

КОЗАК Сергій

Тернопільський національний технічний університет імені Івана Пулюя

<https://orcid.org/0009-0008-7575-0953>

e-mail: serhii_kozak1710@tntu.edu.ua

МИКИТИШИН Андрій

Тернопільський національний технічний університет імені Івана Пулюя

<https://orcid.org/0009-0008-7575-0953>

e-mail: mikitishin@tntu.edu.ua

СТАНЬКО Андрій

Тернопільський національний технічний університет імені Івана Пулюя

<https://orcid.org/0000-0002-5526-2599>

e-mail: stanko.andrii@gmail.com

ОПТИМІЗАЦІЯ ЗБЕРІГАННЯ ТА ОБРОБКИ ДАНИХ ДЛЯ ІОТ-СИСТЕМ ЕКОЛОГІЧНОГО МОНІТОРИНГУ

Широке застосування ІоТ у різноманітних сферах, зокрема у системах для аналізу якості повітря, генерує великі обсяги різноманітних даних. Такі відомості, зокрема інформація про температуру, вологість, вміст газів і частинок у повітрі, постійно накопичуються та потребують ефективної обробки й зберігання. У статті описано засіб обробки та структуризації цих даних, отриманих під час функціонування системи аналізу якості повітря, шляхом застосування спеціалізованого програмного рішення. Запропонований застосунок використовується для упорядкування отриманих даних, які записані у форматі, менш придатному для подальшого аналізу та що потребує більше ресурсів порівняно з обробленими даними. Докладно описано алгоритм первинного сортування даних, їхнього збереження відповідно до структури бази даних, а також продемонстровано приклад організації бази даних для зберігання результатів вимірювання. У перспективі описаний застосунок може стати компонентом складніших ІоТ-систем, зокрема систем розумного дому, автоматизації виробництва або екологічного моніторингу, що забезпечує ширші можливості аналізу акумульованих даних та сприяє ухваленню обґрунтованіших рішень у реальному часі.

Ключові слова: ІоТ, безпроводні системи, системи автоматизації, масиви даних.

KOZAK Serhii, MYKYTYSHYN Andrii, STANKO Andrii

Ternopil Ivan Puluj National Technical University

OPTIMISATION OF DATA STORAGE AND PROCESSING FOR IOT ENVIRONMENTAL MONITORING SYSTEMS

The rapid and widespread adoption of the Internet of Things (IoT) across diverse sectors has led to the generation of vast amounts of heterogeneous data, particularly in applications related to air quality monitoring. These systems continuously collect environmental data such as temperature, humidity, concentrations of gases (e.g., CO₂, NO_x), and particulate matter, resulting in high-frequency, multidimensional datasets. Managing such data efficiently requires not only robust storage solutions but also intelligent preprocessing mechanisms to transform raw inputs into structured, analyzable formats.

This article presents a comprehensive approach to processing and structuring sensor data obtained from an IoT-based air quality analysis system. A specialized software application is introduced, designed to automate the organization and refinement of incoming sensor data. Initially received in unstructured or semi-structured formats that are less suitable for analysis and demand high computational resources, the data undergoes a systematic sorting process. The core algorithm responsible for the primary classification of measurements is described in detail, including logic for separating, labeling, and storing values in accordance with the predefined database schema.

The article provides a detailed overview of the database structure used to archive sensor data, along with practical examples that illustrate the process of converting raw inputs into normalized records ready for query and analysis. Emphasis is placed on scalability and real-time performance, ensuring that the system can accommodate expanding data flows without degradation in response time or accuracy.

Furthermore, the paper discusses the potential for integration of the described application into broader IoT infrastructures, such as smart homes, industrial automation systems, or environmental surveillance networks. By enabling faster data access and more meaningful analysis, the system supports real-time decision-making and long-term trend analysis. The proposed solution not only enhances the operational efficiency of air quality monitoring but also lays the groundwork for more intelligent and responsive environmental management systems.

Keywords: IoT, wireless systems, automation systems, data sets.

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Розповсюдження бездротових мереж зумовило істотне зростання кількості систем, здатних передавати дані бездротовими методами. Разом із цим збільшилися інформаційні потоки, що обробляються такими системами. Велика кількість датчиків, котрі здійснюють регулярні (похвилинні) вимірювання параметрів середовища або стану обладнання, постійно генерує нові дані, які потребують упорядкування та

зберігання задля подальшого аналізу. Відсутність належної обробки цих даних у короткостроковій перспективі може здаватися вигідною (за рахунок меншого споживання ресурсів), проте в довгостроковій перспективі призведе до неорганізованого накопичення надмірних масивів інформації, що ускладнить подальше вилучення корисної інформації і, відповідно, вимагатиме значних обчислювальних ресурсів [5-7].

Тому для ефективного вирішення таких проблем необхідна інтеграція спеціалізованих застосунків, що здійснюють обробку та структурування даних. Обробка даних передбачає використання ключових слів для відокремлення відповідних сегментів, формування нових записів у базі даних із необхідними зв'язками між об'єктами та їх атрибутами, а також надання користувачеві візуальних інструментів (графіків, діаграм) для аналізу зібраної інформації.

Оскільки початкові дані здебільшого представляють собою суцільні стрічки з різними параметрами (інколи у непередбачуваному форматі), важливо передбачити механізм динамічного задання ключових слів, за якими здійснюється сортування. Після відсортування дані повинні бути внесені до бази даних із дотриманням усіх внутрішніх зв'язків, щоб зберегти узгодженість та цілісність. Для підвищення наочності кінцевих результатів у застосунку доцільно реалізувати функцію відображення даних у формі графіків з можливістю використання гнучких фільтрів (за атрибутами, давачами, періодами тощо).

Основним завданням запропонованого застосунку є впорядкування та систематизація великих обсягів даних. Для верифікації коректності та ефективності підходу немає необхідності штучно генерувати гігантські масиви інформації, оскільки це не сприяє науковій цінності дослідження, а радше слугує експериментом для оцінки продуктивності. Практичною базою для тестування стали дані, отримані від системи аналізу якості повітря, де щохвилини фіксуються параметри (температура, вологість, тиск, вміст вуглекислого газу, аміаку, радіації, наявність часток пилу тощо) та записуються у форматі стрічки:

```
{1888,"АНТх0=temperature(C)","Temperature","°C","16.53","2024-08-06 00:00:38"}.
```

Зазначений формат даних збільшує навантаження на систему зберігання та ускладнює подальше вилучення інформації. Отже, перетворення та структуризація таких даних є важливим завданням під час розробки сучасних рішень з обробки великих масивів IoT-даних.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Якщо говорити про світові тренди розвитку інформаційних технологій, то найпопулярнішою, на даний момент, є сфера штучного інтелекту (ШІ) яка вже заходить у всі сфери людського життя від спрощення рутинних завдань до особистих помічників. Хоча більшість провідних компаній збільшили інвестиції в розвиток і розробку ШІ, але при цьому IoT продовжують входити в топ трійку найбільш пріоритетних напрямів для інвестицій. Також передбачається поступове зростання ринку мікропроцесорної техніки після проблемного “ковідного періоду”. Зокрема планується збільшення фінансування в покращення безпеки чипів від фізичного впливу що забезпечить надійнішу передачу даних в тому числі у хмарні сховища. Щодо хмарних сховищ, то поширення і розвиток такого типу технологій значно покращило роботу IoT, через змогу обміну даними не через локальні мережі, а через хмарні сховища. Хмарні сховища також є одним із пріоритетних напрямів фінансування і це в подальшому мало б позбавити деяких проблем які присутні на даний момент. Однією із таких проблем є збільшення загальної затримки в системі, що може не підійти для високоточних процесів, але цілком прийнятне для роботи систем домашньої автоматизації. Другою проблемою може бути вартість до доступу на хмарні сховища, але із збільшенням пропозицій на ринку друга проблема може бути вирішеною в найближчий час.

Ще одним вагомим ривком в розвитку IoT стають нові технологічні рішення спеціалізовані під ШІ. Мається на увазі, що провідні компанії презентують мікрокомп'ютери які специфікуються на роботі із ШІ. Так компанія Siemens представила SIMATIC IPC520A Box PC який оптимізований під роботу із ШІ.

Розвиток 5G став сильним поштовхом до розвитку і впровадження IoT у повсякденне життя людей, а разом із покращенням енергоефективності то це змінило і розширило сфери використання такого типу пристроїв.

Всі ці фактори сприяють росту ринку IoT а отже і систем які їх супроводжують. Передбачуваний ріст витрат на IoT в мільярдах доларів на найближчі роки можна побачити на рисунку 1.

За передбаченнями економістів, основним споживачем такого типу систем в наступні роки має стати Індія, через розширення свого промислового сектору і через тенденцію виводу виробничих потужностей, великих компаній, із Китаю в Індію. В подальшому можна говорити і про перенесення виробництва і в країни Східної Європи, в тому числі і в Україну. Тому як можна побачити світові тренди йдуть до збільшення і вдосконалення IoT і найблище десятиліття має тримати ці темпи зростання, що є фінансово привабливим як для глобальних гравців на цьому ринку, так і для локальних компаній які можуть закривати нестачу можливостей великих компаній чи просто можливою відсутності такої компанії на ринку певної країни чи регіону.

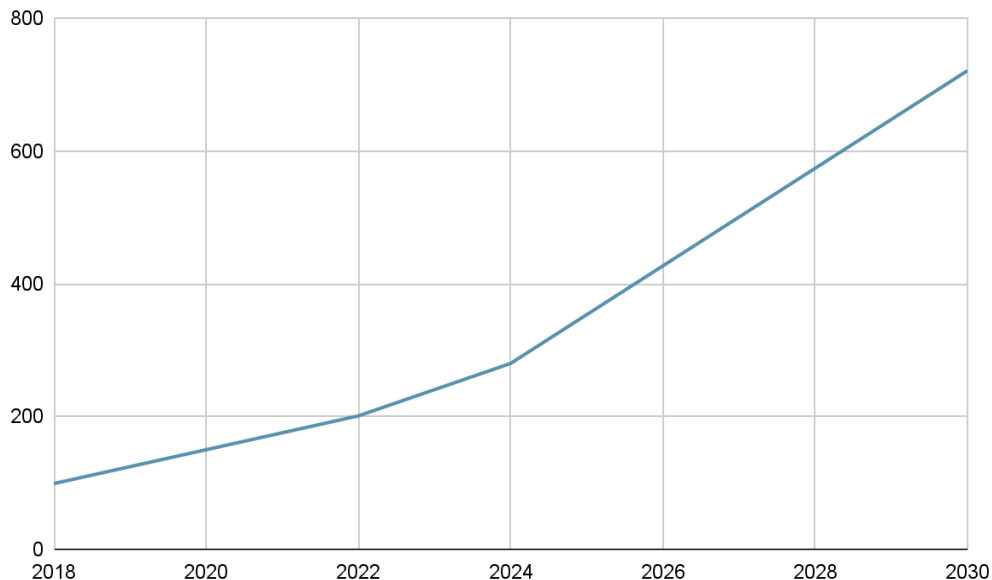


Рис. 1. Загальні світові витрати на IoT в мільярдах доларів США

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою роботи є: розробка застосунків, що дозволяє обробляти, зберігати та візуалізувати дані, генеровані системами аналізу якості повітря та іншими IoT-системами.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Розглянемо ключові критерії, що висуваються до програмного рішення:

- Доступність на різних платформах (ПК, смартфони, планшети тощо).
- Здатність опрацьовувати значні обсяги даних у багатокористувацькому середовищі.
- Гнучка архітектура, яку легко модифікувати чи розширювати новою функціональністю.
- Помірні витрати на розгортання та підтримку.

З огляду на ці вимоги було прийнято рішення реалізувати WEB-застосунок, адже він не прив'язаний до конкретної операційної системи і його можна використовувати на будь-якому пристрої з веббраузером та доступом до мережі Інтернет.

Для створення вебсервера обрано набір серверного ПЗ:

- LAMP (Linux, Apache, MySQL, PHP) або
- WAMP (Windows, Apache, MySQL, PHP)

Оскільки Linux відзначається гнучкістю, безкоштовністю та широким використанням у високонавантажених системах (зокрема, на потужних суперкомп'ютерах), її вибрано як основну операційну систему. Вебсервер Apache має низьку вартість володіння та вирізняється значною поширеністю, а MySQL розглядається як оптимальна СКБД для невеликих та середніх застосунків, забезпечуючи простоту встановлення, високий рівень безпеки та швидкодію.

Серед мов програмування найчастіше використовується PHP, адже це інтерпретована мова, що характеризується високою швидкодією у веб-середовищі та має активну спільноту розробників. Для прискорення процесу написання коду застосовано PHP-фреймворк Laravel, котрий реалізує шаблон MVC, має модульну структуру та багату бібліотеку розширень, а також поширюється безкоштовно.

У межах даного дослідження було розглянуто декілька підходів до архітектури бази даних, зокрема ER-Model та Object-Oriented Data Model, і обрано першу через її популярність та наочність. Перевага ER-моделі полягає у відповідності структур бази реальним чи абстрактним об'єктам, які описують різні аспекти досліджуваної системи.

Основну увагу зосереджено на сутностях, що відображають типи вимірювань: наприклад, температура, вологість, вміст вуглекислого газу та ін. Кожен тип представляється окремою таблицею, в якій зберігаються виміри з відповідних датчиків. На основі ER-моделі розроблено схему, зображену на рисунку 2.

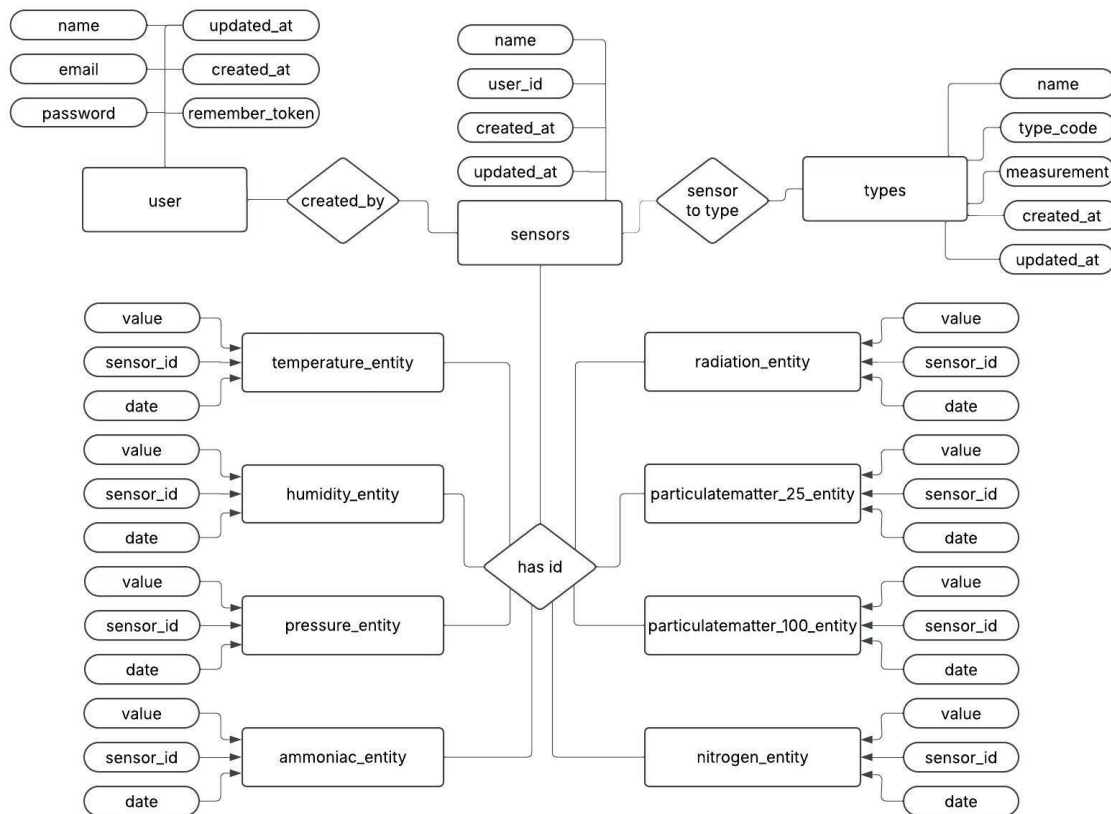


Рис. 2. Схема організації бази даних

Для підвищення безпеки і розмежування доступу передбачено систему авторизації користувачів. Об'єкт "користувач" (user) містить атрибути: name, email, password, remember_token, updated_at та created_at. Завдяки цьому можна реалізувати повноцінний мультикористувацький сценарій зберігання, коли кожен користувач працює із власними даними або поділяє їх з іншими за наявності відповідних прав.

Об'єкти sensors (давачі) описують фізичні джерела інформації й пов'язані з конкретними користувачами. Кожен сенсор може охоплювати кілька різних типів вимірювань (наприклад, одночасне вимірювання температури та вологості), що відображено через додаткову таблицю-зв'язку sensors_to_types.

Основні ж таблиці, в яких фактично зберігаються виміряні величини, мають поля: name, sensor_id та date - структура цих таблиць є однаковою, проте кожна відповідає конкретному типу.

Перш ніж додавати дані до відповідних таблиць, необхідно відокремити їх за типами. Для цього передбачено зчитування початкової інформації з файлу (наприклад, CSV), що робить процес завантаження більш передбачуваним і зручним. Сканування файлу dump безпосередньо ускладнене тим, що він зазвичай містить як структуру БД, так і самі дані, тож саме формат CSV визнаний оптимальним для початкової обробки.

На рисунок 3 показано приклад скрипта, котрий зчитує дані з файлів .csv та імпортує їх до БД.

Практика показала, що існують принаймні два поширені підходи:

1. Попередньо згруповане збереження: сортування та пакування даних у великі масиви з подальшою одночасною вставкою у БД. Це зменшує кількість SQL-запитів, але збільшує час виконання одного "масивного" запиту.

2. Покрокове збереження: дані записуються в БД одразу після їх вилучення та класифікації. У такому разі загальна кількість запитів суттєво зростає, проте виконання кожного з них триває не більш ніж 0.01 секунди.

У досліджуваному застосунку використовується другий підхід, аби забезпечити гнучкість і оперативність обробки. Важливу роль при цьому відіграють змінні \$paramsToTableArray та \$paramsArray, що зберігають ключові слова і зв'язують їх з назвами таблиць у базі даних. Додаючи нові ключові слова, користувач динамічно розширює можливості сортування (див. рисунок 4).

```
public function serialize($path): mixed
{
    $paramsToTableArray = $this->getParamsToTable();
    $paramsArray = $this->getParams();

    if (Storage::exists($path)) {
        // Get the file content
        $content = Storage::get($path);

        // Split the content into lines and parse as CSV
        $lines = explode(PHP_EOL, $content);
        $csvData = array_map('str_getcsv', $lines);
        foreach ($csvData as $csvDataLine) {
            foreach ($paramsArray as $paramName) {
                $tableName = $paramsToTableArray[$paramName];
                $sensorName = explode('=', $csvDataLine[1])[0];

                if (in_array($paramName, $csvDataLine)) {
                    DB::table($tableName)->insert([
                        'value' => $csvDataLine[4],
                        'sensor_id' => Sensor::where('name', $sensorName)->first()->id,
                        'date' => $csvDataLine[5],
                    ]);
                }
            }
        }

        return response()->json($csvData);
    }

    return true;
}
```

Рис. 3. Скрипт зчитування та збереження даних в БД

Dashboard Sensors Types

The image shows a web form titled "Dashboard Sensors Types". It contains three input fields: "Name:", "Type Code:", and "Measurement:". Below these fields are two buttons: "SAVE" (in a dark blue box) and "Cancel".

Рис. 4. Форма створення об'єкта type

Об'єкт type містить атрибути Name, Type Code (що використовується як ключове слово під час парсингу) та Measurement (одиниці вимірювання). Окремо передбачено об'єкт sensor, який слугує проміжною ланкою між type і конкретними відомостями (щоб один сенсор міг одночасно знімати кілька параметрів). Форма на рисунок 5 дає змогу створювати такі сенсори та пов'язувати їх з типами у таблиці sensors_to_types.

The screenshot shows the 'Sensors Types' form in the application dashboard. The form has a 'Name' text input field and a 'Type' dropdown menu. The dropdown menu is open, showing a list of sensor types: Temperature, Humidity, Pressure, Ammoniac, and Carbon. Below the dropdown menu are 'SAVE' and 'Cancel' buttons. A tooltip message 'Please select an item in the list.' is visible near the dropdown menu.

Рис. 5. Форма створення об'єкта sensors

Усі додані об'єкти доступні для перегляду, редагування й видалення у списках, наведених на рисунку 6 та рисунок 7.

The screenshot shows the 'Types' list in the application dashboard. The list contains the following sensor types: Temperature, Humidity, Pressure, Ammoniac, Carbon, Nitrogen, Radiation, Particulate Matter 25, and Particulate matter 100. Each item has a three-dot menu icon to its right. Below the list is an 'ADD TYPE' button.

Types	
Temperature	...
Humidity	...
Pressure	...
Ammoniac	...
Carbon	...
Nitrogen	...
Radiation	...
Particulate Matter 25	...
Particulate matter 100	...

Рис. 6. Список типів, створених користувачем

The screenshot shows the 'Sensors' list in the application dashboard. The list contains the following sensors: MICS-6814, RadKit, AHTx0, BMP280, and SDS011. Each item has a three-dot menu icon to its right. Below the list is an 'ADD SENSOR' button.

Sensors	
MICS-6814	...
RadKit	...
AHTx0	...
BMP280	...
SDS011	...

Рис. 7. Список сенсорів (давачів), створених користувачем

Накопичення великих масивів даних не має цінності без належної аналітики. Тому користувачам запропоновано інструменти візуалізації (найбільш показовою є побудова графіків у часі), що дають змогу відстежувати динаміку змін конкретних показників. Також реалізовано можливість фільтрування за типом, давачем або часовим періодом (фіксованим чи довільно заданим). На рисунк 9 продемонстровано приклад відображення температурних даних.

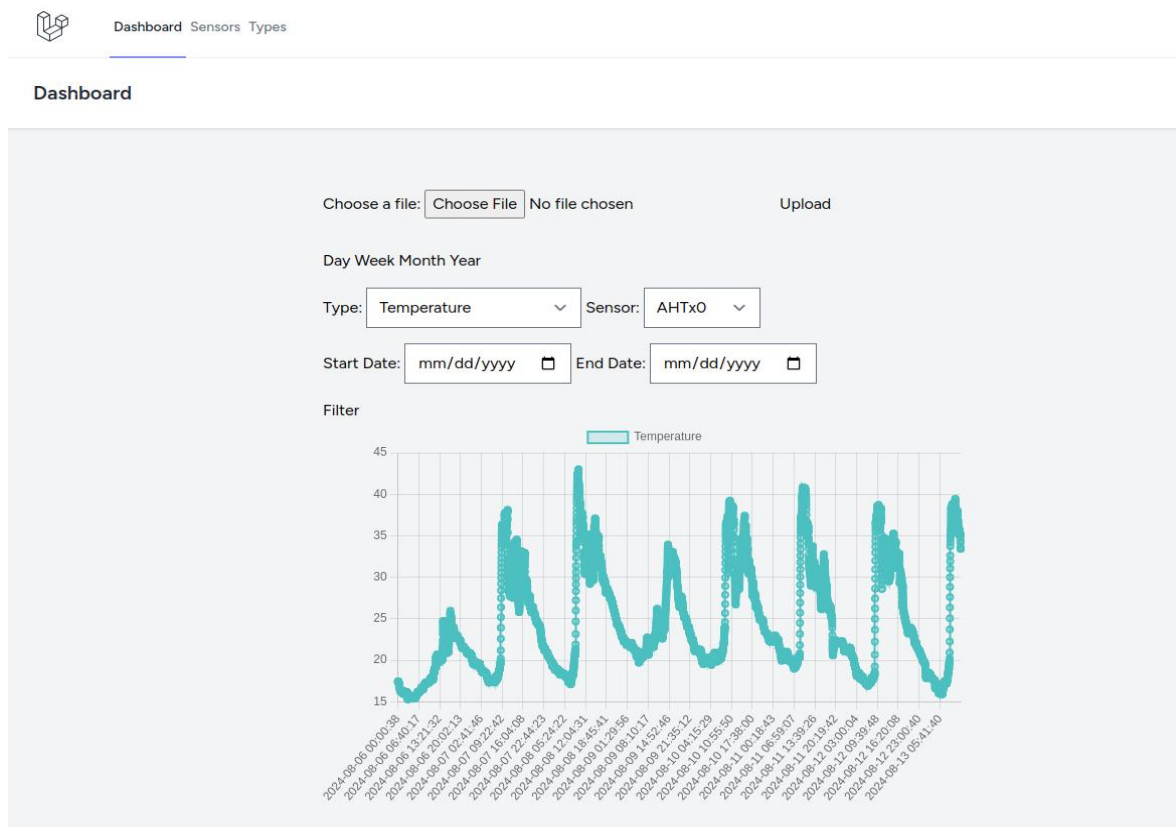


Рис. 8. Графічне відображення даних Temperature

У верхній частині інтерфейсу розміщено кнопки швидкого фільтру (Day, Week, Month, Year), нижче - випадаючі списки для вибору type та sensor, котрі пов'язані між собою. Остаточне ж уточнення періоду виконується вручну, якщо користувачеві потрібні специфічні часові межі.

Апробація застосунку показала відповідність реалізованих функцій початково поставленим вимогам. Використані технології (Apache, MySQL, Laravel тощо) є вільними чи мають невисоку вартість, що зменшує загальні витрати на розробку та впровадження. Популярність зазначених рішень передбачає наявність обширної бази знань і готових рішень для більшості типовими завдань, завдяки чому розробку можна масштабувати і вдосконалювати.

Формат вебзастосунку дозволяє використовувати систему з будь-якого пристрою, що має браузер та підключення до мережі Інтернет, що значно розширює потенційну аудиторію. Як експериментальна складова більших IoT-проектів, цей застосунок може інтегруватися з іншими модулями та сервісами, відповідно, покриваючи ширший спектр потреб.

Щодо продуктивності, другий підхід (покрокове збереження) засвідчив переваги в гнучкості та швидкодії на рівні окремих запитів. Порівняно невелика затримка при кожному записі (менше ніж 0.01 с) істотно не впливає на продуктивність системи навіть за умови, що сенсори генерують дані щохвилини або частіше. У разі зростання обсягів даних можна вдосконалювати архітектуру (наприклад, перейти до масштабування по горизонталі, використовуючи декілька серверів) або впроваджувати черги повідомлень (message queues) для обробки у режимі near real-time.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

Проведений аналіз свідчить, що розроблений застосунок успішно розв'язує завдання обробки великих масивів даних у системах IoT. Завдяки використанню вільнодоступних та поширених технологій (Linux, Apache, MySQL, PHP/Laravel) забезпечується висока швидкість роботи, мінімальні витрати на розгортання та зручність у подальшому супроводі.

Запропонована архітектура бази даних на основі ER-моделі є досить гнучкою для розширення: у разі виникнення нових типів вимірювань чи появи додаткових сенсорів система може бути легко адаптована. Завдяки чіткому розмежуванню об'єктів (type, sensor, user) та використанню таблиці-зв'язки (sensors_to_types) вдається оптимізувати процес обліку різноманітних типів даних.

Алгоритм покрокового збереження даних забезпечує стабільний час відгуку, що є актуальним для систем з частою фіксацією показників. Інтегрована функція візуалізації дає змогу користувачам зручніше відслідковувати зміни параметрів та застосовувати різні фільтри, що спрощує аналіз.

Таким чином, розробка подібних застосунків і впровадження їх у різні IoT-системи (від контролю повітря до розумного дому або індустріального моніторингу) сприяє ефективнішому використанню даних та покращує інструментарій для ухвалення зважених рішень. Подальший розвиток полягає у можливості розширення застосунку для роботи з потоковими (streaming) даними, використання аналітичних модулів на основі машинного навчання та формування автоматизованих звітів у реальному часі.

Література

1. Tamboli, A. (2022). Build your own IoT platform: Develop a flexible and scalable Internet of Things platform (2nd ed.). Apress. doi.org/10.1007/978-1-4842-8073-7
2. Singh, S. K., Cha, J., Kim, T. W., & Park, J. H. (2021). Machine learning based distributed big data analysis framework for next generation web in IoT. *Computer Science and Information Systems*, 18(2), 597-618.
3. Belcastro, L., Carretero, J., & Talia, D. (2024). Edge-Cloud Solutions for Big Data Analysis and Distributed Machine Learning-1. *Future Generation Computer Systems*.
4. State of IoT Spring 2024: 10 emerging IoT trends driving market growth <https://iot-analytics.com/state-of-iot-10-emerging-iot-trends-driving-market-growth/>
5. W. Wiczorek, A. Stanko, A. Mykytyshyn, O. Holotenko, T. Lechachenko, «Real-time air quality management: Integrating IoT and Fog computing for effective urban monitoring». In *CEUR Workshop Proceedings (CITI-2024)*, 2024, vol. 3742, pp. 337–357.
6. Stanko, A. Sverstiuk, L. Matiichuk, U. Polyvana, V. Nykytyuk, «Analytical analysis of approaches to assessing the quality of life in smart cities». In *CEUR Workshop Proceedings (BAIT-2024)*, 2024, pp. 75–91.
7. Didych, A. Stanko, A. Mykytyshyn, M. Mytnyk, «Application of machine learning methods to the prediction of NO₂ concentration in the air environment». In *4th International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP-2024)*, 2024.

References

1. Tamboli, A. (2022). Build your own IoT platform: Develop a flexible and scalable Internet of Things platform (2nd ed.). Apress. doi.org/10.1007/978-1-4842-8073-7
2. Singh, S. K., Cha, J., Kim, T. W., & Park, J. H. (2021). Machine learning based distributed big data analysis framework for next generation web in IoT. *Computer Science and Information Systems*, 18(2), 597-618.
3. Belcastro, L., Carretero, J., & Talia, D. (2024). Edge-Cloud Solutions for Big Data Analysis and Distributed Machine Learning-1. *Future Generation Computer Systems*.
4. State of IoT Spring 2024: 10 emerging IoT trends driving market growth <https://iot-analytics.com/state-of-iot-10-emerging-iot-trends-driving-market-growth/>
5. W. Wiczorek, A. Stanko, A. Mykytyshyn, O. Holotenko, T. Lechachenko, «Real-time air quality management: Integrating IoT and Fog computing for effective urban monitoring». In *CEUR Workshop Proceedings (CITI-2024)*, 2024, vol. 3742, pp. 337–357.
6. Stanko, A. Sverstiuk, L. Matiichuk, U. Polyvana, V. Nykytyuk, «Analytical analysis of approaches to assessing the quality of life in smart cities». In *CEUR Workshop Proceedings (BAIT-2024)*, 2024, pp. 75–91.
7. Didych, A. Stanko, A. Mykytyshyn, M. Mytnyk, «Application of machine learning methods to the prediction of NO₂ concentration in the air environment». In *4th International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP-2024)*, 2024.