

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка вебсайту для моніторингу екосистем парків та скверів за допомогою IoT

Виконав: студент IV курсу, групи СН-42
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

Куцинда Л.Ю.
(підпис) (прізвище та ініціали)

Керівник
(підпис) Небесний Р.М.
(прізвище та ініціали)

Нормоконтроль
(підпис) Липак Г. І.
(прізвище та ініціали)

Завідувач кафедри
(підпис) Боднарчук І.О.
(прізвище та ініціали)

Рецензент
(підпис) Загородна Н.В.
(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 27 » січня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Куцинді Любомиру Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка вебсайту для моніторингу екосистем парків та скверів за допомогою IoT.

Керівник роботи Небесний Руслан Михайлович, доктор філософії, доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 07 » травня 2025 року № 4/7-444

2. Термін подання студентом завершеної роботи 23 червня 2025 р.

3. Вихідні дані до роботи Технічне завдання, цифрові та літературні джерела

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

Вступ. 1 Аналіз предметної області та постановка завдання. 2 Проектування системи на основі інтернету речей. 3 Розробка та реалізація сайту з функціоналом для моніторингу екосистем.

4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел. Додатки.

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

Кваліфікаційна робота на тему: Розробка вебсайту для моніторингу екосистем парків та скверів за допомогою IoT, Актуальність дослідження, Мета і завдання роботи, Концепція «Розумного міста» та IoT у моніторингу екосистем, Методологія та архітектура проектування, Архітектура Системи: Від Датчика до Вебсайту, Ключові компоненти та структура даних, Дизайн та Функціональні особливості, Практична реалізація: Апаратна частина, Практична реалізація: Серверна частина та База даних, Візуалізація даних: Інтерактивна карта Leaflet, Динамічна сторінка «Детальніше», Сторінка «Контакти», Демонстрація ключових функціональних можливостей, Результати: Ефективність та Надійність, Можливості Розвитку, Висновки, Дякую за увагу.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин Віктор Степанович, кандидат техн. Наук кафедри МТ	27.01.2025	20.06.2025

7. Дата видачі завдання 27 січня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

Куцинда Л.Ю.

(прізвище та ініціали)

Керівник роботи

(підпис)

Небесный Р.М.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка вебсайту для моніторингу екосистем парків та скверів за допомогою Іот
// Кваліфікаційна робота освітнього рівня «Бакалавр» // Куцинда Любомир
Юрійович // Тернопільський національний технічний університет імені Івана
Пулня, факультет комп'ютерно-інформаційних систем і програмної інженерії,
кафедра комп'ютерних наук, група СН-42 // Тернопіль, 2025 // С.66, рис. – 20,
табл. – 1, додат. – 7, бібліогр. – 61.

Ключові слова: вебсайт, ІоТ, сенсор, парки та сквери, екологічний
моніторинг, Arduino, Javascript, C++.

Кваліфікаційна робота присвячена дослідженню та розробці вебсайту для
моніторингу екосистем парків та скверів за допомогою технологій інтернет
речей.

В першому розділі кваліфікаційної роботи розглянуто ключові аспекти і
значення «Розумного міста» та моніторингу екосистем зелених зон, зокрема
парків та скверів. Висвітлено роль та потенціал технологій Інтернету речей у
зборі екологічних даних повітряного середовища. Проаналізовано існуючі
підходи та платформи для моніторингу навколишнього середовища, виявлено
їхні недоліки та функціональні прогалини, що обґрунтовує необхідність
створення нової системи.

В другому розділі кваліфікаційної роботи досліджено актуальність та
теоретичні основи проєктування системи моніторингу, включаючи сучасний
стан екологічного моніторингу міських зелених зон, концепцію Інтернету речей
та її застосування, а також переваги та виклики впровадження ІоТ-систем для
моніторингу парків та скверів. Подано методологію проєктування ІоТ-системи
моніторингу екосистем, яка охоплює загальні підходи до проєктування, детальну
архітектуру системи, обґрунтування вибору ключових апаратних компонентів, а
також проєктування структури бази даних MySQL. Визначено функціональні
вимоги до вебсайту, принципи дизайну та інтерфейсу користувача, а також

детально описано проєктування відображення даних на інтерактивній карті Leaflet та механізмів оновлення та взаємодії з даними. Розглянуто підходи до проєктування системи з урахуванням майбутнього тестування та оцінки продуктивності, включаючи забезпечення точності та надійності даних на етапі проєктування.

В третьому розділі кваліфікаційної роботи описано практичну реалізацію вебсайту для моніторингу екосистем парків та скверів за допомогою технологій Інтернету речей. Детально висвітлено реалізацію апаратної частини, що включає пряме підключення датчика температури та вологості, а також розробку програмного забезпечення для ESP01S. Подано налаштування серверного середовища на хостингу Awardspace, створення бази даних MySQL через phpMyAdmin та реалізацію серверних скриптів для прийому та зберігання даних. Описано реалізацію клієнтської частини вебсайту, включаючи структуру головної сторінки, візуалізацію даних на інтерактивній карті Leaflet, а також механізми динамічного оновлення даних та взаємодії з користувачем, зокрема переходи на інші сторінки.

Об'єкт дослідження: розроблений вебсайт для моніторингу екосистем парків та скверів та його компоненти.

Предмет дослідження: процеси реалізації, функціонування та оцінки ефективності системи моніторингу екосистем на основі IoT.

ANNOTATION

Development of a Website for Monitoring Park and Square Ecosystems using IoT // Qualification work of the educational level «Bachelor» // Kutsynda Liubomyr Yuriyovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-42 // Ternopil, 2025 // P.66, fig. – 20, tabl. – 1, annexes. – 7, references – 61.

Keywords: website, IoT, sensor, parks and squares, ecological monitoring, Arduino, Javascript, C++.

This qualification work is dedicated to the research and development of a website for monitoring park and square ecosystems using Internet of Things technologies.

In the first chapter of this qualification endeavor key aspects and the significance of the «Smart City» concept and ecosystem monitoring of green areas, particularly parks and squares, are examined. The role and potential of Internet of Things technologies in collecting environmental air data are highlighted. Existing approaches and platforms for environmental monitoring are analyzed, revealing their shortcomings and functional gaps, which substantiates the necessity of creating a new system.

In the second chapter of this qualification work the relevance and theoretical foundations of designing a monitoring system are investigated, including the current state of environmental monitoring in urban green areas, the concept of the Internet of Things and its application, as well as the advantages and challenges of implementing IoT systems for monitoring parks and squares. The methodology for designing an IoT ecosystem monitoring system is presented, covering general design approaches, a detailed system architecture, the justification for selecting key hardware components, and the design of the MySQL database structure. Functional requirements for the website, design principles, and user interface are defined, and the design of data

visualization on an interactive Leaflet map and mechanisms for data updates and interaction are described in detail. Approaches to system design, considering future testing and performance evaluation, including ensuring data accuracy and reliability at the design stage, are examined.

In the third chapter of this qualification work the practical implementation of the website for monitoring park and square ecosystems using Internet of Things technologies is described. The implementation of the hardware part, including the direct connection of the temperature and humidity sensor, and the development of software for the ESP01S module, are detailed. The setup of the server environment on Awardspace hosting, the creation of the MySQL database via phpMyAdmin, and the implementation of server scripts for data reception and storage are presented. – The implementation of the client-side of the website is described, including the structure of the main page, data visualization on an interactive Leaflet map, and mechanisms for dynamic data updates and user interaction, including transitions to other pages.

Object of research is the developed website for monitoring park and square ecosystems and its components.

Subject of research is the processes of implementation, functioning, and evaluation of the effectiveness of the IoT-based ecosystem monitoring system.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DHT (Digital Humidity and Temperature) – Сенсор для вимірювання температури та вологості.

ISO (англ. International Organization for Standardization) – Міжнародна організація зі стандартизації.

IoT (англ. Internet of Things) – Інтернет речей.

ІКТ – Інформаційно-комунікаційні технології.

KPI (англ. Key Performance Indicator) – Ключові показники ефективності.

МН – Машинне навчання.

ШІ – Штучний інтелект.

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.	12
1.1 Концепція «Розумного міста».....	12
1.1.1 Визначення та основні поняття	12
1.1.2 Технологічні основи «Розумних міст»	14
1.1.3 Міжнародні стандарти та рамки для розвитку розумного міста..	15
1.2 Обмеження традиційних методів моніторингу екосистем	16
1.2.1 Огляд традиційних підходів до моніторингу	16
1.3 Інтернет речей для моніторингу екосистем.....	17
1.3.1 Архітектура системи IoT та технології, що її забезпечують	18
1.3.2 Успішні впровадження IoT у моніторингу довкілля «Розумних міст»	21
1.4 Висновок до першого розділу.....	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБСАЙТУ ДЛЯ МОНІТОРИНГУ ЕКОСИСТЕМ ПАРКІВ ТА СКВЕРІВ.....	23
2.1 Актуальність та теоретичні основи проєктування системи моніторингу	23
2.1.1 Застосування IoT в екологічному моніторингу	24
2.1.2 Переваги та виклики впровадження IoT-систем для моніторингу парків та скверів	24
2.2 Методологія проєктування IoT систем	26
2.2.1 Архітектура IoT системи моніторингу парків та скверів.....	27
2.3 Вибір та обґрунтування апаратних компонентів системи	28
2.4 Проєктування бази даних	28
2.5 Проєктування вебсайту для моніторингу парків та скверів	30
2.5.1 Дизайн та інтерфейс користувача	31
2.5.2 Проєктування відображення даних на карті Leaflet.....	32

2.5.3	Проектування механізмів оновлення та взаємодії з даними	33
2.6	Оформлення висновків для розділу 2	35
РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА. РЕАЛІЗАЦІЯ ВЕБСАЙТУ ДЛЯ		
	МОНІТОРИНГУ ЕКОСИСТЕМ ПАРКІВ ТА СКВЕРІВ.....	37
3.1	Реалізація апаратної частини системи моніторингу.....	37
3.1.1	Підключення датчика DHT11 до модуля ESP01S	38
3.1.2	Програмне забезпечення модуля ESP01S.....	39
3.2	Реалізація програмної частини та бази даних	41
3.2.1	Налаштування серверного середовища та бази даних.....	41
3.2.2	Реалізація серверного скрипту для прийому даних	42
3.3	Реалізація клієнтської частини вебсайту	44
3.3.1	Структура вебсайту та головна сторінка.....	44
3.3.2	Візуалізація даних на карті Leaflet.....	45
3.3.3	Взаємодія з користувачем.....	46
3.4	Оформлення висновків до розділу 3	49
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ		
4.1	Природне середовище та його забруднення.....	51
4.2	Можливість виникнення статичної електрики та заходи боротьби з нею.	53
4.3	Висновок до четвертого розділу.....	56
ВИСНОВКИ		58
ПЕРЕЛІК ДЖЕРЕЛ.....		60
ДОДАТКИ		

ВСТУП

Актуальність теми. Внаслідок глобальних екологічних викликів, урбанізації та зростаючої потреби у створенні комфортного та безпечного міського середовища, моніторинг стану екосистем парків та скверів набуває особливої актуальності. Традиційні методи збору екологічних даних часто є трудомісткими, дорогими та не забезпечують достатньої оперативності та повноти інформації. Розвиток технологій Інтернету речей (IoT) відкриває нові можливості для створення вискоєфективних та доступних систем автоматизованого моніторингу довкілля. Сучасні IoT-пристрої дозволяють здійснювати збір даних про температуру, вологість та інші параметри в режимі реального часу, передаючи їх на сервер для подальшої обробки та візуалізації. Таким чином, розробка та впровадження інтегрованих IoT-систем для екологічного моніторингу міських зелених зон, що включають зручні вебплатформи для візуалізації даних, є актуальним напрямком сучасних досліджень у галузі комп'ютерних наук, природокористування та «розумного міста».

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є підвищення якості управління екосистемами міських зелених зон та підвищення інформованості громадськості шляхом створення сучасної IoT-системи моніторингу та розробки функціонального вебсайту для візуалізації екологічних даних (температури та вологості) у парках та скверах.

Для досягнення поставленої мети потрібно виконати ряд завдань:

- Проаналізувати сучасний стан досліджень у сфері моніторингу екосистем міських зелених зон та роль технологій Інтернету речей у цьому процесі.
- Дослідити та обґрунтувати архітектуру IoT-системи моніторингу, а також спроектувати її ключові апаратні компоненти (датчики та мікроконтролери) та структуру бази даних.

- Розробити функціональні вимоги, принципи дизайну та інтерфейсу користувача для вебсайту моніторингу, включаючи проєктування відображення даних на інтерактивній карті.
- Реалізувати апаратну частину системи, програмне забезпечення мікроконтролера, а також серверну та клієнтську частини вебсайту, включаючи механізми прийому, зберігання та динамічного відображення даних.
- Провести демонстрацію функціональності розробленої системи та оцінити її ефективність і надійність.
- Визначити перспективи подальшого розвитку та масштабування системи.

Практичне значення одержаних результатів. Результати даного дослідження мають значне практичне значення для підвищення ефективності екологічного моніторингу у містах. Розроблений вебсайт та IoT-система дозволяють отримувати актуальні дані про мікроклімат зелених зон, що є критично важливим для прийняття обґрунтованих рішень органами місцевого самоврядування, комунальними службами та екологічними організаціями щодо управління міським середовищем. Система забезпечує прозорість екологічних даних для громадськості, підвищуючи її обізнаність та залученість до питань довкілля. Крім того, реалізація проєкту демонструє можливість створення сучасних, адаптивних та масштабованих рішень для екологічного моніторингу з використанням доступних технологій, що може слугувати основою або шаблоном для впровадження аналогічних систем у різних населених пунктах або на інших природних об'єктах.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.

1.1 Концепція «Розумного міста»

Розумне місто - це новітня модель міського розвитку, що базується на застосуванні інноваційних технологій та інформації для покращення добробуту мешканців, оптимізації роботи та послуг міста [1], а також підвищення його конкурентоспроможності. По суті, «Розумне місто» - це складна соціально-екологічна система, в якій сенсорні мережі та високорівнева аналітика даних відстежують і керують міськими операціями для підвищення стійкості, ефективності та життєстійкості міста [2].

1.1.1 Визначення та основні поняття

Розумне міське середовище використовує інформаційно-комунікаційні технології та інші інструменти для покращення якості життя, якості та ефективності міських послуг, а також задоволення економічних, соціальних та екологічних потреб нинішнього та майбутніх поколінь [3]. Інтеграція «цифрової нервової системи» в міське середовище є основою цієї ідеї, яка сприяє більшій гармонії між діяльністю людини та здоров'ям навколишнього середовища [4]. Застосування технології IoT на прикладі «Розумного міста» показано на рисунку 1.1.

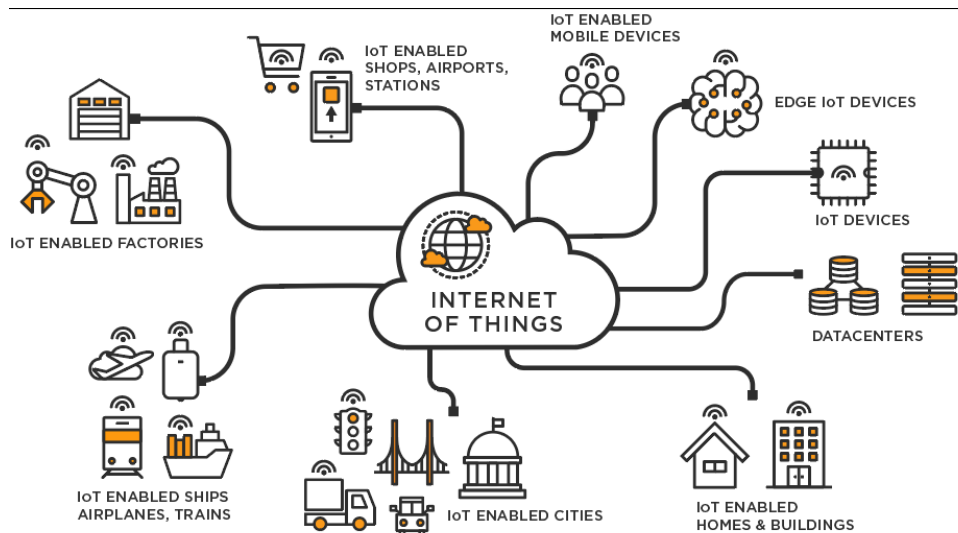


Рисунок 1.1 Схема використання інтернету речей у розумному місті

Основними стовпами, на яких побудована концепція Розумного міста, є ефективність, зв'язок, сталий розвиток та якість життя. Ефективність полягає в найкращому використанні ресурсів та надання послуг, що досягається завдяки автоматизації та оптимізації процесів, таких як інтелектуальне вуличне освітлення, розумна мережа та оптимізований збір відходів [5]. Це досягається для зменшення споживання енергії та води, а також для покращення якості та швидкості міських послуг. Зв'язок дозволяє ефективно обмінюватися даними та комунікувати між різними компонентами та системами міста. ІКТ – це стратегічна інфраструктура, яка об'єднує всі компоненти розумного міста та робить товари та послуги більш ефективними [6]. Це ключовий принцип у руйнуванні старих бар'єрів між відділами та забезпеченні скоординованої роботи міської інфраструктури. Сталий розвиток гарантує довгострокове здоров'я навколишнього середовища, економічну та соціальну рівність та сталий розвиток. Моніторинг навколишнього середовища є ключовим компонентом досягнення сталого та придатного для життя міського середовища [7]. Він найтісніше пов'язаний з інтегрованим управлінням містом, як частина переходу до управління на основі даних, де вибір все більше базується на потоках даних у режимі реального часу, що відображають екологічний, соціальний та економічний стан міста. Його кінцевою метою є краща якість життя, що

досягається завдяки ефективнішим державним послугам, підвищенню безпеки, меншій екологічній небезпеці та кращому доступу до зручностей, таких як парки.

Принцип розумного міста також включає прозорість даних, що передбачає масове управління даними, цифрове, відкрите та прозоре адміністрування за активної участі громадян, що пропонує доступ до даних та чітку політику. Безпека посилюється за допомогою технологічних засобів [8], таких як датчики та камери, а також суворої кібербезпеки для захисту обміну інформацією.

1.1.2 Технологічні основи «Розумних міст»

Цілі розумного міста значною мірою залежать від надійної технологічної основи. Ключові технології включають датчики інтернету речей, аналітику великих даних, штучний інтелект, машинне навчання, інформаційно-комунікаційні технології, хмарні системи управління послугами та розумні мережі. Датчики IoT використовуються для вимірювання різних факторів навколишнього середовища, таких як якість повітря та води, шум та відходи [9]. На рисунку 1.2 зображений принцип роботи типічної системи IoT.

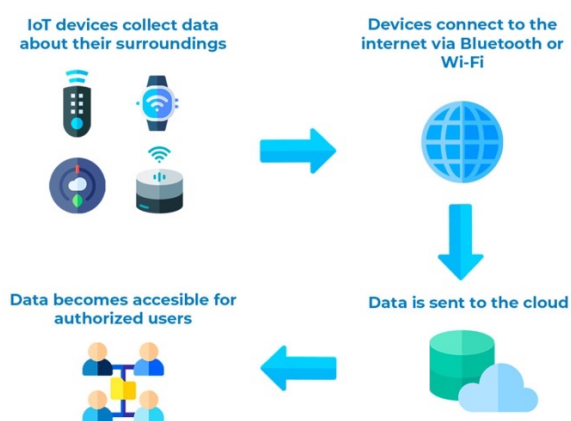


Рисунок 1.2 – Принцип роботи системи IoT

Аналітика великих даних використовується для обробки та аналізу величезних обсягів даних, що генеруються датчиками IoT, виявлення

закономірностей та тенденцій для стратегічного планування. ШІ та МН використовуються для прогнозування (наприклад, прогнозування випадків забруднення), оптимізації міських функцій (наприклад, маршрутів управління відходами, споживання енергії) та забезпечення проактивних стратегій пом'якшення екологічних проблем. ІКТ є важливою платформою, яка поєднує всі елементи розумного міста та оптимізує товари та послуги. Хмарні платформи [10] управління послугами забезпечують платформи для багатьох рішень як важлива центральна нервова система для планування, контролю та управління діяльністю в межах міста з наскрізним аналізом та контролем запитів на послуги. Розумні мережі дозволяють динамічно регулювати енергопостачання відповідно до попиту, зменшуючи втрати та витрати.

Покращення мініатюризації датчиків, а також зниження їх вартості, у поєднанні з бездротовим зв'язком та інноваціями в обробці даних (хмарні обчислення, аналіз великих даних), дозволили встановити системи моніторингу в масштабі всього міста. Це означає, що поєднання цих технологій не тільки знижує їх експлуатаційні витрати, але й дозволяє створювати складні, веб-подібні структури, здатні збирати та обробляти великі обсяги інформації, що є необхідним аспектом для ефективного функціонування розумного міста[11].

1.1.3 Міжнародні стандарти та рамки для розвитку розумного міста

Для забезпечення сумісності і сталості програм для розумних міст, міжнародні рамки та стандарти є важливим питанням. Деякими з ключових установ у цих зусиллях є: «Міжнародна організація зі стандартизації», «Міжнародний союз електрозв'язку», «Міжнародна електротехнічна комісія» та «Інститут інженерів з електротехніки та електроніки». ISO розробляє стандарти для сталого розвитку, інтелектуального та стійкого міста, такі як: ISO 37101, ISO 37120, ISO 37122 та ISO 37123 [12]. ISO розглядає ці стандарти як «перший крок до ідеального та сумісного світу, що базується на принципах «підключи та працюй», де міста можуть поєднувати рішення від різних постачальників, не

боячись прив'язки до певного рівня, застарівання чи безперспективних ініціатив». ІТУ визначає розумні міста та розробляє ключові показники ефективності (KPI) через ініціативу «Об'єднані за розумні стійкі міста», яка спрямована на покращення якості життя в економічному, соціальному та екологічному вимірах [13].

Ці вимоги є життєво необхідними для належної практики, взаємодії та сталого розвитку. Взаємодія гарантується таким чином, що різні пристрої та системи від різних постачальників можуть обмінюватися інформацією та гармонійно функціонувати без прив'язки до певного постачальника. Сталий розвиток встановлює спільну термінологію та спосіб вимірювання, як зазначено у визначенні ІТУ, прийнятому ООН [14]. Належна практика надає чіткі рекомендації щодо практики впровадження та закупівель.

Як видно з їх застосування, поширення систем вимірювання та показників (понад 1152 показники) багатьма установами та організаціями створює проблеми для розвитку спільного розуміння та єдиного ринку рішень і тому вимагає більшої координації між органами стандартизації та містами [15].

1.2 Обмеження традиційних методів моніторингу екосистем

Традиційні методи моніторингу екосистеми міських парків, якими б критичними вони не були, не справляються з масштабом, складністю та динамічною складністю міських екосистем. Традиційні механізми, такі як ручний збір даних та візуальні інструменти, не мають достатньої потужності, що робить повноцінне, перевірене та прогнозне управління навколишнім середовищем неможливим.

1.2.1 Огляд традиційних підходів до моніторингу

Традиційний моніторинг зазвичай здійснюється за допомогою ручних обстежень та підрахунку видів, картографування середовищ існування, польових

спостережень та візуальних досліджень. Ручне обстеження передбачає безпосереднє спостереження та підрахунок видів рослин і тварин. Картування середовищ існування включає ручний опис та класифікацію типів середовищ існування в зелених зонах [16]. Польове спостереження включає безпосередню оцінку умов навколишнього середовища на місці, що в більшості випадків доповнюється обстеженнями, інтерв'ю та спостереженнями за поведінкою з соціально-екологічних міркувань. Візуальний огляд спирається на людський зір (з допомогою інструментів або без них) для оцінки чистоти, виявлення видимих дефектів або визначення загального стану.

1.3 Інтернет речей для моніторингу екосистем

Використання IoT у моніторингу навколишнього середовища надає величезні переваги порівняно з традиційним способом, революціонізуючи управління міськими екосистемами. Пристрої IoT зможуть контролювати параметри навколишнього середовища цілодобово та безперервно, надаючи дані в режимі реального часу про такі параметри, як якість повітря, якість води, рівень шуму та ґрунт [17]. Він забезпечує зворотний зв'язок у режимі реального часу щодо стану навколишнього середовища. IoT автоматично обробляє дані, економить людино-години та оптимізує операції. Окрім того може бути реалізована автоматизація, наприклад, активація водоочисних споруд або очищувачів повітря, коли рівні концентрації забруднюючих речовин перевищують певні порогові значення.

Величезні обсяги даних, які генерують пристрої IoT, надають глибоку інформацію про екологічні тенденції та закономірності. Такий метод дозволяє містам приймати кращі рішення, створювати цілеспрямовані стратегії та прогнозувати екологічні небезпеки, як епізоди забруднення повітря [18]. Автоматизуючи регулярну роботу, оптимізуючи використання ресурсів (наприклад, води, енергії) та забезпечуючи прогнозне обслуговування, IoT може значно знизити експлуатаційні витрати та витрати на технічне обслуговування.

Датчики можуть бути встановлені у віддалених або важкодоступних місцях у парках та громадських просторах, що дозволяє збирати дані з мінімальним прямим втручанням людини.

Моніторинг у режимі реального часу дозволяє виявляти сплески забруднення, зменшувати вплив токсичних частинок та завчасно попереджати про екологічні загрози (наприклад, витоки газу, пожежі, повені), що дозволяє вживати більш швидких та цілеспрямованих заходів.

1.3.1 Архітектура системи IoT та технології, що її забезпечують

Системи Інтернету речей зазвичай структуровані за багаторівневою архітектурою для управління складністю та полегшення потоку даних від фізичних пристроїв до кінцевих користувачів. Хоча існують різні варіації [19] (наприклад, 3-рівневі, 5-рівневі моделі), загальна концепція передбачає чотири основні рівні: сенсорний, мережевий, шар обробки даних та прикладний.

– Сенсорний шар складається з фізичних «речей», вбудованих з датчиками та виконавчими механізмами, відповідальними за взаємодію з фізичним середовищем. Компоненти цього шару включають датчики, які виявляють зміни в навколишньому середовищі та перетворюють фізичні параметри (наприклад, температуру, світло, рух, тиск) в електричні сигнали або дані [20]. Виконавчі механізми — це пристрої, які отримують команди та викликають фізичні зміни в навколишньому середовищі (наприклад, відкриття клапана, увімкнення двигуна). Автоматизація досягається, коли датчики та виконавчі механізми працюють разом без втручання людини [20]. Мікроконтролери, такі як ESP32 та Arduino, зазвичай використовуються для взаємодії з датчиками, локальної обробки даних та забезпечення підключення до мережевого шару.

Широкий спектр та специфіка доступних екологічних датчиків (наприклад, розрізнення між PM2.5, NO2, O3 для якості повітря) дозволяють здійснювати рівень детального та цільового моніторингу, який раніше був

недосяжним традиційними методами [21]. Це призводить до більш точного виявлення проблем та втручання.

Таблиця 1.1 – Поширені екологічні датчики IoT та вимірювані параметри

Датчик	Вимірювані параметри	Типове застосування в моніторингу парків
DHT-11	Температура, вологість	Вимірювання температури та вологості повітря
MQ-135	PM2.5, PM10, PM1, NO ₂ , SO ₂ , CO, O ₃ , CO ₂ , NH ₃	Моніторинг забруднення міського повітря, оцінка ризиків для здоров'я, оцінка переваг очищення повітря рослинністю
TDS	Мутність, Провідність, Розчинений кисень, Хімічні забруднювачі, Рівень поживних речовин, Важкі метали, Токсини	Моніторинг водних об'єктів (ставків, струмків) на предмет забруднення, запобігання цвітінню водоростей, оцінка здоров'я водних екосистем
MAX4466	Рівень звукового тиску (децибели)	Управління рівнем міського шуму, оцінка впливу на комфорт людей та дику природу
YL-69	Вологість ґрунту, рН, Рівень поживних речовин	Оптимізація графіків зрошення, забезпечення здоров'я рослин, управління поверхневим стоком

– Мережевий шар відповідає за забезпечення комунікаційної інфраструктури, яка дозволяє обмінюватися даними між датчиками, виконавчими механізмами, шлюзами та хмарними платформами. Він обробляє величезний обсяг даних, що генеруються датчиками, забезпечуючи надійну та ефективну передачу. Це критично важливо для забезпечення сумісності та взаємодії між різнорідними пристроями від різних виробників.

Технології підключення включають локальні мережі (LAN) / персональні мережі (PAN), такі як Wi-Fi, Bluetooth, Zigbee, Ultra Wideband (UWB). Також використовуються низькоенергетичні глобальні мережі (LPWAN), зокрема LoRaWAN, яка є далекобійною та має низьке енергоспоживання, підходячи для тисяч пристроїв, та NB-IoT, що використовує LTE, має розширене покриття та надзвичайно низьке енергоспоживання. Стільникові мережі, такі як GSM, GPRS, LTE, 4G, 5G, застосовуються для датчиків, що не потребують підключення до шлюзу або високошвидкісної передачі даних [61].

– Шар обробки даних відповідає за прийом, зберігання, обробку та аналіз величезних обсягів необроблених даних, зібраних з пристроїв IoT. Хмарні обчислення забезпечують масштабовану інфраструктуру для зберігання, обробки та аналізу великих обсягів даних IoT. Основними постачальниками хмарних платформ є Microsoft Azure (Event Hubs, Service Bus, IoT Hub), Amazon Web Services (AWS IoT Core, S3), Google Cloud та IBM Bluemix [22].

– Останнім шаром є шар застосунків. Це найвищий шар, що безпосередньо взаємодіє з кінцевими користувачами та надає функціональні можливості для доступу, контролю та візуалізації даних IoT. Користувацькі інтерфейси (UI) зазвичай реалізуються у вигляді мобільних застосунків, веб-порталів або спеціалізованих інформаційних панелей, що пропонують візуальне представлення вимірювань у певному контексті (наприклад, на карті) та дозволяють взаємодіяти з користувачем.

Ключові особливості інформаційних панелей екологічного моніторингу включають візуалізацію даних, огляд моніторингу в реальному часі, системи сповіщень, аналіз історичних даних, управління ризиками та проактивні дії, а також підтримку ГІС. Візуалізація даних передбачає представлення складних даних у легко інтерпретованих графічних форматах, таких як діаграми, карти та індикатори. Це дозволяє швидко інтерпретувати дані в реальному часі та історичні дані. Огляд моніторингу в реальному часі забезпечує безперервний, цілодобовий доступ до поточних даних, зібраних сенсорними мережами.

1.3.2 Успішні впровадження IoT у моніторингу довкілля «Розумних міст»

Численні міста по всьому світу успішно застосували рішення на основі інтернету речей для моніторингу навколишнього середовища, демонструючи практичність та корисність цих технологій. Барселона в Іспанії, впровадила загальноміську мережу датчиків якості повітря та інтелектуальне освітлення, зображено на рисунку 1.3, що призвело до значного зниження рівня NO₂ у районах з інтенсивним рухом транспорту [23].



Рисунок 1.3 – Установка для вимірювання забруднення повітря у Барселоні

Сінгапур використовує датчики якості води у водосховищах та каналах для контролю рівня поживних речовин, запобігання цвітінню водоростей та оптимізації операцій з очищення води. Лос-Анджелес, США, встановили обладнання для моніторингу шуму поблизу основних автомагістралей та аеропортів, що дозволяє динамічно адаптувати обмеження швидкості та розміщувати шумові бар'єри для контролю міського шумового забруднення. Грінополіс в Японія, запровадив систему управління відходами на основі штучного інтелекту (CleanCycle) [24], яка використовує машинне навчання для оптимізації маршрутів та графіків збору, що призводить до 50% зниження

експлуатаційних витрат та підвищення рівня переробки. Ековіль, Данія, запровадив інтелектуальні мережі на основі штучного інтелекту, що призвело до скорочення загального споживання енергії на 20% та підвищення ефективності відновлюваної енергії на 30%.

Ці приклади використання є вагомим доказом реальної цінності моніторингу навколишнього середовища в містах за допомогою інтернету речей. Вони демонструють, що ці системи не лише технологічно доцільні, але й фактично здатні забезпечити вимірні покращення якості навколишнього середовища, ефективності використання ресурсів та управління містом у великих масштабах.

1.4 Висновок до першого розділу

Міські парки та сквери знаходяться на передовій надання ключових екосистемних послуг, таких як очищення повітря, охолодження міського повітряного середовища, управління водними ресурсами та збереження біорізноманіття. Усім цим категоріям загрожують зростаючі екологічні проблеми, спричинені розширенням міст.

Інтернет речей (IoT) пропонує новаторське рішення, безпосередньо вирішуючи ці проблеми. Використовуючи архітектуру, що складається з сенсорного рівня (датчики та виконавчі механізми), мережевого рівня (комунікаційні протоколи та шлюзи), обчислювального рівня (периферійні обчислення, хмарні платформи, аналіз даних, штучний інтелект і машинне навчання) та прикладного рівня (користувацькі інтерфейси та панелі інструментів), IoT пропонує моніторинг у режимі реального часу, автоматизацію, прогностичну аналітику та прийняття рішень на основі даних. Успішний досвід мегаполісів по всьому світу доводить ефективність цього підходу.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБСАЙТУ ДЛЯ МОНІТОРИНГУ ЕКОСИСТЕМ ПАРКІВ ТА СКВЕРІВ

2.1 Актуальність та теоретичні основи проєктування системи моніторингу

Моніторинг довкілля є фундаментальним для розуміння змін у екосистемах та їхнього впливу, що дозволяє приймати обґрунтовані управлінські рішення та запобігати негативним змінам. Зростаючий інтерес до екологічного моніторингу, особливо в умовах зміни клімату та забруднення, вимагає нових, досконаліших інструментів для збору та обробки даних [25]. Традиційні методи, такі як на рисунку 2.1, часто не справляються з великими обсягами інформації, що генерується динамічними екологічними системами, створюючи потребу в точних даних у реальному часі.



Рисунок 2.1 – Метеорологічна будка для вимірювання температури

Міські зелені зони, такі як парки та сквери, є життєво важливими для добробуту мешканців та екології міста [26]. Проєктування систем, що

використовують технології реального часу, дозволяє не лише швидко реагувати на проблеми, але й сприяє довгостроковим покращенням у міській стійкості. Можливість миттєво отримувати дані про середовище парків та скверів міста також дозволяє громадянам планувати свій відпочинок у них. Пробіжки, пікніки, прогулянки та багато інших активностей можуть бути точно затверджені за допомогою інформації про екосистему.

2.1.1 Застосування IoT в екологічному моніторингу

У екологічному моніторингу IoT дозволяє ефективно виявляти джерела забруднення, підвищуючи точність та швидкість отримання даних. Системи на основі IoT-датчиків популярні для моніторингу довкілля в реальному часі, використовуючи мережі LTE або Wi-Fi для передачі даних про якість повітря, води та ґрунту до хмарних платформ. Моніторинг може охоплювати температуру, вологість, якість повітря та ґрунту [27].

Концепція «розумного екологічного моніторингу» (Smart Environmental Monitoring, SEM) поєднує сучасні датчики, машинне навчання (ML) та IoT для підвищення точності та розширення охоплення моніторингу. Це дозволяє створювати бездротові системи віддаленого моніторингу, що мінімізують людське втручання та забезпечують низьку затримку у виявленні загроз. Типова архітектура IoT-систем включає датчики, підключення, обробку даних та застосунки [28].

2.1.2 Переваги та виклики впровадження IoT-систем для моніторингу парків та скверів

Впровадження IoT-систем для моніторингу міських зелених зон пропонує значні переваги. IoT дозволяє безперервно відстежувати стан обладнання та збирати дані з датчиків у реальному часі [29]. Вони є економічно ефективними

та надають надійніші дані для моделювання. Відкриті IoT-рішення, такі як Arduino, сприяють залученню широкого кола осіб до моніторингу.

Концепція «розумних парків» передбачає інтеграцію цифрових технологій та IoT-пристроїв для підвищення функціональності, сталості та покращення досвіду користувачів [30]. Ключові особливості включають:

- Підключення: Безкоштовний Wi-Fi та інтерактивні додатки.
- Сталість: Розумні системи освітлення та автоматизовані інструменти обслуговування.
- Збір даних: Використання IoT-датчиків для моніторингу використання парку, якості повітря та мікроклімату.
- Покращений досвід користувача: Додатки та сайти для доступу до даних.

Переваги охоплюють сприяння здоров'ю, екологічну стійкість, залучення громади, економічну ефективність та підвищення безпеки. У ширшому масштабі «розумного міста», IoT-датчики сприяють швидкому реагуванню на екологічні проблеми та довгостроковим покращенням міської стійкості.

Незважаючи на переваги, впровадження IoT-систем стикається з викликами. Проблеми ідентифікації та масштабованості виникають через велику кількість пристроїв та обсяги даних [31]. Управління даними є значним викликом, оскільки сенсорні мережі генерують великі обсяги даних, які потребують очищення від помилок. Забезпечення безпеки та конфіденційності даних є постійною проблемою. Енергоефективність є критично важливою, оскільки датчики повинні функціонувати роками, вимагаючи енергоефективних схем передачі [60].

Звичайно, це все можна і потрібно впроваджувати на вже існуючій інфраструктурі. У Тернополі однією з таких є відкрита система камер спостереження, що фактично постійно має живлення та з'єднання з інтернетом. Така система ідеально доповнює датчики збору інформації про середовище в парках та скверах.



Рисунок 2.2 – Зображення з камери відкритої системи спостереження

2.2 Методологія проєктування IoT систем

Методологія проєктування IoT-систем для екологічного моніторингу включає кілька взаємопов'язаних етапів: спостереження, оцінка фактичного стану, оцінка прогнозованого стану та підтримка прийняття управлінських рішень [32]. Основною метою проєктування є створення комунікаційної системи розподіленої мережі для збору екологічних даних, що дозволить підвищити ефективність збору інформації, забезпечити її захист та підтримувати надійну роботу системи в умовах масштабування [33].

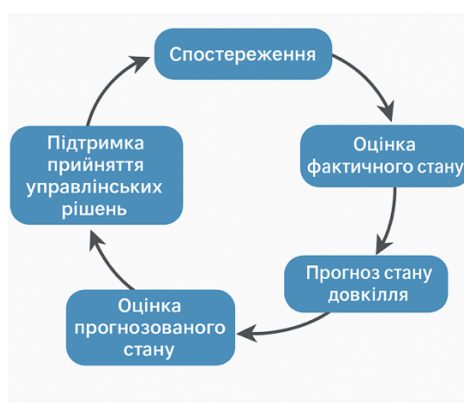


Рисунок 2.3 – Життєвий цикл проєктування IoT системи моніторингу

Для досягнення цієї мети на етапі проєктування виконуються такі завдання:

- Аналіз існуючих рішень у сфері збору та обробки екологічних даних для виявлення їхніх недоліків.
- Розробка вимог до комунікаційної системи розподіленої мережі, що враховує специфіку екологічних даних.
- Створення архітектури системи з урахуванням високих вимог до безпеки, надійності та масштабованості.

Практичне впровадження такої розподіленої мережі дозволяє автоматизувати процеси збору та передачі даних, забезпечити надійну обробку інформації в реальному часі для прийняття управлінських рішень, зменшити витрати на впровадження та обслуговування, а також підвищити рівень прозорості та доступності даних [34].

2.2.1 Архітектура IoT системи моніторингу парків та скверів

Архітектура IoT-системи для моніторингу навколишнього середовища, зокрема парків та скверів, проєктується як багатоварова структура, що забезпечує ефективний збір, передачу, обробку та візуалізацію даних [35].

Архітектура включає наступні основні шари:

- Сенсорний шар: відповідає за збір первинних даних з навколишнього середовища. Для даного проєкту може використовуватись датчик DHT11, який вимірює температуру та вологість.
- Мережевий шар: забезпечує зв'язок та підключення між пристроями в IoT-системі та інтернетом. Для передачі даних використовується модуль ESP01S, що забезпечує Wi-Fi з'єднання [35].
- Шар обробки даних: на цьому шарі відбувається отримання необроблених даних, їхня обробка та підготовка для подальшого аналізу. Для зберігання сенсорних даних використовується база даних MySQL, яка дозволяє ефективно керувати даними. Важливою частиною цього шару є також проєктування механізмів очищення даних від помилок.

– Шар застосунків/візуалізації: Це верхній шар, який безпосередньо взаємодіє з кінцевим користувачем. Він надає зручні інтерфейси, такі як веб-портали, для доступу до IoT-пристроїв та керування ними. Для візуалізації даних на вебсайті можна використати API від Google Maps, або Leaflet, що дозволяє відображати дані на інтерактивній карті.

2.3 Вибір та обґрунтування апаратних компонентів системи

Для проєктування вебсайту моніторингу екосистем парків та скверів за допомогою IoT було обрано та обґрунтовано ключові апаратні та програмні компоненти.

Датчик DHT11 (Температури та Вологості) обрано завдяки його економічній ефективності, простоті використання та достатній точності для загального моніторингу мікроклімату в парках. Він легко інтегрується з мікроконтролерами, такими як Arduino [36]. Діапазон вимірювання температури становить від 0 до 50 °C з точністю ± 2 °C, а вологості – від 20% до 80% з точністю $\pm 5\%$.

Модуль ESP-01S ESP8266 Wi-Fi є самодостатньою системою на кристалі з інтегрованим стеком протоколів TCP/IP, що дозволяє будь-якому мікроконтролеру отримати доступ до мережі Wi-Fi. ESP01S є економічно ефективним рішенням для забезпечення бездротового зв'язку. Його здатність працювати як адаптер UART-to-Wi-Fi дозволяє передавати зібрані дані з DHT11 на віддалений сервер.

2.4 Проєктування бази даних

Для зберігання даних датчиків IoT обрано MySQL як реляційну систему управління базами даних (РСУБД), що буде розміщена на хостингу Awardspace та керуватиметься через phpMyAdmin. MySQL дозволяє ефективно зберігати та

аналізувати дані, отримані від датчиків, такі як температура, вологість, час отримання та місце розташування.

Структури таблиці показана на рисунку 2.4. Для зберігання даних проєктується єдина таблиця `temperature_data` зі структурою, що відповідає потребам моніторингу температури та вологості.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
☐ 1	id	int			Ні	Немає		AUTO_INCREMENT
☐ 2	device_id	varchar(50)	utf8mb4_0900_ai_ci		Так	NULL		
☐ 3	temperature	float			Так	NULL		
☐ 4	humidity	float			Так	NULL		
☐ 5	timestamp	timestamp			Так	CURRENT_TIMESTAMP		DEFAULT_GENERATED

Рисунок 2.4 – Таблиця `temperature_data` в середовищі phpMyAdmin

Для опису функціоналу бази даних створено діаграму, яка показана на рисунку 2.5.

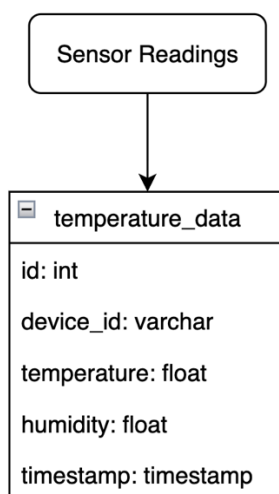


Рисунок 2.5 – Діаграма бази даних

Обґрунтування вибору MySQL та даної структури полягає в забезпеченні ефективного зберігання та аналізу даних, отриманих від датчиків, таких як температура, вологість, ідентифікатор пристрою та мітка часу. Ця спрощена

схема є достатньою для початкового етапу проєкту, забезпечуючи легкість управління та масштабованість для подальших розширень.

2.5 Проєктування вебсайту для моніторингу парків та скверів

Вебсайт для моніторингу екосистем парків та скверів проєктується з урахуванням низки функціональних вимог, що забезпечують ефективне відображення даних, взаємодію з користувачем та підтримку прийняття рішень.

Основні функціональні вимоги до вебсайту включають:

- Відображення даних у реальному часі: вебсайт повинен відображати актуальні показники температури та вологості, отримані від датчиків, безпосередньо на інтерактивній карті.
- Геопросторова візуалізація: система повинна чітко відображати розташування парків та скверів на карті Leaflet [37], використовуючи маркери для позначення точок встановлення датчиків.
- Взаємодія з користувачем: при натисканні на точку датчика на карті, користувач повинен мати можливість отримати детальну інформацію про цей датчик, включаючи його останні показники та час оновлення.
- Перегляд історичних даних: користувачі повинні мати доступ до історичних даних для аналізу тенденцій змін температури та вологості з часом.
- Масштабованість: система має бути спроектована таким чином, щоб забезпечувати легке додавання нових датчиків та моніторингових точок у майбутньому.
- Доступність та зручність використання: інтерфейс вебсайту повинен бути інтуїтивно зрозумілим та доступним для широкого кола користувачів.
- Механізми оновлення даних [38]: вебсайт повинен забезпечувати динамічне оновлення даних на карті, щоб користувачі завжди бачили найактуальнішу інформацію.

2.5.1 Дизайн та інтерфейс користувача

Дизайн та інтерфейс користувача вебсайту моніторингу екосистем парків та скверів проєктується з метою забезпечення ефективної взаємодії та сприйняття інформації. Метою є створення інтуїтивно зрозумілого, візуально привабливого та функціонального інтерфейсу.

Основні принципи дизайну інтерфейсу включають:

- Чистий та неускладнений дизайн: інтерфейс повинен бути мінімалістичним, уникаючи зайвих елементів, що можуть відволікати від основної інформації – карти та даних датчиків [39].
- Інтуїтивна навігація: структура вебсайту повинна бути логічною та простою для навігації.
- Ефективне використання кольору: кольорова схема повинна бути ретельно підібрана для підвищення читабельності та виділення важливих точок даних.
- Чітке маркування: всі ключові області, маркери на карті та точки даних повинні бути чітко та лаконічно позначені.
- Адаптивний дизайн: вебсайт повинен бути адаптований для коректного відображення та функціонування на різних пристроях.
- Акцент на досвіді користувача та залученні: дизайн повинен бути орієнтований на покращення загального досвіду користувача.

Для прикладу такого дизайну можна взяти сервіс Zabbix, інтерфейс якого зображено на рисунку 2.6.

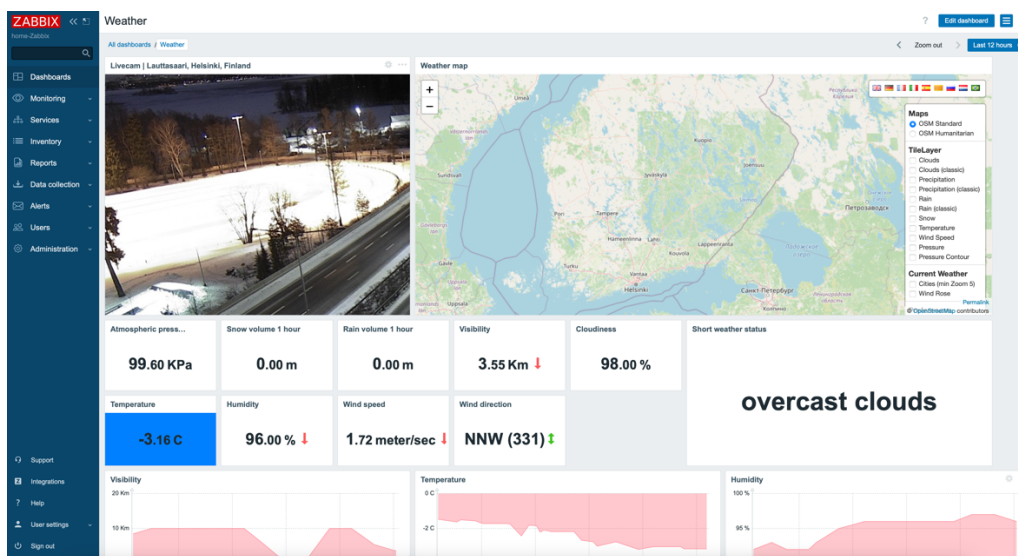


Рисунок 2.6 – Приклад сервісу з моніторингу метеоданих

В дизайні даної системи поєднано функціональність і легкість освоєння для користувачів.

2.5.2 Проєктування відображення даних на карті Leaflet

Відображення даних на карті Leaflet є центральним елементом проєктування вебсайту моніторингу екосистем парків та скверів. Leaflet, як легка та мобільна JavaScript-бібліотека [40], ідеально підходить для створення інтерактивних карт, що відображають дані датчиків у реальному часі.

Процес проєктування відображення даних включає:

- Інтеграція бібліотеки Leaflet: проєктується підключення бібліотеки Leaflet до вебсайту, створюючи контейнер для карти та налаштовуючи її початковий вигляд.
- Завантаження базових шарів карти: для відображення географічного контексту проєктується використання базових шарів карти, таких як OpenStreetMap [41].
- Розміщення зон та маркерів для точок моніторингу: кожен парк або точка встановлення датчика на карті представляється у вигляді полігону і маркера.

– Відображення показників температури та вологості: показники температури та вологості, отримані від датчиків DHT11 [42] та збережені у базі даних, проєктується відображати у спливаючих вікнах (popups) або безпосередньо в іконках маркерів при наведенні або натисканні на них.

– Динамічне завантаження та оновлення даних: для забезпечення відображення даних у реальному часі, вебсайт проєктується використовувати механізми динамічного завантаження даних. JavaScript-код на клієнтській стороні періодично надсилатиме запити до серверної частини для отримання найактуальніших показників датчиків з бази даних.

– Інтерактивні функції: Leaflet надає широкий спектр інтерактивних функцій, які підвищують зручність використання карти, включаючи перетягування карти, масштабування та події кліку.

На рисунку 2.7 зображено приклад оформлення зон та точок на карті Leaflet, що може бути використано в даному проєкті.



Рисунок 2.7 – Мапа leaflet

2.5.3 Проєктування механізмів оновлення та взаємодії з даними

Ефективність системи моніторингу екосистем парків та скверів значною мірою залежить від надійності та швидкості механізмів оновлення та взаємодії з даними, які закладаються на етапі проєктування.

Проектований потік даних у системі: Загальний потік даних у системі проектується за наступною послідовністю:

- Збір даних: Датчик DHT11 збирає показники температури та вологості.
- Передача на мікроконтролер: Дані передаються на мікроконтролер Arduino.
- Бездротова передача: використовуючи модуль ESP01S, показники надсилаються через Wi-Fi на веб-сервер (Awardspace).
- Зберігання в базі даних: веб-сервер приймає дані та зберігає їх у базі даних MySQL (phpMyAdmin) [43].
- Візуалізація на вебсайті: вебсайт, що використовує бібліотеку Leaflet, отримує дані з MySQL та відображає їх на інтерактивній карті.

Для забезпечення відображення даних у реальному або наближеному до реального часу проектуються такі підходи:

- Частота оновлення: датчики проектуються надсилати дані з певною періодичністю, наприклад, раз на 10 секунд.
- Серверна обробка: на серверній стороні проектується скрипт, який відповідатиме за прийом HTTP-запитів від модулів ESP01S, обробку отриманих даних та їх запис у таблицю `temperature_data` бази даних MySQL.
- Клієнтські оновлення: вебсайт проектується використовувати JavaScript для динамічного оновлення даних на карті Leaflet за допомогою асинхронних запитів (AJAX) [44], які періодично звертатимуться до серверного API для отримання найактуальніших показників датчиків.
- Проектування взаємодії користувача з даними: система передбачає кілька рівнів взаємодії користувача з відображеними даними:
 - Натискання на точки моніторингу: Користувач може натиснути на маркер, що представляє парк або точку встановлення датчика на карті, для отримання детальної інформації у спливаючому вікні.

– Перегляд історичних даних: Проєктується можливість розширення функціональності для перегляду графіків історичних даних для обраного датчика.

2.6 Висновки до розділу 2

У цьому розділі було детально розглянуто проєктну частину розробки вебсайту для моніторингу екосистем парків та скверів за допомогою інтернету речей. Обґрунтовано актуальність такого рішення, що дозволяє перейти до екологічного управління, надаючи точну інформацію в реальному часі.

Представлена методологія проєктування IoT-системи включає аналіз існуючих рішень, розробку вимог та створення архітектури з урахуванням безпеки, надійності та масштабованості. Ключові компоненти системи – датчик DHT11, модуль ESP01S, база даних MySQL (розміщена на хостингу Awardspace та керована через phpMyAdmin) та бібліотека Leaflet – були обрані та обґрунтовані з точки зору їхньої економічної ефективності, простоти інтеграції та відповідності функціональним вимогам проєкту. Особливу увагу приділено проєктуванню спрощеної структури бази даних MySQL, що забезпечує ефективне зберігання даних.

Проєктування вебсайту сфокусовано на забезпеченні відображення даних у реальному часі на інтерактивній карті, геопросторовій візуалізації, можливостях взаємодії з користувачем, а також перегляді історичних показань. Дизайн інтерфейсу користувача спрямований на інтуїтивність, чистоту та адаптивність, що сприяє легкому доступу та інтерпретації інформації.

Нарешті, було розглянуто підходи до проєктування системи з урахуванням майбутнього тестування та оцінки продуктивності. Це включає багатоваріантний підхід до тестування, забезпечення точності та надійності даних через калібрування датчиків та корекцію помилок передачі, а також врахування ключових показників продуктивності на етапі проєктування.

У цьому розділі було закладено основу для подальшої розробки та реалізації вебсайту моніторингу екосистем парків та скверів, що буде детально описано у наступному розділі.

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА. РЕАЛІЗАЦІЯ ВЕБСАЙТУ ДЛЯ МОНІТОРИНГУ ЕКОСИСТЕМ ПАРКІВ ТА СКВЕРІВ

3.1 Реалізація апаратної частини системи моніторингу

Практична реалізація системи моніторингу екосистем парків та скверів починається з розгортання апаратної частини, яка відповідає за збір первинних екологічних даних. Основними компонентами, що були використані у цьому проєкті, є датчик температури та вологості DHT11 та модуль ESP01S [45]. Перший датчик зображений на рисунку 3.1.

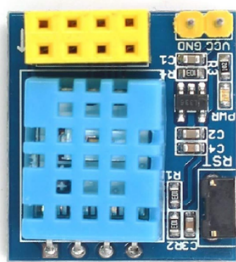


Рисунок 3.1 – Датчик температури та вологості повітря DHT11

Датчик для обробки та зв'язку показаний на рисунку 3.2.

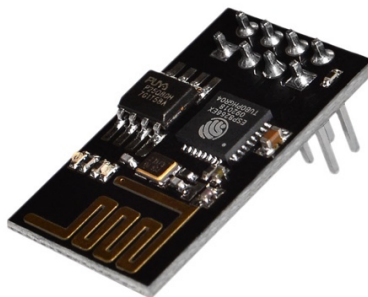


Рисунок 3.2 – Модуль ESP01S

Детальніший опис даних компонентів в наступному пункті.

3.1.1 Підключення датчика DHT11 до модуля ESP01S

Датчик DHT11 є ключовим елементом для збору даних про мікроклімат. Він є недорогим цифровим сенсором, що вимірює температуру та вологість. У цьому проєкті DHT11 безпосередньо підключений до модуля ESP01S, який виступає як мікроконтролер, що обробляє дані та забезпечує бездротовий зв'язок. ESP01S є самодостатньою системою на кристалі (SoC) і може функціонувати як самостійний мікроконтролер.

Принцип роботи DHT11 базується на використанні ємнісного елемента для вимірювання вологості та термістора з негативним температурним коефіцієнтом (NTC) для вимірювання температури [46]. Зміна рівня вологості призводить до зміни ємності, яка потім перетворюється на цифровий сигнал. Для температури використовується термістор, опір якого зменшується зі зростанням температури.

Схема прямого підключення DHT11 до ESP01S: Згідно з наданим рисунком 3.3, підключення виконано на пряму пінами:

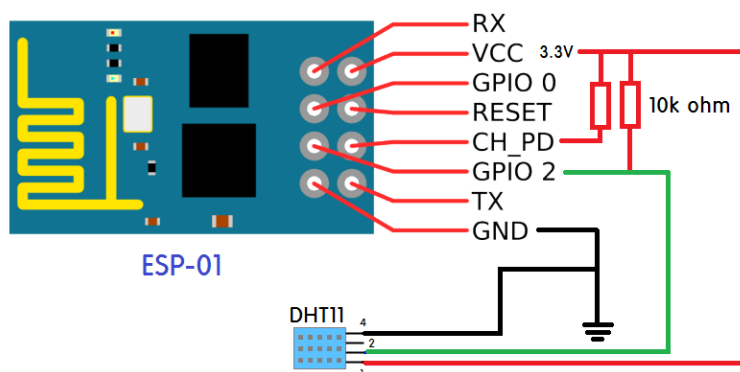


Рисунок 3.3 – Схема підключення датчика до модуля

Підключення виконано на пряму пінами:

- VCC (DHT11): Підключається до виводу 3.3V на модулі ESP01S. Важливо забезпечити стабільне живлення 3.3V – 5V, оскільки ESP01S може споживати значний струм (до 300 мА).

- DATA (DHT11): Підключається до GPIO2-виводу ESP01S [47]. Для стабільної роботи лінії даних DHT11 рекомендується використовувати підтягувальний резистор (pull-up resistor) 5-10 кОм між виводом DATA та VCC.
- GND (DHT11): Підключається до виводу GND на модулі ESP01S.

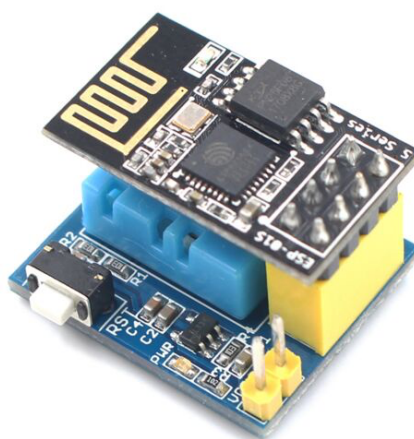


Рисунок 3.4 – Вигляд зібраного датчика

Для програмування модуля ESP01S датчик необхідно відключати [48].

3.1.2 Програмне забезпечення модуля ESP01S

Програмне забезпечення для модуля ESP01S розробляється безпосередньо для цього SoC, оскільки він має достатню обчислювальну потужність та обсяг пам'яті для інтеграції з датчиками та іншими пристроями через свої GPIO-порти. ESP01S постачається з інтегрованим стеком протоколів TCP/IP [49], що дозволяє йому самостійно встановлювати Wi-Fi з'єднання та передавати дані.

Логіка роботи прошивки ESP01S:

- Ініціалізація: Налаштування GPIO-виводу для зв'язку з датчиком DHT11 та ініціалізація Wi-Fi модуля.
- Зчитування даних: Періодичне зчитування показників температури та вологості з датчика DHT11.

- Формування запиту: Зчитані дані, разом з унікальним ідентифікатором пристрою (`device_id`), формуються у HTTP GET або POST запит.
- Передача даних: Модуль ESP01S встановлює Wi-Fi з'єднання з визначеною мережею та відправляє сформований HTTP-запит на вказаний веб-сервер.
- Обробка відповіді: За необхідності, прошивка ESP01S може обробляти відповідь від сервера, підтверджуючи успішність передачі даних.

Код для надсилання показників в базу даних цим датчиком описано в лістингу 3.1.

Лістинг 3.1 – Код для обробки даних з датчика та їх відправки

```
void loop() {
    if (WiFi.status() == WL_CONNECTED) {
        float t = dht.readTemperature();
        float h = dht.readHumidity();

        if (!isnan(t) && !isnan(h)) {
            WiFiClient client;
            HTTPClient http;
            String url = String(serverName) +
            «?device_id=espl&temperature=» + t + «&humidity=» + h;
            http.begin(client, url);
            int httpCode = http.GET();
            if (httpCode > 0) {
                Serial.printf(«HTTP GET... code: %d\n», httpCode);
            } else {
                Serial.printf(«GET failed, error: %s\n»,
                http.errorToString(httpCode).c_str());
            }
            http.end();
        } else {
            Serial.println(«Failed to read from DHT sensor!»);
        }
    }
}
```

Ця апаратна реалізація, де ESP01S виконує функції мікроконтролера та Wi-Fi модуля, забезпечує компактний та ефективний збір і передачу екологічних даних. Повний код подано в додатку А.

3.2 Реалізація програмної частини та бази даних

Після налаштування апаратної частини, наступним етапом є реалізація програмного забезпечення, що включає серверне середовище, базу даних та скрипти для обробки даних.

3.2.1 Налаштування серверного середовища та бази даних

Вебсайт для моніторингу екосистем розміщено на хостингу Awardspace. Цей хостинг надає необхідне середовище для роботи вебсайту, включаючи підтримку PHP та базу даних MySQL. Керування базою даних здійснюється через сервіс адміністрування бд phpMyAdmin, інтерфейс якого зображено на рисунку 3.5.

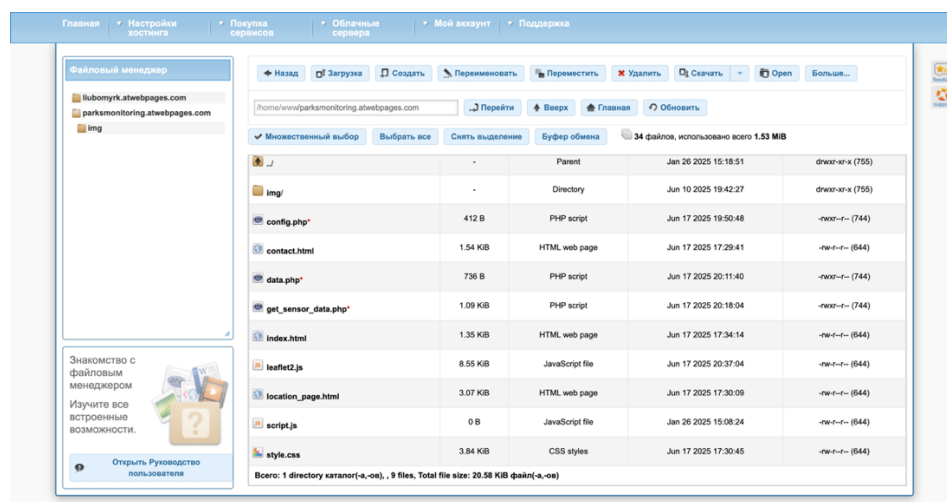
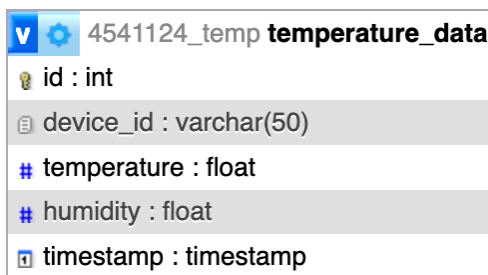


Рисунок 3.5 – Інтерфейс хостингу Awardspace з кореневою папкою сайту

У phpMyAdmin було створено базу даних для проєкту. Згідно з проєктними вимогами та наданим зображенням, для зберігання даних датчиків була створена єдина таблиця `temperature_data`, що показана на рисунку 3.6.



4541124_temp temperature_data	
🔑	id : int
📄	device_id : varchar(50)
#	temperature : float
#	humidity : float
🕒	timestamp : timestamp

Рисунок 3.6 – Таблиця temperature_data в інтерфейсі phpMyAdmin

Опис полів таблиці temperature_data:

- id (int, PRIMARY KEY, AUTO_INCREMENT): Унікальний ідентифікатор кожного запису даних. Автоматично збільшується при додаванні нового запису.
- device_id (varchar(50)): ідентифікатор пристрою/датчика, який надіслав дані (наприклад, «park_central», «square_north»). це дозволяє розрізняти дані від різних точок моніторингу.
- temperature (float): значення температури, отримане від датчика.
- humidity (float): значення вологості, отримане від датчика.
- timestamp (timestamp, DEFAULT CURRENT_TIMESTAMP): час та дата отримання даних. Автоматично встановлюється на поточний час при додаванні запису.

Дана структура таблиці є ефективною для зберігання основних параметрів моніторингу та дозволяє легко розширювати функціональність у майбутньому, додаючи нові поля або пов'язані таблиці за потреби.

3.2.2 Реалізація серверного скрипту для прийому даних

Серверна частина системи відповідає за прийом даних від IoT-пристроїв [50] та їхнє збереження у базі даних MySQL. Для цього використовується два PHP-файли, розміщені на веб-сервері Awardspace: config.php для налаштування підключення до бази даних та data.php для обробки вхідних даних та їхнього запису.

Файл config.php містить параметри для підключення до бази даних MySQL, використовуючи об'єктно-орієнтований підхід PDO (PHP Data Objects) для безпечного та гнучкого з'єднання. Код для даного файлу показаний в лістингу 3.2

Лістинг 3.2 – Код для файлу config.php

```
<?php
$host = «fdb1030.awardspace.net»;
$dbname = «4541124_temp»;
$username = «4541124_temp»;
$password = «*****»;
try {
    $pdo = new PDO(«mysql:host=$host;dbname=$dbname;charset=utf8»,
$username, $password);
} catch (PDOException $e) {
    die(«Database connection failed: « . $e->getMessage());
}
?>
```

Файл data.php відповідає за прийом HTTP GET-запитів від модуля ESP01S, вилучення параметрів device_id, temperature та humidity, а також за їхню валідацію та безпечний запис у таблицю temperature_data бази даних. Використання підготовлених запитів (bindParam) допомагає запобігти SQL-ін'єкціям. Основну частину коду подано в лістингу 3.3, повний код знаходиться в додатку Б.

Лістинг 3.3 – Уривок коду для файлу data.php

```
<?php
include 'config.php';

if (isset($_GET['device_id']) && isset($_GET['temperature']) &&
isset($_GET['humidity'])) {
    $device_id = $_GET['device_id'];
    $temperature = $_GET['temperature'];
    $humidity = $_GET['humidity'];
    $sql = «INSERT INTO temperature_data (device_id, temperature,
humidity)
VALUES (:device_id, :temperature, :humidity)»;
    $stmt = $pdo->prepare($sql);
    $stmt->bindParam(':device_id', $device_id);
    $stmt->bindParam(':temperature', $temperature);
```



```
$stmt->bindParam(':humidity', $humidity);
```

Логіка роботи серверних скриптів:

- Підключення: коли модуль ESP01S надсилає HTTP GET-запит до data.php, цей файл спочатку включає config.php, встановлюючи з'єднання з базою даних.
- Отримання та валідація даних: data.php перевіряє наявність всіх необхідних параметрів (device_id, temperature, humidity) у GET-запиті.
- Безпечний запис: Отримані дані використовуються для підготовки та виконання SQL-запиту на вставку в таблицю temperature_data. PDO та підготовлені запити забезпечують, що дані коректно обробляються та вставляються, мінімізуючи ризики безпеки.
- Відповідь: Скрипт повертає текстову відповідь («OK: Data inserted successfully» або «Error inserting data»), яка може бути використана модулем ESP01S для підтвердження успішності операції.

Ці серверні скрипти є мостом між апаратною частиною системи моніторингу та вебсайтом, забезпечуючи безперервний, безпечний та надійний потік даних від датчиків до бази даних.

3.3 Реалізація клієнтської частини вебсайту

Клієнтська частина системи – це власне вебсайт, доступний користувачам. Вона розроблена з використанням стандартних веб-технологій: HTML для структури, CSS для стилізації та JavaScript для інтерактивності та динамічного відображення даних.

3.3.1 Структура вебсайту та головна сторінка

Головна сторінка вебсайту (index.html) є центральним елементом інтерфейсу користувача. Вона містить інтерактивну карту, див. рисунок 3.7, на

якій відображаються території парків з точками моніторингу температури та вологості.

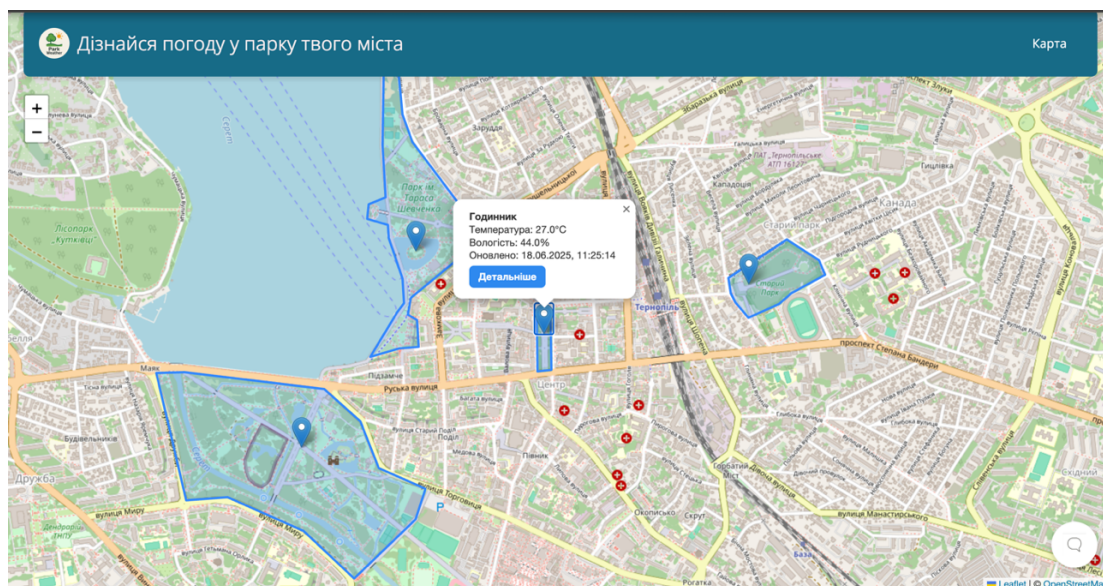


Рисунок 3.7 – Головна сторінка вебсайту з картою

Окрім самого `index.html` на сайті також присутній файл `style.css`, що використовуються для оформлення вебсайту, включаючи розміри карти, шрифти, кольори та загальний вигляд елементів інтерфейсу.

3.3.2 Візуалізація даних на карті Leaflet

Для відображення геопросторових даних та показників датчиків на вебсайті використовується JavaScript-бібліотека Leaflet [51]. Leaflet є провідною відкритою бібліотекою для створення мобільних інтерактивних карт, що відрізняється легкістю, продуктивністю та зручністю використання. Код для файлу `leaflet.js`, що описує функціонал карти з точками на сайті та інше, подано в додатку Ж.

Реалізація відображення даних:

- Ініціалізація карти: у JavaScript-коді ініціалізується об'єкт карти Leaflet, вказується елемент HTML, де буде відображатися карта, та встановлюються початкові координати центру та рівень масштабування.

- Завантаження даних з сервера: за допомогою асинхронних JavaScript-запитів (AJAX)[52] до серверного API (наприклад, до іншого PHP-скрипту, який вибирає дані з таблиці `temperature_data`) отримуються актуальні показники температури та вологості для кожного `device_id`.
- Розміщення маркерів: Для кожного `device_id` (точки моніторингу) на карті розміщується маркер. Координати для цих маркерів заздалегідь визначені в JavaScript-кодi.
- Відображення даних у спливаючих вікнах: При натисканні на маркер відкривається спливаюче вікно (popup)[53], яке відображає поточні показники температури та вологості для цього `device_id`, а також час останнього оновлення. Це дозволяє користувачам швидко отримувати деталі про конкретне місце моніторингу.

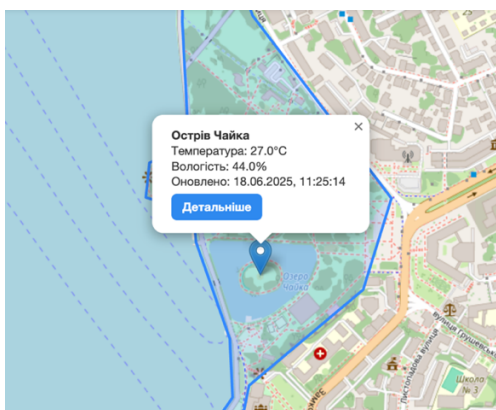


Рисунок 3.8 – Відображення даних з спливаючим вікном

Кнопка «детальніше» переводить на сторінку з детальною інформацією про обрану точку, це буде описано в наступних розділах. Лістинг для сторінки `index.html` подано в додатку В.

3.3.3 Взаємодія з користувачем

Окрім базової взаємодії з картою, вебсайт передбачає переходи до додаткових сторінок для розширеної інформації та зворотного зв'язку.

Перехід на сторінку «Детальніше»: у спливаючому вікні, що з'являється при натисканні на маркер на карті, додано кнопку «Детальніше». Натискання на цю кнопку перенаправляє користувача на сторінку, яка надає розширену інформацію про обрану точку моніторингу. Ця сторінка є динамічною, тобто її вміст формується на основі `device_id`, [54] переданого з головної сторінки.

На сторінці «Детальніше» відображаються:

- Назва точки моніторингу (наприклад, «Острів Чайка»).
- Актуальні показники температури та вологості.
- Посилання на «Камеру» (якщо доступно), що може надавати візуальний моніторинг.
- Графік температури, що відображає історичні дані [55] за певний період (наприклад, за останній день), дозволяючи користувачам аналізувати тенденції.

Код для сторінки детальніше подано у додатку Д. Вигляд сторінки відображено на рисунку 3.9.



Рисунок 3.9 – Сторінка «Детальніше»

У нижньому правому куті інтерфейсу головної сторінки розміщено кнопку, яка забезпечує перехід на сторінку «Contact» (contact.html). Ця сторінка призначена для зворотного зв'язку з користувачами. Вигляд сторінки contact показано на рисунку 3.10.

Рисунок 3.10 – Сторінка з формою контакту

На сторінці «Contact» користувачі можуть заповнити форму, вказавши своє ім'я, електронну пошту та описавши проблему або запитання. Це забезпечує канал для комунікації та підтримки користувачів системи. Код для даної сторінки описано в додатку 3.

Ця реалізація клієнтської частини забезпечує інтуїтивно зрозумілий та інтерактивний інтерфейс для моніторингу екосистем парків та скверів, а також розширює можливості для отримання детальної інформації та зворотного зв'язку. Лістинг style.css з підключеними стилями для всіх сторінок подано у додатку Е.

3.4 Висновки до розділу 3

Розроблений вебсайт успішно демонструє основні функціональні можливості, закладені на етапі проєктування:

- Відображення даних у реальному часі: На головній сторінці сайту чітко видно карту Leaflet, на якій розміщені маркери, що позначають парки. Кожен маркер відображає актуальні показники температури та вологості, отримані від датчиків. Це підтверджує виконання вимоги щодо відображення даних у реальному часі.

- Геопросторова візуалізація: Карта ефективно відображає географічне розташування точок моніторингу, що є критично важливим для розуміння просторового розподілу екологічних параметрів.

- Взаємодія з користувачем: При натисканні на маркер на карті, з'являється спливаюче вікно, що містить детальні показники температури та вологості для конкретного датчика, а також час останнього оновлення. Це забезпечує інтуїтивну взаємодію та доступ до детальної інформації.

- Механізми оновлення даних: Система демонструє динамічне оновлення даних на карті, що свідчить про успішну реалізацію періодичних запитів до серверної частини та оновлення інтерфейсу без перезавантаження сторінки.

- Розширені сторінки: Реалізація сторінок «Детальніше» та «Contact» розширює функціональність вебсайту, надаючи користувачам доступ до історичних даних та можливість зворотного зв'язку.

Проектування системи з використанням модульних компонентів (ESP01S, MySQL, Leaflet) забезпечує її масштабованість. Додавання нових точок моніторингу вимагатиме лише розгортання додаткових апаратних вузлів та оновлення конфігурації на сервері. Архітектура бази даних також дозволяє легко додавати нові записи без значних змін у структурі.

Таким чином, практична реалізація вебсайту для моніторингу екосистем парків та скверів демонструє успішне застосування технологій IoT для вирішення актуальних екологічних завдань. Отримані результати підтверджують ефективність обраних рішень та відкривають широкі перспективи для подальшого вдосконалення та розширення системи.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Природне середовище та його забруднення

Природне середовище – це складна сукупність природних компонентів, таких як повітря, вода, ґрунт, флора, фауна, а також кліматичні та географічні умови, що існують на Землі [54]. Воно є основою для життя всіх організмів, включаючи людину, забезпечуючи нас життєво важливими ресурсами, від кисню та прісної води до родючих ґрунтів і сировини. Взаємодія цих компонентів формує збалансовані екосистеми, де кожен елемент відіграє свою роль у підтримці життєвого циклу планети. Саме природне середовище визначає якість нашого життя, впливаючи на здоров'я, добробут та економічний розвиток суспільства.

Забруднення природного середовища – це внесення в нього або виникнення в ньому нових, зазвичай нехарактерних для нього фізичних, хімічних чи біологічних агентів, або перевищення їхнього природного рівня, що призводить до негативних змін. Ці зміни можуть бути шкідливими для здоров'я людини, живих організмів та екосистем в цілому. Забруднення часто є прямим наслідком людської діяльності, зокрема промисловості, сільського господарства, транспорту та побуту [55].

Забруднення повітря є один з найпоширеніших видів забруднення, що впливає на глобальний клімат та здоров'я людини. Основними джерелами є викиди промислових підприємств (сірчистий газ, оксиди азоту, тверді частинки), вихлопні гази автомобілів (чадний газ, вуглеводні), спалювання викопного палива та лісові пожежі. Наслідками є смог, кислотні дощі, руйнування озонового шару та парниковий ефект, що призводить до зміни клімату. Вдихання забрудненого повітря спричиняє респіраторні захворювання, серцево-судинні проблеми та інші серйозні недуги.

Водні ресурси забруднюються стічними водами промислових підприємств, неочищеними побутовими стоками, змиванням пестицидів та добрив з

сільськогосподарських угідь, а також розливами нафти. Це призводить до загибелі водних мешканців, цвітіння водойм (евтрофікації), поширення небезпечних бактерій та вірусів, роблячи воду непридатною для пиття та використання. Дефіцит чистої питної води стає все гострішою глобальною проблемою.

Родючий ґрунт – це основа сільського господарства та джерело продовольства. Він забруднюється важкими металами з промислових відходів, пестицидами, інсектицидами, гербіцидами та надмірним використанням мінеральних добрив. Крім того, значний вплив має накопичення побутових та промислових відходів на сміттєзвалищах. Забруднені ґрунти втрачають свою родючість, а токсичні речовини можуть потрапляти в харчовий ланцюг, стаючи загрозою для здоров'я людини та тварин.

Хоча і не завжди очевидне, шумове забруднення, спричинене транспортом (автомобілі, літаки, потяги), промисловим обладнанням та будівельними роботами, має значний вплив на живі організми. Воно може призводити до стресу, порушень сну, зниження продуктивності, а також негативно впливати на поведінку та міграцію тварин.

Надмірне або неправильно спрямоване штучне освітлення вночі порушує природні цикли живої природи, впливаючи на поведінку нічних тварин (комах, птахів), їхню міграцію та розмноження. Це також перешкоджає спостереженню за зоряним небом.

Наслідки забруднення природного середовища є масштабними та взаємопов'язаними. Вони включають втрату біорізноманіття, зміну клімату, опустелювання, деградацію ґрунтів, нестачу прісної води та погіршення стану здоров'я населення. Це глобальні виклики, які вимагають скоординованих дій.

Для боротьби із забрудненням необхідний комплексний підхід. Він складається із впровадження чистих технологій на виробництві, використання відновлюваних джерел енергії, розвиток громадського транспорту та електромобілів, ефективне очищення стічних вод та переробку відходів.

Важливим є також раціональне використання природних ресурсів, збереження лісів та водних об'єктів, а також підвищення екологічної свідомості населення.

4.2 Можливість виникнення статичної електрики та заходи боротьби з нею.

Статична електрика є поширеним фізичним явищем, що виникає внаслідок накопичення електричних зарядів на поверхні об'єктів. Це відбувається, коли відбувається контакт та подальше розділення двох різних матеріалів, особливо діелектриків (ізоляторів) [56]. В результаті тертя (трибоелектричний ефект) електрони можуть переміщатися з одного матеріалу на інший, залишаючи один об'єкт позитивно зарядженим (з дефіцитом електронів), а інший – негативно зарядженим (з надлишком електронів). Накопичений заряд залишається на поверхні до тих пір, поки не знайде шлях для розряду, часто у вигляді іскри або удару струмом, якщо об'єкт контактує з провідником або заземленою поверхнею.

Виникнення статичної електрики зумовлене кількома факторами. Найпоширеніша причина – тертя та розділення матеріалів. Наприклад, це відбувається, коли одяг треться об тіло, людина йде по килиму, або коли рідини та сипучі матеріали переміщуються по трубопроводах та ємностях. Чим більша площа контакту та швидкість розділення, тим значніший заряд може накопичитися.

Тип матеріалів також відіграє ключову роль, оскільки деякі матеріали більш схильні до накопичення або віддачі електронів. Матеріали розташовані у трибоелектричному ряду: наприклад, вовна, волосся та скло зазвичай віддають електрони, стаючи позитивно зарядженими, тоді як поліестер, полівінілхлорид та тефлон приймають електрони, набуваючи негативного заряду [57].

Низька відносна вологість повітря значно сприяє накопиченню статичної електрики. Волога в повітрі діє як провідник, допомагаючи розсіювати накопичений заряд. У сухому середовищі, наприклад, взимку при включеному опаленні, заряди не можуть швидко розсіятися і накопичуються.

Також, рух персоналу та обладнання створює умови для генерації статичного заряду. Переміщення людей у синтетичному одязі по певних поверхнях, рух транспортних засобів та робота конвеєрів – все це призводить до тертя та накопичення заряду. Аналогічно, пневматичне транспортування та потоки рідин через труби та шланги можуть викликати значне накопичення заряду через тертя об стінки, особливо якщо транспортуються діелектричні матеріали.

Накопичення статичного заряду може мати як незначні, так і дуже серйозні наслідки. Найбільш очевидний прояв – дискомфорт від неприємних відчуттів іскрового розряду при дотику до металевих предметів.

Однак значно серйознішою загрозою є пошкодження електронних компонентів. Електростатичний розряд (Electrostatic Discharge, ESD) – одна з головних небезпек для чутливої електроніки. Навіть невеликий розряд, який людина може не відчувати, генерує напругу в тисячі вольт, що достатньо для виведення з ладу мікросхем, сенсорів, процесорів та інших компонентів IoT-пристроїв. Це може призвести до прихованих пошкоджень, які проявляться не відразу, або до миттєвого руйнування пристрою [58].

У середовищах, де присутні легкозаймисті пари, газу або пил, таких як промислові об'єкти, хімічні виробництва або заправки, іскра статичного розряду може спричинити вибухи та пожежі. Це найсерйозніша загроза, пов'язана зі статичною електрикою, і важливо враховувати відповідні стандарти безпеки.

Боротьба зі статичною електрикою полягає у застосуванні комплексних заходів, спрямованих на запобігання її виникненню та нейтралізацію накопичених зарядів. Для забезпечення безпеки та правил для уникнення інцидентів пов'язаних з статичною електрикою, застосовуються відповідні національні та міжнародні стандарти. В Україні, наприклад, робота в умовах можливої дії статичної електрики є затвердженою згідно ДСТУ 7302:2013.

Найефективніший та найпоширеніший метод – заземлення, що передбачає створення провідного шляху для відведення електричного заряду в землю. Заземлюють металеві частини обладнання, трубопроводи, резервуари, а також

персонал за допомогою антистатичних браслетів, взуття та килимків. Правильне заземлення забезпечує постійний відтік зарядів, не даючи їм накопичуватися [54].

Застосування антистатичних матеріалів також є ефективним заходом. Це можуть бути провідні або антистатичні підлогові покриття, спецодяг, тара для перевезення електронних компонентів, пакувальні матеріали та покриття для робочих поверхонь. Такі матеріали або не генерують заряд, або дозволяють йому швидко розсіюватися.

У випадках, коли заземлення неможливе або недостатньо, застосовують іонізацію повітря. Іонізатори повітря генерують позитивні та негативні іони, які нейтралізують статичні заряди на поверхнях об'єктів та в повітрі. Деякі матеріали можуть бути оброблені антистатичними добавками або покриттями, які збільшують їхню поверхневу провідність. Це особливо актуально для текстилю, пластмас та фарб.

Нарешті, у промислових умовах важливим є контроль технологічного процесу. Це включає зниження швидкості переміщення матеріалів, використання конфігурацій, що мінімізують тертя, та підтримання оптимальної температури.

Датчики вологості ґрунту, температури повітря, якості повітря, рівня освітленості, а також мікроконтролери та модулі зв'язку (Wi-Fi, LoRa, LTE) – всі ці компоненти, що є серцем IoT-пристроїв, надзвичайно чутливі до статичної електрики. Навіть невеликий розряд під час виробництва, транспортування, встановлення або обслуговування цих пристроїв може пошкодити їх. Це може призвести до виходу з ладу окремих датчиків, спотворення даних або повного виходу з ладу всього пристрою. [59] Таким чином, для забезпечення надійності системи моніторингу, розробники та інсталятори повинні дотримуватися строгих протоколів ESD-захисту, використовуючи антистатичні браслети, килимки та спеціальну упаковку.

Крім того, умови експлуатації в парках та скверах також можуть сприяти накопиченню статичної електрики. Хоча це не промислове середовище, але

переміщення персоналу, що обслуговує пристрої, в сухому одязі по сухій траві або асфальту може генерувати заряд. Самі IoT-пристрої, розташовані в пластикових корпусах (які є діелектриками) і піддаючись впливу вітру та пилу, також можуть накопичувати заряд через трибоелектричний ефект. Хоча ризик вибуху в такому середовищі мінімальний, ризик пошкодження електроніки залишається значним. Отже, при проектуванні корпусів та виборі матеріалів для зовнішніх IoT-пристроїв варто враховувати їхні антистатичні властивості або передбачити можливість розсіювання заряду, наприклад, шляхом заземлення металевих елементів, якщо вони є. Надійність збору даних, що є основою для функціонування вебсайту моніторингу, безпосередньо залежить від працездатності фізичних пристроїв, на яку статична електрика може мати руйнівний вплив.

4.3 Висновок до четвертого розділу

Цей розділ зосередився на двох важливих аспектах безпеки: можливості виникнення статичної електрики та заходах боротьби з нею, а також на проблемі забруднення природного середовища. Обидві теми підкреслюють глибокий зв'язок між діяльністю людини, технологіями та необхідністю забезпечення безпечного і здорового життя.

Ми з'ясували, що статична електрика – це не просто незручність, а потенційна загроза для чутливого електронного обладнання, зокрема IoT-пристроїв, і може призводити до небезпечних ситуацій. Боротьба з нею вимагає комплексних заходів, таких як заземлення, контроль вологості та використання антистатичних матеріалів, що є критично важливим для надійності систем моніторингу екосистем.

Водночас, забруднення природного середовища (повітря, води, ґрунту) є наслідком людської діяльності і створює глобальні виклики, що впливають на зміну клімату, біорізноманіття та здоров'я. Вирішення цих проблем потребує

впровадження сталих технологій, раціонального використання ресурсів та підвищення екологічної свідомості.

ВИСНОВКИ

У даній кваліфікаційній роботі було успішно вирішено завдання розробки вебсайту для моніторингу екосистем парків та скверів за допомогою технологій Інтернету речей. Досягнуті результати підтверджують актуальність та практичну цінність обраної теми, демонструючи потенціал IoT для підвищення ефективності екологічного моніторингу та управління міськими зеленими зонами.

В першому розділі кваліфікаційної роботи подано концепцію «розумного міста» та її основні поняття, що закладає теоретичну основу для розуміння ролі технологій у сучасному міському середовищі. Розглянуто технології «розумних міст» та міжнародні стандарти і рамки для їх розвитку, що дозволило визначити ключові напрямки та найкращі практики у цій сфері. Висвітлено обмеження традиційних методів моніторингу екосистем, підкреслюючи їхню недостатню ефективність у зборі та обробці великих обсягів динамічних екологічних даних. Також, висвітлено успішні впровадження IoT у моніторингу довкілля «розумних міст», демонструючи практичне застосування концепції.

В другому розділі кваліфікаційної роботи досліджено актуальність та теоретичні основи проєктування системи моніторингу, включаючи сучасний стан та виклики екологічного моніторингу міських зелених зон, концепцію IoT та її застосування, а також переваги та виклики впровадження IoT-систем для моніторингу парків та скверів. Це дозволило сформулювати комплексне розуміння контексту проєкту. Обґрунтовано методологію проєктування IoT-системи моніторингу екосистем, включаючи загальні підходи до проєктування, архітектуру системи, вибір та обґрунтування компонентів (датчик DHT11 та модуль ESP01S), а також проєктування бази даних MySQL. Це забезпечило системний підхід до розробки. Сформовано функціональні вимоги до вебсайту, принципи дизайну та інтерфейсу користувача, а також проєктування відображення даних на карті Leaflet та механізмів оновлення та взаємодії з

даними. Це визначило ключові аспекти користувацького досвіду та функціональності.

В третьому розділі кваліфікаційної роботи: – Розроблено апаратну частину системи моніторингу, що включає пряме підключення датчика DHT11 до Wi-Fi модуля ESP01S та програмне забезпечення модуля ESP01S, що дозволило створити компактний та ефективний вузол збору даних. Описано реалізацію програмної частини та бази даних, зокрема налаштування серверного середовища на хостингу Awardspace з phpMyAdmin та реалізацію серверного скрипту (config.php, data.php) для прийому даних, що забезпечило надійне зберігання та обробку інформації. Спроектowano клієнтську частину вебсайту, включаючи структуру головної сторінки, візуалізацію даних на карті Leaflet, а також механізми оновлення даних та взаємодії з користувачем, зокрема переходи на сторінки «Детальніше» та «Contact». Це забезпечило інтуїтивно зрозумілий та функціональний інтерфейс для кінцевого користувача. Протестовано ефективність та надійність системи, включаючи точність даних, продуктивність та виявлені обмеження, а також визначено перспективи та подальший розвиток, що підтверджує життєздатність та потенціал проєкту.

У розділі «Безпека життєдіяльності, основи охорони праці» висвітлено загальні питання безпеки життєдіяльності та основи охорони праці, а також можливості виникнення статичної електрики та заходи боротьби з нею, що є важливим для забезпечення безпеки під час розробки та експлуатації системи.

Дана робота демонструє успішне поєднання теоретичних знань та практичних навичок у створенні функціональної IoT-системи для екологічного моніторингу. Розроблений вебсайт є ефективним інструментом для візуалізації даних про мікроклімат парків та скверів, сприяючи підвищенню обізнаності громадськості та підтримці прийняття рішень щодо збереження міських зелених зон. Отримані результати відкривають широкі перспективи для подальшого вдосконалення системи, включаючи інтеграцію нових типів датчиків, розширену аналітику та впровадження прогнозних моделей, що сприятиме сталому розвитку міського середовища.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Sala E. Smart city standards: a collaboration challenge?. *City diplomacy lab*. URL: <https://www.citydiplomacylab.net/smart-city-standards-a-challenge-for-cooperation%E2%82%AC%91>.
- 2 Sereda, M. K. P. Moving to Smart Cities Through the Standard Indicators ISO 37120. *Multidisciplinary Aspects of Production Engineering – MAPE*. 2020. Vol. 3, no. 1. C. 618. URL: https://www.researchgate.net/publication/344030787_Moving_to_Smart_Cities_Through_the_Standard_Indicators_ISO_37120 (дата звернення: 24.06.2025).
- 3 The 5 pillars of building a smarter city. *Blog C2 ITSM*. URL: <https://blog.c2-itsm.com/how-to-build-smart-city> (дата звернення: 24.06.2025).
- 4 Ultimate Guide: Environmental Monitoring in Smart Cities. *Number Analytics Blog*. URL: <https://www.numberanalytics.com/blog/ultimate-guide-environmental-monitoring-smart-cities> (дата звернення: 24.06.2025).
- 5 Why environmental monitoring sensors are critical for smart cities. *Ignitec*. URL: <https://www.ignitec.com/insights/why-environmental-monitoring-sensors-are-critical-for-smart-cities/> (дата звернення: 24.06.2025).
- 6 The Internet of Things (IoT). *IBM*. URL: <https://www.ibm.com/topics/internet-of-things> (дата звернення: 24.06.2025).
- 7 Sereda, O. S. Integracia prirodoorientovinih risen u miski ekosistemi v umovah strimkoi urbanizacii ta zmin klimatu. 2024. URL: https://www.researchgate.net/publication/382308333_Integracia_prirodoorientovinih_risen_u_miski_ekosistemi_v_umovah_strimkoi_urbanizacii_ta_zmin_klimatu (дата звернення: 24.06.2025).
- 8 Jose, G. V. H., Karina, P., Patricia, H. Urban Green Spaces as a Component of an Ecosystem. *Int J Environ Sci Nat Res*. 2018. Vol. 8, no. 1. C. 555730. URL: https://www.researchgate.net/publication/323692816_Urban_Green_Spaces_as_a_Component_of_an_Ecosystem (дата звернення: 24.06.2025).

- 9 HashStudioz Technologies. IoT in Environmental Monitoring: Smart Tech for Sustainability. *HashStudioz Technologies*. URL: <https://www.hashstudioz.com/blog/iot-in-environmental-monitoring/> (дата звернення: 24.06.2025).
- 10 IoT and Environmental Monitoring with Sensor Networks. *IoT For All*. URL: <https://www.iotforall.com/iot-and-environmental-monitoring-with-sensor-networks> (дата звернення: 24.06.2025).
- 11 Urban Park Biodiversity Monitoring Technologies. *Prism Sustainability Directory*. URL: <https://prism.sustainability-directory.com/scenario/urban-park-biodiversity-monitoring-technologies/> (дата звернення: 24.06.2025).
- 12 Evaluation of Cultural Ecosystem Services in the Urban Parks of Macau from a Cultural Heritage Perspective. *Applied Sciences*. 2024. Vol. 15, no. 7. С. 3946. URL: <https://www.mdpi.com/2076-3417/15/7/3946> (дата звернення: 24.06.2025).
- 13 Industrial-IA. The Strengths and Weaknesses of Visual Inspection. *Industrial-IA*. 2013. URL: <https://industrial-ia.com/the-strengths-and-weaknesses-of-visual-inspection/> (дата звернення: 24.06.2025).
- 14 TXNDT. Visual Inspection NDT techniques rely on the aided and unaided eye. *TXNDT*. URL: <https://www.txndt.com/safety-section/visual-inspection> (дата звернення: 24.06.2025).
- 15 Temperature and Humidity Monitor using Whatsapp and ESP32. *Arduino Project Hub*. URL: <https://projecthub.arduino.cc/sonutest23/temperature-and-humidity-monitor-using-whatsapp-and-esp32-15e844> (дата звернення: 24.06.2025).
- 16 The IoT Academy. IoT Architecture: Detailed Explanation of the 4 IoT Layers. *The IoT Academy Blogs*. URL: <https://www.theiotacademy.co/blog/iot-architecture/> (дата звернення: 24.06.2025).
- 17 GeeksforGeeks. Architecture of Internet of Things (IoT). *GeeksforGeeks*. 2025. URL: <https://www.geeksforgeeks.org/architecture-of-internet-of-things-iot/> (дата звернення: 24.06.2025).

- 18 Lily-osp. Environmental-Monitoring-System. *GitHub*. URL: <https://github.com/lily-osp/Environmental-Monitoring-System/> (дата звернення: 24.06.2025).
- 19 Arduino. Arduino.cc. *Arduino*. URL: <https://www.arduino.cc/> (дата звернення: 24.06.2025).
- 20 Cavli Wireless. How IoT in Smart Cities Powers Urban Transformation: Architecture, Benefits, and Key Solutions. *Cavli Wireless*. URL: <https://www.cavliwireless.com/blog/nerdiest-of-things/iot-smart-city-solutions> (дата звернення: 24.06.2025).
- 21 Milesight. Reliable and Cutting-Edge Smart City Sensors & Solutions. *Milesight*. URL: <https://www.milesight.com/solution/smart-city> (дата звернення: 24.06.2025).
- 22 Prism Sustainability Directory. Sensor Network Integration with Green Infrastructure. *Prism Sustainability Directory*. URL: <https://prism.sustainability-directory.com/scenario/sensor-network-integration-with-green-infrastructure/> (дата звернення: 24.06.2025).
- 23 SaveEcoBot. Станція моніторингу якості повітря SaveEcoSensor 3.0. *SaveEcoBot*. URL: <https://www.saveecobot.com/saveecosensor-3.0> (дата звернення: 24.06.2025).
- 24 Alcaraz, D., Cabello, J., Paruelo, J. M., Delibes, M. Use of Descriptors of Ecosystem Functioning for Monitoring a National Park Network: A Remote Sensing Approach. *Environmental Management*. 2009. Vol. 43. С. 38–48. URL: <https://www.agro.uba.ar/users/paruelo/Publicaciones/2009/Alcaraz%20D.,%20Cabello%20J.,%20Paruelo%20J.M.,%20Delibes%20M.%20Use%20of%20Descriptors%20of%20Ecosystem%20Functioning%20for%20Monitoring%20a%20National%20Park%20Network%20A%20Remote%20Sensing%20Approach.pdf>
- 25 Методологія розробки комунікаційної системи розподіленої мережі зі збору екологічних даних. URL: <https://dglib.nubip.edu.ua/items/ade442ad-9fcf-4e19-b39a-5cc3d562b61e> (дата звернення: 24.06.2025).
- 26 Applications of the Internet of Things (IoT) in Real-Time Monitoring of Contaminants in the Air, Water, and Soil // *IoT*. 2022. Vol. 27, No. 1. Art. 26.

[Електронний ресурс]. URL: <https://www.mdpi.com/2673-4591/27/1/26> (дата звернення: 24.06.2025).

27 Сучасний розвиток інформаційних технологій і зростання інтересу до екологічного моніторингу сприяють появі нових інструментів. [Електронний ресурс]. URL: <https://dglb.nubip.edu.ua/items/ade442ad-9fcf-4e19-b39a-5cc3d562b61e> (дата звернення: 24.06.2025).

28 Architecture of IoT. GeeksforGeeks. [Електронний ресурс]. URL: <https://www.geeksforgeeks.org/architecture-of-internet-of-things-iot/> (дата звернення: 24.06.2025).

29 ESP-01 Pinout, Programming, Datasheet, and Comparison. Xecor. [Електронний ресурс]. URL: <https://www.xecor.com/blog/esp-01-pinout-programming-datasheet-and-comparison> (дата звернення: 24.06.2025).

30 ESP-01S ESP8266 WiFi Module. The Pi Hut. [Електронний ресурс]. URL: <https://thepihut.com/products/esp-01-wifi-serial-transceiver-module-esp8266> (дата звернення: 24.06.2025).

31 DHT11 Sensor Guide. ElectronicWings. [Електронний ресурс]. URL: <https://www.electronicwings.com/sensors-modules/dht11> (дата звернення: 24.06.2025).

32 DHT11 Sensor Definition, Working and Applications. Elprocus. [Електронний ресурс]. URL: <https://www.elprocus.com/a-brief-on-dht11-sensor/> (дата звернення: 24.06.2025).

33 M, Kiran H. Using MySQL for Storing and Analysing Sensor Data in IoT Systems // Journal of Research and Review in Digital Marketing and Communications. 2025. Vol. 2, No. 1. P. 72-78. [Електронний ресурс]. URL: <https://zenodo.org/records/14869969/files/Using%20MySQL%20for%20Storing%20and%20Analysing%20Sensor%20Data%20in%20IoT%20Systems%20.pdf?download=1> (дата звернення: 24.06.2025).

34 Leaflet - a JavaScript library for interactive maps. [Електронний ресурс]. URL: <https://leafletjs.com/> (дата звернення: 24.06.2025).

- 35 Plugins - Leaflet - a JavaScript library for interactive maps. [Электронный ресурс]. URL: <https://leafletjs.com/plugins.html> (дата звернения: 24.06.2025).
- 36 Unlocking Map Data Visualization Basics. Pagon.ai. [Электронный ресурс]. URL: <https://www.pagon.ai/blog/map-data-visualization> (дата звернения: 24.06.2025).
- 37 Internet of Things Testing & IoT Testing Framework. Testfort. [Электронный ресурс]. URL: <https://testfort.com/blog/iot-testing-how-it-works-why-you-cant-do-without-it> (дата звернения: 24.06.2025).
- 38 What is IoT Device Testing | How To Perform It? Testingmind. [Электронный ресурс]. URL: <https://www.testingmind.com/what-is-iot-device-testing-how-to-perform-it/> (дата звернения: 24.06.2025).
- 39 Mastering IoT Device Testing: Essential Strategies & Tools for Success. Luxe Quality. [Электронный ресурс]. URL: <https://luxequality.com/blog/mastering-iot-testing-strategies-tools/> (дата звернения: 24.06.2025).
- 40 Ensuring Reliable and Accurate IoT Data: Challenges, Solutions & Future Trends. Encardio. [Электронный ресурс]. URL: <https://www.encardio.com/blog/ensuring-reliable-accurate-iot-data-challenges-solutions> (дата звернения: 24.06.2025).
- 41 Tech for the Planet: IoT Innovations in Environmental Monitoring and Disaster Management. Wevolver. [Электронный ресурс]. URL: <https://www.wevolver.com/article/tech-for-the-planet-iot-innovations-in-environmental-monitoring-and-disaster-management> (дата звернения: 24.06.2025).
- 42 The Rise of Smart Parks: A New Era for Urban Green Spaces. Creative Systems. [Электронный ресурс]. URL: <https://www.creativesystems.com/the-rise-of-smart-parks-a-new-era-for-urban-green-spaces/> (дата звернения: 24.06.2025).
- 43 Environmental monitoring is crucial for maintaining a sustainable urban environment. Cavli Wireless. [Электронный ресурс]. URL: <https://www.cavliwireless.com/blog/nerdiest-of-things/iot-smart-city-solutions> (дата звернения: 24.06.2025).

44 DHT11 Sensor Guide. ElectronicWings. [Электронный ресурс]. URL: <https://www.electronicwings.com/sensors-modules/dht11> (дата звернения: 24.06.2025).

45 DHT11 Sensor Definition, Working and Applications. Elprocus. [Электронный ресурс]. URL: <https://www.elprocus.com/a-brief-on-dht11-sensor/> (дата звернения: 24.06.2025).

46 ESP-01 Pinout, Programming, Datasheet, and Comparison. Xecor. [Электронный ресурс]. URL: <https://www.xecor.com/blog/esp-01-pinout-programming-datasheet-and-comparison> (дата звернения: 24.06.2025).

47 M, Kiran H. Using MySQL for Storing and Analysing Sensor Data in IoT Systems // Journal of Research and Review in Digital Marketing and Communications. 2025. Vol. 2, No. 1. P. 72-78. [Электронный ресурс]. URL:(<https://zenodo.org/records/14869969/files/Using%20MySQL%20for%20Storing%20and%20Analysing%20Sensor%20Data%20in%20IoT%20Systems%20.pdf?download=1>) (дата звернения: 24.06.2025).

48 Leaflet - a JavaScript library for interactive maps. [Электронный ресурс]. URL: <https://leafletjs.com/> (дата звернения: 24.06.2025).

49 Plugins - Leaflet - a JavaScript library for interactive maps. [Электронный ресурс]. URL: <https://leafletjs.com/plugins.html> (дата звернения: 24.06.2025).

50 Unlocking Map Data Visualization Basics. Pagon.ai. [Электронный ресурс]. URL: <https://www.pagon.ai/blog/map-data-visualization> (дата звернения: 24.06.2025).

51 Applications of the Internet of Things (IoT) in Real-Time Monitoring of Contaminants in the Air, Water, and Soil // IoT. 2022. Vol. 27, No. 1. Art. 26. [Электронный ресурс]. URL: <https://www.mdpi.com/2673-4591/27/1/26> (дата звернения: 24.06.2025).

52 Lezcano Sirre, A. IoT Environmental Monitoring System Using Arduino and NODE MCU ESP8266. 2023. [Электронный ресурс]. URL:(https://upcommons.upc.edu/bitstream/2117/405973/2/TFG_Alejandro_Lezcano_Sirre_IoT.pdf) (дата звернения: 24.06.2025).

53 Ensuring Reliable and Accurate IoT Data: Challenges, Solutions & Future Trends. Encardio. [Електронний ресурс]. URL: <https://www.encardio.com/blog/ensuring-reliable-accurate-iot-data-challenges-solutions> (дата звернення: 24.06.2025).

54 Будстандарт. ДСТУ EN 60079-10-1:2018. Вибухонебезпечні середовища. Частина 10-1. Класифікація зон. Вибухонебезпечні газові середовища. Київ, 2018. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=59393 (дата звернення: 24.06.2025).

55 Міжнародна електротехнічна комісія. IEC 61340-5-1: Electrostatics – Part 5-1: Protection of electronic devices from electrostatic phenomena – General requirements. Geneva, 2016. (Опис загальних вимог до захисту електронних пристроїв від електростатичних явищ).

56 Офіційний сайт Світум. Статична електрика та способи боротьби з ним. URL: <https://svetum.com.ua/ua/blog/sovety-pokupatelyam/staticheskoe-elektrichestvo-i-sposoby-borby-s-nim/> (дата звернення: 24.06.2025).

57 Петров А. В. Основи охорони праці: Навч. посібник. Київ: Знання, 2012. 382 с.

58 Тимошук М. О., Гончарук В. В. Промислова екологія та охорона навколишнього середовища: Підручник. Київ: Вища освіта, 2005. 544 с.

59 Українська Національна Енциклопедія. Забруднення навколишнього середовища. URL: <https://esu.com.ua/article-48356> (дата звернення: 24.06.2025).

60 Кунанець, Н. Е., Небесний, Р. М., & Мацюк, О. В. (2016). Особливості формування цілей соціальних та соціокомунікаційних складових у проектах “Розумних міст”. Вісник Національного університету Львівська політехніка. Серія: Інформаційні системи та мережі.

61 Kunanets, N. E., Nazaruk, M. V., Nebesnyi, R. M., & Pasichnyk, V. V. (2018). Information technology of personalized choice of profession in smart cities. Інформаційні технології і засоби навчання, 277–290.

ДОДАТКИ

Код модуля ESP01S з датчиком DHT11

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
const char* ssid = «*****»;
const char* password = «*****»;
const char* serverName =
«http://parksmonitoring.atwebpages.com/data.php»;
void setup() {
  Serial.begin(115200);
  dht.begin();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(«.»);
  }
  Serial.println(«WiFi connected»);
}
void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    float t = dht.readTemperature();
    float h = dht.readHumidity();
    if (!isnan(t) && !isnan(h)) {
      WiFiClient client;
      HTTPClient http;
      String url = String(serverName) +
«?device_id=espl&temperature=« + t + «&humidity=« + h;
      http.begin(client, url);
      int httpCode = http.GET();
      if (httpCode > 0) {
        Serial.printf(«HTTP GET... code: %d\n», httpCode);
      } else {
        Serial.printf(«GET failed, error: %s\n»,
http.errorToString(httpCode).c_str());
      }
      http.end();
    } else {
      Serial.println(«Failed to read from DHT sensor!»);
    }
  }
  delay(10000);
}
```

Код data.php для прийому показників в базу даних

```
<?php
include 'config.php';

if (isset($_GET['device_id']) && isset($_GET['temperature']) &&
isset($_GET['humidity'])) {
    $device_id = $_GET['device_id'];
    $temperature = $_GET['temperature'];
    $humidity = $_GET['humidity'];

    $sql = «INSERT INTO temperature_data (device_id, temperature,
humidity)
        VALUES (:device_id, :temperature, :humidity)»;

    $stmt = $pdo->prepare($sql);
    $stmt->bindParam(':device_id', $device_id);
    $stmt->bindParam(':temperature', $temperature);
    $stmt->bindParam(':humidity', $humidity);

    if ($stmt->execute()) {
        echo «OK: Data inserted successfully»;
    } else {
        echo «Error inserting data»;
    }
} else {
    echo «Missing parameters»;
}
?>
```

Сторінка index.html

```

<!DOCTYPE html>
<html lang=«en»>
<head>
  <meta charset=«UTF-8»>
  <meta name=«viewport» content=«width=device-width, initial-
scale=1.0»>
  <link rel=«icon» href=«img/thermometer.png»>
  <link rel=«stylesheet» href=«style.css»>
  <link rel=«preconnect» href=«https://fonts.googleapis.com»>
  <link rel=«preconnect» href=«https://fonts.gstatic.com»
crossorigin>
  <link
href=«https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght
@0,300..800;1,300..800&display=swap» rel=«stylesheet»>
  <link rel=«stylesheet»
href=«https://unpkg.com/leaflet@1.9.4/dist/leaflet.css»
  integrity=«sha256-
p4NxAoJBhIIN+hmNHzRCf9tD/miZyoHS5obTRR9BMY=«
  crossorigin=«</>
  <title>ParksMonitoring</title>
</head>
<body>
  <header>
    <div class=«logo»>
      <img src=«img/logo.png» alt=«ParkWeather»>
      <h1>Дізнайся погоду у парку твого міста</h1>
    </div>
    <nav>
      <a href=«index.html»>Карта</a>
    </nav>
  </header>
  <div id=«map»></div>
  <a href=«contact.html» class=«contact-button»>
    <img src=«img/chat-icon.png» alt=«Зв'язок»>
  </a>
  <script src=«https://unpkg.com/leaflet@1.9.4/dist/leaflet.js»
  integrity=«sha256-
20nQCchB9co0qIjJZRGuk2/Z9VM+kNiyxNV1lvTlZBo=«
  crossorigin=«></script>
  <script src=«leaflet2.js»></script>
</body>
</html>

```

Сторінка «Детальніше»

```
<!DOCTYPE html>
<html lang=«uk»>
<head>
  <meta charset=«UTF-8»>
  <meta name=«viewport» content=«width=device-width, initial-
scale=1.0»>
  <title>Інформація про локацію</title>
  <style>
    body {
      font-family: sans-serif;
      margin: 0;
      padding: 20px;
      background: #e0f4f7;
    }
    .container {
      background: #14708b;
      color: white;
      padding: 20px;
      border-radius: 12px;
      max-width: 800px;
      margin: auto;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
    }
    h1 {
      margin-top: 0;
      color: #ffffff;
      text-align: center;
    }
    .info {
      margin-bottom: 20px;
      text-align: center;
      font-size: 1.1em;
    }
    .info p {
      margin: 5px 0;
      text-align: left;
      padding-left: 20px;
    }
    #video {
      margin-bottom: 20px;
      border-radius: 8px;
      overflow: hidden;
    }
    iframe, img {
      width: 100%;
      height: auto;
      display: block;
      border: none;
```

```

        border-radius: 8px;
    }
    canvas {
        background: rgba(255, 255, 255, 0.9);
        border-radius: 8px;
        padding: 15px;
        box-sizing: border-box;
        width: 100% !important;
        height: 450px !important;
        margin-top: 15px;
    }
    h2 {
        text-align: center;
        color: #ffffff;
        margin-top: 25px;
        margin-bottom: 15px;
    }
</style>
</head>
<body>
    <div class=«container»>
        <h1 id=«name»>Завантаження...</h1>
        <div class=«info»>
            <p>Температура: <strong
id=«temp»>Завантаження...</strong></p>
            <p>Вологість: <strong
id=«humidity»>Завантаження...</strong></p>
            <p>Оновлено: <strong
id=«timestamp»>Завантаження...</strong></p>
        </div>

        <div id=«video»>
            <img id=«camera» src=«» alt=«Камера»>
        </div>

        <h2>Графік температури</h2>
        <canvas id=«tempChart» width=«400» height=«300»></canvas>
    </div>

    <script src=«https://cdn.jsdelivr.net/npm/chart.js»></script>
    <script>
        const params = new URLSearchParams(window.location.search);
        const id = params.get('id');

        const locations = {
            sopilche: {
                name: «Алея Сопільче»,
                camera: «locations/Sopilche.png»
            },
            ostrivChaika: {
                name: «Острів Чайка»,
                camera: «locations/prychal-cropped.png»
            },

```

```

        hodynnyk: {
            name: «Годинник»,
            camera: «locations/default.png»
        },
        parkSlavy: {
            name: «Парк Слави»,
            camera: «locations/default.png»
        },
        parkVidrodzhennya: {
            name: «Парк Відродження»,
            camera: «locations/default.png»
        }
    };

    const selectedLocation = locations[id];

    if (!selectedLocation) {
        document.body.innerHTML = '<h2 style=«text-align:center; color:red»>Локацію не знайдено</h2>';
    } else {
        document.getElementById(«name»).textContent =
selectedLocation.name;
        document.getElementById(«camera»).src =
selectedLocation.camera;

        let tempChart;

        async function updateLocationData() {
            try {
                const response = await fetch('/get_sensor_data.php');
                if (!response.ok) {
                    throw new Error(`HTTP error! status:
${response.status}`);
                }
                const sensorDataArray = await response.json();

                if (sensorDataArray && sensorDataArray.length > 0) {
                    const latestData = sensorDataArray[0];
                    const temp =
parseFloat(latestData.temperature).toFixed(1);
                    const hum =
parseFloat(latestData.humidity).toFixed(1);
                    const time = new
Date(latestData.timestamp).toLocaleString();

                    document.getElementById(«temp»).textContent =
` ${temp} °C `;
                    document.getElementById(«humidity»).textContent =
` ${hum} % `;
                    document.getElementById(«timestamp»).textContent =
time;

                    // --- Зміни для графіка ---

```

```

        const graphLabels = [];
        const graphData = [];
        let minTemp = Infinity;
        let maxTemp = -Infinity;

        const recordsToDisplay =
Math.min(sensorDataArray.length, 20); // Збільшено до 20 записів
для більшого діапазону
        for (let i = recordsToDisplay - 1; i >= 0; i--) {
            const record = sensorDataArray[i];
            const recordTime = new Date(record.timestamp);

graphLabels.push(`${recordTime.getHours()}:${String(recordTime.get
Minutes()).padStart(2, '0')}`);
            const currentTemp =
parseFloat(record.temperature);
            graphData.push(currentTemp);

            // Оновлюємо мінімальне та максимальне значення
температури
            if (currentTemp < minTemp) minTemp = currentTemp;
            if (currentTemp > maxTemp) maxTemp = currentTemp;
        }

        // Додаємо запас до мінімального та максимального
значення температури для осі Y
        const padding = 2; // Додаємо 2 градуси до діапазону
        const suggestedMin = Math.floor(minTemp - padding);
        const suggestedMax = Math.ceil(maxTemp + padding);

        if (tempChart) {
            tempChart.data.labels = graphLabels;
            tempChart.data.datasets[0].data = graphData;
            // Оновлюємо діапазон осі Y
            tempChart.options.scales.y.min = suggestedMin;
            tempChart.options.scales.y.max = suggestedMax;
            tempChart.update();
        } else {
            const ctx =
document.getElementById(«tempChart»).getContext(«2d»);
            tempChart = new Chart(ctx, {
                type: «line»,
                data: {
                    labels: graphLabels,
                    datasets: [{
                        label: «Температура (°C)»,
                        data: graphData,
                        borderColor: «#14708b»,
                        backgroundColor: «rgba(255,255,255,0.2)»,
                        tension: 0.3,
                        pointBackgroundColor: «#ffffff»,
                        pointBorderColor: «#14708b»,
                        pointRadius: 5,

```

```

        pointHoverRadius: 7,
    ]],
},
options: {
    responsive: true,
    maintainAspectRatio: false,
    scales: {
        x: {
            grid: {
                color: 'gray'
            },
            ticks: {
                color: 'black'
            }
        },
        y: {
            beginAtZero: false,
            min: suggestedMin, // Встановлюємо
динамічний мінімум
            max: suggestedMax, // Встановлюємо
динамічний максимум
            grid: {
                color: 'gray'
            },
            ticks: {
                color: 'black',
                align: 'start' // Вирівнювання міток по
лівому краю
            }
        }
    },
    plugins: {
        legend: {
            labels: {
                color: 'black'
            }
        },
        tooltip: {
            callbacks: {
                label: function(context) {
                    return `Температура:
${context.raw}°C`;
                }
            }
        }
    }
});
}

console.log('Дані локації та графік оновлено.');
```

```

} else {
```



```

        document.getElementById(«temp»).textContent = «Дані
відсутні»;
        document.getElementById(«humidity»).textContent =
«Дані відсутні»;
        document.getElementById(«timestamp»).textContent =
«Дані відсутні»;
        console.warn('Дані датчика відсутні або некоректні
(масив порожній).');
        if (tempChart) {
            tempChart.destroy();
            tempChart = null;
            document.getElementById(«tempChart»).style.display
= 'none';
        }
    } catch (error) {
        console.error('Помилка при оновленні даних локації:',
error);
        document.getElementById(«temp»).textContent = «Помилка
завантаження»;
        document.getElementById(«humidity»).textContent =
«Помилка завантаження»;
        document.getElementById(«timestamp»).textContent =
«Помилка завантаження»;
        if (tempChart) {
            tempChart.destroy();
            tempChart = null;
            document.getElementById(«tempChart»).style.display
= 'none';
        }
    }
}

updateLocationData();
setInterval(updateLocationData, 10000);
}
</script>
</body>
</html>

```

Код стилів для сайту в файлі style.css

```
body {
    margin: 0;
    overflow-x: hidden;
    background-color: #ffffff;
    display: block;
    width: 100%;
}
header {
    background-color: #176B87;
    position: absolute;
    top: 20px;
    left: 20px;
    right: 20px;
    color: white;
    padding: 10px 20px;
    display: flex;
    align-items: center;
    justify-content: space-between;
    border-radius: 8px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    z-index: 1000;
}
.logo {
    display: flex;
    align-items: center;
    gap: 10px;
}
.logo img {
    height: 40px;
    border-radius: 999px;
}
nav a {
    font-family: 'Open Sans', sans-serif;
    font-size: 16px;
    color: white;
    text-decoration: none;
    margin-left: 20px;
    margin-right: 20px;
    transition: #7a7a7a 0.1s;
}
ul {
    list-style: none;
    padding: 0;
}
h1 {
    font-size: 25px;
    color: #fdf0f0;
    font-family: 'Open Sans', sans-serif;
```

```

        font-weight: 350;
    }

    #map{
        height:100vh;
        width: 100%;

    }
    .leaflet-top.leaflet-left {
        top: 120px !important;
        left: 10px !important;
    }
    .contact-button {
        position: absolute;
        bottom: 30px;
        right: 20px;
        background-color: white;
        border-radius: 50%;
        width: 60px;
        height: 60px;
        display: flex;
        justify-content: center;
        align-items: center;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
        cursor: pointer;
        z-index: 1000;
        transition: transform 0.2s;
    }
    .contact-button:hover {
        transform: scale(1.1);
    }
    .contact-button img {
        width: 30px;
        height: 30px;
    }
    @media (max-width: 768px) {
        header {
            top: 10px;
            left: 10px;
            right: 10px;
        }
        .header h1 {
            margin-left: 20px;
            font-size: 20px;
        }
        #map {
            height: 100vh;
            width: 100%;
        }
        .leaflet-top.leaflet-left {
            top: 520px !important;
            left: 10px !important;
        }
    }

```

```

    }
    @media (max-width: 480px) {
        header h1 {
            display: none;
        }
        #map {
            margin: 0;
            height: 100vh;
            width: 100%;
        }
    }
}

/* Contact page styles */

body#contact-page {
    margin: 0;
    font-family: 'Open Sans', sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
}

.contact-container {
    background-color: #14708b;
    padding: 40px 30px;
    border-radius: 10px;
    width: 90%;
    max-width: 500px;
    box-shadow: 0 0 15px rgba(0,0,0,0.2);
    color: white;
}

.contact-container h2 {
    margin-bottom: 5px;
    font-size: 24px;
}

.contact-container p {
    font-size: 18px;
    margin-bottom: 30px;
}

.contact-container input,
.contact-container textarea {
    width: 100%;
    padding: 12px;
    margin: 15px 0;
    border: none;
    border-bottom: 1px solid #ccc;
    background: transparent;
    color: white;
    font-size: 16px;
}

```

```
.contact-container input::placeholder,  
.contact-container textarea::placeholder {  
  color: #cce5ea;  
}  
  
.contact-container textarea {  
  resize: vertical;  
  min-height: 80px;  
}  
  
.contact-container button {  
  display: block;  
  width: 100%;  
  padding: 14px;  
  background-color: white;  
  color: #14708b;  
  border: none;  
  border-radius: 25px;  
  font-size: 16px;  
  cursor: pointer;  
  margin-top: 20px;  
  transition: background-color 0.3s;  
}  
  
.contact-container button:hover {  
  background-color: #cce5ea;  
}  
  
@media (max-width: 768px) {  
  .contact-container {  
    margin: 20px;  
  }  
}  
  
@media (max-width: 480px) {  
  .contact-container {  
    margin: 10px;  
  }  
}
```

Файл leaflet.js

```
var map = L.map('map').setView([49.55279353355259,  
25.602241945569187], 15);  
  
L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {  
    maxZoom: 19,  
    attribution: '&copy; <a  
href=«http://www.openstreetmap.org/copyright»>OpenStreetMap</a>'  
}).addTo(map);  
  
var parkSopilche = L.polygon([  
    [49.55103430098787, 25.573080426937995], [49.546411685057976,  
25.574759744798726],  
    [49.54651276324743, 25.577080288315933], [49.54528062922883,  
25.581323293172055],  
    [49.54355133105571, 25.584142106803583], [49.54571678293271,  
25.58758727194806],  
    [49.54681839664994, 25.588154650297877], [49.547566687184656,  
25.58551686334872],  
    [49.54926872566721, 25.58449446499284], [49.55043570190557,  
25.582472644354173],  
    [49.55104008077547, 25.574671400986677]  
]).addTo(map);  
  
var parkShevchenka = L.polygon([  
    [49.553211233636446, 25.58656445939148], [49.5516020680742,  
25.585048119527624],  
    [49.55190804803525, 25.58668468621352], [49.55196850419741,  
25.587757565882495],  
    [49.55293644768681, 25.587651934544784], [49.553141808704176,  
25.587140196206253],  
    [49.55374748068571, 25.587330420757997], [49.55386756812314,  
25.587507750973504],  
    [49.55521321202635, 25.590014970717494], [49.5566056608104,  
25.590682435480222],  
    [49.559393829000285, 25.587424945253517], [49.56040346832637,  
25.586849977686107],  
    [49.56219822688597, 25.586593355108768], [49.562855530436,  
25.58581107682577],  
    [49.559286568339466, 25.5857585727493], [49.557401314259366,  
25.585912475543],  
    [49.55714477036242, 25.584982625434368], [49.556614566284914,  
25.584901529574886],  
    [49.55640203199466, 25.586027674158096], [49.55593299913509,  
25.586081766434138],  
    [49.55446240378445, 25.586868816013606], [49.55356990093368,  
25.58693764146131]  
]).addTo(map);  
  
var squareShevchenka = L.polygon([
```

```

        [49.55107976174333, 25.594427275103243], [49.55112435910897,
25.595187597660136],
        [49.55304651996892, 25.595130667663817], [49.553044916610254,
25.59435964884776]
    ]).addTo(map);
var parkSlavy = L.polygon([
    [49.554590280605076, 25.610519720786996], [49.55510496608817,
25.610153441402538],
    [49.555858165121506, 25.608353957482056], [49.55446617604662,
25.605082918588938],
    [49.55391451187749, 25.605233385559387], [49.55358295921866,
25.605501422396173],
    [49.553024787591305, 25.606209450141325], [49.55353290972144,
25.60794499073396],
    [49.553913967463586, 25.60863031700181]
]).addTo(map);
var parkVidrodzhennya = L.polygon([
    [49.55518902201839, 25.639481783350284], [49.55405188595336,
25.638266089164503],
    [49.55344182086995, 25.637739485902642], [49.552604092720024,
25.63691564897065],
    [49.55167118194194, 25.635946241860573], [49.550770461548055,
25.635416935502477],
    [49.549621938354086, 25.635353100144357], [49.547888193443995,
25.635177656989573],
    [49.547606413408836, 25.634971749861425], [49.54757327177686,
25.634294836203875],
    [49.548816866919736, 25.62878005412197], [49.54909892248265,
25.627357788227666],
    [49.549416127418354, 25.628367436664455], [49.54989277772132,
25.62889277879097],
    [49.550585272244525, 25.629409690312606], [49.551408940465336,
25.62988029615962],
    [49.55263561018559, 25.6306992324039], [49.55306088859098,
25.63055065199402],
    [49.55390614852921, 25.63062599558637], [49.55419789700873,
25.631042707852238],
    [49.55664056423814, 25.629430598892338], [49.55690399109547,
25.629940966442653],
    [49.557658409442844, 25.630457556852534]
]).addTo(map);

var allMarkers = [];

var markerSopilche = L.marker([49.548307383031144,
25.581179612518955]).addTo(map)
    .bindPopup('Завантаження даних для Алеї Сопільче...');
allMarkers.push({ marker: markerSopilche, name: 'Алея Сопільче',
id: 'sopilche' });

var markerChaika = L.marker([49.5554209938319,
25.58757689474756]).addTo(map)
    .bindPopup('Завантаження даних для Острова Чайка...');

```

```

allMarkers.push({ marker: markerChaika, name: 'Острів Чайка', id:
'ostrivChaika' });

var markerGodynnyk = L.marker([49.55240858934784,
25.594774222180792]).addTo(map)
    .bindPopup('Завантаження даних для Годинника...');
allMarkers.push({ marker: markerGodynnyk, name: 'Годинник', id:
'hodynnyk' });

var markerParkSlavy = L.marker([49.55422794216372,
25.606226087447563]).addTo(map)
    .bindPopup('Завантаження даних для Парку Слави...');
allMarkers.push({ marker: markerParkSlavy, name: 'Парк Слави', id:
'parkSlavy' });

var markerParkVidrodzhennya = L.marker([49.55140931425392,
25.632383357472605]).addTo(map)
    .bindPopup('Завантаження даних для Парку Відродження...');
allMarkers.push({ marker: markerParkVidrodzhennya, name: 'Парк
Відродження', id: 'parkVidrodzhennya' });

async function updateSensorDataOnMap() {
    try {
        const response = await fetch('/get_sensor_data.php');
        if (!response.ok) {
            throw new Error(`HTTP error! status:
${response.status}`);
        }

        const sensorDataArray = await response.json();

        if (sensorDataArray && sensorDataArray.length > 0) {
            const latestData = sensorDataArray[0];
            const temp =
parseFloat(latestData.temperature).toFixed(1);
            const hum =
parseFloat(latestData.humidity).toFixed(1);
            const time = new
Date(latestData.timestamp).toLocaleString();

            allMarkers.forEach(item => {
                const marker = item.marker;
                const name = item.name;
                const id = item.id;

                const popupContent = `
                    <b>${name}</b>
                    <br>Температура: ${temp}°C
                    <br>Вологість: ${hum}%
                    <br>Оновлено: ${time}
                    <br><a href=«./location_page.html?id=${id}»
target=«_blank»

```



```
        style=«display: inline-block; padding: 6px
12px; background-color: #2d89ef; color: white; border-radius: 6px;
text-decoration: none; font-weight: bold; margin-top: 5px;»>
```

```
        Детальніше
```

```
    </a>`;
```

```
        marker.setPopupContent(popupContent);
```

```
    });
```

```
        console.log('Дані єдиного датчика оновлено для всіх
маркерів на карті.');
```

```
    } else {
```

```
        console.warn('Дані єдиного датчика відсутні або
некоректні (масив порожній).');
```

```
        allMarkers.forEach(item => {
```

```
            const marker = item.marker;
```

```
            const name = item.name;
```

```
            marker.setPopupContent(`<b>${name}</b><br>Дані
відсутні або помилка.`);
```

```
        });
```

```
    }
```

```
    } catch (error) {
```

```
        console.error('Помилка при оновленні даних датчиків на
карті:', error);
```

```
        allMarkers.forEach(item => {
```

```
            const marker = item.marker;
```

```
            const name = item.name;
```

```
            marker.setPopupContent(`<b>${name}</b><br>Помилка
завантаження даних.`);
```

```
        });
```

```
    }
```

```
}
```

```
updateSensorDataOnMap();
```

```
setInterval(updateSensorDataOnMap, 10000);
```

Сторінка contact.html

```
<!DOCTYPE html>
<html lang=«en»>
<head>
  <meta charset=«UTF-8»>
  <meta name=«viewport» content=«width=device-width, initial-
scale=1.0»>
  <link rel=«icon» href=«img/thermometer.png»>
  <link rel=«stylesheet» href=«style.css»>
  <link rel=«preconnect» href=«https://fonts.googleapis.com»>
  <link rel=«preconnect» href=«https://fonts.gstatic.com»
crossorigin>
  <link
href=«https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght
@0,300..800;1,300..800&display=swap» rel=«stylesheet»>
  <title>Contact</title>
</head>
<body id=«contact-page»>
  <header>
    <div class=«logo»>
      <img src=«img/logo.png» alt=«ParkWeather»>
      <h1>Дізнайся погоду у парку твого міста</h1>
    </div>
    <nav>
      <a href=«index.html»>Карта</a>
    </nav>
  </header>
  <div class=«contact-container»>
    <h2>Є запитання?</h2>
    <p>Заповніть форму нижче для зв'язку</p>

    <form action=«mailto:you@example.com» method=«POST»
enctype=«text/plain»>
      <input type=«text» name=«name» placeholder=«Введіть
ваше ім'я» required>
      <input type=«email» name=«email» placeholder=«Введіть
вашу електронну пошту» required>
      <textarea name=«message» placeholder=«Розкажіть про
вашу проблему» required></textarea>
      <button type=«submit»>Надіслати</button>
    </form>
  </div>
</body>
</html>
```