

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Олександр КОВАЛЬ

« ___ » _____ 2021 р.

Дипломна робота

на здобуття ступеня бакалавра

спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерний моніторинг та геометричне

моделювання процесів і систем»

**на тему: «Пошукова GIS система екологічно небезпечних промислових
об'єктів»**

Виконав:

студент IV курсу, групи ТМ-71

Ласкавий Олександр Олександрович _____

Керівник:

Старший викладач

Бандурка Олена Іванівна _____

Консультант:

Рецензент:

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

Студент (-ка) _____

Київ — 2021 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 122 «Комп’ютерні науки»

Освітня програма «Комп’ютерний моніторинг та геометричне моделювання процесів і систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр Коваль
(підпис)

” ___ ” _____ 2021р.

ЗАВДАННЯ

на дипломну роботу студенту

Ласкавому Олександрю Олесандровичу
(прізвище, ім’я, по батькові)

1. Тема роботи «Пошукова GIS система екологічно небезпечних промислових об’єктів»

керівник роботи Бандурка Олена Іванівна, старший викладач

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

затверджена наказом вищого навчального закладу від ”25” травня 2021р. № **1267-с**

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи мова програмування JavaScript, PHP, HTML та таблиця каскадних стилів CSS. Розроблено з використанням фреймворків Bootstrap та CodeIgniter в середовищі розробки NuSpherePhpED.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати проблематику погіршення екологічного стану промислової зони України, розглянути існуючі програмні рішення для моніторингу змін в екологостані країни, розробити універсальний програмний продукт для визначення динаміки рівня забруднення повітря, води та ґрунтових вод. Протестувати програмне забезпечення.

5. Перелік ілюстративного матеріалу зображення архітектури системи та адмін частини, концептуальна схема бази даних, діаграма прецедентів, схема структури файлової системи, представлення інтерфейсу системи та адмін частини, приклади роботи онлайн карти моніторингу, приклади роботи калькуляторів забруднень.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання ” ___ ” _____ 201__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	25.10.21	виконав
2.	Вивчення та аналіз задачі	03.02-11.02.21	виконав
3.	Розробка архітектури та загальної структури системи	12.02-20.02.21	виконав
4.	Розробка структур окремих підсистем	21.02-28.02.21	виконав
5.	Програмна реалізація системи	02.03-11.03.21	виконав
6.	Оформлення пояснювальної записки	12.03-29.05.21	виконав
7.	Захист програмного продукту	13.05.21	виконав
8.	Передзахист	27.05.21	виконав
9.	Захист	18.06.2021	

Студент _____

(підпис)

Ласкавий О.О.

(прізвище та ініціали,)

Керівник роботи _____

(підпис)

Бандурка О.І.

(прізвище та ініціали,)

АНОТАЦІЯ

Дипломну роботу виконано на 73 аркушах, вона містить три додатки, перелік посилань та використаних джерел. У роботі наведено 37 рисунків для ілюстрації схем організації процесів, демонстрації інтерфейсу, та прикладів роботи системи.

Метою дипломної роботи є створення універсального програмного продукту для моніторингу динаміки змін викидів у навколишнє середовище, визначення найбільш еколого небезпечних промислових об'єктів нашої країни. Даний програмний продукт розроблений з використанням фреймворків Bootstrap та CodeIgniter в середовищі розробки NuSpherePhpED у вигляді веб-ресурсу для забезпечення доступу з будь-якого ресурсу.

Ключові слова: Gis-система, пошукова гіс система, промисловість, еколого небезпечний, онлайн моніторинг, калькулятор викидів, динаміка забруднень.

ABSTRACT

Thesis is made on 67 sheets, it contains two additional links, links to links and sources used. During the work 37 drawings were made to create a scheme of organization of processes, demonstration of the interface and preparation of working systems.

The purpose of the thesis is to create a universal software product for monitoring the dynamics of changes in emissions into the environment, which identifies environmentally hazardous industrial facilities in our country. This software is created using the Bootstrap and CodeIgniter frameworks in the NuSpherePhpED development environment in a web resource to provide access to any resource.

Keywords: Gis system, search gis system, industry, environmentally hazardous, online monitoring, emission calculator, pollution dynamics.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	5
ВСТУП.....	6
1. ЗАДАЧА ЕКОЛОГІЧНОЇ GIS-СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ЕКОЛОГІЧНОГО СТАНУ ПРОМИСЛОВОЇ ЗОНИ КРАЇНИ	8
2. ПРОБЛЕМИ МОНІТОРИНГУ ЕКОЛОГІЧНОГО СТАНУ ПРОМИСЛОВОЇ ЗОНИ.....	10
3. ЗАСОБИ РОЗРОБКИ.....	15
3.1 Середовище розробки.....	15
3.2 Вибір хостингу та серверу.....	19
3.3 Вибір програмних додатків.....	22
3.4 Вибір мов програмування.....	23
4 ТЕХНІЧНИЙ ОПИС СИСТЕМИ.....	28
4.1 Опис архітектури.....	28
4.2 Опис баз даних.....	32
4.3 Діаграма прецедентів.....	36
4.4 Структура програмного забезпечення.....	38
5. РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ.....	40
5.1 Системні вимоги.....	40
5.2 Робота адміністратора в системі.....	40
5.3 Можливості гостя на сайті.....	45
ВИСНОВКИ.....	50
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52
Додаток 1.....	54
Додаток 2.....	56
Додаток 3.....	68
Додаток 4.....	73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

HTML (англ. Hypertext Markup Language) — стандартизована мова гіпертекстової розмітки, що використовується для створення веб-сторінок;

CSS (англ. Cascading Style Sheets) — мова опису зовнішнього вигляду блоку інформації ;

JSX (англ. JavaScript eXtensible Markup Language) — синтаксис, який робить можливим використання HTML/XML подібного тексту разом з JavaScript;

XML (англ. Extensible Markup Language) – стандарт побудови мов розмітки структурованих даних.

GIS (англ. Geographic Information System) – система що взаємодіє з місцезнаходженням чи рухом певних об'єктів

ВСТУП

Промисловість відіграє важливу роль в економіці будь-якої країни. Тому кожна країна прагне нарощувати промислову базу. Україна в своєму розпорядженні мала величезну кількість заводів, фабрик та підприємств. Але, нажаль, їх кількість зменшується, а деякі ж з тих що лишилися – не в належному стані.

Зі зростанням активності промислового сектору, все активніше постає питання збереження екологічно стабільного стану навколишнього середовища. Через старе оснащення, неналежну роботу місцевих органів, а саме відсутність контролю, недостатньо якісні очисні та фільтруючі станції щодня у навколишнє середовище потрапляють шкідливі речовини, наприклад CO₂.

Жителі міст в яких знаходяться промислові гіганти щоденно потерпають від надзвичайно забрудненого повітря. Серед таких міст усім нам знайомі Дніпро, Запоріжжя, Маріуполь, Кривий Ріг, Кам'янське.

За останніми дослідженнями через забруднене повітря кожного року помирає близько 9 мільйонів людей на планеті. Причиною кожного третього інсульту, онкологічних захворювань є забруднене повітря.[1]

Не новиною стало потрапляння Києва у п'ятірку міст з найбільш забрудненим повітрям за даними сайту IQair [2].

На жаль, влада не приділяє забрудненому повітрю належної уваги. Мінімальне, що можна зробити, це збільшувати зелені насадження у містах(збільшити кількість хвойних дерев, так як вони фотосинтезують цілий рік), мили вулиці від пилу та бруду(особливо у спекотні дні), розробити інфраструктуру для велосипедного руху, та заохочувати містян пересідати на двоколісний транспорт, перестати спалювати опале листя(адже тонна листя яке спалили виділить близько 40 кілограмів токсичного газу).

Для покращення екологічної ситуації в Україні необхідно проводити регулярний аналіз та моніторинг найбільш небезпечних промислових об'єктів в

нашій країні, і робити це у режимі реального часу. Таким чином ми матимемо дані стосовно того, яке підприємство забруднює найбільше те чи інше місто, та в першу чергу модернізувати їх(можливо за допомогою новітніх очисних споруд).

Основною метою цієї дипломної роботи і є створення такої пошукової GIS-системи, яка дозволить користувачеві відстежувати забруднення повітря в його місті, дізнатися яке підприємство завдає найбільшої шкоди місту, і як змінюється динаміка кількості викинутих забруднюючих речовин. Також на карті відстежувати найбільш забруднені райони свого міста. Актуальність роботи полягає у відсутності настільки об'ємних програмних продуктів для моніторингу повітря в кожному місті України. Унікальністю програмного продукту є його мультирегіональність, яка реалізована за допомогою створеного гео-модулю, який дозволяє жителю кожного міста відстежувати інформацію саме про його рідне місто.

1.ЗАДАЧА ЕКОЛОГІЧНОЇ GIS-СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ЕКОЛОГІЧНОГО СТАНУ ПРОМИСЛОВОЇ ЗОНИ КРАЇНИ

Метою дипломного проекту є створення програмного продукту, який надає змогу вільного доступу контролювати екологічний стан міста, в якому проживають люди, відповідно щомісячним даним, що надаються підприємствами; відстежувати динаміку викидів забруднюючих речовин, дізнаватися про новини на промислових гігантах про аварії, надзвичайні ситуації, встановлення нових очисних споруд та збільшення промислової потужності підприємств. В системі є ряд надзвичайно корисних калькуляторів для розрахунку концентрації шкідливої речовини у повітрі за методикою Гауса [3], порівняння якості води у водозаборах двох річок. За допомогою інтерактивної карти відстежувати найбільш забруднені райони свого міста.

Потенційними користувачами такої системи є звичайні люди- пересічні громадяни, що піклуються про стан свого міста, дбають про його екологічну стійкість та чистоту. Простий інтерфейс дозволяє користування даним ресурсом без вікових обмежень. Адаптивний дизайн сприяє універсальності даної розробки, оскільки дозволяє її використання, незалежно від технічних характеристик пристроїв користувачів, з яких буде здійснений вхід до даного web-застосунку. Наразі, не існує аналогічної мультирегіональної системи моніторингу екологічного стану промислової зони. Існуючі не є універсальними і дають інформацію тільки по певному місту.

Отже створений програмний продукт повинен мати:

- гео-модуль для визначення міста, з якого був здійснений вхід на сайт, для відображення інформації про необхідне місто;
- відображення повної інформації про кожний небезпечних об'єкт в місті;

- регулярне оновлення інформації про промислові об'єкти;
- оперативне інформування жителів про надзвичайну ситуацію в тому чи іншому місті;
- калькулятори для швидкого та зручного розрахунку концентрації шкідливих речовин у повітрі;
- інтерактивну карту з найбільш забрудненими районами в містах
- діаграми для зручного відображення динаміки концентрацій шкідливих речовин.

Для додатку був використаний виділений сервер від хостингу Beget. Web-додаток написаний на фреймворкі CodeIgniter. Шаблон сайту та адмін частини розроблювався на Bootstrap. Для гео-модулю використовувалась відкрита база Sурехgeo. База даних – mySql, а середовище для створення БД – phpMyAdmin. Для створення піддоменів використовувалися директиви DNS хостингу. Для збереження безпеки з'єднання з системою використовується SSL сертифікат Let's Encrypt Wildcard.

Даний ресурс сприятиме активному реагуванню на погіршення екологічного стану їх міста та прийняттю відповідних заходів на містах для виправлення такої жахливої ситуації. Покращення екологічного стану кожного міста сприятиме покращенню стану і всієї країни в цілому.

2. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ПРОДУКТІВ ТА ПРОБЛЕМИ МОНІТОРИНГУ ЕКОЛОГІЧНОГО СТАНУ ПРОМИСЛОВОЇ ЗОНИ

Наразі в Україні близько 55 тисяч промислових підприємств. Зрозуміло, що дуже складно проводити моніторинг такої кількості об'єктів у реальному часі, не кажучи вже про налагодження та удосконалення фільтраційних споруд на них. Частина з цих підприємств навіть не має засобів для вимірювання якості повітря навколо, визначення концентрації шкідливих речовин у викинутому повітрі. Інша частина – дає не дуже достовірні дані. Все це разом значно ускладнює нашу задачу.

Але зовсім недавно з'явився проект SaveEcoBot (рисунок 2.1) яким інвестуються та збираються кошти для встановлення датчиків вимірювання забруднення повітря та трансляції результатів на онлайн карту. Відкритий API дає змогу отримати дані з цих станцій, що стало вихідною інформацією.

SaveEcoBot
Єдиний в Україні екологічний чат-бот

Поеднує дані про забруднення, забруднювачів та інструменти захисту довкілля.

Telegram Viber Messenger

Дізнатись більше >

ПОСКАРЖИТЬСЯ НА ЗАБРУДНЕННЯ
New
Тепер ти можеш подати скаргу на екологічне порушення, і я допоможу тобі сформулювати її за кілька кліків.
Почнемо?

Вода Повітря Ґрунт Відходи Шум

Наша місія - покращити якість повітря в країнах, що розвиваються, шляхом підвищення рівня екологічної обізнаності громадян шляхом надання їм безкоштовного та легкого доступу до індексу якості повітря та інших публічних екологічних даних.

40+ джерел даних 1504.6k користувачів чат-боту 310+ станцій SaveEcoSensor 15 країн у чат-боті 11 мов у чат-боті

Рисунок 2. 1 Save Eco Bot - екологічний чат бот

Наразі це єдиний в Україні екологічний чат-бот який містить дані про забруднення довкілля та інструменти вирішення цих проблем. Серед основних

функції які доступні користувачам даного ресурсу:

- Оцінка впливу на довкілля за допомогою планових звітів підприємств;
- Інформація про дозволи та норми викидів шкідливих речовин різними підприємствами, не залежно від груп забруднювачів;
- Інформація щодо дозволи про користування надрами;
- Дозволи на норми водокористування;
- Звіти та декларації про відходи;

Ще серед аналогів, які вдалося знайти – Codel D-CEM (рисунок 2.2). Назва ресурсу співзвучна з пристроєм за допомогою якого сайт отримує дані. Стационарний газоаналізатор DCEM від компанії CODEL представляє собою надійний промисловий пристрій для безперервного моніторингу забруднень. Даний пристрій зображений на рисунку 2.2, зазвичай використовують на виробництві для самостійного контролю забруднень. А вище згаданий ресурс приймає ці дані та відображає їх в одному місці.



Рисунок 2. 2 Codel D-CEM

Багатоканальний викидоаналізатор за допомогою інфрачервоного датчика вимірює зміни в гарячих газах які вже відпрацювали свої цикли. Сам датчик встановлюється безпосередньо в газовому потоці. Даний пристрій зазвичай використовують в процесах горіння. Codel D-CEM дозволяє моніторити концентрації

основних забруднюючих речовин, таких як CO, NO, NO₂, NO_x, SO₂, CH₄, HCl, N₂O, CO₂ використовуючи інфрачервону спектроскопію, що гарантують відсутність впливу взаємопов'язаних продуктів згорання.

Також можна знайти сайт моніторингу повітря в Києві – asm.kyivcity.gov.ua. Наразі в Києві встановлено лише два пости для моніторингу індексу забруднення повітря. Ці пости знаходяться в найбільш забруднених місцях, поблизу промислових зон. asm.kyivcity.gov.ua (рисунк 2.3) досліджує близько десяти показників, продуктів горіння, чадного газу, неорганічних сполук. Але даний ресурс ще не доведений до фінальної версії, тому ним користуватися не дуже зручно, він не дає повну інформацію про екологічний стан столиці, та не дає порад щодо вирішення основних проблем.

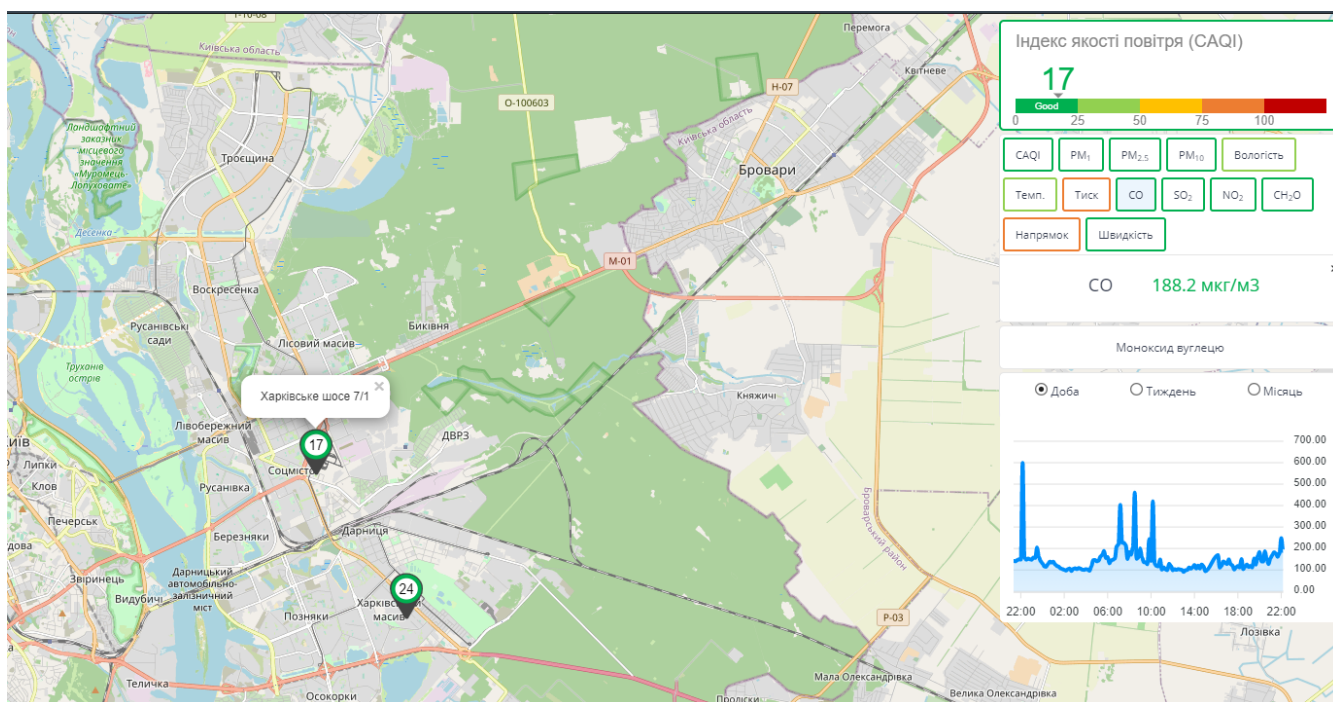


Рисунок 2. 3 Моніторинг показників забруднень в Києві

Проект stopsmog.gov.ua був створений екоактивістами з промислових міст: Запоріжжя, Маріуполя, Дніпра, та Кривого Рогу. Проект інформує гаселення про проблеми промислового забруднення.

Серед основного функціоналу – онлайн камери з промислових об'єктів. Тобто будь-то може подивитися на візуальну кількість викидів від підприємств промислових міст. Подивитися на наявність смогу на міському небі.

Також сайт містить змістовний блог з останніми новинами про екологічні зміни. Це надзвичайно важливо інформувати населення про пагубний вплив забрудненого повітря на здоров'я нації.

Головне, що підприємства починають розуміти, що знаходяться під наглядом, і більш відповідально починають ставитися до екології свого міста. Але знову ж таки, ресурс не дає повної картини, і не є універсальним. Приклад камер відеонагляду на рисунку 2.4.

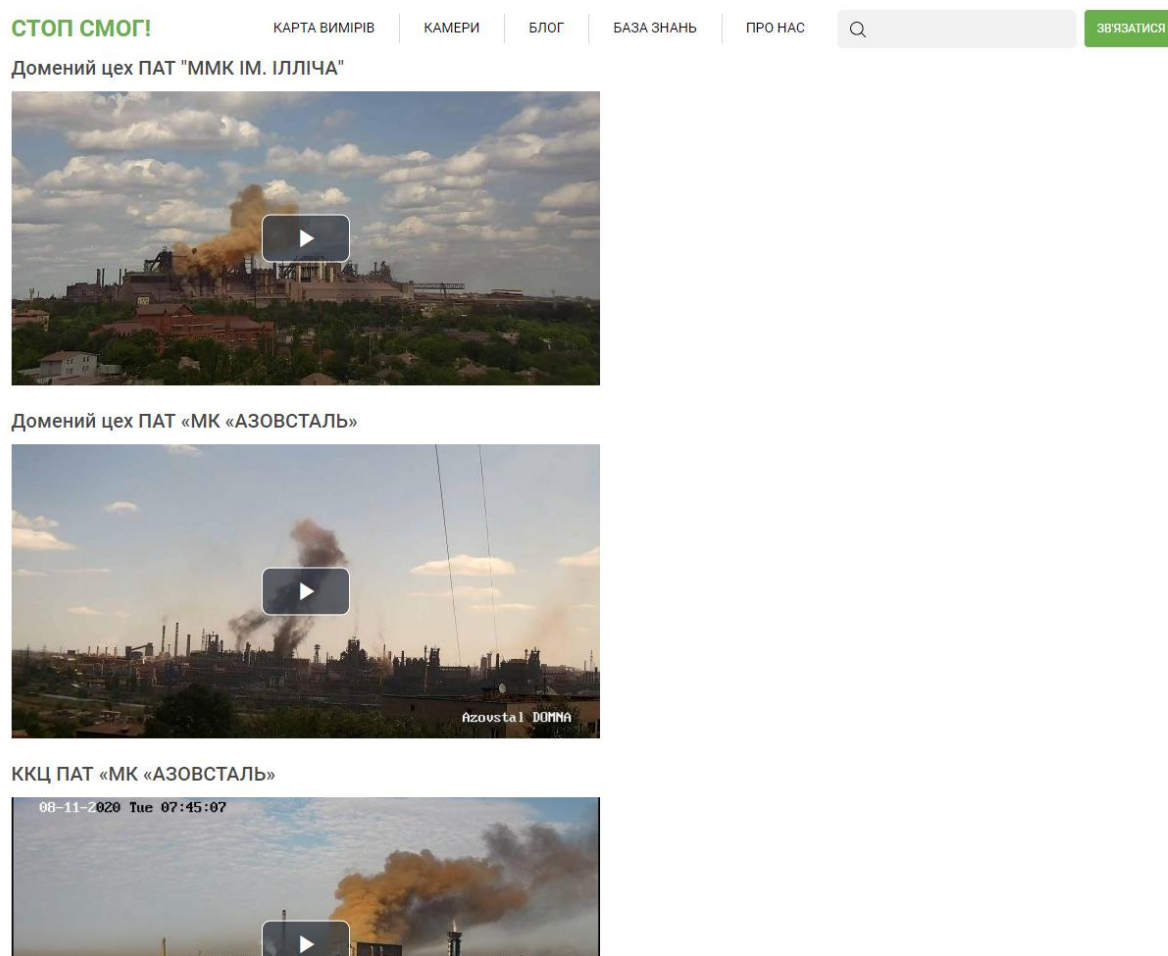


Рисунок 2. 4 Онлайн камери з підприємств

Перші два ресурси – дають інформацію лише про одне місто, «Стоп Смог» - лише транслює відео з промислових гігантів, а SaveEcoBot не дає інформації про природу викидів, кількість забруднюючих речовин від промислових об'єктів, та інформації по ним.

Проаналізувавши аналоги було зрозуміло, що всі ці проекти не виглядають

завершеними, в них безліч недоліків, через які, використовувати дані ресурси немає сенсу. Проте кожен з них має певні позитивні сторони. Тому було вирішено зібрати позитивні сторони аналогів, сумістити їх, та доповнити зручним інтерфейсом, доступністю кожному та адаптивністю. Також не існує жодного ресурсу з онлайн калькуляторами концентрацій шкідливих речовин, методик Гауса, та розрахунками якості води. Жодна компанія в Україні не дає можливості зробити такі розрахунки, хоча вони вкрай важливі.

Саме тому доцільно буде розробити програмний продукт, за допомогою якого можна буде всебічно досліджувати екологічний рівень міста, регіону, чи ситуації в країні в цілому. Серед найнеобхіднішого функціоналу є:

- Можливість переглядати інформацію по окремим містам
- Відстежувати інформацію про динаміку зміни кількості викидів від того чи іншого промислового об'єкту
- Можливість в режимі онлайн часу відстежувати якість повітря
- Розраховувати концентрацію певної речовини у повітрі, за вхідними даними
- Досліджувати якість води у водозаборі

Саме ці можливості зроблять цей ресурс реально корисним для людей. Кожен користувач хоче бачити повну та достовірну інформацію про забруднення, яку, нажаль, влада та власники цих підприємств приховують. А існуючі ресурси не дають повної картини.

Тому на даний момент немає повноцінного ресурсу, який задовольнить усім потребам користувача та буде якісно надавати актуальну інформацію. Дана проблема сприяла прийняттю рішення створити власний ресурс gis-ecology.site, який буде включати в себе все найнеобхідніше для користувача.

3. ЗАСОБИ РОЗРОБКИ

При виборі засобів розробки необхідно було проаналізувати велику кількість ресурсів аби знайти найоптимальніші застосунки до розроблюваної системи. Необхідно аби розробка та удосконалення програми(додавання нових блоків) не складало труднощів.

Для початку визначалось середовище розробки. Це важливий аспект для зручної та швидкої розробки програмних рішень.

Необхідно вибрати та впевнитися в наявності відкритих API у ресурсах з яких необхідно отримувати певні дані. При цьому оброблятися ці дані повинні за мінімальний проміжок часу.

Також потрібно не забувати про використання фреймворків. При розробці такого складного продукту без них не обійтися. Зараз є велика кількість різноманітних фреймворків, проте список лідерів не змінюється вже роки три. Тому з вибором фреймворків труднощів не буде.

Важливим є питання вибору серверу розміщення даного веб-додатку. Оскільки локальний сервер тут однозначно не підійде, доцільно задуматися про хостинг. Останнім часом кількість хостингів росте з геометричною прогресією, проте якість послуг що вони надають не на вищому рівні.

3.1 Середовище розробки

Оскільки операційні системи, з яких буде здійснюватися вхід до системи можуть бути різними, для універсальності проект доцільно виконати у вигляді web-системи, тобто сайту.

Написання коду у звичайному текстовому редакторі є незручним, тому IDE вибрано в якості середовища розробки. IDE -інтегроване середовище розробки, це

надійний інструмент, який робить розробку додатків простою, швидкою та надійною.

Основний функціонал необхідний для створенні даної системи:

- Компілятор – для того щоб перетворити код на виконуваний файл.
- Інтерпретатор – для запуску скриптів без компіляції.
- Інструменти автоматизації – для зручної збірки проекту, для допомоги у написанні вірного синтаксису тощо.

IDE об'єднує в собі всі ці елементи. За даними сайту, в 2020 році найпопулярнішими середовищами розробки були: Visual Studio Code, Atom, Sublime Text 3, Webstorm, Brackets. NuSphere пропонує всеосяжну PHPED PHP IDE, побудовану на найкращих у своєму роді компонентах. NuSphere PhpED забезпечує все необхідне для швидкого створення та доставки професійних веб-сайтів та програм.

NuSphere тісно співпрацює із спільнотою відкритих джерел для сприяння розвитку та впровадженню технологій з відкритим кодом по всьому світу. Редактор - NuSphere PhpED є надзвичайно зручним інтегрованим середовищем.

Редактор включає наступний функціонал:

- розширений редактор;
- налагоджувач PHP;
- підтримка останніх версій php;
- інтеграція з фреймворками для роботи;
- підключення баз даних;
- з'єднання для передачі даних по FTP.

Якщо аналізувати чому саме був вибраний даний редактор, то серед основних переваг можна виділити швидкість, прозорість та простоту. Є кілька речей які мають значення при створенні програмного продукту з погляду швидкості – перемикання проектів, отримання пропозицій по адаптації коду, швидкі підказки, автозаповнення. Оскільки розроблюваний проект досить складний, такі функції середовища розробки зроблять процес кодингу більш приємним.

Середовище NuSphere має надзвичайно зручний майстер проектів,

зображений на рисунку 3.1. Він допоможе провести глибокі налаштування вашого проекту. Створити локальну версію, чи підключитись до віддаленого серверу, запустити на існуючому веб-сервері чи створити власний.

Навіть при помилках при створенні чи підключенні проекту буде досить просто виправити проблеми у разі їх виникнення.

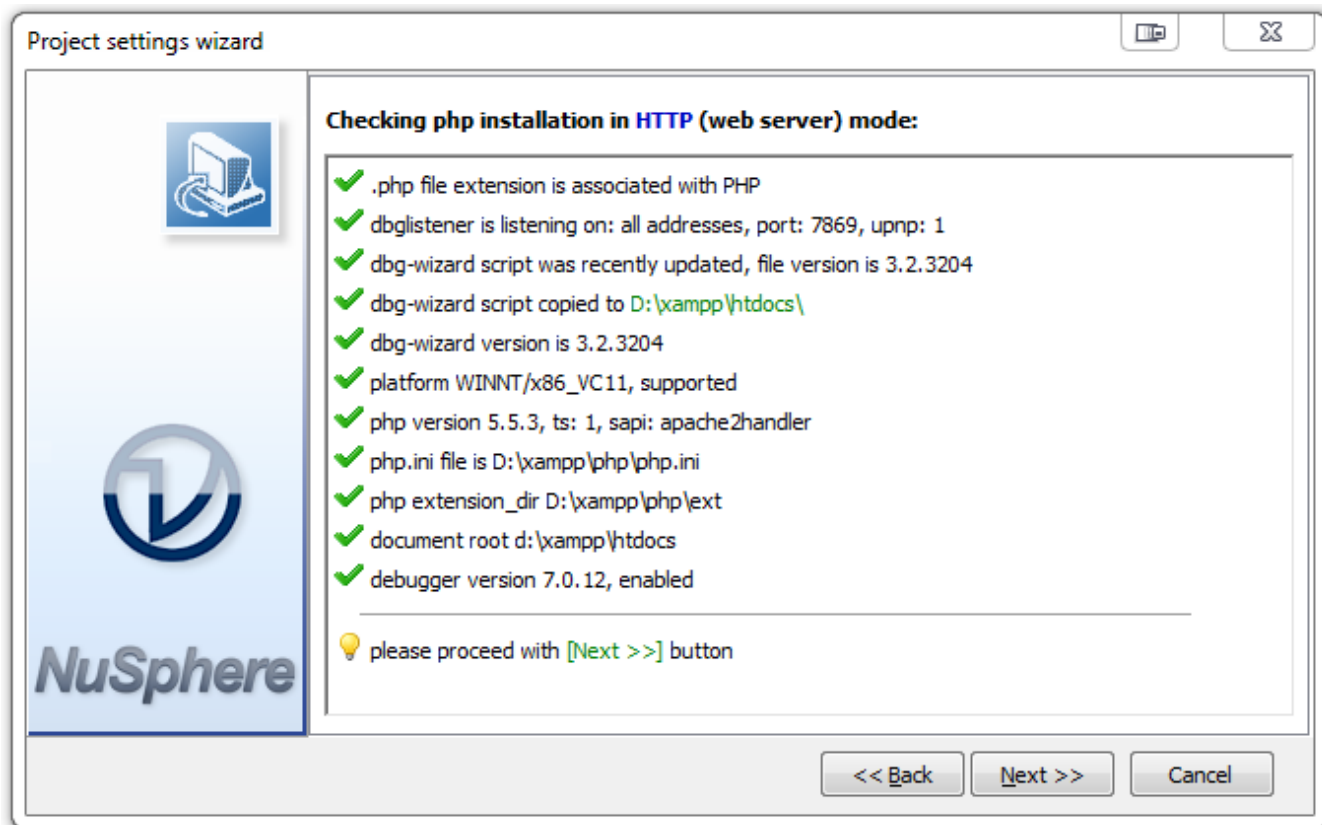


Рисунок 3. 1 Майстер проектів NuSphere

При написанні чи редагуванні коду важливими функціями є динамічне підсвічування синтаксису, швидкий пошук та навігація по коду, автодоповнення. При пошуку необхідної функції, редактор відображає подібні функції з детальним описом їх роботи та параметрами які необхідні. У підказках наведені приклади синтаксису та входів для роботи з даним елементом. Для швидкої роботи, за допомогою комбінації можна відразу додати шаблон роботи з цією функцією, що значно пришвидшить процес написання всієї системи. Приклад роботи підказок наведено на рисунку 3.2.

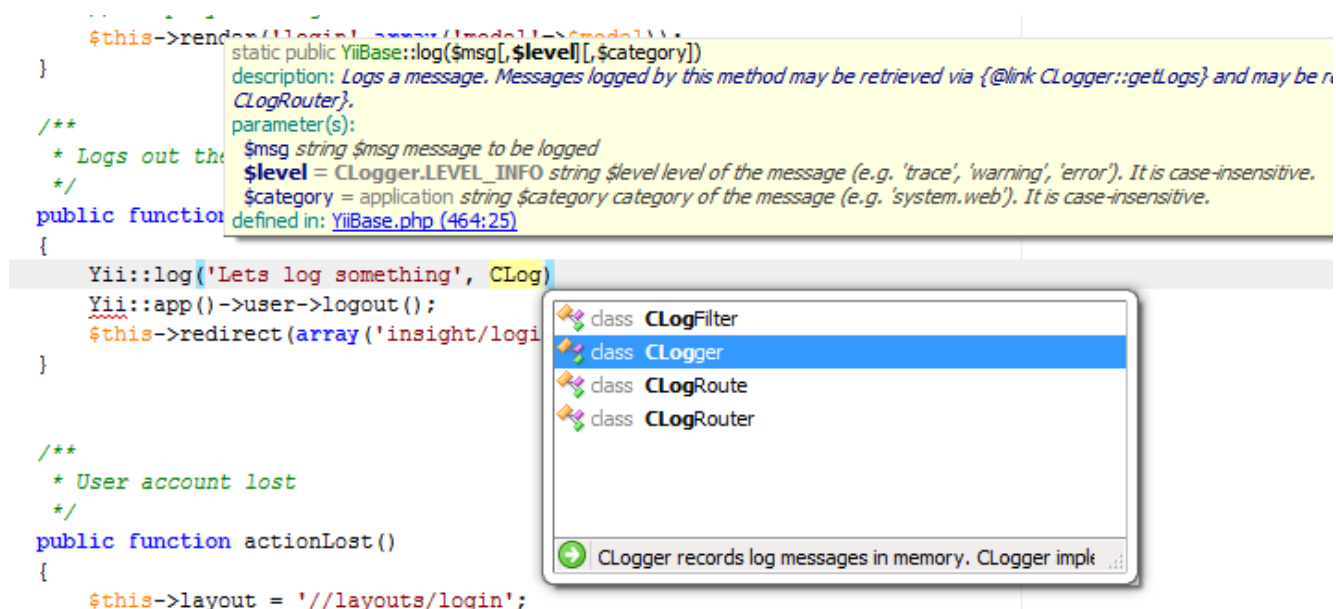


Рисунок 3. 2 Автопідказки в редакторі

Важливим етапом розробки програмного продукту є налагодження. В NuSphere налагоджувач встановлюється як модуль PHP. Майстер проекту автоматично встановить модуль для вас, якщо ви працюєте на локальному сервері. Для віддаленого сервера вам доведеться зробити це самостійно. Знову ж, тут дуже допоможе майстер, як це зробити, пропонуючи правильні шляхи та вміст файлів для вашого сервера. PhpED автоматично прослуховує вхідні підключення для налагодження та допомагає отримати з'єднання через брандмауери та маршрутизатори, якщо це необхідно.

Налагоджувач (рисунок 3.3) має всі функції, на які ми можемо очікувати; один крок, біг до або з положення курсору. Точки зупинки можуть мати додаткові умови, які оцінюються кожного разу, коли їх торкаються.

Налагоджувач можна запустити з IDE за допомогою "Запустити в налагоджувачі", але його також можна запускати в будь-який час, просто додавши параметр DBGSESSID до запиту. Цей параметр також зберігається як сеансовий файл cookie, щоб кожен наступний запит можна було автоматично налагодити.

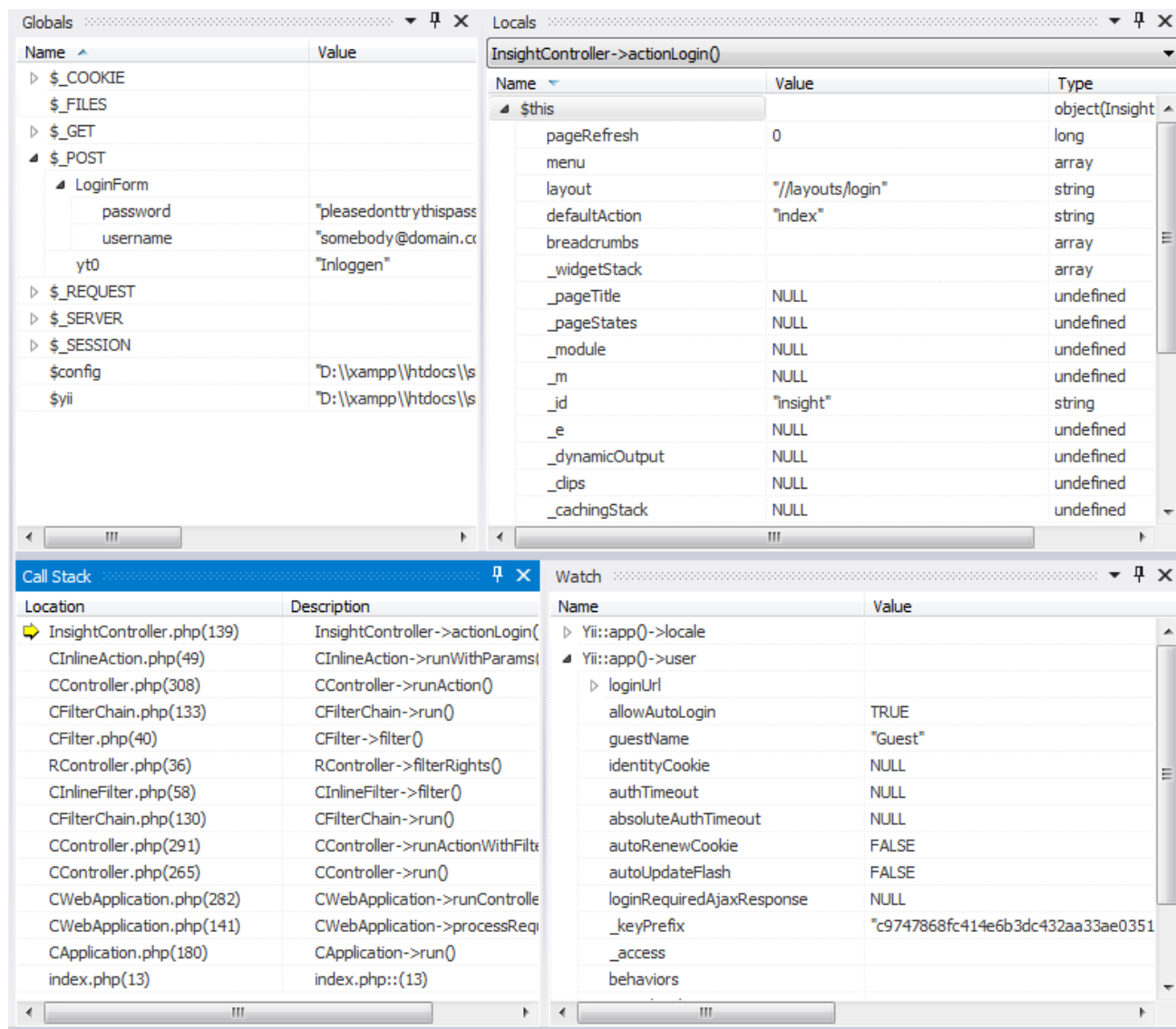


Рисунок 3. 3 Налгоджувач у редакторі

Як видно з рисунку 3.3, в налагоджувачі є велика кількість корисної інформації про глобальні та локальні змінні, про їх роботу, типи та значення.

3.2 Вибір хостингу та серверу

Для вільного доступу користувачів до ресурсу він повинен бути локальною версією. Тому необхідно вибрати хостинг, на якому можливо розмістити сайт.

На сьогоднішній день можна знайти величезну кількість хостингів на будь який смак. Зараз можна навіть орендувати сервер на іншій частині світу. Це не є

проблемою. Проте серед великої кількості хостингів було вирішено зупинитися на такому як Beget.

Для роботи обраний хостинг Beget, оскільки це гарний хостинг для будь-яких потреб: як для розміщення одного простого сайту, так і для компаній, які займаються розробкою веб-ресурсів, для розміщення сотні таких сайтів.

Оскільки даний проект не прогнозує величезних навантажень на сервер, брати VPS, тобто виділений віртуальний сервер немає необхідності. Для проекту необхідне стандарте рішення – розміщення на сервері хостингу.

Розроблений сервер має наступні характеристики:

- CPU'S:96 * Intel(R) Xeon(R) Gold 6240R CPU @ 2.40GHz;
- ОЗУ вільно: 102540 / 644426 Мб;
- Навантаження: 39.27 load average;
- Uptime: 218 днів.

Оскільки поки що користувачів на сайті не велика кількість, то такі параметри серверу розміщення цілком підійдуть. Як видно на рисунку нижче, навантаження на сервер в нормі і не є критичними.

При необхідності, можна підвищити технічні характеристики серверу. Меню аналізу навантаження на сервері можна бачити на рисунку 3.4. На це не потрібно витратити безліч часу, все робиться за допомогою однієї кнопки в особистому кабінеті користувача хостингом.

Цей хостинг надає всі необхідні інструменти для роботи: робота з базами даних через phpMyAdmin, передача даних по FTP, підключення вже готових CMS до сайту, управління записами DNS, та багато іншого.

Серед позитивних аспектів при використанні даного хостингу є детальне налаштування хостингу під себе. На мій погляд це дуже важлива річ. Допустимо використовувати 10 сайтів, але тим більше мені не потрібно для них багато Гб на серверах. У інших хостинг-провайдерів прийшлося придбати більш дорогий тарифний план, а тут досить перейти в «Управління послугами».



Полезная информация

[Описание раздела Статистика](#)
[Ограничения по нагрузке](#)
Нагрузка на сайты: 65 CP / сутки

Нагрузка на MySQL: 2500 CP / сутки

[Увеличить лимиты по нагрузке](#)
 OS: Ubuntu Server Edition

 Apache:

 MySQL: 5.7.21-20-beget

 Nginx: 1.13.4

 Perl: 5.14.2

 PHP: 5.2 5.3 5.6 7.0 7.1 7.2 7.3 7.4 8.0

 Python: 2.7.3

Сервер: sweet.beget.com

CPU'S: 96 * Intel(R) Xeon(R) Gold
6240R CPU @ 2.40GHz

ОЗУ свободно: 135805 / 644426 Мб.

Нагрузка: 35.93 load average

Uptime: 243 дней

Нагрузка на сервере (load average):

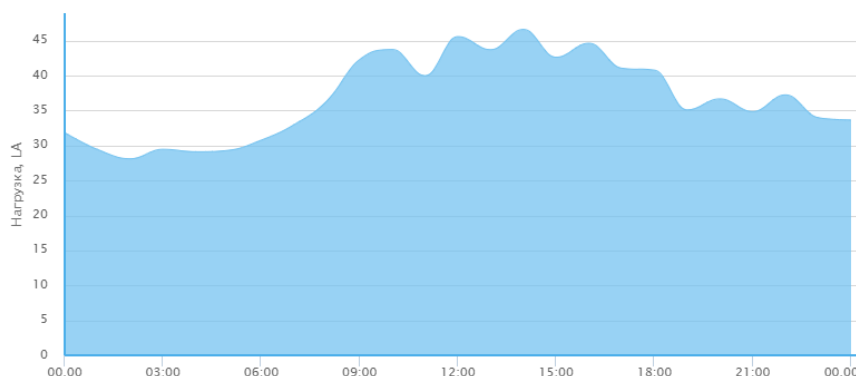


Рисунок 3. 4 Інформація про сервер

Серед позитивних аспектів при використанні даного хостингу є детальне налаштування хостингу під себе. На мій погляд це дуже важлива річ. Допустимо використовувати 10 сайтів, але тим більше мені не потрібно для них багато Гб на серверах. У інших хостинг-провайдерів прийшлося придбати більш дорогий тарифний план, а тут досить перейти в «Управління послугами».

Також плюсом є те що цей хостинг підтримує SSL сертифікати Lets Encrypt. Це безкоштовний сертифікат для підключенню по безпечному з'єднанню.

Для розміщення розробленого веб—ресурсу необхідно вибрати доменне ім'я та придбати його. Для цього можна скористатися сайтом для реєстрація доменів — піс.ua[15]. Надається можна переглянути актуальну інформацію про домени, які з них вільні, а які вже зайняті. Пошукати та вибрати необхідне вам ім'я. Тут відразу можна і придбати домен, а вже потім розмістити його на своєму хостингу, для цього й необхідна робота з DNS записами.

3.3 Вибір програмних додатків

На сьогоднішній день, більшість програмних додатків є платними, або ж демо версіями з обмеженим функціоналом. Проте можна знайти безкоштовні продукти або ж ресурси, які надають дані через API. Багато хто дає продукти з відкритим кодом, що значно полегшить роботу, ще й кошти не потрібно витратити.

Серед додатків, які використані в роботі – syrpxgeo.net[4]. Цей ресурс необхідний при розробці гео-модулю для визначення міста з якого користувач заходить на сайт та редиректу на необхідний йому міський піддомен. Syrex Geo – це ресурс який по ір-адресі користувача видає повну інформацію про нього: країну, регіон, місто в якому він знаходиться та приблизні координати, а це саме те, що мені потрібно.

Робота з цим ресурсом можлива за допомогою Rest Api, а саме через запити по HTTPS. Rest Api дозволяє отримати інформацію про ір-користувача у JSON форматі, а вже потім обробити отримані дані та перенаправити користувача на регіональний піддомен. За один запит можна отримати інформацію про 100 ір-адрес максимально, а цього мені достатньо для якісної роботи системи.

В якості системи управління базами даних була вибрана – MySQL. Це продукт всім відомої корпорації Oracle. Наразі це найпопулярніша та найвикористовуваніша СУБД. Її можна зустріти в 90% CMS систем та сайтах. І це не дивно, через основних переваг можна виділити: підтримку багатьох одночасних запитів, фіксувати записи певної довжини, зручна робота з форматами чисел, рядків, міток часу, та головне, її підтримують мабуть всі хостинги.

Для розробки шаблону сайту та да адмін частини використовувався фреймворк Bootstrap. Фреймворк – це певний набір бібліотек, які призначені для полегшення розробки програмних продуктів. Можна сказати, фреймворк створює певний кістяк програми, який використовують для з'єднання різних компонентів програми.

Серед найпопулярніших html/css фреймворків можна виділити Bootstrap. Він був створений ще у 2011 році, і до сьогоднішнього дня є безперечним лідером. Отже не дивно чому я вибрав саме його. Серед основних його переваг слід виділити адаптивність. Мабудь саме через неї цей фреймворк став настільки популярним. Bootstrap дає змогу створити стильний, лаконічний дизайн, і при цьому, шаблон буде автоматично підлаштовуватися під параметри екрану. Велика кількість стилів, шаблонів, кастомізація по сторінкам.

Використовуючи Bootstrap для створення сайту, займе в декілька разів менше часу, а сам процес буде комфортним и простим. Для використання цього фреймворку, його необхідно підключити до сторінки, і після цього всі його інструменти будуть у вашому розпорядженні.

Для створення сайту використовувався фреймворк CideIgniter. Це також всесвітньовідомий фреймворк на мові php для розробки веб-систем. Цікавою його особливістю є те, що компоненти завантажуються і процедури виконуються лише по запити до них, а не глобально. CideIgniter має вбудований компонент кешування, тому це дозволить досягти максимальної продуктивності. Основними перевагами даного фреймворку є:

- велика кількість бібліотек, які можна підключити;
- не потребує складного налаштування;
- не має певних строгих правил кодування сторінки;
- компоненти програми мають відносно слабку зв'язність;
- повна та чітка документація по роботі;
- висока продуктивність.

3.4 Вибір мов програмування

Оскільки створювана система буде мати вигляд веб-ресурсу, то серед необхідних для програмування мов будуть html, css, php, javascript.

Мова HTML (HyperText Markup Language) – іншими словами, мова розмітки гіпертексту. Її використовують для того, аби браузер міг зрозуміти як саме необхідно показати сторінку.[5] Вся мова HTML складається з тегів, які перетворюються браузером на візуальні об'єкти які ми бачимо у вигляді повноцінного сайту. Також для розмітки сторінки використовуються атрибути, складові тегів. Вони задають певні параметри тегам, як саме необхідно відобразити певний блок, де це зробити і як оформити. HTML - це мова, якою написано більшість веб-сайтів. HTML використовується для створення сторінок і забезпечення їх функціональності. Код, який використовується для того, щоб зробити їх візуально привабливими, відомий як CSS [6]

Мова програмування Php – ще одна мова програмування для створення веб сторінок. Головне її завдання – генерувати html-сторінку на стороні веб-серверу. Тобто код написаний на цій мові інтерпретується сервером у вигляд html коду, який вже передається на сторону користувача. PHP розпочався як невеликий проект з відкритим кодом, який еволюціонував у міру того, як все більше людей дізнавалося, наскільки це корисно. Расмус Лердорф випустив першу версію PHP ще в 1994 році. PHP - це рекурсивна аббревіатура від "PHP: Гіпертекстовий препроцесор". PHP - це мова сценаріїв на стороні сервера, яка вбудована в HTML. Він використовується для управління динамічним вмістом, базами даних, відстеженням сеансів, навіть побудовою цілих сайтів електронної комерції. Він інтегрований з низкою популярних баз даних, включаючи MySQL, PostgreSQL, Oracle, Sybase, Informix та Microsoft SQL Server.

Мова PHP дуже приємна у виконанні, особливо коли вона компілюється як модуль Apache на стороні Unix. Після запуску сервер MySQL виконує навіть дуже складні запити з величезними наборами результатів за час встановлення записів.

PHP підтримує велику кількість основних протоколів, таких як POP3, IMAP та LDAP. PHP4 додав підтримку Java та архітектури розподілених об'єктів (COM та CORBA), що вперше надало можливість розвитку n-рівня. Синтаксис PHP подібний до мови програмування C. [7]

Мова програмування php безперечно є лідером серед мов програмування для створення динамічних веб-сторінок. Мова швидко набрала популярність через свою простоту, високу швидкість та кросплатформовість. Більш того, більшість існуючих CMS систем написано за допомогою Php.[8]

Створення жодного сайту не обійдеться без використання JavaScript. Це кросплатформова, об'єктно-орієнтована мова програмування високого рівня. Використовується для створення сценаріїв для веб-додатків. Це необхідно для взаємодії сайту з користувачем, керування відображенням та обміну даними з сервером.[9]

Частини коду, написані на JavaScript називають скриптами, які підтримуються усіма основними браузерами, тому цю мову й називають універсальною. [10]

За допомогою скриптів можна збільшити інтерактивність сайту, зробити його більш простим для користувача, який вперше відвідав даний портал. Засобами JavaScript можна досягти концентрації уваги користувача на необхідних нам блоках. Ця мова зручна й тим, що ми отримуємо відклик від користувача миттєво та без перезавантаження сторінки.

JavaScript спочатку був створений, щоб «оживити веб-сторінки». Програми на цій мові називаються сценаріями. Їх можна записати прямо в HTML веб-сторінки та запускати автоматично під час завантаження сторінки. Сценарії надаються та виконуються як звичайний текст. Їм не потрібна спеціальна підготовка чи складання для запуску. У цьому аспекті JavaScript сильно відрізняється від іншої мови, яка називається Java.[16]

CSS – або ж таблиця каскадних стилів, мова стилю сторінки. Її використовують для візуального оформлення сторінки. Стили – це сукупність правил відображення певного блоку сторінки. З їх допомогою ми можемо концентрувати увагу користувача на необхідних нам блоках.[11]. Для використання стилів на сайті необхідно певному тегу присвоїти стиль оформлення. Також це можна робити до всієї сторінки, або ж окремого файлу.

Для обміну даними між серверами чи комп'ютерами використовується текстовий формат JSON. Він слугує для опису структури даних. Його найчастіше використовують при отриманні даних з серверу, при підключенні по Арі. Саме це було необхідно мені при отриманні даних про ір-користувача моєї системи.

JSON, або JavaScript Object Notation, - це мінімальний, читабельний формат для структурування даних. Він використовується в основному для передачі даних між сервером та веб-додатком, як альтернатива XML. Squarespace використовує JSON для зберігання та організації вмісту сайту, створеного за допомогою системи управління вмістом.

Популярність JSON також призвела до вбудованої підтримки JSON у багатьох базах даних. Реляційні бази даних, такі як PostgreSQL та MySQL, тепер мають вбудовану підтримку для зберігання та запити даних JSON. Бази даних NoSQL, такі як MongoDB та Neo4j, також підтримують JSON, хоча MongoDB використовує трохи модифіковану двійкову версію JSON за кадром.

У цій статті ми швидко розглянемо JSON і обговоримо, звідки він з'явився, його переваги перед XML, його недоліки, коли слід використовувати його та коли слід розглядати альтернативи. Але спочатку давайте зануримося в безглуздий, як виглядає JSON на практиці.

JSON - це загальний формат даних з мінімальною кількістю типів значень: рядки, числа, логічні значення, списки, об'єкти та null. Хоча позначення є підмножиною JavaScript, ці типи представлені у всіх поширених мовах програмування, що робить JSON хорошим кандидатом для передачі даних через мовні прогалини. Формат отриманих JSON даних можна бачити на рисунку 3.5.

Наведена нижче структура чітко визначає деякі атрибути людини. Він включає ім'я та прізвище, кількість входів людини в систему, чи є ця людина автором, список компаній, з якими людина працює, і список домашніх тварин цієї людини (у цьому випадку лише одну).

```
{
  "firstName": "Jonathan",
  "lastName": "Freeman",
  "loginCount": 4,
  "isWriter": true,
  "worksWith": ["Spantree Technology Group", "InfoWorld"],
  "pets": [
    {
      "name": "Lilly",
      "type": "Raccoon"
    }
  ]
}
```

Рисунок 3. 5 Дані у форматі JSON

Структура, подібна наведеній вище, може бути передана із сервера у веб-браузер або мобільний додаток, який потім виконає певну дію, наприклад, відображення даних або збереження їх для подальшого використання. Кількість даних при такій передачі не обмежена, лише можуть спостерігатися затримки при відповіді сервера, при великих навантаженнях чи низькій спроможності самого серверу.

4 ТЕХНІЧНИЙ ОПИС СИСТЕМИ

Аби чітко розуміти функціонал веб-додатку, слід детально описати архітектуру, взаємодію користувача з системою та іншими користувачами. Для цього використовують UML діаграми. Серед основних – діаграма прецедентів, діаграма класів, схема архітектури бази даних, опис візуальної складової та багато іншого.

4.1 Опис архітектури

Архітектура створеного веб-додатку проста та зрозуміла, оскільки цей сайт має бути універсальним і підходити для використання різними людьми, як за віком, так і за досвідом користування сайтами. Іншими словами, створений сайт повинен бути зрозумілим як для молодого покоління, так і для людей похилого віку які турбуються про стан свого міста.

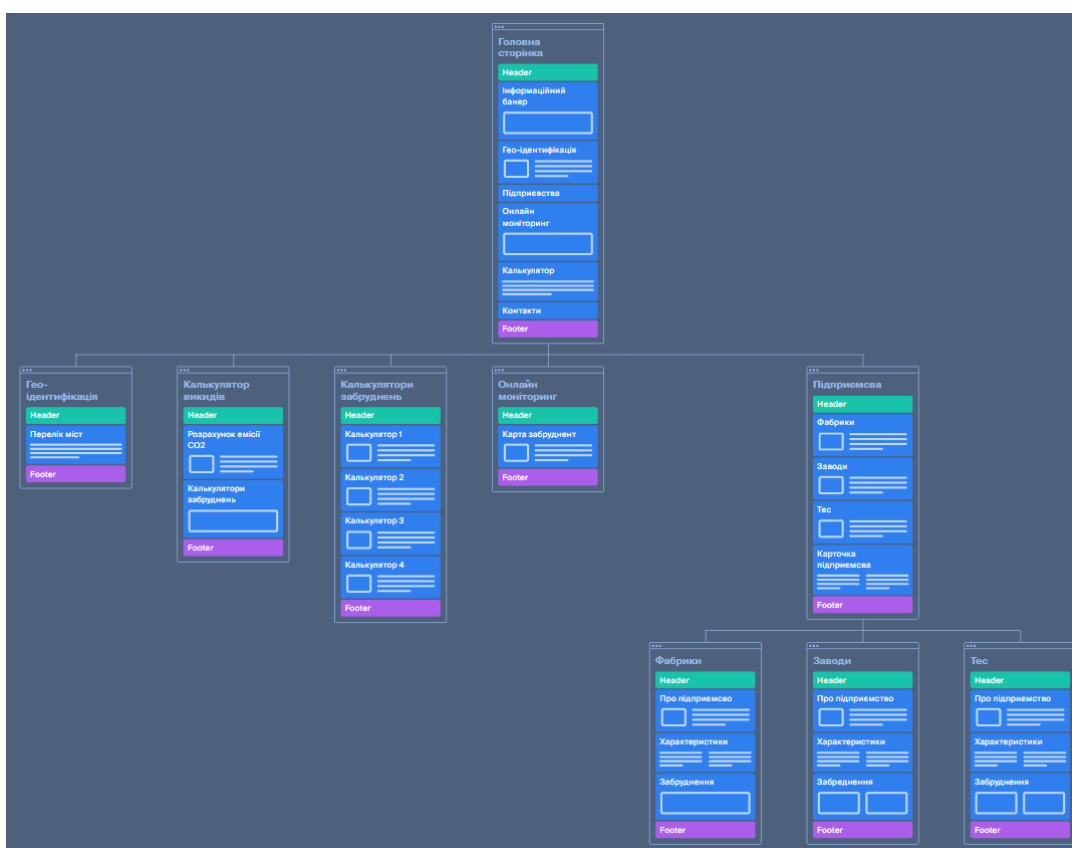


Рисунок 4. 1 Загальний вигляд веб-додатку

Для демонстрації архітектури веб-додатку доцільно використати ресурс ostopus.do. Це дуже зручний набір інструментів для веб-майстрів. З його допомогою можна візуалізувати sitemap вашого сайту, наглядно показати архітектуру побудови сайту. Архітектуру веб-додатку зображена на рисунку 4.1.

Даний ресурс – найкращий варіант для створення подібних схем. Візуальний конструктор карт сайту.

Ви можете створити свою візуальну карту сайту або структуру програми, використовуючи інтерактивний потік, користуючись усіма перевагами перетягування.

Є можливість візуалізувати за допомогою каркасів. За допомогою каркасів низької точності ви можете візуалізувати вміст кожного блоку, полегшуючи розуміння макета сторінки та підготовку вмісту для вашого майбутнього проекту.

Впорядкування вмісту сторінки, виділивши блоки вмісту кольором. Є можливість розробити свої кольорові коди та застосуйте їх до своїх проектів.

Додайте легенду до власної схеми, щоб легко передавати значення кольорів своїй команді або клієнтам. Завжди будьте на одній сторінці: визначте точні описи окремих кольорів, щоб все було чітко.

За даною схемою можна визначити основні сторінки сайту, та принцип навігації. Як ми бачимо, головна сторінка сайту має інформаційні банери, гео-ідентифікацію користувача, навігаційне меню, яке складається з посилань на такі сторінки: підприємства вашого міста, онлайн моніторинг, калькулятор викидів та контакти.

Розділ Гео-ідентифікації складається з переліку міст, в яких вже впровадження дана система. За замовчуванням користувач, зайшовши на сайт відразу потрапить на піддомен свого міста. Але якщо він захоче змінити місто та ознайомитися з інформацією про екологічний стан сусіднього міста йому необхідно з даного списку вибрати що саме його цікавить. Після вибору користувач буде перенаправлений на вже вибраний ним піддомен за допомогою редиректу. Серед редиректів є чотири основних типи:

- 301: Переспрямування 301 - це постійне переспрямування і, безумовно, найпоширеніший тип переспрямування. Це переспрямування на стороні сервера. Введення URL-адреси у браузер або натискання посилання надсилає запит на сторінку на сервер веб-сайту. Перенаправлення 301, "переміщено назавжди", являє собою набір інструкцій, які виконуються, коли запит потрапляє на сервер, автоматично перенаправляючи на іншу сторінку. Реалізація 301 переспрямування є фокусом більшості зусиль щодо SEO під час міграції сайту. Цей конкретний тип повідомляє системі, що сторінка назавжди перейшла на нову сторінку вашого сайту.
- 302: 302 переадресації не такі поширені, як 301 редиректи, і вважаються „тимчасовим переміщенням”. Впровадження їх, як правило, відбувається під час обслуговування сайту, але оскільки більшість сайтів мають режим обслуговування, де розробники та адміністратори можуть перевіряти зміни до їх запуску, сьогодні такий тип переспрямування використовується рідко.
- 300: Переспрямування “Множинний вибір” пропонує кілька варіантів для одного ресурсу. Він використовується рідко, але у нього є деякі важливі програми - багатомовні сайти та різні розширення файлів - два гарні приклади.
- 101: "Зміна протоколу" раніше була більш поширеною для таких речей, як HTTP на HTTPS, але зараз це обробляється "RewriteEngine", який використовує 301s [17]

Калькулятори викидів – наступний розділ який допоможе провести розрахунки за даними користувача. На сайті представлені різноманітні калькулятори для визначення екологічного стану різних середовищ: повітря, води, ґрунтових вод.

Третій розділ, який можна побачити в архітектурі розроблюваного сайту – онлайн моніторинг. Це свого роду карта індексації забруднення повітря. Проект SaveEcoBot займається розміщенням датчиків визначення індексу забруднення повітря, та об’єднанням цих даних в один вигляд. Саме з цього ресурсу за допомогою

API дістається інформація з таких датчиків в різних містах. Далі інформація накладається на карту, і ми можемо спостерігати повноцінну картину. [18]

Останнім розділом є інформація про підприємства в вибраному місті. Сам розділ ще поділяється на три підрозділи для зручнішої навігації. Тут відображені найнебезпечніші об'єкти вказаного міста, інформація про них, опис речовин-забрудників.

Адмін частина сайту має простий дизайн для зручності роботи. Навігація дозволяє переходити для редагування основної інформації про сайт, редагуванню карточок підприємств, додавання нового міста в базу, та нових промислових об'єктів. Також ми можемо відслідковувати інформацію про сервер, технічні помилки, перенавантаження та інші технічні моменти. Архітектура адмін частини зображена на рисунку 4.2.



Рисунок 4. 2 Архітектура адміністративної частини

Даний веб-сайт взаємодіє з декількома зовнішніми ресурсами. Серед них, ресурс для отримання інформацію про IP-адресу користувача, інформацію про місцезнаходження відвідувача сайту, чи базу датчиків забруднення.[19]

4.2 Опис баз даних

В базі даних налічується 7 таблиць, зв'язаних між собою ключами. На рисунку 4.3 зображено концептуальну схему БД. З її допомогою можна чітко прослідкувати зв'язки між таблицями, які реалізуються за допомогою ключів.[20]

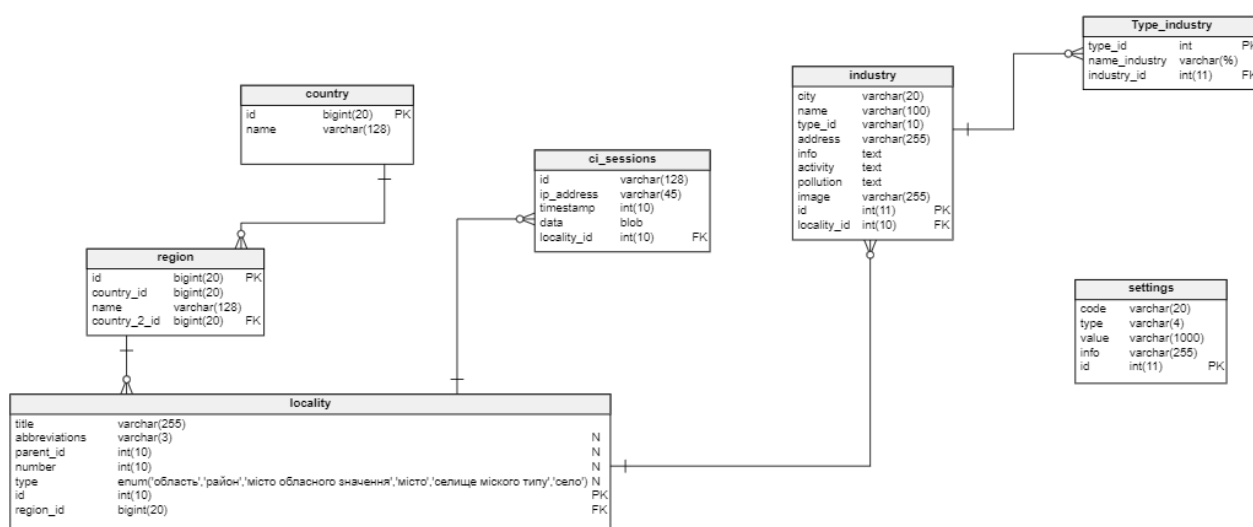


Рисунок 4. 3 Концептуальна схема бази даних

Таблиця “country” (рисунок 4.4) створення для зберігання бази країн, має два поля: Поле “id” – числове поле, значення ідентификатора країни в базі даних. Поле “name” – символічне поле, для відображення назви країни.

Columns		+ Add column		
Name	Type	N	PK	
id	bigint(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	⌵ ×
name	varchar(128)	<input type="checkbox"/>	<input type="checkbox"/>	⌵ ×

Рисунок 4. 4 Таблиця country

Таблиця “region” (рисунок 4.5) створена для зберігання бази регіонів має чотири поля:

Поле “id” – числове поле, вказує ідентифікатор регіону в базі.

Поле “country_id” – числове поле, існує для з’єднання країни та регіону в базі даних.

Поле “name” – символічне поле, в якому вказується назва регіону.

Поле “country_2_id” – числове поле для звернення до резервної бази країн.

Таблиця “locality” (рисунок 4.6) для отримання повного місця знаходження користувача у форматі: країна -> регіон -> область -> назва населеного пункту -> аббревіатура для відображення скороченої назви -> тип населеного пункту. Таблиця містить шість полів.

Name	Type	N	PK	
id	bigint(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	⌵ ×
country_id	bigint(20)	<input type="checkbox"/>	<input type="checkbox"/>	⌵ ×
name	varchar(128)	<input type="checkbox"/>	<input type="checkbox"/>	⌵ ×
country_2_id	bigint(20)	<input type="checkbox"/>	<input type="checkbox"/>	⌵ ×

Рисунок 4. 5 Таблиця region

Поле “id” – числове поле, ідентифікатор таблиці.

Поле “title” – символічне поле, для зберігання назви населеного пункту.

Поле “abbreviations” – символічне поле, містить скорочену назву(аббревіатуру) населеного пункту.

Поле “parent_id”- числове поле, ідентифікатор області до якої належить дане місто.

Поле “number”- числове поле, в якому зберігається номер населеного пункту.

Поле “type”-перераховуємий тип, для вказання типу населеного пункту.

Таблиця ci_session (рисунок 4.7) – містить в собі інформацію про кожний візит на даний ресурс. Показує ip, з якого був вхід, час, та відображає місце з якого здійснювався вхід, містить п’ять полів.

Columns						+ Add column
Name	Type	N	PK			
title	varchar(255)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×	
abbreviations	varchar(3)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	▼	×	
parent_id	int(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	▼	×	
number	int(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	▼	×	
type	enum("облас")	<input checked="" type="checkbox"/>	<input type="checkbox"/>	▼	×	
id	int(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▼	×	
region_id	bigint(20)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×	

Рисунок 4. 6 Таблица locality

Поле “id” – поле зберігання ідентифікатора заходу на сайт.

Поле ”ip_address” – символічне поле для зберігання ip адреси користувача.

Поле “timestamp” – поле містить дату та час заходу користувача на створений ресурс.

Поле “data” – поле містить інформацію у бінарному вигляді про сесію одного користувача.

Поле “locality_id” – числове поле для відображення точного місця, з якого виконався вхід на створений сайт.

Columns						+ Add column
Name	Type	N	PK			
id	varchar(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▼	×	
ip_address	varchar(45)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×	
timestamp	int(10)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×	
data	blob	<input type="checkbox"/>	<input type="checkbox"/>	▼	×	
locality_id	int(10)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×	

Рисунок 4. 7 Таблица ci_sessions

Таблиця “type_industry” (рисунок 4.8) створена для зберігання інформації про типи промислових об’єктів. Вони необхідні для більш зручної навігації по сайту. Таблиця містить 3 поля:

Поле “type_id” – містить інформацію про ідентифікатор типу підприємства.

Поле “name_industry” – символічне поле для зберігання назви типу.

Поле “industry_id” – для з’єднання певних підприємств з типами.

Columns					+ Add column
Name	Type	N	PK		
type_id	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▼	×
name_industry	varchar	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
industry_id	int(11)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×

Рисунок 4. 8 Таблиця type_industry

Наступною таблицею є “industry” (рисунок 4.9). Саме в ній зберігається інформація про підприємства які розміщені на сайті. В цій таблиці десять полів:

Columns					+ Add column
Name	Type	N	PK		
city	varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
name	varchar(100)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
type_id	varchar(10)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
address	varchar(255)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
info	text	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
activity	text	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
pollution	text	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
image	varchar(255)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
id	int(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▼	×
locality_id	int(10)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×

Рисунок 4. 9 Таблиця industry

Поле “id” – числове поле, значення ідентифікатору підприємства.

Поле “city” – символічне поле, для вказання місця розташування об’єкта промисловості. Також необхідне для роботи гео-модулю.

Поле “name” – символічне поле, зберігає назву підприємства.

Поле “type” – символічне поле, зберігає тип підприємства.

Поле “address” – символічне поле для вказання точної адреси підприємства.

Поле “info” – текстове поле, в якому зберігається детальна інформація про кожне підприємство.

Поле “activity” – текстове поле, зберігає інформацію про сферу діяльності підприємства.

Поле “pollution” – текстове поле, призначене для зберігання інформацію про викиди даного підприємства в навколишнє середовище.

Поле “image” – символічне поле, містить посилання на зображення підприємства.

Останньою таблицею є “settings” (рисунок 4.10). Вона зберігає налаштування сайту, а саме: контактну інформацію, посилання на соціальні мережі, назву сайту.

Поле “id” – ідентифікатор налаштування.

Поле “code” – для зберігання інформацію про редагуємий блок.

Поле “type” – для зберігання інформації про тип налаштування.

Поле “value” – символічне поле що містить саме значення налаштування

Поле “info” – символічне поле, назва налаштування, використовується в адмін частині сайту.

Columns					+ Add column
Name	Type	N	PK		
code	varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
type	varchar(4)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
value	varchar(1000)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
info	varchar(255)	<input type="checkbox"/>	<input type="checkbox"/>	▼	×
id	int(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▼	×

Рисунок 4. 10 Таблица settings

Кожна таблиця має як обов’язкові поля для заповнення, так і не обов’язкові. Більшість полей не можуть бути нульовими.

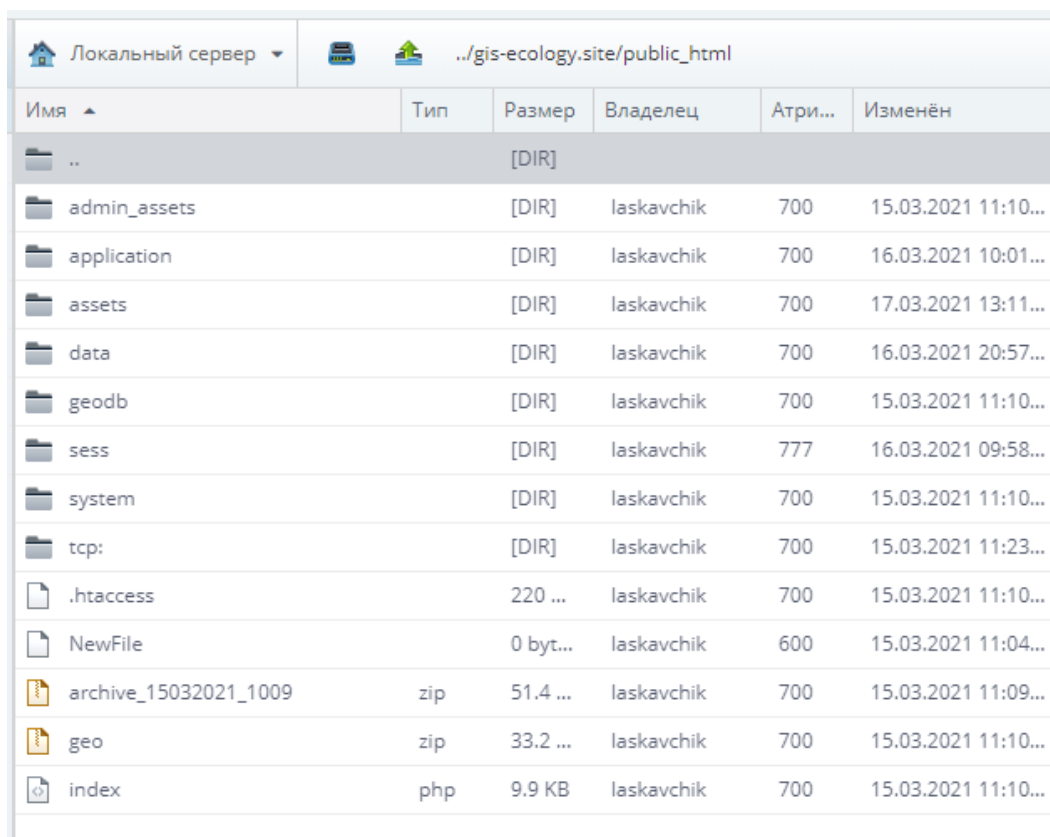
4.3 Діаграма прецедентів

Для наглядного відображення відношень між дійовими особами в системі використовується діаграма прецедентів. Цей граф показує відношення між акторами, та прецеденти з якими вони взаємодіють.

4.4 Структура програмного забезпечення

Сама файлова система сайту складається з шести основних каталогів. Перший – admin-assets. В ньому зберігаються всі файли шаблону адмін частини сайту: вхід, редагування даних, створення нових блоків та інше.

Сама файлова система сайту складається з шести основних каталогів (рисунок 4.12). Перший – admin-assets. В ньому зберігаються всі файли шаблону адмін частини сайту: вхід, редагування даних, створення нових блоків та інше.



Имя	Тип	Размер	Владелец	Атри...	Изменён
..	[DIR]				
admin_assets	[DIR]		laskavchik	700	15.03.2021 11:10...
application	[DIR]		laskavchik	700	16.03.2021 10:01...
assets	[DIR]		laskavchik	700	17.03.2021 13:11...
data	[DIR]		laskavchik	700	16.03.2021 20:57...
geodb	[DIR]		laskavchik	700	15.03.2021 11:10...
sess	[DIR]		laskavchik	777	16.03.2021 09:58...
system	[DIR]		laskavchik	700	15.03.2021 11:10...
tcp:	[DIR]		laskavchik	700	15.03.2021 11:23...
.htaccess		220 ...	laskavchik	700	15.03.2021 11:10...
NewFile		0 byt...	laskavchik	600	15.03.2021 11:04...
archive_15032021_1009	zip	51.4 ...	laskavchik	700	15.03.2021 11:09...
geo	zip	33.2 ...	laskavchik	700	15.03.2021 11:10...
index	php	9.9 KB	laskavchik	700	15.03.2021 11:10...

Рисунок 4. 12 Файлова система сайту

Наступний каталог – application. Тут зберігається сам функціонал сайту: калькулятори, карти онлайн моніторингу та інше.

Каталог assets включає в себе файли шаблону самого сайту: стилі, шрифти, фотографії, блоки сайту, та скрипти.

Data – зберігає завантажувальні файли: це можуть бути певні документи, відео фрагменти, банери та якісь фотографії.

Каталог `geodb` – це бібліотека та база даних для геолокації. Це необхідно для коректної роботи гео-модуля, котрий визначає місцезнаходження користувача.

`System` – каталог системних файлів фреймворку. Підключення баз даних, бібліотеки, мовні налаштування, шрифти – все це зберігається в цьому каталозі.

Кожен з каталогів має велику кількість підкаталогів, які в свою чергу також розгалужені. Сама структура файлової системи дуже складна, тому детально зупинятися на кожному каталозі немає сенсу.

В корні файлової системи сайту є технічні файли: `.htaccess`, `index`, `robots`, вони призначені для встановлення технічної конфігурації сайту.

Файл (або файли) `.htaccess` встановлять правила для веб-сервера лише в каталозі, де він знаходиться, та його дочірніх каталогах, без глобальних змін у роботі всього сервера. Можливість використання `.htaccess` прописується директивою (тобто командою) - `AllowOverride` - у головному конфігураційному файлі `httpd.conf`, назва якого прямо вказує на те, що налаштування всередині `.htaccess` мають пріоритет над параметрами в `httpd.conf` (якщо інше не обмежено всіма тими самими директивами `AllowOverride`). За допомогою цієї директиви ви можете дозволити все, а можете - лише деякі дії.

5. РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

Створений програмний продукт потребує підключення до мережі інтернет.

Для доступу необхідно перейти за адресою <https://gis-ecology.site>

5.1 Системні вимоги

Оскільки був створений веб-ресурс, то сурових вимог до системних вимог немає. Сайт зможуть відкрити користувачі будь-якої операційної системи: Windows, Linux, Android чи iOS.

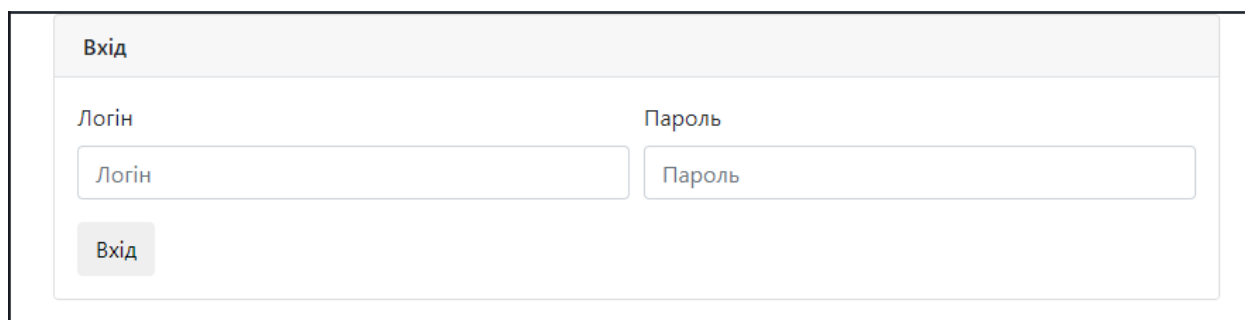
Сайт повністю адаптивний, а отже немає й різниці чи буде користувач заходити в комп'ютера(десктопна версія), чи з мобільного пристрою(мобільна версія).

Сайт підтримує всі основні браузері незалежно від їх версії. Тому зайшовши з будь-якого браузера користувач отримає якісну версію сайту.

5.2 Робота адміністратора в системі

Адмін частина сайту призначена для редагування та управління інформацією на сайті. Доступ до адмін частини звісно ж обмежений.

Для переходу до адмін частини часту необхідно перейти за посиланням: <https://gis-ecology.site/login>. Тут знаходиться блок входу до системи (рисунок 5.1).



Вхід	
Логін	Пароль
<input type="text" value="Логін"/>	<input type="text" value="Пароль"/>
<input type="button" value="Вхід"/>	

Рисунок 5. 1 Блок авторизації

Для входу необхідно ввести логін та пароль (рисунок 5.2). При невірному вводі цих даних, доступ до адмін частину буде заблокований, а користувач побачить інформаційне повідомлення виду «Логін/пароль невірний». Кількість спроб для входу необмежена.

Рисунок 5. 2 Спроба доступу з невірними даними

Після введення вірних даних, перед користувачем відкриється головна сторінка адмін частини сайту. Серед основних блоків є Налаштування та Об'єкти (рисунок 5.3).

#	Назва	Місто	Адреса	Операції
7	Придніпровська ТЕС	dnipro	Дніпро - Самарський район	<input type="button" value="Віддалити"/> <input type="button" value="Закрити"/>
8	завод Євраз	dnipro	Дніпро - Новокодакський район	<input type="button" value="Віддалити"/> <input type="button" value="Закрити"/>
9	Інтерпай сталь	dnipro	Дніпро - Індустріальний район	<input type="button" value="Віддалити"/> <input type="button" value="Закрити"/>
10	КП «Дніпроводоканал»	dnipro	Дніпро - Центральний район	<input type="button" value="Віддалити"/> <input type="button" value="Закрити"/>
11	Дніпросталь	dnipro	Дніпро - Індустріальний район	<input type="button" value="Віддалити"/> <input type="button" value="Закрити"/>
12	ТОВ «Дніпропетровський завод стінових матеріалів»	dnipro	Дніпро - Індустріальний район	<input type="button" value="Віддалити"/> <input type="button" value="Закрити"/>
13	агрегатний завод	dnipro	Дніпро - Центральний район	<input type="button" value="Віддалити"/> <input type="button" value="Закрити"/>
14	Дніпрометиз	dnipro	Дніпро - Слобожанський район	<input type="button" value="Віддалити"/> <input type="button" value="Закрити"/>
21	Вуглегірська ТЕС	donetsk	м Світлодарськ, Донецька область	<input type="button" value="Віддалити"/> <input type="button" value="Закрити"/>
22	Слов'янська ТЕС	donetsk	м Миколаївка, Донецька область	<input type="button" value="Віддалити"/> <input type="button" value="Закрити"/>

Рисунок 5. 3 Головна сторінка адмін частини

Блок Налаштування відповідає за загальні налаштування сайту: назву сайту, контактні дані(адреса, номери телефону, email, посилання на соціальні мережі), та банери на головній сторінці сайту. Блок налаштувань зображений на рисунку 5.4.

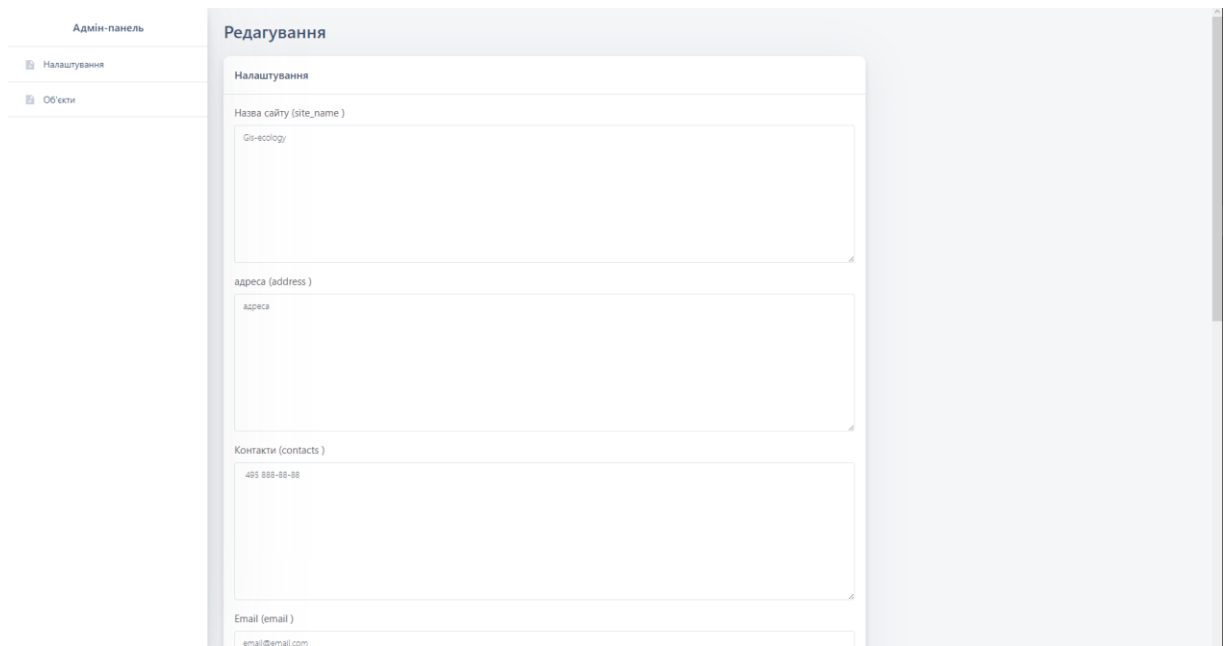


Рисунок 5. 4 Засоби редагування інформації на сайті

Блок об'єктів включає в себе список всіх підприємств що є в базі даних сайту, меню їх редагування, видалення чи додавання нового (рисунок 5.5).

#	Назва	Місто	Адреса	Операції
7	Придніпровська ТЕС	dnipro	Дніпро - Самарський район	<input type="button" value="✖ Видалити"/> <input type="button" value="⋮ Змінити"/>
8	завод Євраз	dnipro	Дніпро - Новокодакський район	<input type="button" value="✖ Видалити"/> <input type="button" value="⋮ Змінити"/>
9	Інтерпайп сталь	dnipro	Дніпро - Індустріальний район	<input type="button" value="✖ Видалити"/> <input type="button" value="⋮ Змінити"/>
10	КП «Дніпроводоканал»	dnipro	Дніпро - Центральний район	<input type="button" value="✖ Видалити"/> <input type="button" value="⋮ Змінити"/>

Рисунок 5. 5 Меню об'єктів

Для більш зручного відображення, в таблиці виводяться дані не тільки про назву підприємства, а й про місто, та адресу де воно знаходиться.

Для видалення об'єкту з бази даних необхідно натиснути кнопку «Видалити». Після її натискання з'явиться попереджувальне повідомлення про підтвердження видалення об'єкту з бази даних. Це створено аби виключили можливість ненароком видалити якісь важливі дані. Процес видалення, та попередження при видаленні зображено на рисунку 5.6.

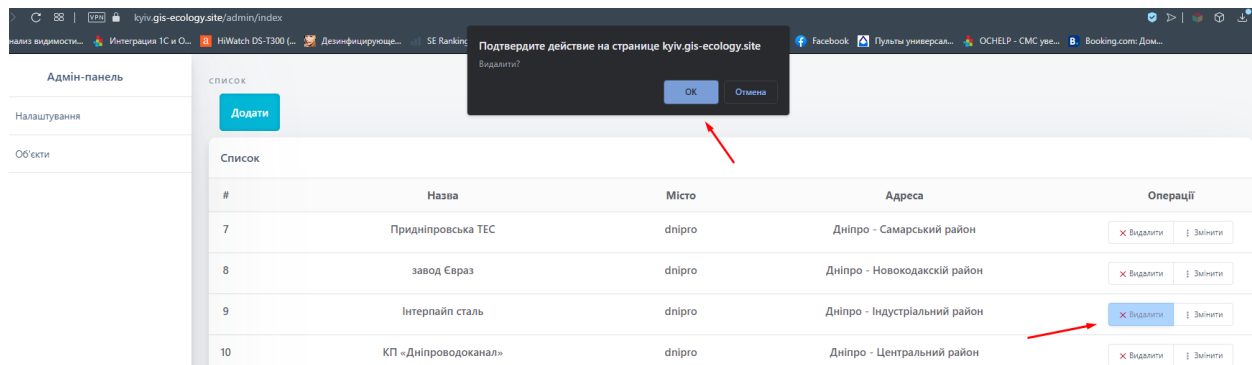


Рисунок 5. 6 Видалення даних про об'єкт

Для редагування даних необхідно скористатися кнопкою «Змінити». Після її натискання перед користувачем з'явиться меню редагування інформації про підприємство (рисунок 5.7).

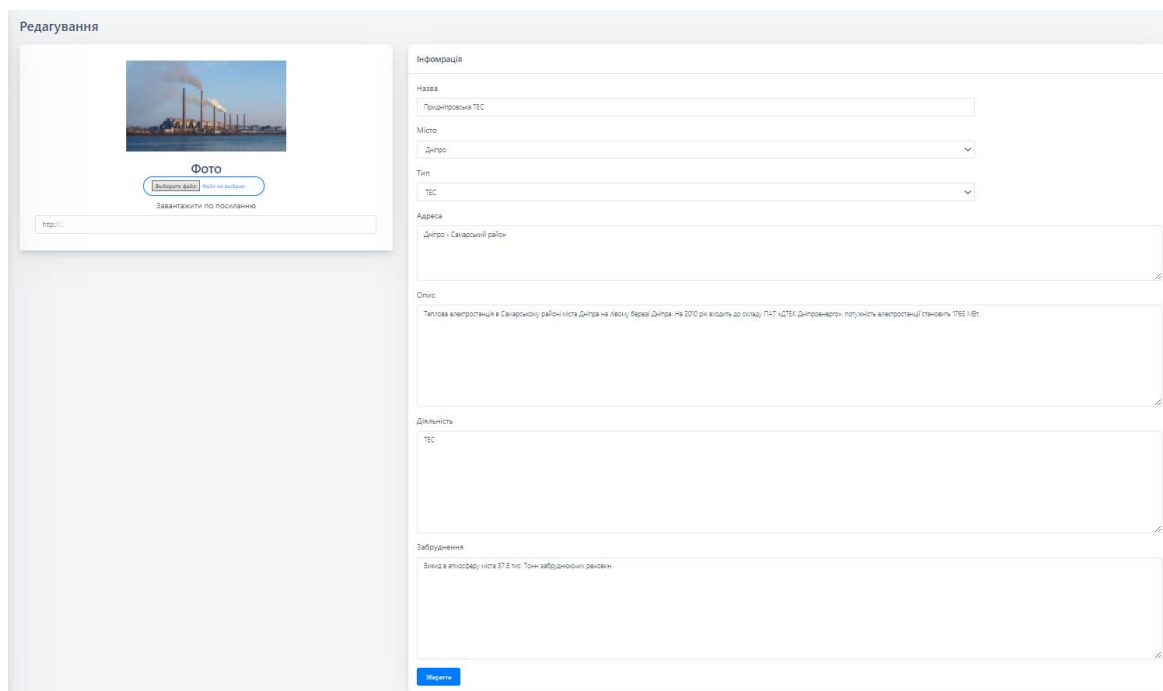


Рисунок 5. 7 Редагування інформації про підприємство

Серед основних даних для заповнення є фотографія або посилання на зображення підприємства, назва, точна адреса та контакти, опис підприємства, опис діяльності та устаткування та остання інформація про забруднення навколишнього середовища. Після редагування даних необхідно натиснути «зберегти», для збереження змін.

Для додавання нового підприємства необхідно в головному меню натиснути кнопку «Додати». Перед користувачем відкриється блок додавання нового

підприємства. Даний функціонал зображено на рисунку 5.8.

Рисунок 5. 8 Додавання нового підприємства

Для початку необхідно завантажити зображення підприємства. Зробили це можна як завантаживши вже існуюче фото на пристрої чи вказати посилання на фото. Процес додавання фотографії зображено на рисунку 5.9.

Рисунок 5. 9 Додавання фото

Далі слід вказати назву підприємства, контактні дані, додати опис даної структури(вказати що це за підприємство, чим займається, потужність та ін.), рід

діяльності, та останні дані про кількість викидів в навколишнє середовище. З випадуючого списку слід вибрати місто, в якому знаходиться це підприємство.

Також користувач адмін частини може додавати міста в базу даних, створювати для них субдомени, та наповнювати базу підприємств.

5.3 Можливості гостя на сайті

Для відображення сайту звичайному користувачеві слід перейти за посиланням: <https://gis-ecology.site>.

Після входу на головний домен, користувача буде перенаправлено на міський піддомен сайту того міста, з якого зробив вхід користувач. Якщо система не зможе визначити місто користувача, то з'явиться список міст для вибору(рисунок 5.10).



Рисунок 5. 10 Вибір міста для відображення інформації

У навігаційному меню є можливість перейти до розділів: «Підприємства вашого міста», «Онлайн моніторинг», «Калькулятор викидів», «Контакти».

Перейшовши до розділу підприємств, перед користувачем відкривається список карточок еколого-небезпечних підприємств його міста. Праворуч, можна бачити певний фільтр, для більш зручної орієнтації на сайті. За допомогою фільтра можна вибрати певну категорію підприємств: «Фабрики», «Заводи», «ТЕС» (рисунок 5.11).

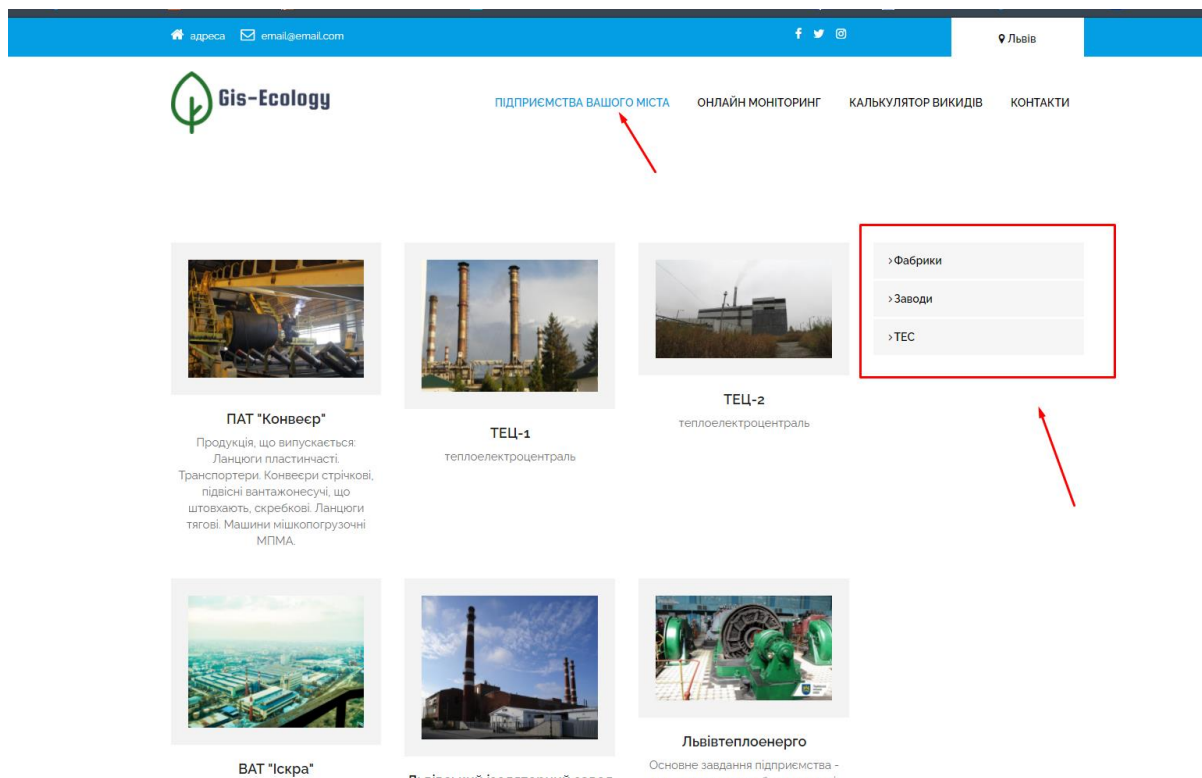


Рисунок 5. 11 Розділ підприємств вашого міста

Натиснувши на карточку підприємства що вас зацікавило, користувач потрапить на сторінку вибраної установи.

На сторінці підприємства користувач може ознайомитися з інформацією про вибраний завод, прочитати характеристики, та подивитися останні дані по забрудненню навколишнього середовища цим підприємством (рисунок 5.12).

За допомогою зручної навігації користувач може перейти до будь-якої сторінки сайту за мінімальну кількість переходів, що не буде ускладнювати роботі даної системи.

**ВАТ "Іскра"**

м. Львів, вул. Вулицька 14

- > Фабрики
- > Заводи
- > ТЕС

Про підприємство

Характеристики

Забруднення

Іскра - єдина в Україні потужна світлотехнічна компанія, що має повний технологічний цикл виробництва - від наукових розробок до випуску компонентів і готової високоякісної продукції, яка займає сьогодні 70% українського ринку.

Рисунок 5. 12 Картка підприємства

Розділ онлайн моніторинг слугую для відстеження якості повітря у вашому місті в режимі реального часу(рисунок 5.13).

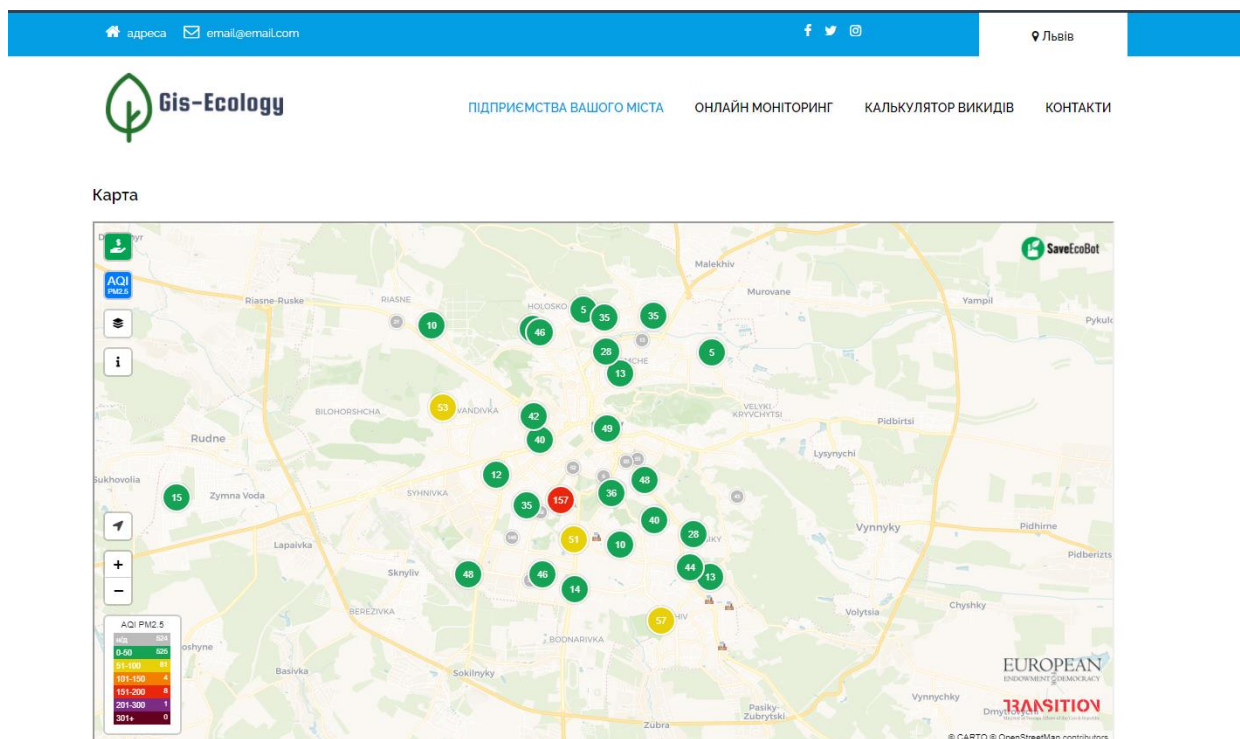


Рисунок 5. 13 Онлайн моніторинг якості повітря

На карті відображені інформація зі всіх станцій оцінки якості повітря, за

знаходяться в вашому місті. Як видно на рисунку 5.13, всі станції відображаються певним кольором. Сірий колір означає, що в даний момент відсутнє з'єднання з цим датчиком оцінки якості повітря. Зелений – каже що якість повітря гарна, і це найкращий час аби подихати реально свіжим повітрям. Жовтий колір сигналізує про незначне погіршення якості повітря. Оранжевий – якість повітря ще гірша, людям з хворобами дихальних шляхів не рекомендують довго перебувати на вулиці. Червоний колір каже що повітря забруднене, краще утриматися від довгих прогулянок. Фіолетовий колір сигналізує про надзвичайно забруднене повітря, при такому рівні перебувати на вулиці небезпечно. Останнім кольором є коричневий, ці каже що індекс забруднення повітря вищий за норму більш ніж в 6 разів, цей колір сигналізує про якусь надзвичайну ситуацію(рисунок 5.14).

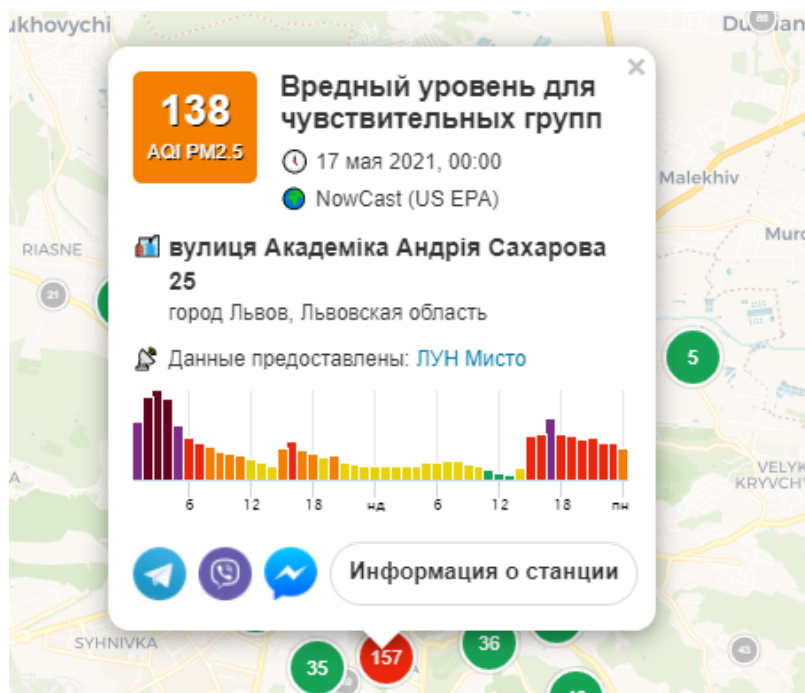


Рисунок 5. 14 Детальна інформація про датчик

Натиснувши на будь-який датчик, можна ознайомитися з його інформацією. Це і місце розташування датчика, і останній час оновлення, і динаміка зміни якості повітря. Також за допомогою інтерактивного меню, є змога надіслати цю інформацію за допомогою месенджера будь-кому.

Розділ з калькуляторами дає змогу самостійно розрахувати концентрацію шкідливих речовин, індекс забруднення повітря, якості води та інше(рисунок 5.15).

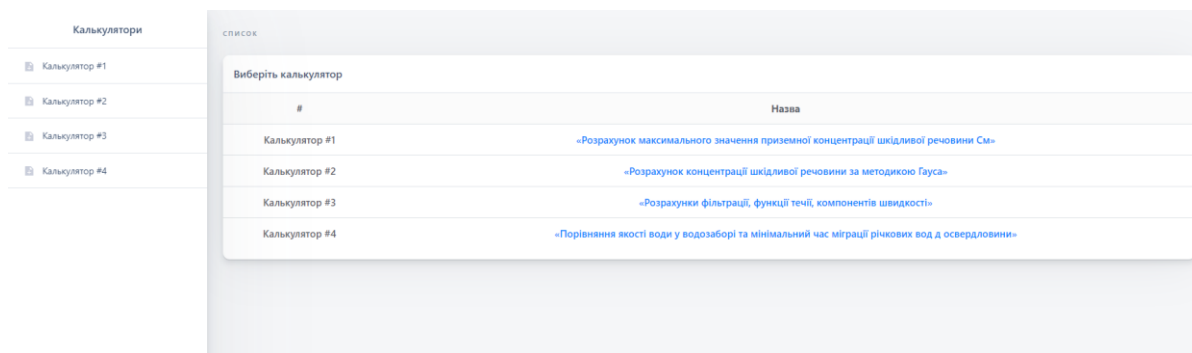


Рисунок 5. 15 Калькулятори забруднень

У головному меню перед користувачем буде список доступних калькуляторів. Вибравши необхідний, ми побачимо інформацію про метод розрахунку, формулу(рисунок 5.16).

Рисунок 5. 16 Розрахунок максимального значення приземної концентрації шкідливої речовини

Ліворуч є історія калькулятора, тобто детальний опис величин. Праворуч – дані для самостійного заповнення. Якщо у формулі присутні константи, то вони вже будуть внесені у необхідне поле, але при необхідності їх можна змінити.

Останнім розділом є Контакти. Він слугує для зв'язку користувача з менеджерами сайту. За допомогою цього розділу можна пропонувати адміністратору сайту додавати нові дані(міста, підприємства), чи пропонувати співпрацю для досягнення загальної цілі – зроби країну екологічно чистою.

ВИСНОВКИ

В ході виконання роботи було проведено детальний аналіз предметної області. Для виконання поставлених вимог було проаналізовано понад десять аналогів, досліджені їх переваги та недоліки. Для розробки використовувались мови програмування JavaScript, PHP, HTML та таблиця каскадних стилів CSS. Сама система розроблялась з використанням фреймворків Bootstrap та CodeIgniter в середовищі розробки NuSpherePhpED. Для тестів роботи системи використовувались фреймворки Jest та AnalyzeMe. Адаптивність системи перевірялась на вбудованому в середовище розробки вікні адаптиву. Веб-ресурс успішно пройшов тести на більш ніж 210 моделях адаптивності.

Актуальність даного додатку була доведена за допомогою досліджень проблематики моніторингу екологічного стану промислової зони.

Перед розробкою програмного рішення були проаналізовані існуючі системи, які частково допомагають моніторити забруднення атмосфери. Проте серед аналогів немає жодної цілісної системи яка б давала повну інформацію. Серед основних недоліків вже існуючих програмних рішень можна виділити:

- відсутність мультирегіональності, а саме відображення інформації лише про певне місто;
- відсутність повної інформації про природу забруднень та речовини що викидаються;
- велика вартість ліцензії на користування даними ресурсами;
- відсутність карти онлайн моніторингу.

Всі ці недоліки були взяті за основу при розробці функціоналу даного ресурсу, та доповнені іншими, не менш важливими функціями.

Перед розробкою робочої версії були створені прототипи програмного продукту, які і лягли в основу розробки.

Створений програмний продукт є універсальним і дає повну інформацію по будь-якому місту України.

Ресурс має обширний функціонал для зручної роботи:

- гео-модуль для визначення міста, з якого був здійснений вхід на сайт, для відображення інформації про необхідне місто;
- відображення повної інформації про кожний небезпечних об'єкт в місті;
- регулярне оновлення інформації про промислові об'єкти;
- оперативне інформування жителів про надзвичайну ситуацію в тому чи іншому місті;
- калькулятори для швидкого та зручного розрахунку концентрації шкідливих речовин у повітрі;
- інтерактивну карту з найбільш забрудненими районами в містах
- діаграми для зручного відображення динаміки концентрацій шкідливих речовин.

На даний момент ресурс є вседоступним для використання в мережі Інтернет, тобто призначений для більшості населення нашої країни. Саме інтернет ресурс є найбільш зручним та універсальним видом даного продукту. Саме так його зможе використати найбільша кількість користувачів.

Таким чином були здобуті навички розробки веб-ресурсів, роботи з API сторонніх ресурсів, налаштування серверів для розміщення сайту. Також були отримані навички тестування програмного продукту, що є невід'ємною частиною створення будь якого програмного рішення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Супрун У. Як забруднене повітря впливає на здоров'я [Електронний ресурс] / Уляна Супрун. – 2019. – Режим доступу до ресурсу: suprun.doctor.
2. Дослідження якості повітря [Електронний ресурс] – Режим доступу до ресурсу: <https://www.iqair.com/ru/>.
3. Прогнозування ризиків для здоров'я населення від забруднення атмосферного повітря викидами підприємств : дис. канд. біол. наук / . – Дніпро, 2017. – 264 с.
4. Визначення місцезнаходження за ір [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: syrexgeo.net.
5. Дакетт Д. HTML і CSS. Розробка і дизайн веб-сайтів / Джон Дакетт., 2017. – 463 с.
6. Що таке html і чому його має знати кожен [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://skillbox.ru/media/code/chto_takoe_html/.
7. Занстра М. PHP. Об'єкти, шаблони і методики програмування / Мет Занстра. – Москва: Вільямс, 2015. – 576 с. – (4).
8. Мова програмування php [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://web-creator.ru/articles/php>.
9. JavaScript. The Definitive Guide, 2008. – 992 с. – (4).
10. Кантор І. Введение в JavaScript [Електронний ресурс] / Ілья Кантор. – 2019. – Режим доступу до ресурсу: <https://learn.javascript.ru/intro>.
11. Duckett J. HTML and CSS: Design and Build Websites / Jon Duckett., 2020. – 670 с. – (1).
12. Elliott E. Programming JavaScript Applications: Robust Web Architecture With Node, Html5, And Modern Js Libraries / Eric Elliott., 2013. – 253 с. – (First Edition).
13. Meeks E. D3.js in Action: Data visualization with JavaScript / Elijah Meeks., 2017. – 375 с. – (Second Edition).
14. Verou L. CSS Secrets: Better Solutions to Everyday Web Design Problems / Lea Verou., 2015. – 354 с. – (Third Edition).
15. Ніс. Придбання доменного імені [Електронний ресурс] / ніс – Режим доступу до ресурсу: <https://nic.ua>.

16. An Introduction to JavaScript [Электронный ресурс] – Режим доступа до ресурсу:
<https://javascript.info/intro>.
17. How Redirects Work [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.digitalthirdcoast.com/blog/301-101-redirects-work>.
18. Matthias B. Webhooks Events for REST APIs / Biehl Matthias., 2017. – 130 с. – (1).
19. Timothy R. IP Address Management: Principles and Practice / Rooney Timothy. –
London. – 448 с. – (1).
20. Allen T. SQL All-In-One For Dummies, 3rd Edition / Taylor Allen., 2019. – 768 с. –
(3).

Додаток 01

Пошукова GIS система екологічно небезпечних промислових об'єктів

Специфікація

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ТМ-71165

Аркушів 1

2021

Позначення	Найменування	Відмітки
Документація		
УКР.НТУУ"КПІ"_ТЕФ_АПЕ ПС_ТМ7171165	Записка.docx	Пояснювальна записка
Компоненти		
УКР.НТУУ"КПІ"_ТЕФ_АПЕ ПС_ТМ7171165	web.php	Основний компонент

Додаток 02

Пошукова GIS система екологічно небезпечних промислових об'єктів

Текст програмного модулю

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ТМ-71165

Аркушів 11

Київ 2021

```

<div class="main-content-container container-fluid px-4">
  <!-- Page Header -->
  <div class="page-header row no-gutters py-4">
    <div class="col-12 col-sm-8 text-center text-sm-left mb-0">
      <span class="text-uppercase page-subtitle"></span>
    </div>
  </div>

  <div class="row">
    <div class="col-lg-6">
      <div class="card card-small mb-4 pt-3">
        <div class="card-header border-bottom text-center">
          <div class="mb-3 mx-auto">
            <h4 class="mb-0">Розрахунок концентрації шкідливої речовини за
методикою Гауса</h4>

          </div>
          <ul class="list-group list-group-flush">
            <li class="list-group-item px-4">
              <strong class="text-muted d-block mb-2">Розрахунок проводиться
за формулами</strong>
               </div>
            </li>
            <li class="list-group-item p-4">
              <strong class="text-muted d-block mb-2">Опис величин</strong>
              <span>
                <p><p>
<p class="MsoNormal">
<p class="MsoNormal"><strong style="mso-bidi-font-weight: normal;"><span style="mso-ansi-language:
UK;">C(</span></strong><strong style="mso-bidi-font-weight:
normal;"><span style="mso-ansi-language: EN-US;" lang="EN-US">x</span><span
lang="RU">,</span></strong><strong style="mso-bidi-font-weight:
normal;"><span style="mso-ansi-language:
EN-US;" lang="EN-US">y</span><span
lang="RU">,</span></strong><strong style="mso-bidi-font-weight:
normal;"><span style="mso-ansi-language: EN-US;" lang="EN-US">z</span></strong><strong style="mso-
bidi-font-weight:
normal;"><span style="mso-ansi-language:
UK;">)</span></strong><span lang="RU"> -</span><span style="mso-ansi-language:
UK;">концентрація речовини в точці з координатами </span><span style="mso-ansi-
language: EN-US;" lang="EN-US">x</span><span lang="RU">,</span><span style="mso-ansi-
language: EN-US;" lang="EN-US">y</span><span lang="RU">,</span><span style="mso-ansi-
language: EN-US;" lang="EN-US">z</span><span style="mso-ansi-language: UK;"><span
style="mso-spacerun: yes;">&nbsp;&nbsp;&nbsp;</span>мг/м3</span></p>
<p class="MsoNormal"><strong style="mso-bidi-font-weight: normal;"><span style="mso-ansi-
language: EN-US;" lang="EN-US">Q</span><span lang="RU">-</span></strong><span lang="RU">
</span><span style="mso-ansi-language: UK;">інтенсивність викиду речовини
з джерела забруднення г/с</span></p>
<p class="MsoNormal"><strong style="mso-bidi-font-weight: normal;"><span style="mso-ansi-
language: UK;">K </span></strong><span style="mso-ansi-language: UK;">&ndash;
коефіцієнт перерахунку =1*10<sup>6</sup></span></p>
<p class="MsoNormal"><strong style="mso-bidi-font-weight: normal;"><span style="mso-ansi-
language: EN-US;" lang="EN-US">V</span></strong><span lang="RU"> &ndash;
</span><span style="mso-ansi-language: UK;">Вертикальні умови розсіювання домішки в
атмосфері</span> <span lang="RU"></span></p>
<p class="MsoNormal"><strong><span style="font-size: 12.0pt; line-height: 107%;"
lang="RU">&sigma; <sub>y</sub> &sigma; </span><span lang="RU">z </span></strong><span style="mso-
fareast-font-family: &quot;Times New Roman&quot;; mso-fareast-theme-font:
minor-fareast; mso-ansi-language: UK;">&ndash; відхилення розсіювання в
горизонтальному та вертикальному напрямку. Внаслідок вітру наприклад, м</span></p>
<p class="MsoNormal"><strong><span style="mso-fareast-font-family: &quot;Times New

```

```

Roman&quot;; mso-fareast-theme-font: minor-fareast; mso-ansi-language:
UK;">u<sub>s</sub></span></strong><strong><sub><span style="mso-fareast-font-family:
&quot;Times New Roman&quot;; mso-fareast-theme-font: minor-fareast;">
</span></sub></strong><span style="mso-fareast-font-family: &quot;Times New
Roman&quot;; mso-fareast-theme-font: minor-fareast; mso-ansi-language: UK;"><span
style="mso-spacerun: yes;">&nbsp;&nbsp;&nbsp;</span>&ndash; швидкість вітру на необхідній
висоті</span><span style="mso-fareast-font-family: &quot;Times New Roman&quot;; mso-
fareast-theme-font: minor-fareast;" lang="RU"> (2.3)<em style="mso-bidi-font-style:
normal;"><sup></sup></em></span></p>
<p class="MsoNormal"><strong style="mso-bidi-font-weight: normal;"><span style="mso-
fareast-font-family: &quot;Times New Roman&quot;; mso-fareast-theme-font: minor-
fareast;" lang="RU"><span style="mso-spacerun: yes;">&nbsp;</span>h<sub>s</sub>
&ndash; </span></strong><span style="mso-fareast-font-family: &quot;Times New
Roman&quot;; mso-fareast-theme-font: minor-fareast; mso-ansi-language: UK;">висота
джерела викиду<br /> <strong>u</strong></span><strong><span style="mso-fareast-font-
family: &quot;Times New Roman&quot;; mso-fareast-theme-font: minor-fareast;">
</span></strong><strong><sub><span style="mso-fareast-font-family: &quot;Times New
Roman&quot;; mso-fareast-theme-font: minor-fareast; mso-ansi-language:
UK;">ref</span></sub></strong><span style="mso-fareast-font-family: &quot;Times New
Roman&quot;; mso-fareast-theme-font: minor-fareast;"> <span lang="RU">&ndash;
</span></span><span style="mso-fareast-font-family: &quot;Times New Roman&quot;; mso-
fareast-theme-font: minor-fareast; mso-ansi-language: UK;">Приземна швидкість вітру
<br /> </span><strong><span style="mso-fareast-font-family: &quot;Times New
Roman&quot;; mso-fareast-theme-font: minor-fareast; mso-ansi-language: EN-US;"
lang="EN-US">z</span></strong><strong><span style="mso-fareast-font-family:
&quot;Times New Roman&quot;; mso-fareast-theme-font: minor-fareast;" lang="EN-US">
</span></strong><strong><sub><span style="mso-fareast-font-family: &quot;Times New
Roman&quot;; mso-fareast-theme-font: minor-fareast; mso-ansi-language:
UK;">ref</span></sub></strong><span style="mso-fareast-font-family: &quot;Times New
Roman&quot;; mso-fareast-theme-font: minor-fareast;"> </span><span style="mso-fareast-
font-family: &quot;Times New Roman&quot;; mso-fareast-theme-font: minor-fareast; mso-
ansi-language: UK;">&ndash; Висота на якій визначається приземна швидкість, м<br />
<strong style="mso-bidi-font-weight: normal;">p</strong> &ndash; Поправочний
коефіцієнт</span></p>
</p>
</p> </p>
</span>
</li>
</ul>
</div>
</div>
<div class="col-lg-6">
<div class="card card-small mb-4">
<div class="card-header border-bottom">
<h6 class="m-0">Калькулятор</h6>
</div>
<ul class="list-group list-group-flush">
<li class="list-group-item p-3">
<div class="row">
<div class="col">
<form>
<div class="form-row">
<div class="form-group col-md-12 border-bottom">
<label for="feFirstName"> Тип розрахунку.</label>
<div class="form-group">
<input type="radio" name="type" value="1"
checked="checked"> Розрахувати за стандартною формулою (2.1)<br>
<input type="radio" name="type" value="2"> Джерело
наземне (2.2)<br>
</div> </div>
<div class="form-group col-md-12 ">

```

```

        <label for="feLastName"><b>Q</b> -інтенсивність
викиду речовини з джерела забруднення г/с</label>
        <input type="text" class="form-control" id="q"
value="">

        </div>
        <div class="form-group col-md-12 ">
        <label for="feLastName"><b>Y</b> -координата</label>
        <input type="text" class="form-control" id="y"
value="">

        </div>

        <div class="form-group col-md-12 ">
        <label for="feFirstName"><b>V</b> Вертикальні умови
розсіювання домішки в атмосфері</label>
        <div class="form-group">
        <input type="text" class="form-control" id="v"
value="">
        </div>
        </div>

        </div>
        <div class="form-row">
        <div class="form-group col-md-6">
        <label for="feEmailAddress"><b>σ y</b> - відхилення
розсіювання в горизонтальному напрямку </label>
        <input type="number" class="form-control" id="d_y"
value=""> </div>
        <div class="form-group col-md-6">
        <label for="feEmailAddress"><b>σ z</b> - - відхилення
розсіювання в вертикальному напрямку </label>
        <input type="number" class="form-control" id="d_z"
value="">
        </div>
        </div>

        <div class="form-row">
        <div class="form-group col-md-12">
        <label for="feEmailAddress"><b>Hs</b> - Висота
джерела викиду </label>
        <input type="number" class="form-control" id="hs"
value=""> </div>

        <div class="form-group col-md-6">
        <label for="feEmailAddress"><b>U ref</b> - Приземна
швидкість вітру </label>
        <input type="number" class="form-control" id="uref"
value=""> </div>
        <div class="form-group col-md-6">
        <label for="feEmailAddress"><b>Z ref</b> - Висота на
якій визначається приземна швидкість </label>
        <input type="number" class="form-control" id="zref"
value="10">
        </div>
        </div>

        <div class="form-row">
        <div class=" col-md-12">
        <label for="feEmailAddress"><b>P</b> - Поправочний

```

```

коєфіцієнт =<b id='p-koef'>0.07</b></label> </div>
    <div class=" col-md-6">
        <label for="feEmailAddress">Місцевість</label><br>
        <input type="radio" name="mis" value="1"
checked="checked" onchange="SetKoeff()"> Сільська <br>
        <input type="radio" name="mis" value="2"
onchange="SetKoeff()"> Міська <br>
    </div>
    <div class=" col-md-6">
        <label for="feEmailAddress">Стабільність
атмосфери</label><br>
        <input type="radio" name="stab" data-m1='0.07' data-
m2='0.15' checked="checked" onchange="SetKoeff()"> A<br>
        <input type="radio" name="stab" data-m1='0.07' data-
m2='0.15' value="2" onchange="SetKoeff()"> B<br>
        <input type="radio" name="stab" data-m1='0.10' data-
m2='0.20' value="3" onchange="SetKoeff()"> C<br>
        <input type="radio" name="stab" data-m1='0.15' data-
m2='0.25' value="4" onchange="SetKoeff()"> D<br>
        <input type="radio" name="stab" data-m1='0.35' data-
m2='0.30' value="5" onchange="SetKoeff()"> E<br>
        <input type="radio" name="stab" data-m1='0.55' data-
m2='0.30' value="6" onchange="SetKoeff()"> F<br>
    </div>
</div>
</div>
    <button type="button" class="btn btn-accent"
onclick="Calc()">Провести розрахунок</button>
</form>
<div id="result">
<h3>Результат</h3>
Концентрація шкідливої речовини C = <b id="res_1">-
</b><br><br>
    Швидкість вітру Us = <b id="res_2">-</b><br><br>
</div>
</div>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript">
var p_koef = 0.07;
function SetKoeff(){
    if($('input[name=mis]:checked').val()==1){
        p_koef = $('input[name=stab]:checked').data('m1');
    }else{
        p_koef = $('input[name=stab]:checked').data('m2');
    }
    $('#p-koef').html(p_koef);
}
function Calc(){
    var q = $('#q').val()*1;
    var k = Math.pow(10, 6);

```

```

var v = $('#v').val()*1;
var y = $('#y').val()*1;
var d_y = $('#d_y').val()*1;
var d_z = $('#d_z').val()*1;
var hs = $('#hs').val()*1;
var uref = $('#uref').val()*1;
var zref = $('#zref').val()*1;

var us = uref* Math.pow((hs/zref), p_koef);

if($('#input[name=type]:checked').val()==1){
    var exp = -0.5*Math.pow(y, 2)/Math.pow(d_y, 2);
    var res_1 = (q*k*v*Math.exp(exp))/(2*3.1415*us*d_y*d_z)
}else{

    var res_1 = q/(3.1415*us*d_y*d_z)
}

$('#res_1').html(res_1.toFixed(3));
$('#res_2').html(us.toFixed(3));

return false;

//var res_1 = Math.round(((($('#f_cr').val()*1)/100)*(Math.pow(10,
6)/($('#f_qir').val()*1)),2);
//$('#res_1').html(res_1);
var res_2 = (3.1415*Math.pow((($('#d').val()*1),
2)*($('#w0').val()*1)/4);
$('#res_2').html(res_2.toFixed(3));
var k_a = $('#input[name=k_a]:checked').val()*1;
var k_f = $('#input[name=k_f]:checked').val()*1;
var masa = $('#masa').val()*1;
var h = $('#h').val()*1;
var k_m = $('#k_m').val()*1;
var k_n = $('#k_n').val()*1;
var k_nn = $('#k_nn').val()*1;
var t1 = $('#t1').val()*1;
var t2 = $('#t2').val()*1;

var res_1_1 = k_a*k_f*masa*k_m*k_n*k_nn;
console.log(res_1_1)
var res_1_2 = Math.pow(h, 2)*Math.cbrt((res_2/(t1-t2)))
console.log(res_1_2)

var res_1 = res_1_1/res_1_2;
$('#res_1').html(res_1.toFixed(3));

return false;
}

</script>

define ('SXGEO_FILE', 0);
define ('SXGEO_MEMORY', 1);
define ('SXGEO_BATCH', 2);
class SxGeo {
    protected $fh;

```

```

protected $iplc;
protected $info;
protected $range;
protected $db_begin;
protected $b_idx_str;
protected $m_idx_str;
protected $b_idx_arr;
protected $m_idx_arr;
protected $m_idx_len;
protected $db_items;
protected $country_size;
protected $db;
protected $regions_db;
protected $cities_db;

public $id2iso = array(
    '', 'AP', 'EU', 'AD', 'AE', 'AF', 'AG', 'AI', 'AL', 'AM', 'CW', 'AO', 'AQ',
'AR', 'AS', 'AT', 'AU',
    'AW', 'AZ', 'BA', 'BB', 'BD', 'BE', 'BF', 'BG', 'BH', 'BI', 'BJ', 'BM',
'BN', 'BO', 'BR', 'BS',
    'BT', 'BV', 'BW', 'BY', 'BZ', 'CA', 'CC', 'CD', 'CF', 'CG', 'CH', 'CI',
'CK', 'CL', 'CM', 'CN',
    'CO', 'CR', 'CU', 'CV', 'CX', 'CY', 'CZ', 'DE', 'DJ', 'DK', 'DM', 'DO',
'DZ', 'EC', 'EE', 'EG',
    'EH', 'ER', 'ES', 'ET', 'FI', 'FJ', 'FK', 'FM', 'FO', 'FR', 'SX', 'GA',
'GB', 'GD', 'GE', 'GF',
    'GH', 'GI', 'GL', 'GM', 'GN', 'GP', 'GQ', 'GR', 'GS', 'GT', 'GU', 'GW',
'GY', 'HK', 'HM', 'HN',
    'HR', 'HT', 'HU', 'ID', 'IE', 'IL', 'IN', 'IO', 'IQ', 'IR', 'IS', 'IT',
'JM', 'JO', 'JP', 'KE',
    'KG', 'KH', 'KI', 'KM', 'KN', 'KP', 'KR', 'KW', 'KY', 'KZ', 'LA', 'LB',
'LC', 'LI', 'LK', 'LR',
    'LS', 'LT', 'LU', 'LV', 'LY', 'MA', 'MC', 'MD', 'MG', 'MH', 'MK', 'ML',
'MM', 'MN', 'MO', 'MP',
    'MQ', 'MR', 'MS', 'MT', 'MU', 'MV', 'MW', 'MX', 'MY', 'MZ', 'NA', 'NC',
'NE', 'NF', 'NG', 'NI',
    'NL', 'NO', 'NP', 'NR', 'NU', 'NZ', 'OM', 'PA', 'PE', 'PF', 'PG', 'PH',
'PK', 'PL', 'PM', 'PN',
    'PR', 'PS', 'PT', 'PW', 'PY', 'QA', 'RE', 'RO', 'RU', 'RW', 'SA', 'SB',
'SC', 'SD', 'SE', 'SG',
    'SH', 'SI', 'SJ', 'SK', 'SL', 'SM', 'SN', 'SO', 'SR', 'ST', 'SV', 'SY',
'SZ', 'TC', 'TD', 'TF',
    'TG', 'TH', 'TJ', 'TK', 'TM', 'TN', 'TO', 'TL', 'TR', 'TT', 'TV', 'TW',
'TZ', 'UA', 'UG', 'UM',
    'US', 'UY', 'UZ', 'VA', 'VC', 'VE', 'VG', 'VI', 'VN', 'VU', 'WF', 'WS',
'YE', 'YT', 'RS', 'ZA',
    'ZM', 'ME', 'ZW', 'A1', 'XK', 'O1', 'AX', 'GG', 'IM', 'JE', 'BL', 'MF',
'BQ', 'SS'
);

public $batch_mode = false;
public $memory_mode = false;

public function __construct($db_file = 'SxGeo.dat', $type = SXGEO_FILE){
    $this->fh = fopen($db_file, 'rb');
    // Сначала убеждаемся, что есть файл базы данных
    $header = fread($this->fh, 40); // В версии 2.2 заголовок увеличился на 8
байт
    if(substr($header, 0, 3) != 'SxG') die("Can't open {$db_file}\n");
    $info
    =
    unpack('Cver/Ntime/Ctype/Ccharset/Cb_idx_len/nm_idx_len/nrange/Ndb_items/Cid_len/nmax

```

```

_region/nmax_city/Nregion_size/Ncity_size/nmax_country/Ncountry_size/npack_size',
substr($header, 3));
    if($info['b_idx_len'] * $info['m_idx_len'] * $info['range'] *
$info['db_items'] * $info['time'] * $info['id_len'] == 0) die("Wrong file format
{$db_file}\n");
    $this->range = $info['range'];
    $this->b_idx_len = $info['b_idx_len'];
    $this->m_idx_len = $info['m_idx_len'];
    $this->db_items = $info['db_items'];
    $this->id_len = $info['id_len'];
    $this->block_len = 3 + $this->id_len;
    $this->max_region = $info['max_region'];
    $this->max_city = $info['max_city'];
    $this->max_country = $info['max_country'];
    $this->country_size = $info['country_size'];
    $this->batch_mode = $type & SXGEO_BATCH;
    $this->memory_mode = $type & SXGEO_MEMORY;
    $this->pack = $info['pack_size'] ? explode("\0", fread($this->fh,
$info['pack_size'])) : '';
    $this->b_idx_str = fread($this->fh, $info['b_idx_len'] * 4);
    $this->m_idx_str = fread($this->fh, $info['m_idx_len'] * 4);

    $this->db_begin = ftell($this->fh);
    if ($this->batch_mode) {
        $this->b_idx_arr = array_values(unpack("N*", $this->b_idx_str)); //
Быстрее в 5 раз, чем с циклом
        unset ($this->b_idx_str);
        $this->m_idx_arr = str_split($this->m_idx_str, 4); // Быстрее в 5
раз чем с циклом
        unset ($this->m_idx_str);
    }
    if ($this->memory_mode) {
        $this->db = fread($this->fh, $this->db_items * $this->block_len);
        $this->regions_db = $info['region_size'] > 0 ? fread($this->fh,
$info['region_size']) : '';
        $this->cities_db = $info['city_size'] > 0 ? fread($this->fh,
$info['city_size']) : '';
    }
    $this->info = $info;
    $this->info['regions_begin'] = $this->db_begin + $this->db_items * $this-
>block_len;
    $this->info['cities_begin'] = $this->info['regions_begin'] +
$info['region_size'];
}

protected function search_idx($ipn, $min, $max){
    if($this->batch_mode){
        while($max - $min > 8){
            $offset = ($min + $max) >> 1;
            if ($ipn > $this->m_idx_arr[$offset]) $min = $offset;
            else $max = $offset;
        }
        while ($ipn > $this->m_idx_arr[$min] && $min++ < $max){};
    }
    else {
        while($max - $min > 8){
            $offset = ($min + $max) >> 1;
            if ($ipn > substr($this->m_idx_str, $offset*4, 4)) $min =
$offset;
            else $max = $offset;
        }
    }
}

```

```

        while ($ipn > substr($this->m_idx_str, $min*4, 4) && $min++ <
$max){};
    }
    return $min;
}

protected function search_db($str, $ipn, $min, $max){
    if($max - $min > 1) {
        $ipn = substr($ipn, 1);
        while($max - $min > 8){
            $offset = ($min + $max) >> 1;
            if ($ipn > substr($str, $offset * $this->block_len, 3)) $min =
$offset;
            else $max = $offset;
        }
        while ($ipn >= substr($str, $min * $this->block_len, 3) && ++$min <
$max){};
    }
    else {
        $min++;
    }
    return hexdec(bin2hex(substr($str, $min * $this->block_len - $this-
>id_len, $this->id_len)));
}

public function get_num($ip){
    $ipln = (int)$ip; // Первый байт
    if($ipln == 0 || $ipln == 10 || $ipln == 127 || $ipln >= $this->b_idx_len
|| false === ($ipn = ip2long($ip))) return false;
    $ipn = pack('N', $ipln);
    $this->iplc = chr($ipln);
    // Находим блок данных в индексе первых байт
    if ($this->batch_mode){
        $blocks = array('min' => $this->b_idx_arr[$ipln-1], 'max' => $this-
>b_idx_arr[$ipln]);
    }
    else {
        $blocks = unpack("Nmin/Nmax", substr($this->b_idx_str, ($ipln - 1) *
4, 8));
    }
    if ($blocks['max'] - $blocks['min'] > $this->range){
        // Ищем блок в основном индексе
        $part = $this->search_idx($ipn, floor($blocks['min'] / $this-
>range), floor($blocks['max'] / $this->range)-1);
        // Нашли номер блока в котором нужно искать IP, теперь находим нужный
блок в БД
        $min = $part > 0 ? $part * $this->range : 0;
        $max = $part > $this->m_idx_len ? $this->db_items : ($part+1) *
$this->range;
        // Нужно проверить чтобы блок не выходил за пределы блока первого
байта
        if($min < $blocks['min']) $min = $blocks['min'];
        if($max > $blocks['max']) $max = $blocks['max'];
    }
    else {
        $min = $blocks['min'];
        $max = $blocks['max'];
    }
    $len = $max - $min;
    // Находим нужный диапазон в БД
    if ($this->memory_mode) {

```

```

        return $this->search_db($this->db, $ipn, $min, $max);
    }
    else {
        fseek($this->fh, $this->db_begin + $min * $this->block_len);
        return $this->search_db(fread($this->fh, $len * $this->block_len),
$ipn, 0, $len);
    }
}

protected function readData($seek, $max, $type){
    $raw = '';
    if($seek && $max) {
        if ($this->memory_mode) {
            $raw = substr($type == 1 ? $this->regions_db : $this-
>cities_db, $seek, $max);
        } else {
            fseek($this->fh, $this->info[$type == 1 ? 'regions_begin' :
'cities_begin'] + $seek);
            $raw = fread($this->fh, $max);
        }
    }
    return $this->unpack($this->pack[$type], $raw);
}

protected function parseCity($seek, $full = false){
    if(!$this->pack) return false;
    $only_country = false;
    if($seek < $this->country_size){
        $country = $this->readData($seek, $this->max_country, 0);
        $city = $this->unpack($this->pack[2]);
        $city['lat'] = $country['lat'];
        $city['lon'] = $country['lon'];
        $only_country = true;
    }
    else {
        $city = $this->readData($seek, $this->max_city, 2);
        $country = array('id' => $city['country_id'], 'iso' => $this-
>id2iso[$city['country_id']]);
        unset($city['country_id']);
    }
    if($full) {
        $region = $this->readData($city['region_seek'], $this->max_region,
1);
        if(!$only_country)
            $country = $this-
>readData($region['country_seek'], $this->max_country, 0);
        unset($city['region_seek']);
        unset($region['country_seek']);
        return array('city' => $city, 'region' => $region, 'country' =>
$country);
    }
    else {
        unset($city['region_seek']);
        return array('city' => $city, 'country' => array('id' =>
$country['id'], 'iso' => $country['iso']));
    }
}

protected function unpack($pack, $item = ''){
    $unpacked = array();
    $empty = empty($item);
    $pack = explode('/', $pack);

```

```

$pos = 0;
foreach($pack AS $p){
    list($type, $name) = explode(':', $p);
    $type0 = $type[0];
    if($empty) {
        $unpacked[$name] = $type0 == 'b' || $type0 == 'c' ? '' : 0;
        continue;
    }
    switch($type0){
        case 't':
        case 'T': $l = 1; break;
        case 's':
        case 'n':
        case 'S': $l = 2; break;
        case 'm':
        case 'M': $l = 3; break;
        case 'd': $l = 8; break;
        case 'c': $l = (int)substr($type, 1); break;
        case 'b': $l = strpos($item, "\0", $pos)-$pos; break;
        default: $l = 4;
    }
    $val = substr($item, $pos, $l);
    switch($type0){
        case 't': $v = unpack('c', $val); break;
        case 'T': $v = unpack('C', $val); break;
        case 's': $v = unpack('s', $val); break;
        case 'S': $v = unpack('S', $val); break;
        case 'm': $v = unpack('l', $val . (ord($val[2]) >> 7 ? "\xff"
: "\0")); break;

        case 'M': $v = unpack('L', $val . "\0"); break;
        case 'i': $v = unpack('l', $val); break;
        case 'I': $v = unpack('L', $val); break;
        case 'f': $v = unpack('f', $val); break;
        case 'd': $v = unpack('d', $val); break;

        case 'n': $v = current(unpack('s', $val)) / pow(10, $type[1]);
break;

        case 'N': $v = current(unpack('l', $val)) / pow(10, $type[1]);
break;

        case 'c': $v = rtrim($val, ' '); break;
        case 'b': $v = $val; $l++; break;
    }
    $pos += $l;
    $unpacked[$name] = is_array($v) ? current($v) : $v;
}
return $unpacked;
}

public function get($ip){
    return $this->max_city ? $this->getCity($ip) : $this->getCountry($ip);
}
public function getCountry($ip){
    if($this->max_city) {
        $tmp = $this->parseCity($this->get_num($ip));
        return $tmp['country']['iso'];
    }
    else return $this->id2iso[$this->get_num($ip)];
}
public function getCountryId($ip){
    if($this->max_city) {

```

```

        $tmp = $this->parseCity($this->get_num($ip));
        return $tmp['country']['id'];
    }
    else return $this->get_num($ip);
}
public function getCity($ip){
    $seek = $this->get_num($ip);
    return $seek ? $this->parseCity($seek) : false;
}
public function getCityFull($ip){
    $seek = $this->get_num($ip);
    return $seek ? $this->parseCity($seek, 1) : false;
}
public function about(){
    $charset = array('utf-8', 'latin1', 'cp1251');
    $types = array('n/a', 'SxGeo Country', 'SxGeo City RU', 'SxGeo City EN',
'SxGeo City', 'SxGeo City Max RU', 'SxGeo City Max EN', 'SxGeo City Max');
    return array(
        'Created' => date('Y.m.d', $this->info['time']),
        'Timestamp' => $this->info['time'],
        'Charset' => $charset[$this->info['charset']],
        'Type' => $types[$this->info['type']],
        'Byte Index' => $this->b_idx_len,
        'Main Index' => $this->m_idx_len,
        'Blocks In Index Item' => $this->range,
        'IP Blocks' => $this->db_items,
        'Block Size' => $this->block_len,
        'City' => array(
            'Max Length' => $this->max_city,
            'Total Size' => $this->info['city_size'],
        ),
        'Region' => array(
            'Max Length' => $this->max_region,
            'Total Size' => $this->info['region_size'],
        ),
        'Country' => array(
            'Max Length' => $this->max_country,
            'Total Size' => $this->info['country_size'],
        ),
    );
}
}

```

Додаток 03

Пошукова GIS система екологічно небезпечних промислових об'єктів

Апробація

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ТМ-71165

Аркушів 4

2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали ХІХ Міжнародної
науково-практичної конференції
молодих вчених і студентів
м. Київ, 20–23 квітня 2021 року

ТОМ 2



Київ- 2021

УДК 004.91

Студент 4 курсу, гр. ТМ-71 Ласкавий О.О.
Ст.викл., к.ф.-м.н. Бандурка О.І.

ПОШУКОВА GIS СИСТЕМА ЕКОЛОГІЧНО НЕБЕЗПЕЧНИХ ПРОМИСЛОВИХ ОБ'ЄКТІВ

Промисловість завжди мала не абиякий вплив на економічний розвиток різних країн світу, і Україна, у цьому плані, не виняток. Промисловий сектор відіграє чи не найважливішу роль в економіці нашої держави. Розвиток цієї галузі сприяє появі нових інвестицій, дозволяє збільшити кількість експорту, а також сприяє збільшенню кількості робочих місць, що, у свій час, дуже позитивно впливає на загальний економічний стан України.

Для більш наглядного представлення про вплив промисловості на економіку нашої держави, приведемо наступні дані. Станом на 2017 рік експорт промислової продукції досягав близько 32 мільярдів доларів США, що складає 73% з усього товарного експорту України. З усього зайнятого населення, у промисловості задіяно 15%. [1]

Окрім цього, через старе оснащення та відсутність належних засобів для фільтрації відходів, підприємства важкої промисловості щодня викидають в атмосферу шкідливі речовини, як наприклад CO₂. (фор.2)

У результаті викидів жителі таких промислових міст як Дніпро, Кам'янське, Кривий Ріг, Запоріжжя та Маріуполь щодня потерпають від надзвичайно забрудненого повітря [3].

Варто також зауважити, що столиця нашої країни, місто Київ, вже не раз потрапляла у десятку міст світу з найбільш забрудненим повітрям, через велику кількість екологічно небезпечних підприємств та високу кількість транспортних засобів. Наразі ситуація покращилась і Київ займає 58 сходинку у рейтингу найбільш забруднених міст згідно з даними сайту IQair [4].

Для того, аби покращити екологічну ситуації в Україні, вкрай необхідно проводити регулярний аналіз промислових найбільш екологічно небезпечних

об'єктів у реальному часі. Таким чином можна дізнатись, який з промислових об'єктів спричиняє найбільшу загрозу для навколишнього середовища та модернізувати їх, у першу чергу, за допомогою встановлення різноманітних інноваційних фільтрів.

В Пошуковій GIS системі використовувалися різноманітні методики для визначення викидів в атмосферу, такі як: «Методика визначення валових викидів забруднюючих речовин в атмосферу від котлових установок ТЕС» (за даними вимірів їх концентрацій в димових газах та розрахунковим методом), «Методика розрахунку викидів шкідливих речовин в атмосферу підприємств побутового обслуговування» (за витратами використовуваних матеріалів), «Методика розрахунку викидів шкідливих речовин при спалюванні газу».

Одним з основних джерел викиду забруднюючих речовин в атмосферу є теплосилові установки. Валовий викид j -ї забруднюючої речовини E_j , т, що надходить у атмосферу з димовими газами теплосилової установки за проміжок часу P , визначається як сума валових викидів цієї речовини під час спалювання різних видів палива, у тому числі під час їх одночасного спільного спалювання(1):

$$E_j = \sum_i E_{ji} = 10^{-6} \sum_i k_{ji} B_i (Q_i^y)_i \quad (1)$$

де E_{ji} - валовий викид j -ї забруднюючої речовини під час спалювання i -го палива за проміжок часу P , т;

k_{ji} - показник емісії j -ї забруднюючої речовини для i -го палива, г/ГДж;

B_i - витрата i -го палива за проміжок часу P , т;

$(Q_i^y)_i$ - нижча робоча теплота згоряння i -го палива, МДж/кг.

Найбільшої шкоди нашому повітрю завдає діоксид вуглецю (вуглекислий газ CO₂) відноситься до парникових газів і є основним газоподібним продуктом окислення вуглецю органічного палива. Обсяг викиду CO₂ безпосередньо пов'язано із вмістом вуглецю в паливі та ступенем окислення вуглецю палива в установці спалювання. Показник емісії діоксиду вуглецю, k_{CO_2} , г/ГДж, під час спалювання органічного палива визначається за формулою:

$$k_{CO_2} = \frac{44}{12} \times \frac{C^y}{100} \times \frac{10^6}{Q_i^y} \varepsilon_C = 3,67 k_C \varepsilon_C \quad (2)$$

де C^y - масовий вміст вуглецю в паливі на робочу масу, %;

Q_i^y - нижча робоча теплота згоряння палива, МДж/кг;

ε_C - ступінь окислення вуглецю палива;

k_C - показник емісії вуглецю палива, г/ГДж.

Пошукова GIS система екологічно небезпечних промислових об'єктів як раз і допоможе спостерігати та аналізувати вплив різних підприємств на навколишнє середовище. Зрозуміло що керівництво країни не приділяє так багато уваги, як потрібно до таких екологічно небезпечних об'єктів. Пошукова GIS система – це чудовий інструмент для місцевих груп самоврядування, які турбуються про свої міста. Люди зможуть детально прослідкувати що забруднюють їх місто, як змінюється кількість викидів з кожним роком. І я сподіваюсь, зможуть повпливати на екологічний стан свого міста.

Перелік посилань:

1. Розвиток промисловості для забезпечення зростання та оновлення української економіки [Електронний ресурс] // НАН України. – 2018. – Режим доступу до ресурсу: <http://ief.org.ua/docs/sr/301.pdf>.
2. Перелік об'єктів, що можуть спричинити виникнення надзвичайної ситуації техногенного і природного характеру та вплинути на стан захисту населення і територій, проекти будівництва яких підлягають державній експертизі з питань техногенної безпеки [Електронний ресурс] // Кабінет Міністрів України – Режим доступу до ресурсу: <https://www.kmu.gov.ua/npas/155861368>.
3. Експерти визначили, де в Україні найчистіше і найбрудніше повітря [Електронний ресурс] – Режим доступу до ресурсу: <https://www.radiosvoboda.org/a/news-mista-zabrudnennya-povitrya/29437963.html>.
4. Air quality and pollution city ranking [Електронний ресурс] – Режим доступу до ресурсу: <https://www.iqair.com/world-air-quality-ranking>.

Додаток 04

Пошукова GIS система екологічно небезпечних промислових об'єктів

Опис програмного модулю

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ТМ-71165

Аркушів 8

2021

АНОТАЦІЯ

Додаток містить опис пошукової GIS системи для визначення екологічного стану промислової зони.

У додатку описані такі функції, як:

- 1) Введення даних
- 2) Розрахунок викидів
- 3) Виведення результату

Даний програмний продукт розроблений з використанням фреймворків Bootstrap та CodeIgniter в середовищі розробки NuSpherePhpED у вигляді веб-ресурсу для забезпечення доступу з будь-якого ресурсу.

ЗМІСТ

1. Загальні відомості.....	76
2. Функціональне призначення	77
3. Опис логічної структури.....	78
4. Використовувані технічні засоби.....	79
5. Завантаження.....	80
6. Вхідні та вихідні дані.....	81

1. ЗАГАЛЬНІ ВІДОМОСТІ

У четвертому додатку описано роботу пошукової GIS-системи для визначення екологічного стану промислової зони.

Розроблювана програма може працювати на будь-якому пристрої, не залежно від операційної системи.

Даний програмний продукт розроблений з використанням фреймворків Bootstrap та CodeIgniter в середовищі розробки NuSpherePhpED у вигляді веб-ресурсу для забезпечення доступу з будь-якого ресурсу

2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Розроблена автоматизована система дозволяє визначати кількість шкідливих викидів від підприємств, оцінювати рівень забруднення води, повітря та ґрунтових вод.

Даний програмний продукт може використовуватись у вищих навчальних при проведенні різного роду стартап конкурсів, а також на підприємствах. Рекомендовано до використання викладачам університетів або запрошеним експертам з різних сфер.

3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ

При першому візиті на сайт, гео модуль визначить ваше місцезнаходження та перенаправить на необхідний міський піддомен. Для користування адмін частиною, необхідно здійснити вхід, ввівши коректні дані.

На сайті представлені модулі розрахунків забруднення повітря, води, та ґрунтових вод. Відображена карта онлайн моніторингу стану повітря.

4. ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для додатку був використаний виділений сервер від хостингу Beget. Web-додаток написаний на фреймворкі CodeIgniter. Шаблон сайту та адмін частини розроблювався на Bootstrap. Для гео-модулю використовувалась відкрита база Sурехgeo. База даних – mySql, а середовище для створення БД – phpMyAdmin. Для створення піддоменів використовувалися директиви DNS хостингу. Для збереження безпеки з'єднання з системою використовується SSL сертифікат Let's Encrypt Wildcard.

Даний програмний продукт розроблений з використанням фреймворків Bootstrap та CodeIgniter в середовищі розробки NuSpherePhpED у вигляді веб-ресурсу для забезпечення доступу з будь-якого ресурсу.

5. ЗАВАНТАЖЕННЯ

Для того аби потрапити на даний ресурс, необхідно перейти за посиланням:
<https://gis-ecology.site/>

Адмін частина сайту знаходиться за посиланням: <https://gis-ecology.site/login>

У результаті запуску користувач отримає доступ до ресурсу, редагуванню даних, та використанню функціоналу.

6. ЗАВАНТАЖЕННЯ

Вхідними даними є дані з щомісячних та щоквартальних звітів підприємств по кількості викидів в навколишнє середовище. Для роботи з калькуляторами – дані вводить користувач, покладаючись на звіти.

Онлайн карта моніторингу працює з даними про індекс якості повітря, які отримує з датчиків встановлених у містах.

У результаті отримуються дані про рівень забруднення, екологічний стан міста, онлайн карта з найбільш забрудненими районами міста.