $i++) {
                                ?>
                                <option value="<?php echo $enum[$i]
?>">
                                <?php echo $enum[$i] ?>
                                </option>
                                <?php
                                }
                                ?>
                                </select>

                                <label        for="avgTreeDiameter"><b>Середній
діаметр деревостю</b></label>
                                <select                                id='avgTreeDiameter'
name="avgTreeDiameter">
                                <option        value=""            disabled
selected>Оберіть середній діаметр деревостю</option>

```

```

        <?php
            $enum_params =
mysqli_fetch_assoc(mysqli_query($dbc, "SHOW COLUMNS FROM wood_damage
WHERE Field = 'diameter'"));
            preg_match("/^enum\(\\'(.*)\'\)/",
$enum_params['Type'], $res);
            $enum = explode("'", "'", $res['1']);
            for ($i = 0; $i < count($enum); $i++) {
                ?>
                <option value="<?php echo $enum[$i]
?>">
                    <?php echo $enum[$i] ?>
                </option>
            }
        ?>
    </select>

    <label for="startPerimeter"><b>Початковий
периметр зони загорання, м</b></label>
    <input type="number" placeholder="Введіть
початковий периметр загорання" name="startPerimeter" required>

    <button
id="calculateButton">Розрахувати</button>
    </div>
</div>
<?php
}
?>
</section>
<section id="resultSection">
    <div class="nhr-container container">
        <div id="mapImg" class="pagePart"></div>
        <div class="pagePart">
            <div id="resultTable" class="vertical"></div>
            <div id="damageTable" class="vertical"></div>
        <?php
            if (!isset($_GET['id_cat'])) {

```

```

        ?>
        <button id="saveButton" class="save-
button">Зберегти розрахунок</button>
        <?php
        }
        ?>
    </div>
</div>
</section>
</div>

<?php include('../php/module/footer.php'); ?>

<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.0/jquery.min.js'><
/script>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/skycons/1396634940/skycons.mi
n.js'></script>
<script src="/js/helpers/readJSON.js"></script>
<script src="/js/calc/getWether.js"></script>
<script src="/js/helpers/readJSON.js"></script>
<script src="/js/getters/localGetter.js"></script>
<script src="/js/getters/serverGetter.js"></script>
<script src="/js/helpers/drawOnCanvas.js"></script>
<script src="/js/helpers/drawTable.js"></script>
<script src="/js/lib/html2canvas.min.js"></script>
<script src="/js/calc/workWithMap.js"></script>
<script src="/js/module/forestFire.js"></script>
<script src="/js/calc/forestfire.js"></script>
</body>

</html>

```

earthquakeCalculator.php

```

<?php
session_start();
include("../config/conection.php");
?>

<!DOCTYPE html>

```

```

<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet"
href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css" />
  <script
src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js"></script>
  <link rel="stylesheet" href="../../../css/main.css">
  <title>Document</title>
</head>

<body>
  <?php include('../../../php/module/header.html'); ?>
  <div class="Site-content">
    <section>
      <?php
      if (!isset($_GET['id_cat'])) {
        ?>
        <div class="nhr-container container">
          <div id="map" class="pagePart"></div>
          <div class="form-container">
            <div id="msg"></div>

            <label for="fdate"><b>Час виникнення
землетрусу</b></label>

            <div class="checkbox">
              <input type="checkbox" id="dateNow"
name="dateNow">

              <label for="dateNow" style="display:
inline">На момент розрахунку</label>
            </div>
            <input name="fdate" type="datetime-local"
required>

            <label for="fname"><b>Варіант вводу
координат землетрусу</b></label>

            <select name="inputType" id="inputType"
required>

```

```

                <option value="" disabled selected>--
Оберіть варіант вводу--</option>
                <option value="marker">Маркер на
мапі</option>
                <option value="coordinates">Координати
місця виникнення</option>
            </select>

            <label for="latitude"><b>Широта</b></label>
            <input type="number" placeholder="Введіть
широту" name="latitude" required disabled>

            <label
for="longititude"><b>Довгота</b></label>
            <input type="number" placeholder="Введіть
довготу" name="longititude" required disabled>

            <label
for="magnitude"><b>Магнітуда</b></label>
            <input type="number" placeholder="Введіть
магнітуду" name="magnitude" required>

            <label for="depth"><b>Глибина
гіпоцентру</b></label>
            <input type="number" placeholder="Введіть
глибину гіпоцентру" name="depth" required>

            <button
id="calculateButton">Розрахувати</button>
        </div>
    </div>
    <?php
    }
    ?>
</section>
<section id="resultSection" style="display: none">
    <div class="nhr-container container">
        <div id="mapImg" class="pagePart"></div>
        <div class="pagePart">
            <div id="resultTable" class="vertical"></div>
            <div id="damageTable" class="vertical"></div>

```

```

        <div                                id="lifeSupportSystems"
class="vertical"></div>
        <?php
            if (!isset($_GET['id_cat'])) {
                ?>
                <button    id="saveButton"    class="save-
button">Зберегти розрахунок</button>
                <?php
                    }
                ?>
            </div>
        </div>
    </section>
</div>

<?php include('../php/module/footer.php'); ?>

<script src="/js/helpers/readJSON.js"></script>
<script src="/js/getters/localGetter.js"></script>
<script src="/js/getters/serverGetter.js"></script>
<script src="/js/helpers/drawOnCanvas.js"></script>
<script src="/js/helpers/drawTable.js"></script>
<script src="/js/lib/html2canvas.min.js"></script>
<script src="/js/calc/workWithMap.js"></script>
<script src="/js/module/earthquake.js"></script>
<script src="/js/calc/earthquake.js"></script>
</body>

</html>

```

historyPage.php

```

<?php
session_start();
include("../config/conection.php");
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">

```

```

        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
        <link rel="stylesheet"
href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css" />
        <script
src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js"></scrip
t>
        <script
src="https://kit.fontawesome.com/461af6709f.js"
crossorigin="anonymous"></script>
        <link rel="stylesheet" href="../../../css/main.css">
        <link rel="stylesheet" href="../../../css/history.css">
        <link rel="stylesheet"
href="../../../css/lib/leaflet.awesome-markers.css">
        <title>Document</title>
</head>

<body>
    <?php include('../../../php/module/header.html'); ?>

    <div class="Site-content">
        <section>
            <div class="history-map-container">
                <div id="map" class=""></div>
                <div class="filter-container">
                    <label for="catType"><b>Вид
катастрофи</b></label>
                    <select name="catType" id="catType">
                        <option value="" disabled
selected>--Оберіть вид катастрофи--</option>
                        <option value="0">Всі</option>
                        <option value="1">Викид
НХР</option>
                        <option
value="2">Землетрус</option>

```



```

        <option
value="3">Пожежа</option>
    </select>

    <h3>Часовий проміжок</h3>

    <div class="checkbox">
        <input          type="checkbox"
id="allTime" name="allTime">
            <label          for="allTime"
style="display: inline">За весь час</label>
        </div>

        <label
for="fromDate"><b>Від</b></label>
        <input          id="fromDate"
name="fromDate" type="datetime-local">

        <label
for="toDate"><b>До</b></label>
        <div class="checkbox">
            <input          type="checkbox"
id="dateNow" name="dateNow">
                <label          for="dateNow"
style="display: inline">До поточного моменту</label>
            </div>
            <input id="toDate" name="toDate"
type="datetime-local">

            <button
id="applyFilterButton">Застосувати фільтр</button>
        </div>
    </div>

</section>

```

```

</div>

<?php include('../php/module/footer.php'); ?>

<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.0/jquery.min.js'></script>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/skycons/1396634940/skycons.min.js'></script>
<script src='../js/lib/leaflet.awesome-markers.min.js'></script>
<script src='../js/helpers/readJSON.js'></script>
<script
src='../js/getters/serverGetter.js'></script>
<script src='../js/calc/historyMap.js'></script>
</body>

</html>

```

workWithMap.js

```

async function workWithMap(coordinates, scale) {
  let latitude = coordinates.latitude;
  let longitude = coordinates.longitude;

  map.setView([latitude, longitude], scale);

  let promise = new Promise((resolve, reject) => {
    setTimeout(() => resolve("готово!"), 1000)
  });
  let result = await promise;

  goFunc();

  return new Promise((resolve, reject) => {
    setTimeout(() => resolve("готово!"), 1000)
  });
}

```

```

}

let map;

(() => {
  const params = new
URLSearchParams(document.location.search.substring(1));
  const id_cat = params.get("id_cat");

  if (!id_cat) {
    map = L.map('map').setView([50.452554, 30.561522], 5);

L.tileLayer('https://{s}.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png', {
  attribution: '&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
contributors'
}).addTo(map);
L.control.scale().addTo(map);
}
})();

function goFunc() {
  html2canvas(document.querySelector("#map"), {
    allowTaint: true,
    useCORS: true,
  }).then(function (canvas) {
    let mapImg = document.querySelector('#mapImg');
    let cnvs = document.querySelector("canvas");
    if (cnvs) cnvs.remove();
    canvas.id = 'canvas';
    mapImg.appendChild(canvas);
  })
};

function drawOnMap(drawFuncName, color, canvas, latitude,
longitude, scale, radius, angle) {
  let ctx = canvas.getContext('2d');

  if( drawFuncName == 'drawArc'){
    let rad = pixelValue(latitude, radius * 1000, scale)

```

```

        drawArc(color, ctx, [canvas.width / 2, canvas.height / 2],
rad);
    } else if (drawFuncName == 'drawEllipse') {
        let sPixelA = pixelValue(latitude, radius[0], scale);
        let sPixelB = pixelValue(latitude, radius[1], scale);

        drawEllipse(color, ctx, [canvas.width / 2, canvas.height /
2], [sPixelA, sPixelB], angle);

        return;
    } else if (drawFuncName == 'drawFireZone'){
        let sPixelA = pixelValue(latitude, radius[0], scale);
        let sPixelB = pixelValue(latitude, radius[1], scale);

        drawFireZone(color, ctx, [canvas.width / 2, canvas.height /
2], [sPixelA, sPixelB], angle);

        return;
    }
}

function metersPerPixel(latitude, zoomLevel) {
    let earthCircumference = 40075016.6;
    let latitudeRadians = latitude * (Math.PI / 180);
    return earthCircumference * Math.cos(latitudeRadians) /
Math.pow(2, zoomLevel + 8);
};

function pixelValue(latitude, meters, zoomLevel) {
    return meters / metersPerPixel(latitude, zoomLevel);
};

```

parceCanvas.js

```

function unitArrayToRegularArray(unitArray) {
    let regularArray = [];
    let rgbaArray = [];

    for(let i = 0; i < unitArray.data.length; i+=4){
        regularArray.push([unitArray.data[i],unitArray.data[i+1],
unitArray.data[i+2], unitArray.data[i+3]]);
    }

    let counter = 0;

    for (let i = 0; i < unitArray.height; i++){

```

```

        let tmpArray = [];
        for (let j = 0; j < unitArray.width; j++){
            tmpArray.push(regularArray[j+counter]);
        }
        rgbaArray.push(tmpArray);
        counter += unitArray.width;
    }

    return rgbaArray;
}

function getData() {
    let c = document.getElementById("canvas");
    let ctx = c.getContext("2d");
    let data = ctx.getImageData(0, 0, c.width, c.height)

    let arryaOfRGBAValue = unitArrayToREGuralArray(data);
    return arryaOfRGBAValue;
}

function findPixelsInZone(data, zoneColor, pixelColor){
    let totalCounter = 0;
    let positionOfLastZonePixel;

    for (let row of data){
        let tmpCounter = 0;
        let countPixel = false;
        let addToTotalCount = false;
        for (let i = 0; i < row.length - 1; i++){
            if(row[i][0] === zoneColor.r && row[i][1] === zoneColor.g &&
row[i][2] === zoneColor.b){
                positionOfLastZonePixel = i;
            } else if(!countPixel && i - positionOfLastZonePixel === 1){
                countPixel = !countPixel;
            }

            if(countPixel){
                if(row[i][0] === pixelColor.r && row[i][1] === pixelColor.g
&& row[i][2] === pixelColor.b){
                    tmpCounter++;
                }
                if(row[i + 1][0] === zoneColor.r && row[i + 1][1] ===
zoneColor.g && row[i + 1][2] === zoneColor.b){
                    totalCounter += tmpCounter;
                    break;
                }
            }
        }
    }

    return totalCounter;
}

```

nhr.js

```

let zoom = 12;
const routerName = "dbrouter",

```

```

        height = 1, //                ВЫСОТА ОБВАЛОВАНИЯ
с формы
        place = {
            //                инфа о местности
с формы
            type: "відкрита",
            place: "Ліс",
            leng: 1,
        },
        zoneBorderColorRGB = {
            r: 255,
            g: 0,
            b: 0,
        },
        forestColorRGB = {
            r: 178,
            g: 194,
            b: 157,
        },
        waterColorRGB = {
            r: 144,
            g: 204,
            b: 203,
        },
        fieldColorRGB = {
            r: 222,
            g: 221,
            b: 190,
        };

        async function calculate(id, nhrType, nhrQuantity, provGasMasl,
date, imageData) {
            const accidentDate = new Date(date);
            const unixAcidentDate = accidentDate.getTime() / 1000;
            const accidentTime = accidentDate.getHours() + ":" +
accidentDate.getMinutes(); //        время аварии

            const factoryStr = await apiRequest("factory", routerName,
{
                id,
            });
            const factoryArr = factoryStr.split("|");

```

```

        const latitude = +factoryArr[0], // широта
              longitude = +factoryArr[1], // долгота
              peopleNum = +factoryArr[2], // кількість людей
на об'єкті
              city = factoryArr[3]; // город в котрому
находиться об'єкт

        const wetherObjJSON = await getWetherData(latitude,
longitude, unixAcidentDate);
        const wetherObj = JSON.parse(wetherObjJSON);

        const temperature = wetherObj.currently.temperature, //
температура воздуха
              windSpeed = wetherObj.currently.windSpeed, //
скорость ветра
              windAzimut = wetherObj.currently.windBearing; //
угол/азимут ветра

        let cloudiness = wetherObj.currently.summary; //
облачность

        cloudiness =
            cloudiness == "Ясно" ?
                cloudiness :
            cloudiness == "Хмарно" ?
                cloudiness :
                "Напівясно";

        const timeOfDay = getTimeOfDay(accidentTime); //Время суток

        const degreeOfVerticalStability = await
apiRequest("vertical", routerName, {
            timeOfDay,
            cloudiness,
            windSpeed,
        }); //ступінь вертикальної стійкості

        const tablDeptZone = await apiRequest("tabledeptvalue",
routerName, {
            nhrQuantity,
            nhrType,

```

```

        degreeOfVerticalStability,
        temperature,
        windSpeed,
    }); //Гт - табличне значення глибини зони

    const ksx = await apiRequest("nhr_cloud", routerName, {
        nhrType,
        height,
    }); //Ксх - коефіцієнт, що враховує тип сховища і
характеризує зменшення глибини розповсюдження хмари НХР при виливі "у
піддон"

    const kzm = await apiRequest("koef_kzm", routerName, {
        degreeOfVerticalStability,
        palce: place["place"],
    }); //Кзм - коефіцієнт зменшення глибини розповсюдження
хмари НХР для кожного 1 км довжини закритої місцевост

    const deptDecr = getDeptDecr(place.leng, kzm); //Гзм -
величина, на яку зменшується глибина розповсюдження хмари НХР

    const deptCalculatedZone =
getDeptCalculatedZone(tablDeptZone, ksx, deptDecr); //розрахункової
глибини отримане значення Гр

    const transferSpeed = await apiRequest("v_transfer",
routerName, {
        windSpeed,
        degreeOfVerticalStability,
    }); //W - швидкість переносу повітряних мас

    const deptTramsfer = getDeptTransfer(transferSpeed);
//глибини переносу повітряних мас Гп

    const forecastDeptZone = getForecastDeptZone(
        deptCalculatedZone,
        deptTramsfer
    ); //за фактичну прогнозовану глибину зони забруднення,
тобто Гпзхз

    //zoom = forecastDeptZone > 10 ? 11 : forecastDeptZone < 2
? 13 : zoom;

```



```

if (forecastDeptZone > 5)
    zoom = 11;
else if (forecastDeptZone < 2)
    zoom = 13;

const widthForecastZone = getWidthForecastZone(
    degreeOfVerticalStability,
    forecastDeptZone
); //Ширина прогнозованої зони хімічного забруднення
(Шпзхз)

const F = getF(windSpeed); //Коефіцієнт Ф, який залежить
від швидкості вітру

const squarePossibleZone =
getSquarePossibleZone(forecastDeptZone, F); //Площа зони можливого
хімічного забруднення (ЗМХЗ).const

const K = getK(degreeOfVerticalStability); //де К -
коефіцієнт

const N = 4; //N - час, на який розраховується глибина ПЗХЗ
const squarePredictedZone = getSquarePredictedZone(K,
forecastDeptZone, N); //Площа прогнозованої зони хімічного забруднення
(ПЗХЗ)

const pollutionDuration = await
apiRequest("pollution_duration", routerName, {
    nhrType,
    height,
    temperature,
    windSpeed,
});

const percentOfLoss = await apiRequest("population_loss",
routerName, {
    place: place.type,
    provGasMasl,
});
const peopleLoss = getPeopleLoss(peopleNum, percentOfLoss);

```

```

const meterperPixel = metersPerPixel(latitude, zoom);

if (!imageData) {
    await workWithMap({
        latitude,
        longitude,
    },
        zoom
    );
}

const canvas = document.querySelector('#canvas');
if (imageData) {
    const context = canvas.getContext('2d');

    base_image = new Image();
    base_image.src = `../../saved_img/${imageData}`;
    setTimeout(() => {
        context.drawImage(base_image, 0, 0);
        drawRasults(nhrType, nhrQuantity,
forecastDeptZone, widthForecastZone, squarePredictedZone,
squarePossibleZone, pollutionDuration, peopleLoss, meterperPixel, city)
    }, 2000);
    } else {
        const semiaxisA = (forecastDeptZone * 1000) / 2;
        const semiaxisB = (widthForecastZone * 1000) / 2;
        drawOnMap('drawEllipse', zoneBorderColorRGB,
canvas, latitude, longitude, zoom, [semiaxisA, semiaxisB], windAzimut);
        drawRasults(nhrType, nhrQuantity, forecastDeptZone,
widthForecastZone, squarePredictedZone, squarePossibleZone,
pollutionDuration, peopleLoss, meterperPixel, city)
    }
};

async function drawRasults(nhrType, nhrQuantity, forecastDeptZone,
widthForecastZone, squarePredictedZone, squarePossibleZone,
pollutionDuration, peopleLoss, meterperPixel, city) {
    createTable('resultTable', [
        [
            "Тип НХР, кількість, тон",
            "Глибина ПЗХЗ, км",
            "Ширина ПЗХЗ, км",

```

```

        "Площа ПЗХЗ, км.кв.",
        "Площа МЗХЗ, км.кв.",
        "Тривалість уражаючої дії, хв",
        "Втрати людей, чол",
    ],
    [
        nhrType + ", " + nhrQuantity,
        forecastDeptZone,
        widthForecastZone,
        squarePredictedZone,
        squarePossibleZone,
        pollutionDuration,
        `Всього: ${peopleLoss.totalLoss}, смерт.-
        ${peopleLoss.death}, серед.- ${peopleLoss.moderateSeverity}, легкі-
        ${peopleLoss.lightSeverity}`
    ],
    ], 'Таблиця 1 - Результати оцінки хімічної обстановки');

    const data = getData();

    const fieldInZone = findPixelsInZone(data,
zoneBorderColorRGB, fieldColorRGB);
    const forestInZone = findPixelsInZone(data,
zoneBorderColorRGB, forestColorRGB);
    const waterInZone = findPixelsInZone(data,
zoneBorderColorRGB, waterColorRGB);

    const fieldSqrt = getSqrt(meterPerPixel, fieldInZone,
data.length);
    const forestSqrt = getSqrt(meterPerPixel, forestInZone,
data.length);
    const waterSqrt = getSqrt(meterPerPixel, waterInZone,
data.length);

    const peopleLossDamage = getPeopleLossDamage(peopleLoss);

    const cityInfoStr = await apiRequest("agriculture",
routerName, {
        city,
    });
    const cityInfo = cityInfoStr.split("|");

```

```

const agricultureDamage = getAgricultureDamage(
    fieldSqrt,
    cityInfo[0],
    cityInfo[1]
);

const foresDamage = getForestDamage(forestSqrt,
cityInfo[2]);

const waterDamage = 900 * waterSqrt;

createTable('damageTable', [
    [
        'Витрати на відшкодування життя та здоров'я
населення',
        'Збитки від порушення сільськогосподарських
угідь',
        'Збитки від втрати деревини та інших
лісових ресурсів',
        'Збитки рибного господарства'
    ],
    [
        peopleLossDamage.toFixed(2),
        agricultureDamage.toFixed(2),
        foresDamage.toFixed(2),
        waterDamage.toFixed(2)
    ]
], 'Таблиця 2 - Результати оцінки прогнозованого еколого-
економічного збитку');
}

```

getWether.js

```

const goButton = document.querySelector("#go");
const KEY = "199cb0b508acd58820ca7ed8fea3afb3";

function getWetherData(latitude, longitude){
    const
url=`https://api.darksky.net/forecast/${KEY}/${latitude},${longitude}?lang=uk
&units=auto`;

    return getJSONWether(url);
}

```

forestfire.js

```

//const for calc
const V0 = 0.4, // V0 - скорость распространения пожара
на равнине в безветренную погоду, м/мин
    k = 0.16, // k - коэффициент, учитывающий
раздувающее влияние пламени
    C = 3,
    routerName = "dbrouter"; // C - коэффициент,
учитывающий удельную теплоемкость горючих материалов

const zoom = 11,
    zoneBorderColorRGB = {
        r: 255,
        g: 0,
        b: 0,
    },
    zoneBorderColorRGB2 = {
        r: 0,
        g: 0,
        b: 255,
    },
    zoneBorderColorRGB3 = {
        r: 255,
        g: 255,
        b: 0,
    };

//variable
const hourOne = 24,
    hourTwo = 48;

async function calculate(latitude, longitude, fireType,
burnabilityClass, avgHeghtCarbon, avgTreeDiameter,
startPerimeter, date, imageData) {
    const accidentDate = new Date(date);

```

```

    const unixAcidentDate = accidentDate.getTime() /
1000;

    const wetherObjJSON = await getWetherData(latitude,
longitude, unixAcidentDate);
    const wetherObj = JSON.parse(wetherObjJSON);

    const temperature = wetherObj.currently.temperature,
//
    температура воздуха
    windSpeed = wetherObj.currently.windSpeed, //
скорость ветра
    windAzimut = wetherObj.currently.windBearing, //
угол/азимут ветра
    humidity = wetherObj.currently.humidity; //
ВЛАЖНОСТЬ

    let dewPoint = getDewPoint(temperature, humidity),
    indicator = getIndicator(temperature, dewPoint),
    wetherClass = getWetherClass(indicator),
    linearFireFrontSpeed =
getLinearFireFrontSpeed(fireType, windSpeed),
    linearFireWingSpeed =
getLinearFireWingSpeed(windSpeed),
    linearFireRearSpeed =
getLinearFireRearSpeed(windSpeed),
    perimeterGainOne =
getPerimeterGain(linearFireFrontSpeed, hourOne),
    newPerimeterOne =
getPerimeterInHours(startPerimeter, perimeterGainOne),
    newSquareOne =
getSquareInHours(newPerimeterOne),
    perimeterGainTwo =
getPerimeterGain(linearFireFrontSpeed, hourTwo),
    newPerimeterTwo =
getPerimeterInHours(startPerimeter, perimeterGainTwo),

```

```

        newSquareTwo =
getSquareInHours(newPerimeterTwo),
        unusableWood = getUnusableWood(fireType,
burnabilityClass);

        let woodDamage = await apiRequest("wood_damage",
routerName, {
            avgTreeDiameter,
            avgHeghtCarbon,
            burnabilityClass
        });
        let woodDamageCharJSON = await
apiRequest("wood_damage_char", routerName, {
            woodDamage,
        });

        let woodDamageChar = JSON.parse(woodDamageCharJSON);
        if (!imageData) {
            await workWithMap({
                latitude,
                longitude,
            },
                zoom
            );
        }

        let radiusCircle = getRadius(startPerimeter);
        let sCircle = Math.ceil((Math.PI *
Math.pow(radiusCircle, 2)) / 10000);
        let radiusNewCircleOne = getRadius(newPerimeterOne);
        let axisOne = getAxis(radiusNewCircleOne,
linearFireFrontSpeed, linearFireRearSpeed, hourOne)
        let semiaxisOne = getSemiaxis(axisOne);
        let semiordinatOne = radiusNewCircleOne;
        let radiusNewCircleTwo = getRadius(newPerimeterTwo);

```

```

        let axisTwo = getAxis(radiusNewCircleTwo,
linearFireFrontSpeed, linearFireRearSpeed, hourTwo)
        let semiaxisTwo = getSemiaxis(axisTwo);
        let semiordinatTwo = radiusNewCircleTwo;
        const canvas = document.querySelector('#canvas');
        if (imageData) {
            const context = canvas.getContext('2d');

            base_image = new Image();
            base_image.src = `../../saved_img/${imageData}`;
            setTimeout(() => {
                context.drawImage(base_image, 0, 0);
                drawRasults(sCircle, startPerimeter,
newSquareOne, newPerimeterOne, newSquareTwo, newPerimeterTwo,
woodDamageChar, wetherClass, linearFireFrontSpeed,
linearFireWingSpeed, linearFireRearSpeed, unusableWood);
            }, 2000);
        } else {
            drawOnMap('drawArc', zoneBorderColorRGB, canvas,
latitude, longitude, zoom, radiusCircle / 1000);
            drawOnMap('drawFireZone', zoneBorderColorRGB2,
canvas, latitude, longitude, zoom, [semiaxisOne,
semiordinatOne], windAzimut);
            drawOnMap('drawFireZone', zoneBorderColorRGB3,
canvas, latitude, longitude, zoom, [semiaxisTwo,
semiordinatTwo], windAzimut);
            drawRasults(sCircle, startPerimeter,
newSquareOne, newPerimeterOne, newSquareTwo, newPerimeterTwo,
woodDamageChar, wetherClass, linearFireFrontSpeed,
linearFireWingSpeed, linearFireRearSpeed, unusableWood);
        }
    };

    async function drawRasults(sCircle, startPerimeter,
newSquareOne, newPerimeterOne, newSquareTwo, newPerimeterTwo,

```



```

woodDamageChar,      wetherClass,      linearFireFrontSpeed,
linearFireWingSpeed, linearFireRearSpeed, unusableWood) {
    createTable('resultTable', [
        ['Час',
            'Площа лісової пожежі, Га ',
            'Периметр лісової пожежі, км',
        ],
        [
            'На момент виявлення(червоний)',
            sCircle,
            startPerimeter,
        ],
        [
            'Через добу після виявлення(синій)',
            newSquareOne / 1000,
            newPerimeterOne,
        ],
        [
            'Через дві доби після виявлення(жовтий)',
            newSquareTwo / 1000,
            newPerimeterTwo,
        ],
    ], 'Таблиця 1 - Розповсюдження пожежі в часових
проміжках');

```

```

let res;
if (fireType = 3) {
    res = woodDamageChar[0][0] + ' Відсоток
непригодної деревини = ' + woodDamageChar[0][1];
} else {
    res = unusableWood;
}

createTable('damageTable', [

```

```

        ['Клас пожежної небезпеки',
         'Швидкість фронту пожежі, м/с',
         'Швидкість флангу пожежі, м/с',
         'Швидкість тилу пожежі, м/с',
         'Відсоток деревини непригодної для
реалізації'
        ],
        [
            wetherClass,
            linearFireFrontSpeed,
            linearFireWingSpeed,
            linearFireRearSpeed,
            res
        ]
    ], 'Таблиця 2 - Загальні відомості про пожежу');
}

function getS(square) {
    return square * 10000;
}

function getSemiaxis(axis) {
    return axis / 2;
}

function getAxis(radiusCircle, linearFireFrontSpeed,
linearFireRearSpeed, hours) {
    return radiusCircle * 2 + linearFireFrontSpeed *
hours + linearFireRearSpeed * hours;
}

function getRadius(perimeter) {
    return perimeter / (2 * Math.PI);
}

```

```

function getDewPoint(temperature, humidity) {
    const F = getF(temperature, humidity);
    return (237.7 * F) / (17.27 - F);
}

function getF(temperature, humidity) {
    return (17.23 * temperature) / (237.7 + temperature)
+ Math.log(humidity / 100);
}

function getIndicator(noontemperature, dewPoint) {
    return ((noontemperature - dewPoint) *
noontemperature) * 14;
}

function getWetherClass(indicator) {
    if (indicator <= 300)
        return 1; //отсутствие опасности
    else if (indicator > 300 && indicator <= 1000)
        return 2; //малая пожарная опасность
    else if (indicator > 1000 && indicator <= 4000)
        return 3; //средняя пожарная опасность
    else if (indicator > 4000 && indicator <= 12000)
        return 4; //высокая пожарная опасность
    else if (indicator >= 12000)
        return 5; //чрезвычайная опасность
}

function getLinearFireFrontSpeed(fireType, windSpeed) {
    switch (+fireType) {
        case 1:
            return 120;
        case 2:
            return 250;
        case 3:

```

```

        return Math.ceil(((V0 + k * windSpeed) *
Math.pow(1 + (windSpeed / Math.sqrt(Math.pow(windSpeed, 2) +
Math.pow(C, 2))), 2)) * 30);
        default:
            return;
    }
}

function getLinearFireWingSpeed(windSpeed) {
    return Math.ceil((V0 + k * windSpeed) * 30);
}

function getLinearFireRearSpeed(windSpeed) {
    return Math.ceil(((V0 + k * windSpeed) * Math.pow(1
- (windSpeed / Math.sqrt(Math.pow(windSpeed, 2) + Math.pow(C,
2))), 2)) * 30);
}

function getPerimeterGain(linearFireFrontSpeed, hours) {
    return 3.3 * linearFireFrontSpeed * hours;
}

function getPerimeterInHours(startPerimeter,
perimeterGain) {
    return Math.ceil(+startPerimeter + perimeterGain);
}

function getSquareInHours(newPerimeter) {
    return Math.ceil(4 * Math.pow(10, -6) *
Math.pow(newPerimeter, 2));
}

function getUnusableWood(fireType, burnabilityClass) {
    if (+fireType == 2 && +burnabilityClass == 1) {
        return 50;
    }
}

```

```
        } else if (+fireType == 2 && +burnabilityClass == 2)
    {
        return 70;
    } else if (+fireType == 3 && +burnabilityClass == 1)
    {
        return 30;
    } else if (+fireType == 3 && +burnabilityClass == 2)
    {
        return 60;
    } else {
        return;
    }
}
```

earthquake.js

```
const routerName = "dbrouter",
    zoom = 10,

    epicentrZoneBorderColorRGB = {
        r: 242,
        g: 12,
        b: 157,
    },

    dangerZoneBorderColorRGB = {
        r: 255,
        g: 0,
        b: 0,
    },

    midleDangerZoneBorderColorRGB = {
        r: 255,
        g: 111,
        b: 0,
    };
```

```

    async function calculate(latitude, longitude, magnitude, depth,
imageData) {
    if (!imageData) {
        await workWithMap({
            latitude,
            longitude,
        },
            zoom
        );
    }

    const canvas = document.querySelector('#canvas');
    if (imageData) {
        const context = canvas.getContext('2d');

        base_image = new Image();
        base_image.src = `../../saved_img/${imageData}`;
        setTimeout(() => {
            context.drawImage(base_image, 0, 0);
            drawRasults(magnitude, depth)
        }, 2000);
    } else {
        drawOnMap('drawArc', epicentrZoneBorderColorRGB, canvas,
latitude, longitude, zoom, 1);
        drawOnMap('drawArc', dangerZoneBorderColorRGB, canvas,
latitude, longitude, zoom, 10);
        drawOnMap('drawArc', midleDangerZoneBorderColorRGB, canvas,
latitude, longitude, zoom, 20);
        drawRasults(magnitude, depth)
    }
}

async function drawRasults(magnitude, depth) {
    let epicentrIntensity = getIntensityEpicentr(magnitude, depth),
        firstZone = getIntensity(magnitude, depth, 10),
        secondZone = getIntensity(magnitude, depth, 20);

    let epicentrEffect, firstEffect, secondEffect;

    if (epicentrIntensity < 1) {
        epicentrEffect = await apiRequest('earthquakeeffects',
routerName, {

```

```

        intensity: 1
    })),
    firstEffect = await apiRequest('earthquakeeffects',
routerName, {
        intensity: 1
    })),
    secondEffect = await apiRequest('earthquakeeffects',
routerName, {
        intensity: 1
    });
} else {
    epicentrEffect = await apiRequest('earthquakeeffects',
routerName, {
        intensity: epicentrIntensity
    })),
    firstEffect = await apiRequest('earthquakeeffects',
routerName, {
        intensity: firstZone
    })),
    secondEffect = await apiRequest('earthquakeeffects',
routerName, {
        intensity: secondZone
    });
}

createTable('resultTable', [
    ['',
        'Інтенсивність землетрусу ',
        'Наслідки землетрусу',
    ],
    [
        'В епіцентрі виникнення',
        epicentrIntensity,
        epicentrEffect,
    ],
    [
        'В радіусі 10км від епіцентру',
        firstZone,
        firstEffect,
    ],
    [
        'В радіусі 20км від епіцентру',

```

```

        secondZone,
        secondEffect,
    ],
    ], 'Таблиця 1 - Наслідки землетрусу в кожній зоні, в залежності від
інтенсивності на поверхні');

let deathArr = [
    [,
        'Тип будівлі',
        'Втрати населення в епіцентрі, %',
        'Втрати населення в радіусі 10км, %',
        'Втрати населення в радіусі 20км, %',
    ]
];

let epicentrDeath,
    firstZoneDeath,
    secondZoneDeath;

if (epicentrIntensity < 7) {
    let epicentrDeathJSON = await apiRequest('earthquakeadeath',
routerName, {
        intensity: 0
    }),
        firstZoneDeathJSON = await apiRequest('earthquakeadeath',
routerName, {
            intensity: 0
        }),
        secondZoneDeathJSON = await
apiRequest('earthquakeadeath', routerName, {
            intensity: 0
        });

    epicentrDeath = JSON.parse(epicentrDeathJSON);
    firstZoneDeath = JSON.parse(firstZoneDeathJSON);
    secondZoneDeath = JSON.parse(secondZoneDeathJSON);
} else {
    let epicentrDeathJSON = await apiRequest('earthquakeadeath',
routerName, {
        intensity: epicentrIntensity
    }),

```



```

        firstZoneDeathJSON = await apiRequest('earthquakeadeath',
routerName, {
            intensity: firstZone
        }),
        secondZoneDeathJSON = await
apiRequest('earthquakeadeath', routerName, {
            intensity: secondZone
        });

        epicentrDeath = JSON.parse(epicentrDeathJSON);
        firstZoneDeath = JSON.parse(firstZoneDeathJSON);
        secondZoneDeath = JSON.parse(secondZoneDeathJSON);
    };

    for (let i in epicentrDeath) {
        deathArr.push([
            epicentrDeath[i][0],
            epicentrDeath[i][1],
            firstZoneDeath[i][1],
            secondZoneDeath[i][1]
        ])
    };

    createTable('damageTable', deathArr, 'Таблиця 2 - Втрати населення в
кожній зоні, в залежності від інтенсивності на поверхні, %');

    let systemArr = [
        [,
            'Система',
            'Стійкість систем в епіцентрі, %',
            'Стійкість систем в радіусі 10км, %',
            'Стійкість систем в радіусі 20км, %',
        ]
    ];

    let epicentrSystem,
        firstZoneSystem,
        secondZoneSystem;

    if (epicentrIntensity < 7) {
        let epicentrSystemJSON = await apiRequest('earthquakeakesystem',
routerName, {

```

```

        intensity: 0
    })),
    firstZoneSystemJSON = await
apiRequest('earthquakeakesystem', routerName, {
    intensity: 0
}),
    secondZoneSystemJSON = await
apiRequest('earthquakeakesystem', routerName, {
    intensity: 0
});

    epicentrSystem = JSON.parse(epicentrSystemJSON);
    firstZoneSystem = JSON.parse(firstZoneSystemJSON);
    secondZoneSystem = JSON.parse(secondZoneSystemJSON);
} else {
    let epicentrSystemJSON = await apiRequest('earthquakeakesystem',
routerName, {
        intensity: epicentrIntensity
    })),
    firstZoneSystemJSON = await
apiRequest('earthquakeakesystem', routerName, {
        intensity: firstZone
    }),
    secondZoneSystemJSON = await
apiRequest('earthquakeakesystem', routerName, {
        intensity: secondZone
    });

    epicentrSystem = JSON.parse(epicentrSystemJSON);
    firstZoneSystem = JSON.parse(firstZoneSystemJSON);
    secondZoneSystem = JSON.parse(secondZoneSystemJSON);
};

for (let i in epicentrSystem) {
    systemArr.push([
        epicentrSystem[i][0],
        epicentrSystem[i][1],
        firstZoneSystem[i][1],
        secondZoneSystem[i][1]
    ])
}

```

```

};

    createTable('lifeSupportSystems', systemArr, 'Таблиця 3 - Стійкість систем життєзабезпечення, %', 'Примітка: У чисельнику -% систем життєзабезпечення, здатних до функціонуванню негайно, а в знаменнику - після відновлювальних робіт в Протягом доби.');
```

```

}

function getIntensityEpicentr(magnitude, depth) {
    return Math.round(1.5 * magnitude - 3.5 * Math.log10(depth) + 3);
}

function getIntensity(magnitude, depth, radius) {
    return Math.round(1.5 * magnitude - 3.5 * Math.log10(Math.sqrt(Math.pow(radius, 2) + Math.pow(depth, 2))) + 3);
}
}

```

historyMap.js

```

const map = L.map('map').setView([49.0384, 31.4513], 6);
L.tileLayer('https://{s}.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png', {
    attribution: '&copy; OpenStreetMap contributors',
    href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
}).addTo(map);
L.control.scale().addTo(map);

const markersLayer = L.layerGroup().addTo(map);

const allTimeCheckbox = document.querySelector('#allTime'),
    dateNow = document.querySelector('#dateNow'),
    fromDate = document.querySelector('#fromDate'),
    toDate = document.querySelector('#toDate'),
    catType = document.querySelector('#catType'),
    applyFilterButton = document.querySelector("#applyFilterButton");

(async () => {
    const catListJSON = await apiRequest('cat_list', 'dbrouter', {
        id_cat_type: 0,
        from_date: undefined,
        to_date: undefined,
    });

```

```

    const catList = JSON.parse(catListJSON);

    fillMap(catList);
  }) ();

  applyFilterButton.addEventListener('click', async () => {
    const catListJSON = await apiRequest('cat_list', 'dbrouter', {
      id_cat_type: catType.value,
      from_date: fromDate.value !== "" ? fromDate.value : null,
      to_date: toDate.value !== "" ? toDate.value : null,
    });

    const catList = JSON.parse(catListJSON);

    fillMap(catList);
  })

  allTimeCheckbox.addEventListener('change', () => {
    fromDate.value = null;
    toDate.value = null;
    dateNow.checked = false;
    fromDate.disabled = allTimeCheckbox.checked;
    toDate.disabled = allTimeCheckbox.checked;
    dateNow.disabled = allTimeCheckbox.checked;
  })

  dateNow.addEventListener('change', () => {
    if (dateNow.checked) {
      const now = new Date();
      now.setMinutes(now.getMinutes() - now.getTimezoneOffset());
      toDate.value = now.toISOString().slice(0, 16);
      toDate.disabled = true;
    } else {
      toDate.value = null;
      toDate.disabled = false;
    }
  });

  function fillMap(catList) {
    markersLayer.clearLayers();

```

```

catList.forEach(cat => {
  let markerColor, icon;

  switch(Number(cat.id_cat_type)) {
    case 1:
      icon = 'flask';
      markerColor = 'blue';
      break;
    case 2:
      icon = 'mountain';
      markerColor = 'green';
      break;
    case 3:
      icon = 'fire-alt';
      markerColor = 'red';
      break;
    default:
      break;
  }

  const marker = L.marker([cat.latitude, cat.longitude], {icon:
L.AwesomeMarkers.icon({icon,          markerColor,          prefix:          'fa'})
}).addTo(markersLayer);

  const link = createLink(cat.id_cat_type, cat.id_cat);

  marker.bindPopup(
    `Вид катастрофи: ${cat.name}<br/>
    <b>Дата:</b> ${cat.cat_date}<br/>
    <b>Користувач:</b> ${cat.uname ?? 'Гість'}<br/>
    <a href="${link}">Переплянути деталі</a>
    `
  )
})
}

function createLink(id_cat_type, id_cat) {
  let page = '';

  switch(Number(id_cat_type)) {
    case 1:

```

```
        page = "nhrCalculator.php";
        break;
    case 2:
        page = "earthquakeCalculator.php";
        break;
    case 3:
        page = "fireCalculator.php";
        break;
    default:
        break;
}

return `~/pages/${page}?id_cat=${id_cat}`;
}
```