

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Веб-орієнтована система моніторингу параметрів мікроклімату
житлового приміщення»**

Виконав студент групи ІПЗм-22-1
ШКІЛЬНЮК Дмитро Олегович

Керівник роботи:
ЛОБАНЧИКОВА Надія Миколаївна

Рецензент:
ВАКАЛЮК Тетяна Анатоліївна

Житомир – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
 ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
 ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

зав. кафедри інженерії
 програмного забезпечення

Тетяна ВАКАЛЮК

«18» вересня 2023 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ШКІЛЬНЮК Дмитро Олегович**

Керівник роботи: **ЛОБАНЧИКОВА Надія Миколаївна**

Тема роботи: **«Веб-орієнтована система моніторингу параметрів мікроклімату житлового приміщення»**,

затверджена Наказом закладу вищої освіти від **«18» вересня 2023 р. № 479/с**

Вихідні дані для роботи: Предметом дослідження є використання сучасних новітніх технологій для розробки системи моніторингу. Об'єктом дослідження є сучасні технології для розробки веб-додатків та приладів збору інформації.

Консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Лобанчикова Н. М.	22.09.2023р.	22.09.2023р.
2	Лобанчикова Н. М.	22.09.2023р.	22.09.2023р.
3	Лобанчикова Н. М.	22.09.2023р.	22.09.2023р.

РЕФЕРАТ

Кваліфікаційна магістерська робота присвячена вивченню процесу розробки веб додатку з метою спостереження за параметрами мікроклімату житлового приміщення.

Пояснювальна записка складається зі вступу, трьох розділів, переліку умовних позначень та списку використаної літератури. Записка дипломної роботи містить 33 ілюстрацій, 10 таблиць та 38 посилання на джерела бібліографії. Повний обсяг роботи становить 64 сторінок, з них 61 сторінки основного тексту.

Кваліфікаційна магістерська робота на тему: «Веб-орієнтована система моніторингу параметрів мікроклімату житлового приміщення» складається з теоретичного та програмного комплексу системи та пояснювальної записки.

У першому розділі обґрунтовано вибір технологій та інструментів для розробки системи, враховано аналіз ринкових рішень, забезпечено оптимальний функціонал та ефективність, що сприятиме успішній розробці системи моніторингу параметрів мікроклімату житлового приміщення.

У другому розділі визначено етапи проектування системи моніторингу житлового приміщення, зазначено ключові компоненти та вимоги користувачів. Здійснено обґрунтований вибір технологій та додаткових рішень для ефективної реалізації функціональності та забезпечення швидкості та інтуїтивності веб-інтерфейсу.

У 3 розділі було детально викладено практичну реалізацію веб-додатку для система моніторингу параметрів мікроклімату житлового приміщення. джерело даних та реалізовано практичну реалізацію веб-орієнтованої системи. Використано ключові технології та рішення з розділу 2, для ефективного відображення та аналізу даних.

Пакет програмного забезпечення було протестовано.

КЛЮЧОВІ СЛОВА: МОНІТОРИНГ, ВЕБ-ЗАСТОСУНОК, ГРАФІКИ, СИСТЕМА, МІКРОКЛІМАТ.

ABSTRACT

The qualifying master's thesis is devoted to the study of the process of developing a web application for the purpose of monitoring the microclimate parameters of a living space.

The explanatory note consists of an introduction, three sections, a list of notations and a list of used literature. The thesis note contains 33 illustrations, 10 tables and 38 references to bibliographic sources. The full volume of the work is 64 pages, including 61 pages of the main text.

The qualifying master's thesis on the topic: "Web-oriented system for monitoring the microclimate parameters of a residential premises" consists of a theoretical and software complex of the system and an explanatory note.

In the first section, the choice of technologies and tools for the development of the system is substantiated, the analysis of market solutions is taken into account, optimal functionality and efficiency are ensured, which will contribute to the successful development of a system for monitoring the microclimate parameters of a residential premises.

In the second section, the design stages of the residential monitoring system are defined, the key components and user requirements are specified. A reasoned choice of technologies and additional solutions was made to effectively implement the functionality and ensure the speed and intuitiveness of the web interface.

In the 3rd chapter, the practical implementation of the web application for the monitoring system of the microclimate parameters of the residential premises was described in detail. data source and the practical implementation of the web-oriented system was realized. Key technologies and solutions from Chapter 2 are used for effective data display and analysis.

The software package has been tested.

KEY WORDS: MONITORING, WEB APPLICATION, GRAPHICS, SYSTEM, MICROCLIMATE.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1. Аналіз параметрів моніторингу мікроклімату приміщень.....	11
1.2. Порівняльний аналіз існуючих систем моніторингу	14
1.3. Обґрунтування архітектури системи моніторингу	21
1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	25
1.5. Постановка задачі	29
Висновки до першого розділу.....	30
РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ПАРАМЕТРІ МІКРОКЛІМАТУ ЖИТЛОВОГО ПРИМІЩЕННЯ	32
2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи.....	32
2.2. Проектування бази даних системи та структури файлів системи.....	35
2.3. Проектування алгоритмів роботи системи	40
2.4. Обґрунтування засобів реалізації для веб-орієнтованої системи моніторингу параметрів мікроклімату житлового приміщення	41
Висновки до другого розділу.....	44
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ПАРАМЕТРІВ МІКРОКЛІМАТУ ЖИТЛОВОГО ПРИМІЩЕННЯ	45
3.1. Встановлення та налаштування системи	45
3.2. Реалізація веб-додатку	46
3.3. Результат виконання роботи.....	53
Висновок з третього розділу	56
ВИСНОВОКИ.....	57
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	58
ДОДАТКИ.....	62

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

СМ – Система моніторингу

СМЖ – Система моніторингу житлового приміщення

IoT (Internet of Things) – Інтернет речей

IDE – Середовище розробки (написання коду)

OSPanel – Open Server Panel

RBAC – Контроль доступу на основі ролей

ВСТУП

Актуальність теми. Моніторинг – система постійного спостереження за явищами і процесами, що проходять в навколишньому середовищі, суспільстві, результати якого слугують для обґрунтування управлінських рішень по забезпеченню безпеки людей та об'єктів/суб'єктів. «Моніторити» можливо будь що: фінансовий ринок, геолокацію (GPS), технічний стан авто, довкілля, здоров'я тощо. Маючи інформацію, яку дає система моніторингу, можна: спланувати подальші дії, запобігти небажаному результату, чи просто дізнатися про поточну ситуацію з предметом моніторингу.

На сьогоднішній день в більшості житлових і, не тільки, приміщеннях є прилади для спостереження за окремими аспектами мікроклімату, зокрема:

- термометр;
- датчик газу, або сигналізатор газу;
- гігрометр;
- барометр.

Все це часто невеликі прилади, які стоять десь на полиці і показують вам дані з допомогою стрілки або числового циферблату. Щоб переглянути інформацію з ним потрібно бути біля них.

Розвиток інформаційних технологій та сучасний рівень науки і техніки дозволяють ефективно впроваджувати інноваційні підходи у різні галузі життя суспільства. Однією з важливих складових цього процесу є інформатизація суспільства, яка охоплює широкий спектр застосувань, включаючи в себе використання способів автоматизації та систем моніторингу.

Зокрема, в сучасних умовах надзвичайно актуальною стає необхідність впровадження систем моніторингу параметрів мікроклімату житлових приміщень. Це важливо для забезпечення комфорту та безпеки мешканців, а також для оптимізації витрат енергоресурсів. Системи моніторингу дозволяють в реальному часі вимірювати та аналізувати різні параметри, такі як температура,

вологість, рівень CO₂ та інші, забезпечуючи тим самим можливість швидкого реагування на будь-які зміни.

Автоматизовані системи контролю за мікрокліматом можуть бути інтегровані з різними пристроями та смарт-технологіями, що дозволяє вдало поєднувати комфорт та енергоефективність. Наприклад, системи "розумного будинку" можуть автоматично регулювати опалення, кондиціонування повітря та освітлення відповідно до вимог користувачів та отриманих даних з моніторингових систем.

Розробка та впровадження таких систем має великий потенціал у покращенні якості життя громадян, зменшенні впливу людської діяльності на навколишнє середовище та раціональному використанні ресурсів. Зокрема, це стає актуальним в умовах зростання світового інтересу до енергоефективності та сталого розвитку.

Знайдення нових моделей, методів та засобів автоматизації обробки інформації та моніторингу параметрів мікроклімату в різних об'єктах віддалено є актуальною задачею сьогодення.

Виходячи з вищевикладеного, розробка веб-орієнтованої системи моніторингу параметрів мікроклімату житлового приміщення, що направлена на аналіз поточного стану параметрів мікроклімату та інформування власників приміщень про поточну ситуацію в окремому приміщенні, або будівлі в цілому є актуальною.

Мета даної магістерської кваліфікаційної роботи полягає у розробці веб-орієнтованої системи моніторингу параметрів мікроклімату житлового приміщення.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз приладів вимірювання параметрів мікроклімату, архітектури та інструментів для розробки веб-системи моніторингу.
- розробити алгоритм для збору, обробки та передачі даних від сенсорів до веб-системи моніторингу. Забезпечити допустиму частоту зчитування та передачі даних для забезпечення реального часу та мінімізації витрат енергії.

- здійснити проектування та програмну реалізацію веб-орієнтованої системи моніторингу параметрів мікроклімату житлового приміщення.

Об’єктом дослідження є сучасні технології побудови веб-орієнтованої системи моніторингу параметрів мікроклімату житлового приміщення

Предметом дослідження є методи та алгоритми розробки системи моніторингу параметрів мікроклімату житлового приміщення з функціями АПІ (application programming interface), збору інформації про навколишнє середовище.

Наукова новизна отриманих результатів:

- Проведений аналіз приладів вимірювання параметрів мікроклімату, архітектури та інструментів для розробки веб-системи моніторингу.
- Розроблено алгоритм збору та аналізу даних з сенсорів мікроклімату:
- Запропоновано метод для забезпечення взаємодії між датчиками мікроклімату та веб-системою для віддаленого моніторингу.

Практичне значення отриманих результатів полягає у програмній реалізації запропонованого алгоритму та методів у вигляді веб-орієнтованої системи моніторингу параметрів мікроклімату житлового приміщення.

Значимість дослідження. Розробка та впровадження веб-орієнтованої системи моніторингу параметрів мікроклімату житлових приміщень має ключове значення з ряду аспектів:

- Підвищення комфорту та здоров'я мешканців:

Вивчення та оптимізація параметрів мікроклімату дозволить підтримувати оптимальні умови для здоров'я та комфорту мешканців. Це важливо для підвищення якості життя та зменшення ризику розвитку захворювань, пов'язаних з неадекватними умовами мікроклімату.

- Енергоефективність та сталість експлуатації:

Використання автоматизованої системи моніторингу дозволяє ефективно використовувати енергію для регулювання параметрів мікроклімату, зменшуючи витрати та покращуючи сталість експлуатації житлових будівель.

- Внесок у розвиток Інтернету речей (IoT):

Дослідження сприяє розвитку та вдосконаленню технологій Інтернету речей, що є актуальним напрямком в сучасному інформаційному суспільстві. Веб-орієнтована система моніторингу є частиною інфраструктури для забезпечення зручного та ефективного використання ресурсів.

- Внесок у розвиток "розумних" будівель та міст:

Дослідження вносить свій внесок у розвиток концепції "розумного будинку" та "розумного міста", де системи моніторингу та автоматизації сприяють створенню ефективної та екологічно-дружньої інфраструктури.

- Сприяння екологічній стійкості та збереженню ресурсів:

Впровадження веб-орієнтованої системи моніторингу дозволяє не лише покращити умови життя, але й сприяє більш ефективному використанню енергії та ресурсів, сприяючи збереженню екологічної рівноваги та стійкості навколишнього середовища.

Методологія роботи. Під час дослідження було використано загальнонаукові та емпіричні наукові методи, що ґрунтуються на системному підході, такі як абстрагування, аналіз, синтез, порівняння, систематизація, методи об'єктно-орієнтованого проектування та моделювання

Для розробки системи було вирішено використати стек технологій WAMP, а саме операційну систему Windows, веб-сервер OpenServer з модулем Apache, систему керування БД MySQL та мову програмування PHP. Для швидкої та якісної розробки було обрано PHP-фреймворк Yii2. Як суміжні технології було використано середовище розробки Visual Studio Code, систему керування версіями Git, менеджер PHP-розширень Composer.

Особистий внесок здобувача. Усі результати роботи отримані автором самостійно.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз параметрів моніторингу мікроклімату приміщень

Речі за якими можливо проводити моніторинг в житловому приміщенні можна поділити на 3 категорії:

- Сам мікроклімат (температура/вологість):

Моніторинг температури та вологості, дозволить користувачам забезпечити оптимальні умови для власного здоров'я, комфорту, рослин та інших елементів екосистеми в житловому приміщенні (рис. 1.1).



Рис. 1.1. Зовнішній вигляд датчику температури та вологості повітря

Але для різних кімнат оптимальні умови можуть відрізнятись. Ознайомитися з оптимальними параметрами мікроклімату можна в таблиці 1.1.

- Безпека (пошук вогню/газу):

Впровадження системи виявлення вогню/диму дозволить оперативно реагувати на потенційні небезпеки в вигляді пожегів та забезпечить безпеку мешканців приміщення (рис 1.2). Також вона допоможе в виявленні витoku чадного газу. У чадного газу немає ані запаху, ані кольору – людина не може його відчувати. Виявити небезпечну концентрацію CO в повітрі можуть лише спеціалізовані датчики (рис. 1.3).

Таблиця 1.1

Оптимальні норми різних параметрів мікроклімату в залежності від виду приміщень.

Пора року	Назва приміщення	Температура повітря, °C		Відносна вологість, %	
		оптимальна	допустима	оптимальна	допустима, не більш
Холодна	Жила кімната	20-22	18-24 (20-24)	45-30	60
	Кухня	19-21	18-26	Не нормується	Не нормується
	Туалет	19-21	18-26	Не нормується	Не нормується
	Ванна	24-26	18-26	Не нормується	Не нормується
	Приміщення для відпочинку та навчання	20-22	18-24	45-30	60
	Міжквартирний коридор	18-20	16-22	45-30	60
	Вестибюль	16-18	14-20	Не нормується	Не нормується
	Комори	16-18	12-22	Не нормується	Не нормується
Тепла	Жила кімната	22-25	20-28	60-30	65

Нормальна концентрація чадного газу в домашніх умовах — менше 0,001% (10 ppm). Концентрація понад 0,052% (520 ppm) протягом двох годин викликає головний біль, запаморочення, нудоту, а концентрація на рівні 0,1% (1000 ppm) призводить до втрати свідомості. За концентрації CO в повітрі 0,15% (1500 ppm) можливим є летальний випадок.



Рис. 1.2. Зовнішній вигляд датчику диму



Рис. 1.3. Зовнішній вигляд датчиків угарного газу

– Флора (вологість ґрунту):

Система моніторингу вологості ґрунту стане незамінним інструментом для тих, хто займається рослинництвом чи утриманням домашніх рослин, дозволяючи вчасно реагувати на потреби рослин у воді. Зачасту такі датчики є вбудованими елементами системи автополиву, яка завдяки доступу до води має можливість відразу при необхідності виконувати полив (рис 1.4).



Рис. 1.4. Приклад датчиків угарного газу

Аналіз категорій моніторингу в житловому приміщенні виявив важливі аспекти, такі як мікроклімат (температура/вологість), безпека (виявлення вогню/газу) та вологість ґрунту для рослин. Ця інформація дозволяє забезпечити оптимальні умови для здоров'я, комфорту та безпеки мешканців, а також ефективно управляти умовами в приміщенні та доглядати за рослинами.

1.2. Порівняльний аналіз існуючих систем моніторингу

Системи моніторингу (СМ) зазвичай застосовуються для конкретних цілей і областей в житті, де необхідно стежити за реальним станом об'єкта або процесу. Наприклад, на біржах криптовалют СМ слугує для перегляду історії змін ціни на криптовалюту та її поточного стану. Аналогічно, у географічних інформаційних системах, таких як Google Maps, СМ виявляється у вигляді інтерактивної картографії, де дороги можуть змінювати свій колір в залежності від рівня трафіку ("Затори").

В сфері побуту Система Моніторингу Житлового приміщення (СМЖ) часто входить до складу систем розумного будинку. Наприклад, Аях – це комплекс системи безпеки, де СМЖ постійно спостерігає за багатьма факторами які можуть зашкодити власнику, або його майну. Основні можливості Аях включають:

- Спостереження за рухами та розбиттям вікон (рис. 1.5):

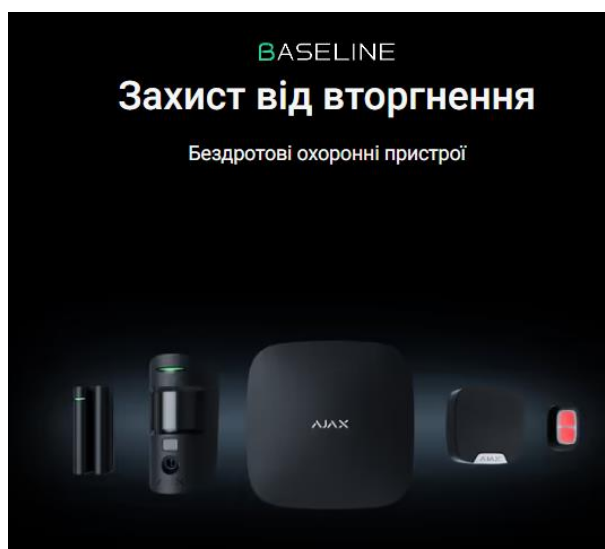


Рис. 1.5. Комплекс Ајах для захисту від вторгнень

Система реагує на неправомірний доступ або спроби вторгнення, сповіщаючи власника про потенційну загрозу.

- Виявлення задимлення, вогню (рис 1.6):

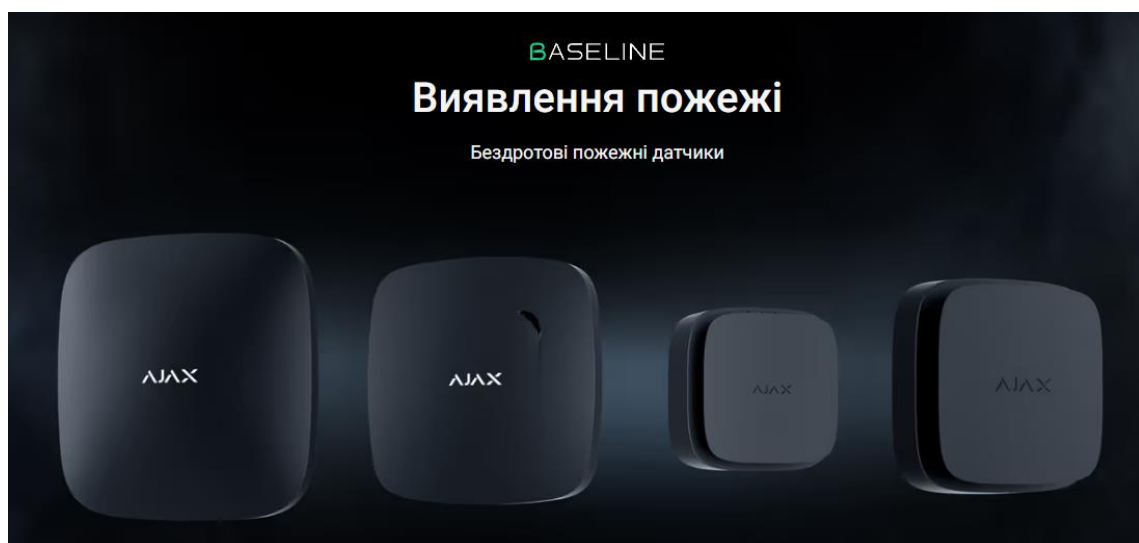


Рис. 1.6. Комплекс Ајах для виявлення пожеж

Ајах реагує на виникнення пожежі чи задимлення, запобігаючи потенційним катастрофам. В парі з модулем інтеграції, який буде описано потім, система отримує можливість самостійно реагувати на можливу небезпеку (перекриття газу, ввимкнення живлення в будинку)

- Виявлення затоплення (рис. 1.7):



Рис. 1.7. Комплекс Ajax для захисту від вторгнень

При виявленні ознак протікання води Ajax може самостійно перекрити подачу води в відповідному місці або в будинку в цілому.

- Моніторинг витрат електроенергії:

З використанням розумних розеток, Ajax дозволяє контролювати та оптимізувати витрати електроенергії.

Ajax також надає можливість розширити мережу пристроїв стороніми для самого Ajax пристроями за допомогою модулів інтеграції та має доступ до веб-інтерфейсу, десктопних та мобільних додатків для зручного керування. Однак, серед недоліків варто відзначити високу вартість пристроїв, зокрема, хабу – центральної частини системи, який може обійтися користувачеві відповідно до обраної моделі, в мінімум 6899 грн на момент написання.

У відмінну від попереднього згаданого Ajax, сервіс "Народний моніторинг" представляє собою інший підхід до систем моніторингу. Специфікою цього сервісу є те, що він не постачає власних готових пристроїв для збору даних, але замість цього дозволяє користувачам використовувати власні датчики та пристрої для збирання різноманітних даних.

"Народний моніторинг" вражає різноманітністю своєї бази даних, яка включає інформацію з усього світу (рис 1.8). Користувачі можуть зберігати дані, що стосуються температури, вологості, тиску, рівня забрудненості повітря, а

навіть рівня радіації та інших параметрів довкілля. Це робить сервіс універсальним і відкритим для різних застосувань, від домашнього використання до наукових досліджень.

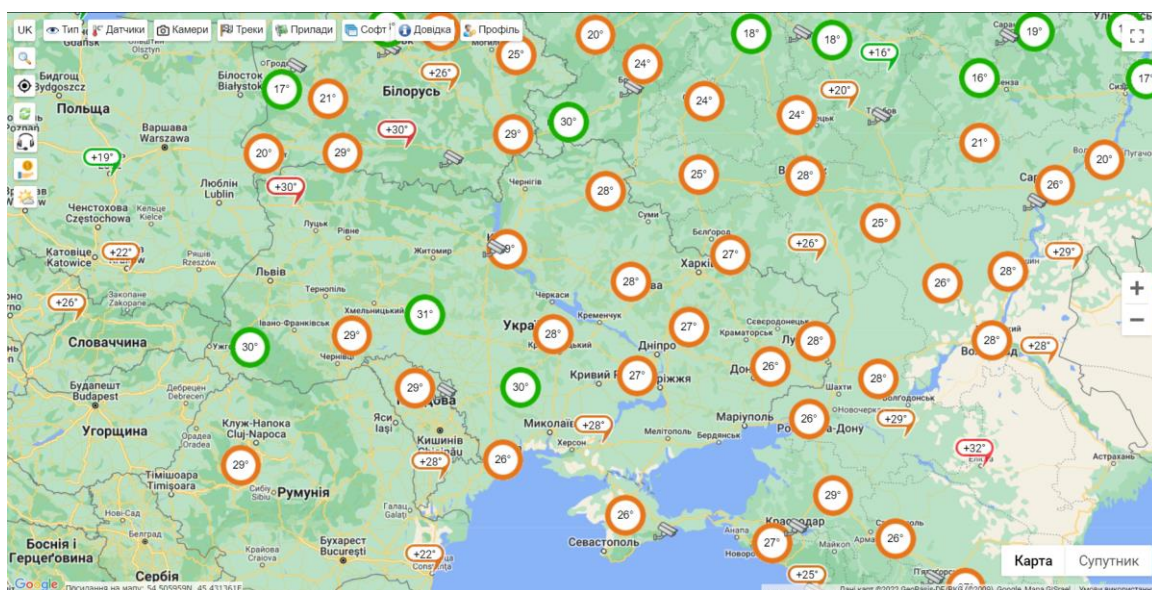


Рис. 1.8. Інтерфейс сервісу «Народний моніторинг»

Однак, важливим недоліком "Народного моніторингу" є обмеження на часовий інтервал відправки даних - лише раз в п'ять хвилин. Крім того, безкоштовна версія сервісу дозволяє зберігати дані лише з одного приладу, що може бути обмеженням для тих, хто має кілька датчиків або пристроїв в різних місцях.

Такий підхід вказує на те, що "Народний моніторинг" розрахований на широкий коло користувачів, які мають власні датчики та бажають активно вносити дані в загальний пул інформації для подальшого аналізу та використання.

Ще одним прикладом є Bosch Smart Home, який представляє собою сучасну систему розумного будинку від відомого німецького виробника Bosch. Ця система забезпечує широкий спектр можливостей для автоматизації та контролю параметрів житлового приміщення. Насамперед, Bosch Smart Home визначається високою якістю виробів та інтеграцією різних аспектів розумного житла.



Рис. 1.9. Bosch Smart Home

Основні можливості Bosch Smart Home:

- Зручний інтерфейс та додатки:

Система надає інтуїтивно зрозумілі додатки для смартфонів, планшетів та комп'ютерів, які дозволяють користувачам контролювати різні аспекти свого будинку в режимі реального часу.

- Енергозбереження:

Bosch Smart Home дозволяє ефективно керувати споживанням енергії в будинку, регулюючи опалення, кондиціонування повітря та освітлення за допомогою автоматичних регуляторів.

- Безпека:

Система включає в себе функції безпеки, такі як виявлення руху, системи відеоспостереження та сигналізації, забезпечуючи відчуття безпеки для мешканців.

- Автоматизація:

Bosch Smart Home може автоматизувати різні процеси, від керування шторами та термостатами до управління водопостачанням та системами безпеки.

- Інтеграція з іншими пристроями:

Система може інтегруватися з іншими смарт-пристроями та платформами, такими як Google Assistant, Amazon Alexa та Apple HomeKit, щоб забезпечити більше можливостей та зручності.

Однак, слід відзначити, що Bosch Smart Home може виявитися вартісною для деяких користувачів, адже це високотехнологічна система, яка може вимагати певних витрат на покупку та встановлення.

Усього враховуючи, Bosch Smart Home є високоякісною та надійною системою розумного будинку, яка відповідає високим стандартам якості виробника Bosch.

Проаналізувавши приклади, розглянемо порівняльну характеристику їх СМЖ, наведену у таблицях 1.2-1.4.

Таблиця 1.2

Аналіз характеристик аналогів

Назва	Зручність GUI	Платформа	Роль СМЖ	Графічне оформлення
Аjax	Дуже добре	Desktop Web Mobile	Інструмент	Дуже добре
«Народний моніторинг»	Добре	Web	Ключовий елемент	Нормально
Bosch Smart Home	Добре	Mobile	Інструмент	Добре

Порівняння аналогів за роботу з даними.

Таблиця 1.3

Робота з даними

Назва	Поточними	Попередніми
Аjax	Вказуються при перегляді пристрою	Лише записи в сповіщеннях
«Народний моніторинг»	Відсутня	Графіки та таблиці
Bosch Smart Home	Вказуються при перегляді пристрою	Не вдалося знайти інформацію

Порівняння аналогів за реакцією на різні події.

Таблиця 1.4

Реакцією на різні події

Назва	Немає зв'язку з датчиками	«Подія»	У пристрій розряджено
Ajax	На ваші пристрої приходить відповідне сповіщення	Відправлення сповіщень та запуск сценарію, умовами якого є дана подія	На ваші пристрої приходить відповідне сповіщення
«Народний моніторинг»	Сповіщення на сервіси відповідно до налаштувань профілю	Відсутня	Відсутня
Bosch Smart Home	На ваші пристрої приходить відповідне сповіщення	Відправлення сповіщень та запуск сценарію, умовами якого є дана подія	На ваші пристрої приходить відповідне сповіщення

Враховуючи різноманіття систем моніторингу житлових приміщень і їхніх функціональних особливостей, ключовою частиною проекту є коректне визначення та управління подіями. У даному контексті, термін "подія" асоціюється із спрацюванням датчика або виникненням аномальних значень, що викликають необхідні дії.

Проаналізувавши аналогічні системи, слід відзначити, що основна відмінність у використанні систем моніторингу житлових приміщень полягає в подальших діях, які вони виконують після виявлення подій. Додатково, система "Народний моніторинг" відкриває можливість використання сторонніх додатків для взаємодії та отримання сповіщень на події, що робить її більш гнучкою у плані інтеграції з іншими сервісами.

Враховуючи це, проект системи моніторингу житлового приміщення визначається наступними ключовими рисами:

- Зручний інтерфейс:

Система повинна мати інтуїтивно зрозумілий інтерфейс, що спрощує взаємодію для кінцевого користувача.

- Перегляд даних:

Можливість переглядати як поточні, так і архівні дані є ключовою функцією, що дозволяє аналізувати динаміку параметрів приміщення.

- Сповіщення про події:

Система повинна надавати можливість налаштовувати сповіщення на події, такі як перевищення параметрів датчика, що допомагає оперативно реагувати на зміни.

- Доступ через інтернет:

Можливість отримання доступу до системи через Інтернет розширює функціональні можливості та зручність використання, особливо для віддаленого контролю.

Реалізація цих характеристик дозволить створити високоефективну та зручну систему моніторингу житлових приміщень, що відповідає сучасним вимогам та потребам користувачів.

1.3. Обґрунтування архітектури системи моніторингу

Архітектура системи, яку ми розробляємо, відповідає концепції Internet of Things (IoT). IoT охоплює кілька аспектів, включаючи самі пристрої, що підключені до мережі та взаємодіють між собою, машини до машини (M2M) для обміну даними без участі людини, а також великі дані, які генеруються пристроями та використовуються для підвищення комфорту або прийняття бізнес-рішень.

Архітектура IoT системи включає кілька рівнів:

- Physical Layer — фізичний рівень. Цей рівень представляє два типи операцій - збір інформації (датчики) та здійснення механічної роботи (виконавчі механізми);

- Edge Layer – рівень периферійного обчислення. Цей рівень зазвичай підключається до одного датчика або виконавчого механізму. Він забезпечує мінімальну функціональність для перетворення аналогової інформації в цифрову та/або навпаки. Для підключення датчиків існують ті ж вимоги щодо ціни та споживаної потужності;
- Local Network Layer – рівень периферійної комунікації;
- Gateway Layer - рівень шлюзу. У IoT-рішенні існує кілька причин наявності рівня шлюзу:
 - Якщо Backend отримуватиме необроблену інформацію, це збільшить його потужність і витрати будуть дуже великі;
 - Робота Backend не може гарантувати реакцію реального часу для великої кількості периферійних пристроїв;
 - Через обмеження безпеки деяка інформація не може бути надіслана до Backend і не може постійно контролюватись людиною. До такої інформації відносяться дані камер вуличного спостереження, медична інформація та ін;
- Wide Network Layer – рівень зовнішнього зв'язку. Цей шар поділяє периферійну та Backend частини загального рішення. Шлюз в основному підключений до Backend з використанням мобільного бездротового зв'язку, такий як 4G / 5G, але іноді використовується дротовий доступ до Інтернету;
- Security Layer - рівень безпеки. Цей рівень забезпечує функції AAA (Authentication, Authorization and Accounting — автентифікація, авторизація та облік) та шифрування/дешифрування разом з іншими послугами, пов'язаними з Інтернетом;
- Middleware Layer – рівень усередині серверного зв'язку. Цей шар забезпечує внутрішню Cloud функціональність балансування навантаження, черги повідомлень та передачі потокової інформації. Компоненти цього шару мають бути дубльовані та автоматично масштабуватися;
- Etl layer - рівень збору, обробки та зберігання даних. Внутрішній рівень ETL (витяг, перетворення та завантаження) є третьою операцією ETL. Перший

був у периферійному пристрої, другий – у шлюзі. Back End ETL накопичує дані з усіх периферійних пристроїв та шлюзів та відповідає за наступні операції;

- Big Data and Analytic Layer – рівень аналітики. Залежить від конкретної програми IoT. Великі дані та аналітичний рівень отримуватимуть ситуативну інформацію з усього набору периферійних пристроїв. Ця частина менш стандартизована, тому що вона сильно відрізняється від одного додатка до іншого через різні завдання рішень;

- Notification layer — рівень сповіщення. На цьому рівні може існувати кілька компонентів, але всі вони мають алгоритм сповіщення про підписку. Клієнтська програма підписується на необхідні події і, коли це відбувається, отримує інформаційний сигнал — повідомлення;

- Presentation Layer - рівень вистави. Додаток IoT може мати два потоки: M2M (від машини до машини) та M2P (від машини до людини). Рівень уявлення, пов'язаний із потоком M2M, де Back End обробляє інформацію та надає її клієнту або інженеру служби підтримки;

- Configuration Layer – рівень конфігурацій. Цей рівень відноситься до обох потоків - M2M і M2P і працює як сховище для трьох типів статусів периферійних пристроїв:

- Актуальний стан периферійного пристрою;
- Новий стан периферійного пристрою, який буде завантажено;
- Проміжний статус периферійного пристрою вказує на процес оновлення від старих станів до нових. Часто цей статус відсутній.

Не все описане вище буде виконане при розробці проекту, але воно нам потрібне щоб розуміти принцип роботи подібних систем.

Для розробки веб-додатку можна використовувати різні патерни, такі як MVC (Model-View-Controller, рис. 1.10), ECS (Entity-Component-System, рис. 1.11), MVP (Model-View-Presenter, рис. 1.12), та MVVM (Model-View-ViewModel, рис. 1.13).

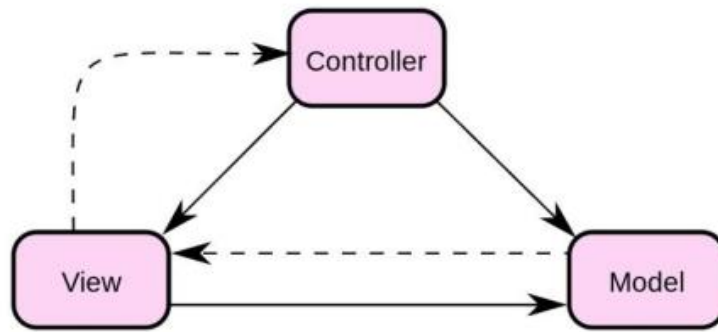


Рис. 1.10. Model-View-Controller

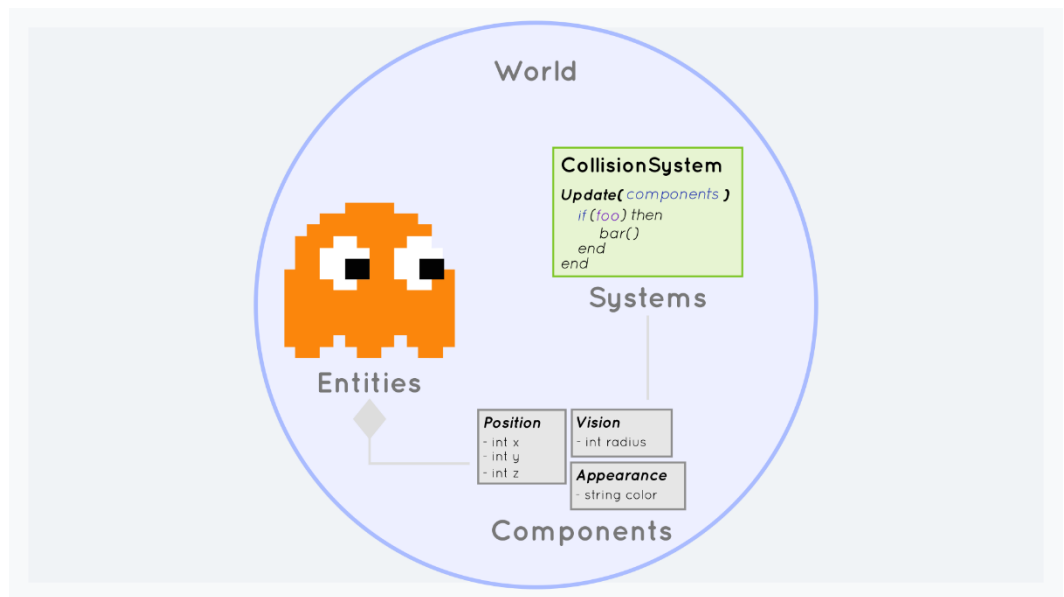


Рис. 1.11. Entity-Component-System

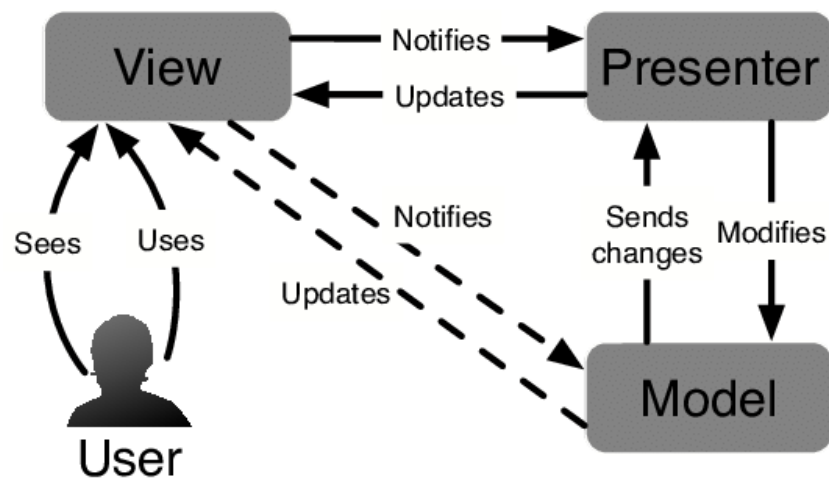


Рис. 1.12. Model-View-Presenter

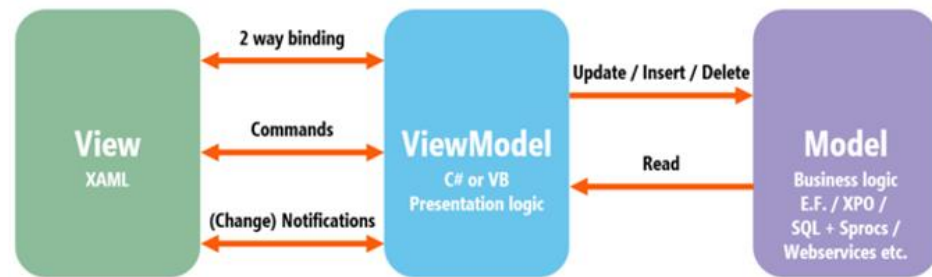


Рис. 1.13. Model-View-ViewModel

Серед них, для розробки веб-сайту, патерн MVC є популярним вибором. Він дозволяє відокремити бізнес-логіку від інтерфейсу користувача, полегшуючи повторне використання коду та підтримку додатку. У MVC модель відповідає за дані та їх структуру, вид відповідає за інтерфейс, а контролер керує компонентами та реагує на дії користувача.

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Інструменти для розробки можна розділити на дві основні частини: сторону сайту (веб-додаток) та сторону приладів.

Веб-додатки мають дві ключові складові:

- Фронтенд (Frontend):
 - Те, що користувач бачить на екрані та з чим взаємодіє.
 - Використовуються HTML, CSS та JavaScript.
 - Фреймворки: React.js, Angular, Vue.js.
- Бекенд (Backend):
 - Включає базу даних, API, логіку сайту та інтеграції.
 - Використовуються мови програмування: PHP, Java, Python, Go, C#, Ruby.
 - Також може бути розроблений на Node.js та інших JavaScript-фреймворках.

Фронтенд відповідає за інтерфейс користувача, а бекенд - за обробку даних та бізнес-логіку. Спільно вони дозволяють користувачам взаємодіяти з сайтом, виконувати операції, такі як реєстрація, оплата замовлення, коментування тощо.

Основні інструменти фронтенд-розробника - HTML для розмітки сторінок, CSS для стилізації та JavaScript для взаємодії. HTML визначає структуру сторінок, CSS - їхній вигляд, а JavaScript додає динаміку та інтерактивність.

Мова для бекенда обирається в залежності від обсягу та завдань проекту. PHP, Java, Python, Go, C# та Ruby - це лише деякі з популярних мов. PHP, наприклад, підходить для невеликих проектів, має велику кількість фреймворків (Yii, CodeIgniter, Laravel) і підтримує різні бази даних.

Для ефективного зберігання та обробки даних у нашому проекті ми обрали базу даних MySQL. Це обумовлено кількома перевагами, які нам важливі:

- Масштабованість: MySQL здатна оптимально обробляти великі обсяги даних, включаючи до 50 мільйонів записів та більше.
- Переносність: MySQL підтримує роботу на різних платформах, таких як Unix, Linux, Windows, OS/2, Solaris, та Mac OS.
- Безпека: База даних має систему контролю доступу, а також забезпечує шифрування даних під час їх передавання.
- Швидкість функціонування: MySQL славиться високою швидкістю роботи та оптимізацією запитів.
- Зручність експлуатації: Легка у встановленні та адмініструванні, MySQL забезпечує зручні інструменти для роботи.

Для розробки сайту ми вибрали фреймворк Yii2. Цей об'єктно-орієнтований фреймворк для PHP реалізує парадигму MVC (Model-View-Controller), дозволяючи максимально використовувати концепцію повторного використання коду та прискорюючи процес веб-розробки.

Для розміщення та отримання даних від пристроїв ми використовуватимемо сервер, який спочатку буде розміщено локально. У цьому контексті для локальної розробки ми вибрали Open Server. Це портативне програмне середовище, спеціально створене для веб-розробників, зручне для

розробки, налагодження та тестування веб-проектів. OSPanel володіє рядом переваг:

- Можливості програми:
 - Непомітна робота у треї Windows.
 - Швидкий старт та зупинка.
 - Автостарт сервера під час запуску програми.
 - Багатофункціональна керуюча утиліта.
 - Можливість монтування віртуального диска.
 - Підтримка керування через командний рядок тощо.
- Надійність: OSPanel існує вже більше 10 років та зарекомендував себе як надійний інструмент для веб-розробки.
- Скромні системні потреби:
 - Мінімальне навантаження на ресурси системи.
- Зручний інструментарій:
 - Простий та зручний інтерфейс.
- Велика кількість модулів готових до використання:
 - Додаткові інструменти для полегшення розробки.

Вибір цих інструментів зумовлений їхньою ефективністю, надійністю та можливістю прискорення процесу веб-розробки.

Для написання коду вибір середовища (IDE) може не відігравати ключову роль, оскільки головне – це результат, а не те, яким інструментом користуватися. Проте, для зручності розробника ми обрали використання Visual Studio Code, розробленої Microsoft, з такими перевагами, як:

- Зручний інтерфейс.
- Безкоштовність.
- Велика кількість користувацьких модулів.
- Підтримка будь-якої мови програмування (якщо встановлено відповідний модуль).

Також важливо використовувати Composer, інструмент для управління залежностями в PHP, який спростить управління бібліотеками та полегшить встановлення Yii.

На стороні приладів для виконання поставленої задачі ми обрали платформу Arduino, зокрема плати Arduino Uno або Arduino Nano (рис. 1.14). Arduino є ідеальною платформою, яка може бути запрограмована для виконання різних завдань, є легкою в зборці та доступною за ціною.



Рис. 1.14. Плати Arduino

Написання коду для Arduino буде виконуватися в середовищі Arduino IDE, яке має велику кількість готових до використання бібліотек, можливість перевірки коду перед завантаженням, наявність прикладів та зручний інтерфейс.

Загалом, вибір архітектури та інструментів розробки здійснено з урахуванням ефективності, надійності та можливості прискорення процесу веб-розробки та реалізації системи зі збору та обробки даних.

1.5. Постановка задачі

Основною метою розглядуваної системи моніторингу житлового приміщення (далі – СМЖ) є створення інтегрованого інформаційного середовища, що дозволить користувачеві отримувати різноманітну інформацію про стан житлового простору, в якому встановлена система. Впровадження СМЖ буде сприяти:

- Спрощенню доступу до інформації з різних частин будівлі, кімнати чи квартири.
- Зменшенню кількості факторів, за якими людина повинна слідкувати вручну.
- Створенню можливості для прогнозування подальших змін у мікрокліматі та умовах житла.

У випускній роботі передбачається розробка веб-додатку у формі веб-сайту, де користувачі матимуть можливість отримувати різноманітну інформацію у вигляді графіків, таблиць та списків.

Основними етапами реалізації цього завдання є:

1.5.1. Розробка серверної частини/сайту системи:

1.5.1.1. Спроектування Баз Даних (БД): Необхідно розробити оптимальну структуру бази даних, що відображатиме параметри мікроклімату та інші дані, зібрані в системі моніторингу.

1.5.1.2. Розробка інтерфейсу користувача: Створити інтуїтивно зрозумілий і естетичний веб-інтерфейс з можливістю перегляду даних в зручному форматі

– Відображення останніх даних у легко сприйнятливому вигляді.

– Графічне відображення динаміки змін параметрів мікроклімату з часовими мітками.

1.5.1.3. Створення точки входу для отримання інформації:

- Реалізація механізму прийому даних від датчиків через

визначену точку входу.

1.5.2. Розробка програмного забезпечення для периферії:

1.5.2.1.Спостереження за датчиками: Розробка програм для зчитування даних від датчиків, їх моніторингу та виявлення аномалій.

1.5.2.2.Збирання та відправлення даних на сервер: Реалізація процесу збору та відправлення даних на центральний сервер для подальшого аналізу.

1.5.2.3.Збереження даних на пристрої: Розробка механізму збереження даних у випадку неможливості їх передачі на сервер.

Результатом виконання цих завдань стане функціональний веб-додаток системи моніторингу житлового приміщення, що надасть користувачам зручний та ефективний інструмент для контролю та оптимізації умов у приміщенні.

Висновки до першого розділу

Визначено основні аспекти архітектури та інструментів для розробки системи. Проведено аналіз вже існуючих на ринку рішень. Вирішено використовувати архітектурний підхід, розподіливши систему на фронтенд (веб-додаток) та бекенд (база даних, API, логіка сайту). Вибрано ключові технології для кожної частини, такі як HTML, CSS, JavaScript для фронтенду та MySQL для бази даних на бекенді.

Для полегшення розробки визначено використання фреймворку Yii2 для PHP. Обрано серверну платформу Open Server для розгортання та тестування веб-проектів локально. Важливим моментом є врахування факторів, таких як можливості програми, надійність, скромні системні вимоги та зручний інструментарій.

Для написання коду визначено використання Visual Studio Code як IDE з великою кількістю користувацьких модулів. Для управління залежностями в PHP буде використовуватись Composer.

Щодо сторони приладів, обрано платформу Arduino, конкретно плату Arduino Uno та Arduino Nano для забезпечення різних функцій.

Загальний висновок полягає в тому, що здійснено обґрунтований вибір технологій та інструментів для розробки системи, враховуючи аналіз існуючих продуктів, ефективність, зручність та надійність. Сформульовані критерії дозволяють створити ефективний веб-проект та автоматизовану систему збору даних на основі Arduino.

РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ПАРАМЕТРІ МІКРОКЛІМАТУ ЖИТЛОВОГО ПРИМІЩЕННЯ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Система моніторингу житлового приміщення створюється з метою надати користувачу можливість отримати інформацію в приміщенні.

Вимоги користувачів.

Користувачі мають доступ до наступних функцій:

1. Швидкий перегляд даних;
2. Перегляд історії змін даних;
3. Перегляд графіку з порівнянням даних;
4. Отримувати повідомлення при певних випадках;



Рисунок 2.1 – Діаграма варіантів використання

Функціональні вимоги:

1. Користувач матиме можливість переглядати інформацію датчиків в приладах, історію змін.
2. Система працює без участі користувача.

Нефункціональні вимоги:

1. Сприйняття:

Час відповіді системи для звичайних запитів не повинен перевищувати 0.1 секунди, а для більш складних запитів — 0.5 секунди.

Інтерфейс веб-додатку повинен бути інтуїтивно зручним для користувача та не вимагати від нього додаткової підготовки.

2. Надійність

3. Доступність - час, потрібний для обслуговування системи не повинен перевищувати 10 % від загального часу роботи.

Аналіз функціональних вимог дозволив виділити наступні сутності, що забезпечать реалізацію програмного комплексу системи. На рис. 2.2. наведено приблизну діаграму класів, які будуть використовуватись.

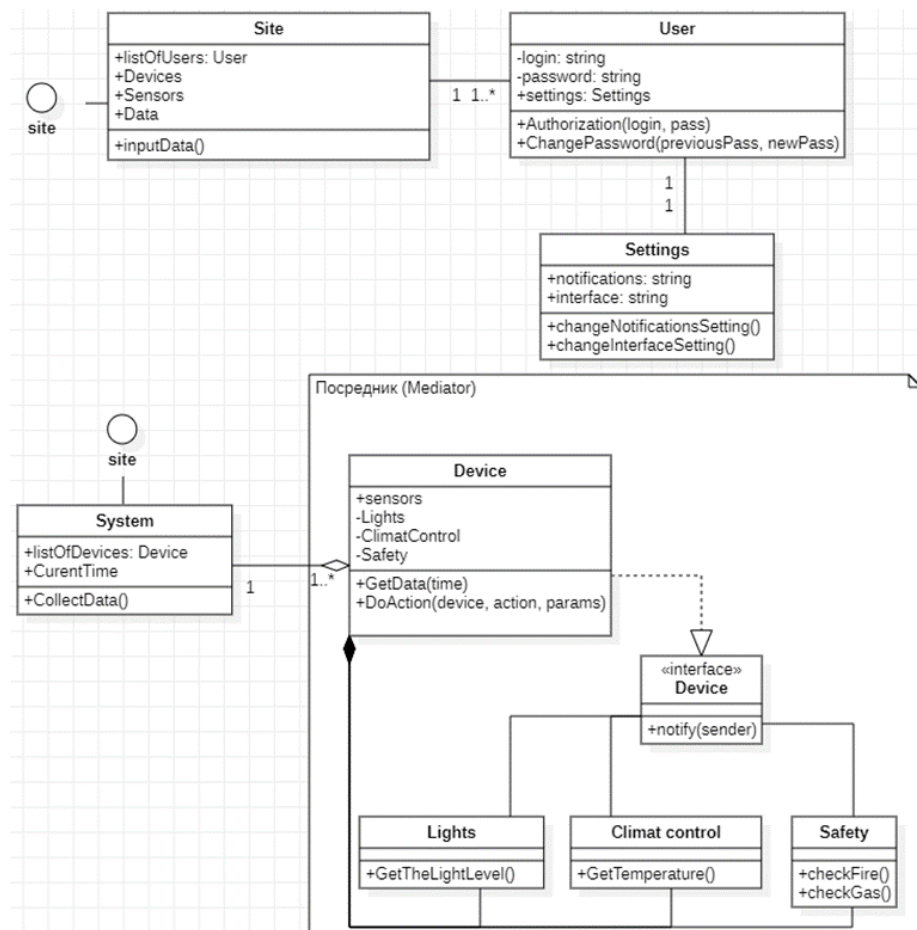


Рис. 2.2. Діаграма класів системи

На спроектованій діаграмі також можна виділити класи, що були

спроектовані з використанням шаблону «Посередник».

Посередник – це поведінковий патерн проектування, який дозволяє зменшити зв'язаність безлічі класів між собою завдяки переміщенню цих зв'язків в один клас-посередник.

Патерн посередник змушує об'єкти спілкуватися не безпосередньо, а через окремий об'єкт-посередник, який знає, кому потрібно перенаправити той чи інший запит. Завдяки цьому компоненти системи залежатимуть тільки від посередника, а не від десятків інших компонентів.

Нижче наведений детальний опис ключових класів та наявних в них методів та атрибутів.

System - клас, який визначає методи та атрибути для системи. Його атрибути:

- listOfDevices – список пристроїв;
- CurentTime – поточний час. Використовується для встановлення часової мітки на отримані від датчиків дані.

Методи класу System:

- CollectData() – метод що по чергово опитує датчики і відправляє все на сайт. Також записує дані в файл якщо відправити не вдалося.

Site - клас, який визначає методи та атрибути що має сайт. Його атрибути:

- listOfUsers – список користувачів;
- Devices – список датчиків на сайті;
- Devices – список відомих сайту датчиків;
- Data – всі дані отримані від датчиків.

Методи класу Site:

- inputData() – скоріше не метод, а реакція сайту на вхідні дані з System.

Клас User - клас, який визначає спільні методи для всіх користувачів. Його атрибути:

- login - поле типу String, яке зберігає ім'я користувача;
- password - поле типу String, яке зберігає пароль;
- settings – посилання на об'єкт налаштувань користувача.

Методи класу User:

- Authorization(login, pass) – вхід до системи;
- ChangePassword(previousPass, newPass) – зміна пароля.

Клас Settings - клас, який визначає спільні методи та атрибути для налаштувань користувачів. Його атрибути:

- notifications: - поле типу String, яке зберігає налаштування сповіщень користувача в форматі JSON;
- interface - поле типу String, яке зберігає налаштування інтерфейсу користувача в форматі JSON.

Клас Device - клас, який визначає спільні методи для всіх пристроїв. Його атрибути:

- sensors - поле, яке зберігає список сенсорів пристрою;
- Lights - поле, яке зберігає компонент;
- ClimatControl - поле, яке зберігає компонент;
- Safety - поле, яке зберігає компонент.

Методи класу Device:

- GetData – отримати дані з пристрою.

2.2. Проектування бази даних системи та структури файлів системи

Для збереження інформації у системі використовується база даних MySQL версії 5.7. В таблицях 2.1-2.6 буде описано структуру БД.

Таблиця 2.1

Детальний опис таблиць

Таблиця	Призначення
user	Таблиця користувачів, генерується автоматично фреймворком Yii
notifications	Таблиця всіх повідомлень для користувачів
settings	Налаштування сайту для кожного користувача
device	Перелік відомих сайту пристроїв
sensorslist	Перелік відомих сайту датчиків

Таблиця	Призначення
sensors	Проміжна таблиця, створена для забезпечення зв'язку “Many-to-many” між таблицями device та sensorslist
device_data	Дані отримані з всіх пристроїв
migration	Таблиця фреймворку Yii, відповідає за контроль міграцій. Міграції по своїй суті це зміна структури БД.
auth_assignment	Автоматично згенеровані таблиця контролю доступу на основі ролей (RBAC). RBAC забезпечує простий, але потужний централізований контроль доступу.
auth_item	
auth_item_child	
auth_rule	

Розглянемо детальніше деякі таблиці та поля, які було створено. Таблиці та поля створені фреймворком буде пропущено, детальніше про них можна дізнатися з офіційного сайту[29][30].

Таблиця 2.2

Детальний опис таблиці device

Поле	Тип	Опис
id	int	Порядковий номер приладу в системі
device_name	varchar(255)	Ім'я приладу, що буде відображатись користувачу
update_at	datetime	Остання дата оновлення даних зв'язаних з цим приладом

Таблиця 2.3

Детальний опис таблиці sensorslist

Поле	Тип	Опис
id	int	Порядковий номер приладу в системі
sysname	varchar(32)	Системне ім'я приладу необхідне для виводу даних в інтерфейсі

Поле	Тип	Опис
title	varchar(64)	Назва вкладок для даного датчика, що буде відображатись користувачу
name	varchar(64)	Ім'я приладу, що буде виводитись серед переліку інших датчиків
hint	string	Опис датчику який виводиться при перегляді інформації про датчик, або в якості спливаючої підказки.
unit	varchar(10)	Одиниця вимірювання, що додається до відображених даних.
quickhint	varchar(64)	Виводиться як назва планки (кругового)
quickhint2	varchar(64)	Якщо датчик надає 2 типи даних виводиться в якості

Таблиця 2.4

Детальний опис таблиці device_data

Поле	Тип	Опис
id	int	Порядковий номер даних в системі
created_at	datetime	Дата створення даних
value	float	Відформатоване значення з датчику
value_type	varchar(16)	Системне поле, необхідне для правильного відображення даних
sensor_id	int	ID сенсору з якого отримано дані
device_id	int	ID пристрою, з якого отримано дані

Для збереження рівню заряду створена окрема таблиця, хоча вона відрізняється від таблиці device_data лише відсутністю деяких стовпців, тому її опис буде також пропущено.

Таблиця 2.5

Детальний опис таблиці settings

Поле	Тип	Опис
id	int	Порядковий номер даних в системі
user_id	int	Посилання на користувача, для якого створено налаштування
interface	json	Налаштування інтерфейсу головного модуля
notifications	json	Налаштування сповіщень користувача
sensor_id	int	ID сенсору з якого отримано дані
device_id	int	ID пристрою, з якого отримано дані

Таблиця 2.6

Детальний опис таблиці notifications

Поле	Тип	Опис
id	int	Порядковий номер даних в системі
user_id	int	Посилання на користувача, для якого створено налаштування
type	char(20)	Тип повідомлення. Необхідно для вибору візуального вигляду
title	text	Заголовок повідомлення
message	text	Текст повідомлення
is_read	int	Чи прочитане повідомлення

На рис. 2.3. наведено діаграму бази, створену на основі описаної вище інформації

- runtime - ця папка призначена для зберігання даних часу виконання
- test - ця папка включає всі тести (приймальні, юніт, функціональні)
- views - ця папка призначена для уявлень, що відображаються контролерами. Макетна папка призначена для шаблону сторінки.
- web - точка входу з Інтернету.

2.3. Проектування алгоритмів роботи системи

Система моніторингу розглядає користувача як спостерігача, який може бачити, але не втручатися в процес. Алгоритм користувача в системі описаний на рисунку в додатку А.

Відповідно до цього алгоритму, після успішного входу користувач потрапляє на головний екран, звідки може перейти до списку приладів, переглянути конкретний пристрій чи повернутися на головний екран. Крім того, користувач може у будь-який момент зайти в меню налаштувань і змінити параметри.

Оскільки головною метою системи є збір даних, розглянемо послідовність дій системи при необхідності відправити дані (рис. 2.4). Згідно з цією діаграмою, зібрані дані пристроєм форматуються та зберігаються в системі до моменту успішної передачі на сервер.

У випадку невдачі (відсутність зв'язку з сервером або помилка на сервері) дані зберігаються локально до моменту відправки, після чого вони видаляються. Це забезпечує системі надійність, що дані будуть успішно передані на сервер.

Оскільки дані потрібно не лише передати на сервер, а й зберегти, надалі буде розглянуто алгоритм збереження даних на сервері (див. додаток Б).

Після отримання даних сервер потім додає дані з пристрою в таблицю даних, без необхідності попереднього їх перетворення, оскільки це вже було зроблено пристроєм в момент отримання даних з датчиків.

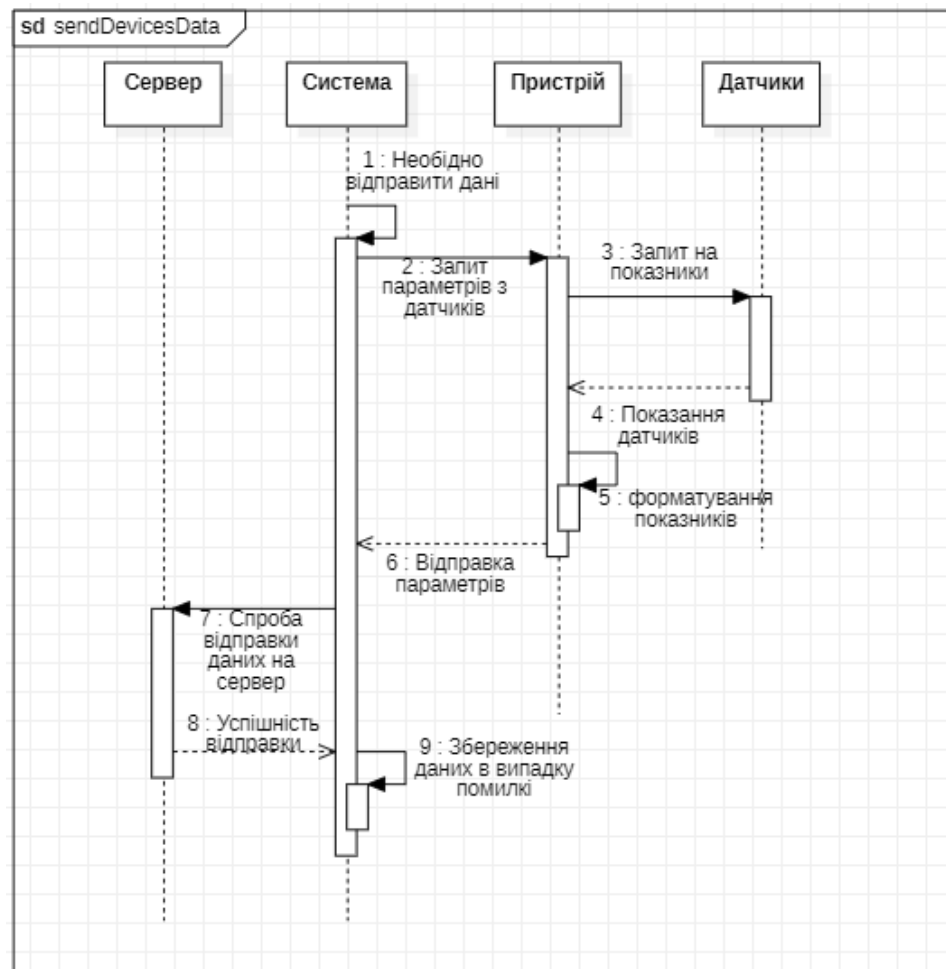


Рис. 2.4. Алгоритм збору та аналізу даних з сенсорів мікроклімату

Потім відбувається перевірка "підписки" користувача на певну подію, пов'язану із даним пристроєм. Це надає можливість створити для кожного зв'язку "користувач-прилад" власні налаштування повідомлень.

2.4. Обґрунтування засобів реалізації для веб-орієнтованої системи моніторингу параметрів мікроклімату житлового приміщення

Використовуючи фреймворк Yii2, який дозволяє підключати модулі, розроблені іншими користувачами, ми спростимо розробку, використовуючи готові рішення для задоволення наших потреб. Наприклад, для ключового елементу системи - графіків, ми вирішили використовувати JavaScript-скрипт ApexCharts.js[26]. Цей скрипт дозволяє створювати різноманітні графіки, такі як лінійні, стовпчасті, теплові карти, прогрес-бари та інші.

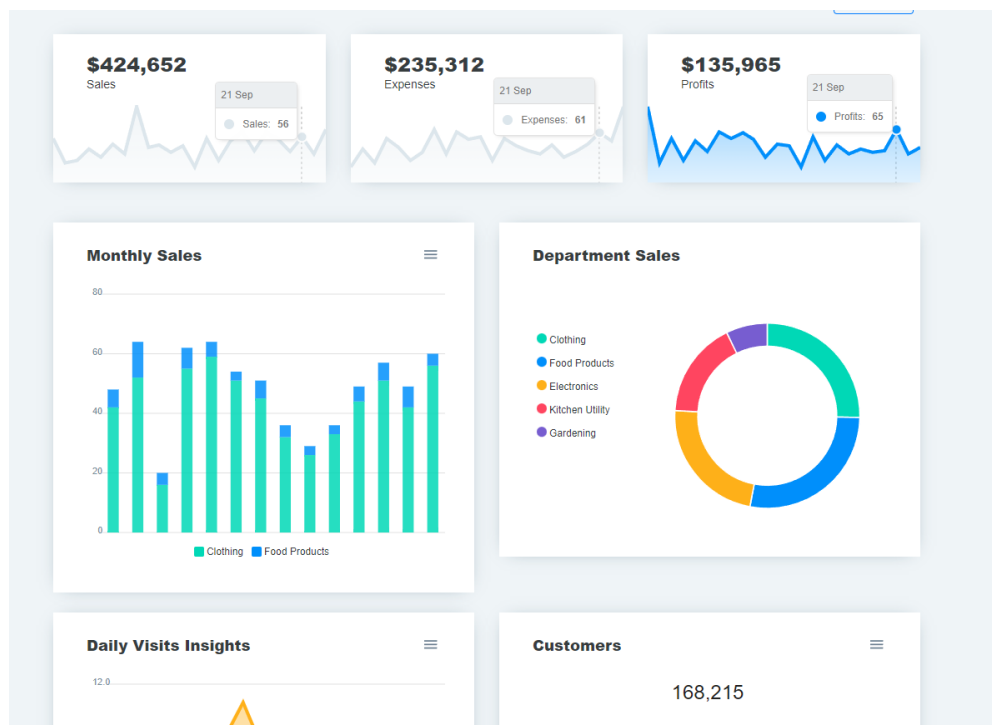


Рис. 2.5. ApexCharts.js дашборд

У процесі розробки також будуть використані спеціально розроблені для Yii2 віджети та модулі від Krajee[33]. Ці рішення мають значний набір готових компонентів, які легко встановлюються за допомогою Composer.

Один із важливих для нас модулів - це модуль Grid, який замінює аналогічний стандартний модуль у Yii і надає можливість використовувати більш гнучку таблицю з розширеним функціоналом, таким як вивід додаткового рядка інформації про елемент таблиці.

Library

* Resize table columns just like a spreadsheet by dragging the column edges.

		Name	Color	Author	Buy Amount (\$)	Sell Amount (\$)	Buy - Sell (\$S)	Buy % (100 * B/S)	Status	Actions
				Any author					Select...	
1		The Great Gatsby	Blue	F. Scott Fitzgerald	209.00	300.00	509.00	58.94		
2		The Grapes of Wrath	Red	John Steinbeck	104.00	155.50	259.50	59.92		

Book Details The Grapes of Wrath

Published: 1939-07-18

Buy Amount Breakup

#	Breakup	Price
1	Base Price	93.60
2	Tax @ 6%	6.24
3	Shipping @ 4%	4.16
Total		104.00

Sell Amount Breakup

#	Breakup	Price
1	Base Price	139.95
2	Tax @ 6%	9.33
3	Shipping @ 4%	6.22
Total		155.50

3		Nineteen Eighty-Four	Orange	George Orwell	79.00	98.50	177.50	55.49		
4		Ulysses	Green	James Joyce	69.00	98.50	167.50	58.81		
5		Lolita	Pink	Vladimir Nabokov	58.00	85.50	143.50	59.58		
6		Catch-22	Cyan	Joseph Heller	85.00	105.50	190.50	55.38		
7		The Catcher in the Rye	Red	J.D. Salinger	105.00	125.50	230.50	54.45		
8		Beloved	Green	Toni Morrison	130.00	154.99	284.99	54.38		
9		The Sound and the Fury	Brown	William Faulkner	106.00	125.50	231.50	54.21		
10		To Kill a Mockingbird	Orange	Harper Lee	79.00	98.50	177.50	55.49		
Total					1,024.00	1,347.99	2,371.99	56.67	(amounts in \$)	

* The page summary displays SUM for first 3 amount columns and AVG for the last.

12

Download Selected

Рис. 2.6. Демонстрація Grid від Krajee

Також серед обраних модулів для виконання нашого завдання ми використали модуль TabX[35], який дозволяє створювати вкладки на сторінці, аналогічні вкладкам браузера. Це дає можливість динамічно відображати графіки або виділяти інформацію окремими вкладками (див. рисунок 2.7).

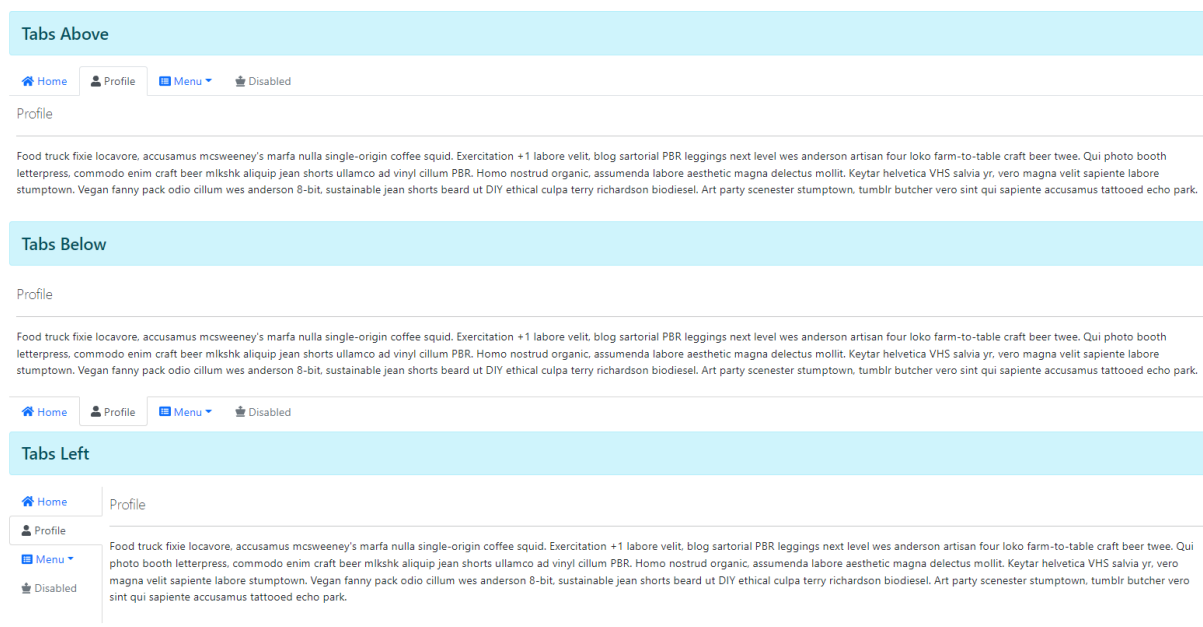


Рис. 2.7. Демонстрація TabX від Krajee

Кожен елемент масиву \$item представляє собою готову для відображення сторінку HTML. Це дозволяє нам передбачити дані на сторінці заздалегідь або використовувати Аях для отримання їх при необхідності. З метою спрощення завдання був обраний перший варіант.

Крім того, на нашому веб-сайті реалізовано відображення сповіщень за допомогою двох модулів від Krajee: PopoverX[36] та Alert[37].

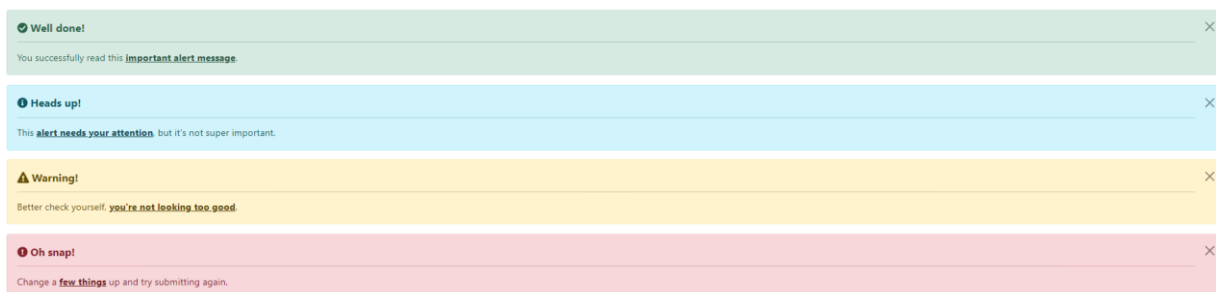


Рис. 2.8. Демонстрація Alert від Krajee

Всі описані модулі та скрипти будуть використані для реалізації веб-додатку в майбутньому.

Висновки до другого розділу.

У другому розділі викладено основні етапи проектування системи моніторингу житлового приміщення. Визначено варіанти використання та об'єктно-орієнтовану структуру системи. Розглянуті вимоги користувачів, функціональні та нефункціональні вимоги.

Підкреслено важливість забезпечення швидкості відповіді системи, інтуїтивно зручного інтерфейсу веб-додатку та надійності системи.

Описано алгоритм взаємодії користувача з системою, зокрема послідовність дій при відправці даних і їх подальшому збереженні на сервері. Виділено використання фреймворку Yii2 та деяких готових рішень, таких як ApexCharts.js, для реалізації графіків, а також віджетів та модулів від Krajee для полегшення розробки і забезпечення ряду функціональних можливостей.

У цілому, розділ висвітлює етапи розробки системи моніторингу та визначає ключові компоненти, інструменти та технології, які будуть використовуватись для її успішної реалізації.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ПАРАМЕТРІВ МІКРОКЛІМАТУ ЖИТЛОВОГО ПРИМІЩЕННЯ

3.1. Встановлення та налаштування системи

Система моніторингу житлового приміщення спроектована для локального використання і має просту структуру, яку можна побачити на схемі розгортання (рис 3.1).

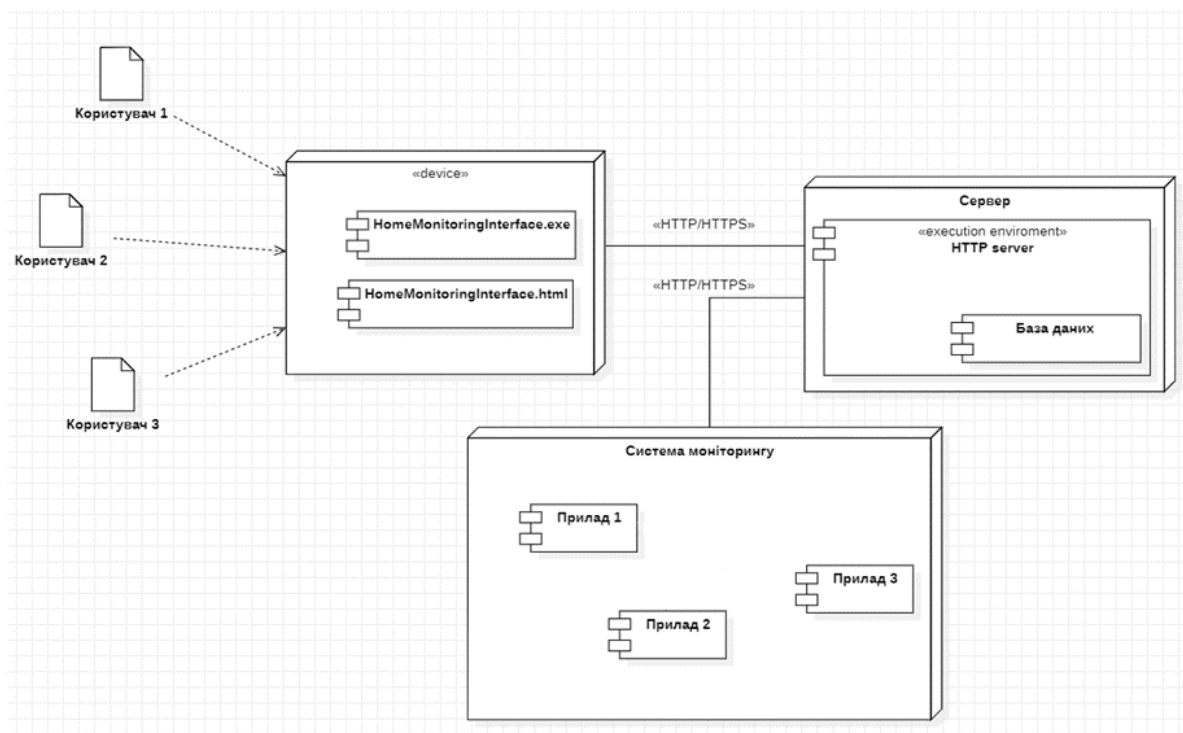


Рис. 3.1. – Діаграма розгортання

Для встановлення системи потрібні наступні елементи:

- Сервер: В даному випадку в якості сервера використовується OpenServer. Він дозволяє розміщувати сайт на комп'ютері та містить базу даних для зберігання інформації від системи моніторингу.
- Система моніторингу: Плата Arduino в парі з WiFi модулем esp8266, використовується для збору інформації.
- <<Device>>: Це може бути будь-який пристрій з веб-браузером, який надає доступ до системи.

Оскільки веб-додаток системи виконується в середовищі OpenServer, мінімальні системні вимоги визначаються відповідно до вимог програми:

- Операційні системи: Windows 7 SP1 x64 / Windows Server 2008 R2 SP1 або новіші.
- Вільні апаратні ресурси: Мінімум 500 МБ оперативної пам'яті та 10 ГБ вільного місця на диску.
- Системне програмне забезпечення: MSVC++ 2005-2022 Redistributable Packages (в комплекті).

3.2. Реалізація веб-додатку

Як було описано в розділі 2, для графіків, ми вирішили використовувати JavaScript-скрипт ApexCharts.js[26].

Оскільки для відображення графіків використовується JavaScript, сервер не обов'язково буде графіки самостійно. Браузер користувача виконує цю задачу, а серверу потрібно лише передати розмітку та відповідний скрипт створення графіку.

Для передачі розмітки і скрипта використовується файл _dataChart.php. Його виконання передає інформацію про датчик у змінній \$sensor та самі дані у змінній \$data. Оскільки ці змінні є об'єктами відповідних моделей, вони можуть використовувати їх функції для коректного відображення даних.

Файл _dataChart.php:

```
<?
$name = $sensor->sysname;
?>
<div id="<?= $name ?>Chart" class="chart">

</div>
<script>
    let <?= $name ?>ChartOptions = {
        chart: {
            locales: ourlocale,
            defaultLocale: "ua",
            height: 380,
            type: "line",
            foreColor: '#6D6D6D'
```

```

    },
    series: [{
        <? if ($sensor->id != 1) { ?>
            name: "<?= $sensor->quickhint ?>",
            data: [<?
                foreach ($data as $data_item) :
                    if ($data_item->sensor_id == $sensor->id) :
                        ?>['<?= str_replace(' ', 'T', $data_item["created_at"])
?>', <?= $data_item["value"] ?>],
                        <?
                            endif;
                        endforeach; ?>
                    ]
                <? } else { ?>
                    name: "Температура",
                    data: [<?
                        foreach ($data as $data_item) :
                            if ($data_item->sensor_id == $sensor->id &&
$>data_item->value_type == "temperature") :
                                ?>['<?= str_replace(' ', 'T', $data_item["created_at"])
?>', <?= $data_item["value"] ?>],
                                <?
                                    endif;
                                    endforeach; ?>
                            ]],{
                                name: "Вологість",
                                data: [<?
                                    foreach ($data as $data_item) :
                                        if ($data_item->sensor_id == $sensor->id && $data_item-
>value_type == "humidity") :
                                            ?>['<?= str_replace(' ', 'T', $data_item["created_at"])
?>', <?= $data_item["value"] ?>],
                                            <?
                                                endif;
                                                endforeach; ?>
                                        ]
                                    <? } ?>
                                }
                            ],
                            grid: {
                                borderColor: '#6D6D6D'

                            },
                            stroke: {
                                curve: 'smooth'
                            },
                            yaxis: {
                                min: 0,
                                max: 45
                            },

```

```

xaxis: {
  type: 'datetime',
  labels: {
    datetimeFormatter: {
      year: 'yyyy',
      month: 'MMM \'yy',
      day: 'dd MMM',
      hour: 'HH:mm'
    },
    datetimeUTC: false
  }
},
noData: {
  text: "No data text",
  align: "center",
  verticalAlign: "middle",
}
};
var <?= $name ?>Chart = new ApexCharts(document.querySelector("#<?= $name
?>Chart"), <?= $name ?>ChartOptions);
<?= $name ?>Chart.render();
</script>

```

Скрипт буде створений для відображення обраних типів даних у формі кругового індикатора при швидкому перегляді (рисунок 3.2), формі графіка під час докладного перегляду на сторінці пристрою (рисунок 3.3), або разом на головному екрані (рисунок 3.4).

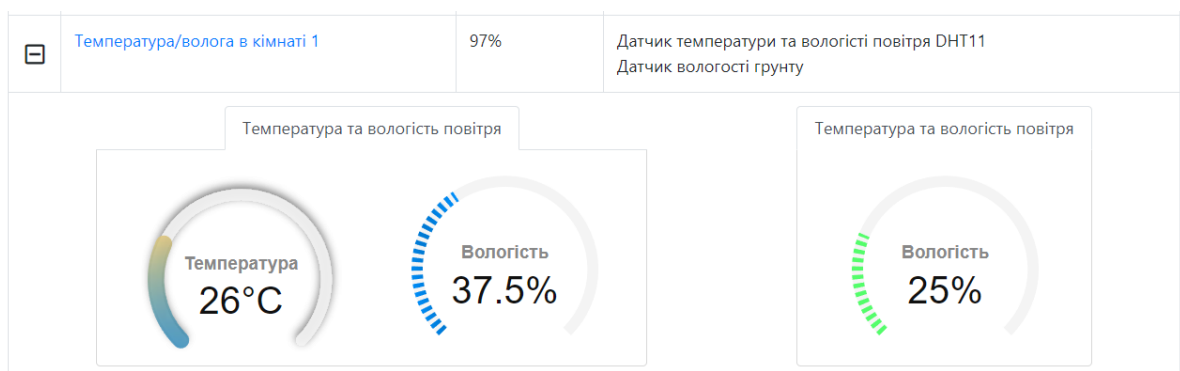


Рис. 3.2. Швидкий перегляд даних в списку приладів

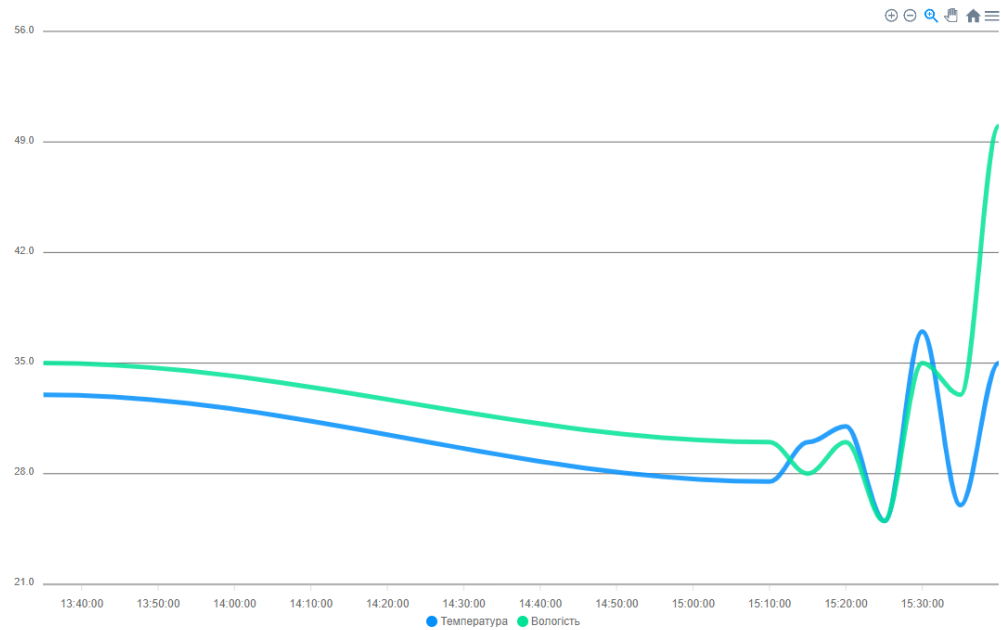


Рис. 3.3. Детальний перегляд даних приладу

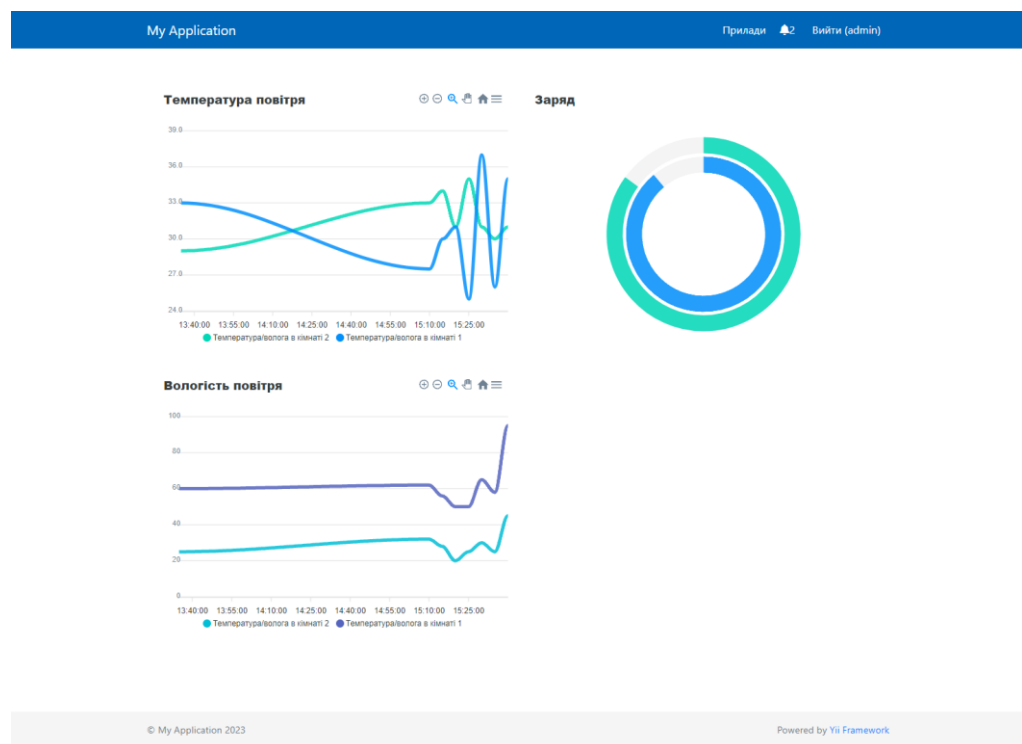


Рис. 3.4. Головний екран веб-додатку

Далі перейдемо до модуля Grid, який ми використаємо в списку приладів для відображення його в вигляді таблиці з можливістю розкрити кожен рядок для швидкого перегляду даних. Для створення такої рядка, згідно з документацією[34], одну з колонок потрібно налаштувати наступним чином:

```

[
    'class' => 'kartik\grid\ExpandRowColumn',
    'width' => '50px',
    'value' => function ($model, $key, $index, $column) {
        return GridView::ROW_COLLAPSED;
    },
    // uncomment below and comment detail if you need to render via
ajax
    // 'detailUrl' => Url::to(['/site/book-details']),
    'detail' => function ($model, $key, $index, $column) {
        return Yii::$app->controller->renderPartial('_expand-row-
details', [
            'model' => $model,
            'sensors' => Sensors::find()->where(['device_id' => $model-
>id])->orderBy('sensor_id')->asArray()->all(),
        ]);
    },
    'headerOptions' => ['class' => 'kartik-sheet-style'],
    'expandOneOnly' => true,
    'detailRowCssClass' => ''
],

```

В елемент з бистрим переглядом даних «detail» ми записуємо функцію, яка після виконання поверне нам готову сторінку, на якій потім JavaScript функція виведе дані в таблиці в описаному раніше вигляді (рис. 3.5).

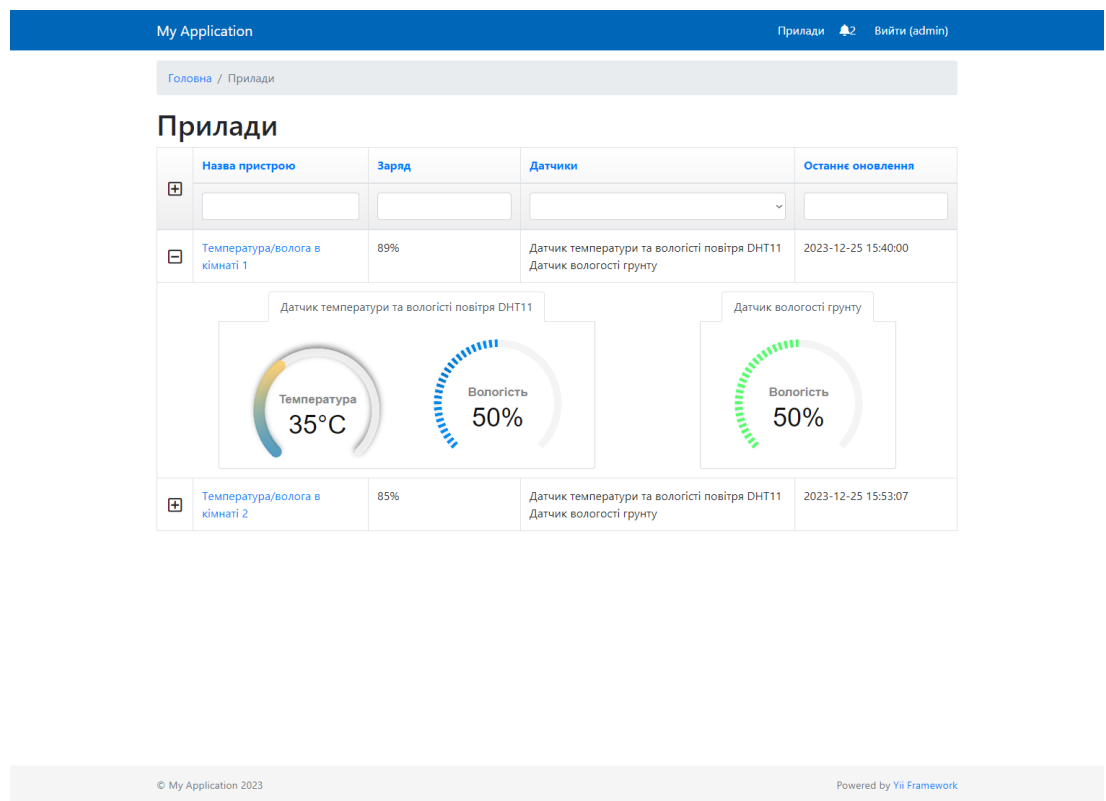


Рис. 3.5. Детальний перегляд даних приладу

На сторінці детального перегляду приладу ми використовуємо модуль TabX, для розділення інформації (рис. 3.6). Структура створення подібних вкладок виглядає наступним чином:

```
$items = [];
foreach ($model->sensors as $sensor) {
    $sensor = SensorsList::findOne($sensor['sensor_id']);
    $items[] = [
        'label' => $sensor->title,
        'content' => $this->render('charts/_dataChart', ['data' => $model-
>data, 'sensor' => $sensor]),
    ];
}
$items[] = [
    'label' => 'Рівень заряду',
    'content' => $this->render('charts/_chargeChart', ['data' => $model-
>remainingCharges]),
];
echo TabsX::widget([
    'items' => $items,
    'position' => TabsX::POS_ABOVE,
    'align' => TabsX::ALIGN_CENTER,
    'bordered' => true,
    'encodeLabels' => false
]);
```



Рис. 3.6. Розділення інформації між вкладками

Коли дані що прийшли з пристрою будуть виходити за рамки встановлені системою, ми будемо генерувати повідомлення для користувача з допомогою

PopoverX[37] та Alert[38] (рис. 3.7). Для створення вікон повідомлень і попапу з ними використовується наступний фрагмент коду:

```
$notificationsList = Notifications::find()->asArray()->where(['user_id' =>
Yii::$app->user->getId(), 'is_read' => 0])->all();
// var_dump($notificationsList);
$notifications = "";
foreach ($notificationsList as $notification) :
    $notifications .= Alert::widget([
        'type' => Notifications::$notificationTypeList[$notification['type']],
        'title' => $notification['title'],
        'body' => $notification['message'],
        'closeButton' => [
            'id' => 'nav-close-notofication-' . $notification['id'],
            'data-notificationid' => $notification['id'],
            'data-something' => "bar",
        ],
        'showSeparator' => true,
        'delay' => 0
    ]);
    $this->registerJs('$(\'#nav-close-notofication-' . $notification['id'] .
'\').on(\'click\', function(e){
    $.ajax({
        dataType: "json",
        type: "POST",
        url: "/notificationread",
        data: { "id" : "' . $notification['id'] . '" },
        success: function (data) {
            console.log(data);
        },
        error: function (errorMessage) {
            alert("Невдалося прочитати повідомлення");
        }
    });
    $("#notfi_count").text($("#notfi_count").text()-1);
});');
endforeach;
?>
```

```
<form class="form-inline float-right">
    <?= PopoverX::widget([
        'header' => 'Сповіщення',
        'placement' => PopoverX::ALIGN_BOTTOM_RIGHT,
        'size' => 'md',
        'content' => $notifications,
        'closeButton' => false,
        'toggleButton' => [
            'label' => Html::tag(
                'span',
                '',
```

```

        ['class' => 'fas fa-bell']
    ) .
    Html::tag(
        'span',
        count($notificationsList),
        ['id' => 'notfi_count']
    ),
    'class' => 'btn btn-link logout'
],
'options' => ['class' => 'navNotification '],
'pluginOptions' => [
    'dialogCss' => ['z-index' => 1051], // will overlay the popover over
the navbar
]
])
?>
</form>

```

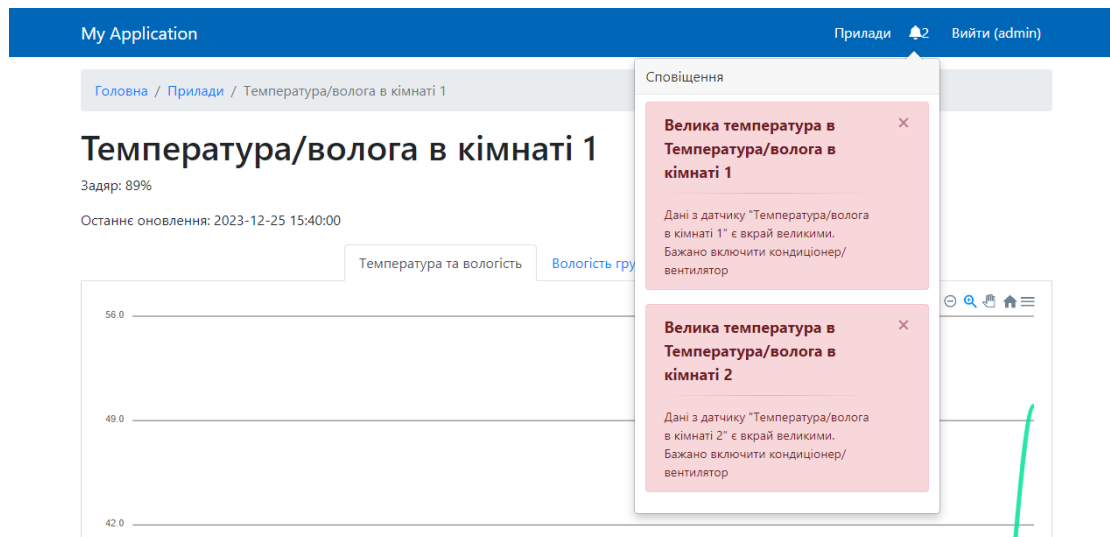


Рис. 3.7. Розділення інформації між вкладками

3.3. Результат виконання роботи

При вході на веб-сайт користувача зустріне вікно авторизації (рис. 3.8), де йому необхідно ввести свої облікові дані. Реєстрація нового користувача не передбачена, ініціює її адміністратор, який створює обліковий запис для користувача.

My Application

Вхід

Заповніть поля нижче щоб авторизуватись:

Логін

❗

Логін cannot be blank.

Паролю

☒ Запам'ятати мене

Login

© My Application 2023

Powered by [Yii Framework](#)

Рис. 3.8. Екран входу в веб-додаток

Після входу користувач переходить до головного меню (рис. 3.9), на якому виводиться інформація з датчиків 5 останніх оновлених приладів. Проте на момент написання звіту приладів в тестовому режимі лише 2.

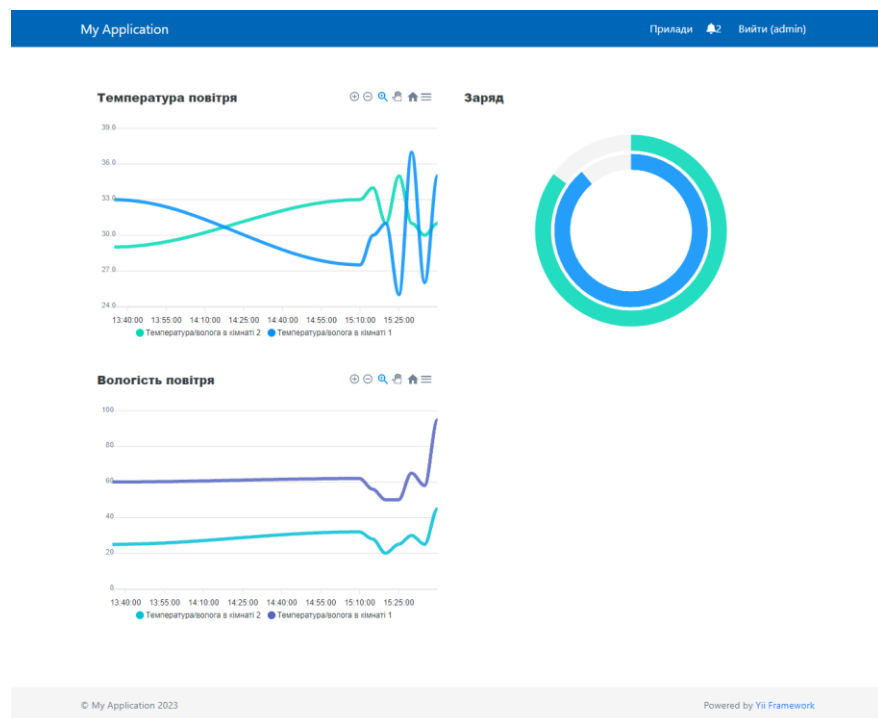


Рис. 3.9. – Головний екран веб-додатку

На даний момент після переходу на головний екран у користувача є вибір між переходом до списку приладів, переглядом сповіщень, та виходом з аккаунта. При переході до списку приладів буде відкритий екран з таблицею,

описаною в попередньому розділі. На рисунку 3.10 уже використано одну з можливостей нового Grid, швидкий перегляд, розкриттям чього перегляду у всіх елементів.

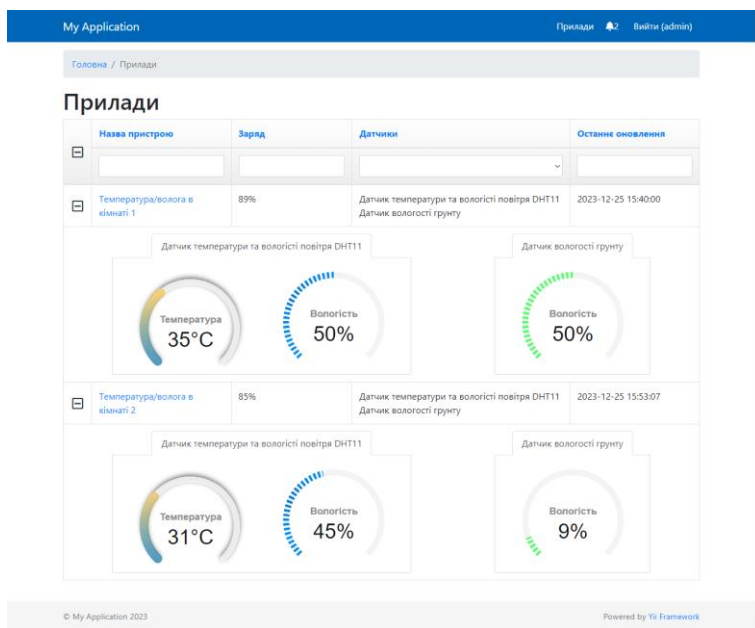


Рис. 3.10. – Екран списку приладів, з розкритими швидким переглядом

У разі, якщо користувач клікне на назву конкретного пристрою, він буде перенаправлений на сторінку детального перегляду. На цій сторінці вся інформація щодо даного пристрою відображається у вигляді графіка, таблиці чи списку (рис. 3.11).

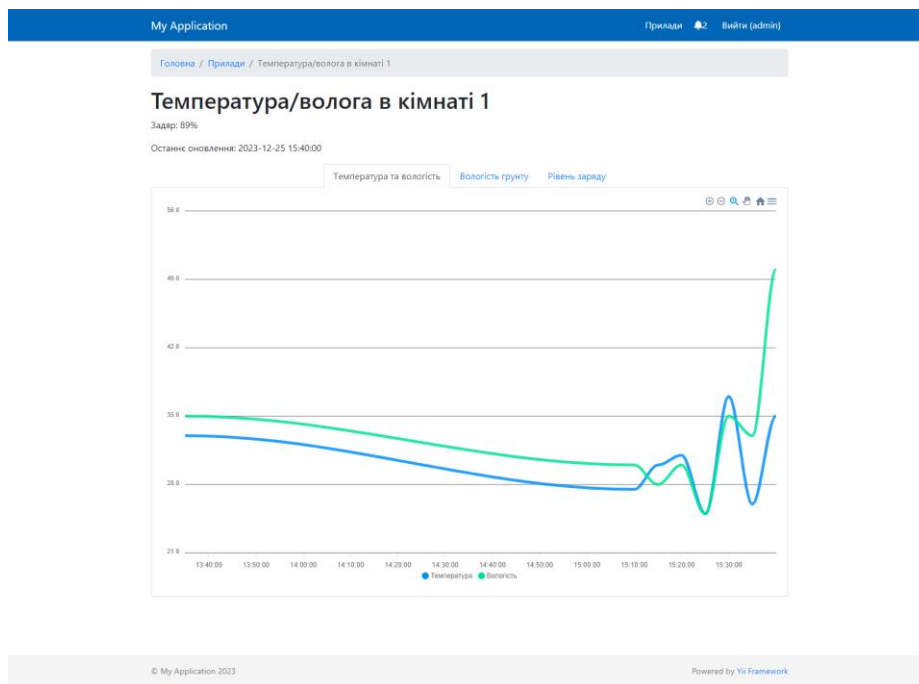


Рис. 3.11. – Екран списку приладів, з розкритими швидким переглядом

На даний момент після переходу на головний екран у користувача є вибір між переходом до списку приладів, переглядом сповіщень, та виходом з аккаунта. При переході до списку приладів буде відкритий екран з таблицею, описаною в попередньому розділі.

Висновок з третього розділу

У третьому розділі магістерської кваліфікаційної роботи "Реалізація веб-додатку для системи моніторингу житлових приміщень" була детально представлена практична реалізація розробленого веб-додатку. Описано встановлення та конфігурацію основних компонентів системи, зокрема сервера OpenServer, а також зазначено необхідні елементи, такі як плата Arduino та WiFi модель esp8266, що використовуються для збору інформації.

Зазначено, що використано JavaScript-скрипт ApexCharts.js для відображення графіків, і надано приклад коду для передачі розмітки та скрипту графіку. Описано використання модуля ExpandRowColumn для розгортання деталей приладів та використання вкладок для розділення інформації на різні аспекти.

У розділі "Результат виконання роботи" продемонстровано реальний вигляд веб-додатку та його функціональні можливості. Користувач може легко переглядати стан приладів, детально вивчати дані та отримувати повідомлення про події.

ВИСНОВКИ

В рамках цієї магістерської кваліфікаційної роботи було представлено підхід до розробки та реалізації веб-орієнтованої системи моніторингу параметрів мікроклімату житлового приміщення, що призначеної для збору та відображення даних щодо параметрів мікроклімату житлового простору.

1. Проведено аналіз проблем та обґрунтовано необхідність впровадження системи моніторингу для ефективного контролю за різними параметрами мікроклімату у приміщенні. Також були визначені ключові вимоги до системи, серед яких зручний інтерфейс, можливість відслідковування даних та надійність передачі інформації.

2. Проведено аналіз архітектури системи та основні етапи взаємодії користувача із системою. Проаналізовано можливі послідовності дій користувача в системі, починаючи від входу до головного екрану та закінчуючи передачею даних на сервер. Також детально описано використання фреймворку Yii2 та різноманітні модулі, які сприяють швидкій та ефективній розробці. Зокрема, використано готові рішення для графіків (ApexCharts.js), таблиць (Krajee Grid), та вкладок (TabX). Такий підхід дозволяє значно полегшити та прискорити процес створення функціональних елементів системи.

3. Здійснено опис інтерфейсу та порядок роботи з системою. Було підкреслено, що система моніторингу призначена переважно для локального використання, і використовує просту структуру для забезпечення зручності та легкості в управлінні. Описано процес встановлення та налаштування системи, вказані мінімальні системні вимоги для різних компонентів.

Інтерфейс користувача є інтуїтивно зрозумілим та забезпечує зручний доступ до важливої інформації. Використання сучасних технологій та готових рішень дозволяє покращити продуктивність розробки та забезпечити ефективну взаємодію із системою. Інтеграція віджетів та модулів від Krajee сприяє розширенню можливостей та поліпшенню візуального представлення даних.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Adiline Macriga, "Smart Home Monitoring and Controlling System Using Android Phone," International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 11, pp. 426,427, November 2013.
2. Hall, Fraser, et al. "Smart homes: security challenges and privacy concerns." arXiv preprint arXiv:2010.15394 (2020).
3. Blyler J. 8 Critical IoT Security Technologies. Home | Electronic Design. URL: <https://www.electronicdesign.com/markets/automation/article/21805420/8-critical-iot-security-technologies>.
4. Davidovic, Boban, and Aleksandra Labus. "A smart home system based on sensor technology." Facta Universitatis, Series: Electronics and Energetics 29.3 (2015): 451-460.
5. Bluetooth Low Energy (BLE): A Complete Guide. Novel Bits. URL: <https://novelbits.io/bluetooth-low-energy-ble-complete-guide/>.
6. Basu, Debraj, et al. "Wireless sensor network based smart home: Sensor selection, deployment and monitoring." 2013 IEEE Sensors Applications Symposium Proceedings. IEEE, 2013.
7. Bregman, David, and Arik Korman. "A universal implementation model for the smart home." International Journal of Smart Home 3.3 (2009): 15-30
8. Jiang , Hongbo, et al. "Smart home based on WiFi sensing: A survey." IEEE Access 6 (2018): 13317-13325
9. Guo, Xiao, et al. "Review on the application of artificial intelligence in smart homes." Smart Cities 2.3 (2019): 402-420.
10. Soliman, Moataz, et al. "Smart home: Integrating internet of things with web services and cloud computing." 2013 IEEE 5th international conference on cloud computing technology and science. Vol. 2. IEEE, 2013
11. Dengler, Sebastian, Abdalkarim Awad, and Falko Dressler. "Sensor/actuator networks in smart homes for supporting elderly and handicapped people." 21st International Conference on Advanced Information Networking and Applications

12. Workshops (AINAW'07). Vol. 2. IEEE, 2007. Zhongyang, Y. Jiadi, Z. Yanmin, C. Yingying, L. Minglu // IEEE International
13. Dominguez A., Bailey J., Djermanović D. Saving Data on Android: Learning Room, Firebase and SQLite with Kotlin / A. Dominguez, J. Bailey, D. Djermanović // Razeware LLC, 2019 — P. 306.
14. Alimdjanovna, Xalilova Muhabbat. "CLIMATE CONTROL AND LIGHT CONTROL IN A SMART HOME." European Journal of Interdisciplinary Research and Development 8 (2022): 149-155.
15. Harrison B.L., Consolvo S., Choudhury . Using multi-modal sensing for human activity modeling in the real world / B.L.Harrison, S.Consolvo, T.Choudhury // Handbook of Ambient Intelligence and Smart Environments, Springer, 2010. — P. 463-478.
16. Гашинська А.С. Сучасні системи створення мікроклімату чистих приміщень лікувально-профілактичних закладів [електронний ресурс]: Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fbtegp/all-fbtegp-2018/paper/download/3903/3251>
17. Jiankai Dong, Haixia Lan, Yu Liu, Xiaotao Wang, Chenchen Yu. “Indoor environment of nearly zero energy residential buildings with conventional air conditioning in hot-summer and cold-winter zone”. Energy and Built Environment vol. 3, iss. 2, 2022 pp. 129-138. <https://doi.org/10.1016/j.enbenv.2020.12.001>.
18. Golinko I., Drevetskiy V. An optimization of a digital controller for a stochastic control system. IAPGOŚ vol. 2019. pp 74-77 <https://doi.org/10.35784/IAPGOS.240>.
19. Korbut V., Voznyak O., Myroniuk K., Sukholova I. “Examining a device for air distribution by the interaction of counter non-coaxial jets under alternating mode”. Eastern European Journal of Enterprise Technologies vol. 2 iss 8(86). 2017, pp. 30-38. <https://doi.org/10.15587/1729-4061.2017.96774>.
20. Sukholova I., Vozniak O. “Modeliuvannia povitrorozpodilu v prymishchenni u zminnomu rezhymi za dopomohoiu modeli turbulentnosti Spalarta-Almarasa”. Visnyk NU “LP” “Teoriia i praktyka budivnytstva” iss. 844 pp. 26-33.

<https://science.lpnu.ua/sctp/all-volumes-and-issues/volume8442016/modelyuvannya-povitrorozpodilu-vprimishchenni-u>.

21. Dovhaliuk V., Shyshyna M., Moskvitina A. “Rozrakhunok system ventyliatsii ta okholodzhennia povitria hromadskykh budivel”. Vents Magazine no 2 2018, p.70-73.

https://ukrblog.vents.ua/wpcontent/uploads/2018/12/VM_2_2018_UA1.pdf.

22. Dziubenko V. H., Mileikovskiy V. O., Sachenko I. A. “Expansion of the Range of Wet Air I-D Diagram for Environmental Safe Heat Production”. Екологічна безпека та природокористування vol. 26 no 2 2018, p. 15-22.

<https://doi.org/10.32347/2411-4049.2018.2>.

23. Model CR-1A hygrometer with autofill: Operating manual. Buck Research Instruments L.L.C., 2012. 26 P. URL: <http://www.hygrometers.com/wp-content/uploads/CR-1A-users-manual-2009-12.pdf>. Access date 20.02.2022. Archived 04.04.2023. URL:

<http://web.archive.org/web/20230404194623/https://www.hygrometers.com/wp-content/uploads/CR1A-users-manual-2009-12.pdf>.

24. Dovhaliuk V., Mileikovskiy V. New Approach for Refined Efficiency Estimation of Air Exchange Organization. International Journal of Engineering & Technology, 2018, Vol. 7, Iss. 3.2, p. 591-596. <https://doi.org/10.14419/ijet.v7i3.2.14596>

25. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування. [Чинний від 2014-01-01]. – К. : Інститут «УкрНДІспецбуд»: 2013. – 147 с. – (Нормативний документ Держбуд України. Державні будівельні норми, редакція від 25.01.2013 року).

26. ApexCharts.js - Open Source JavaScript Charts for your website [Електронний ресурс] – Режим доступу до ресурсу: <https://apexcharts.com>

27. Сукач, С.В. Автоматизована система контролю та керування параметрами повітряного середовища приміщень / С.В. Сукач, О.В. Мозговой, М.А.Кобилянський, О.Л.Величко // Електромеханічні і енергозберігаючі системи. Щоквартальний науково-виробничий журнал.— Кременчук: КрНУ,

2012. – Вип. 4/2012 (20). – С. 127–132.

28. Newest 'yii2' Questions - Stack Overflow [Електронний ресурс] – Режим доступу до ресурсу: <https://stackoverflow.com/questions/tagged/yii2>

29. Security: Authorization | The Definitive Guide to Yii 2.0 | Yii PHP Framework [Електронний ресурс] – Режим доступу до ресурсу: <https://www.yiiframework.com/doc/guide/2.0/en/security-authorization>

30. User, yii\web\User | API Documentation for Yii 2.0 | Yii PHP Framework [Електронний ресурс] – Режим доступу до ресурсу: <https://www.yiiframework.com/doc/api/2.0/yii-web-user>

31. Метод і засоби контролю та управління якістю повітряного середовища у приміщеннях. Монографія / С.В.Сукач, Ю.І.Шульга. – Кременчук: Видавець ПП Щербатих О.В., 2013. – 192 с

32. Michael Dory, Adam Parrish, Brendan Berg. Introduction to Tornado. — O'Reilly Media, Inc., 2012. —138 с.

33. OpenAI (2021) OpenAI API, OpenAI. OpenAI. URL: <https://openai.com/api/>

34. Krajee Yii Extensions - © Kartik [Електронний ресурс] – Режим доступу до ресурсу: <https://demos.krajee.com>

35. GridView - © Kartik [Електронний ресурс] – Режим доступу до ресурсу: <https://demos.krajee.com/grid>

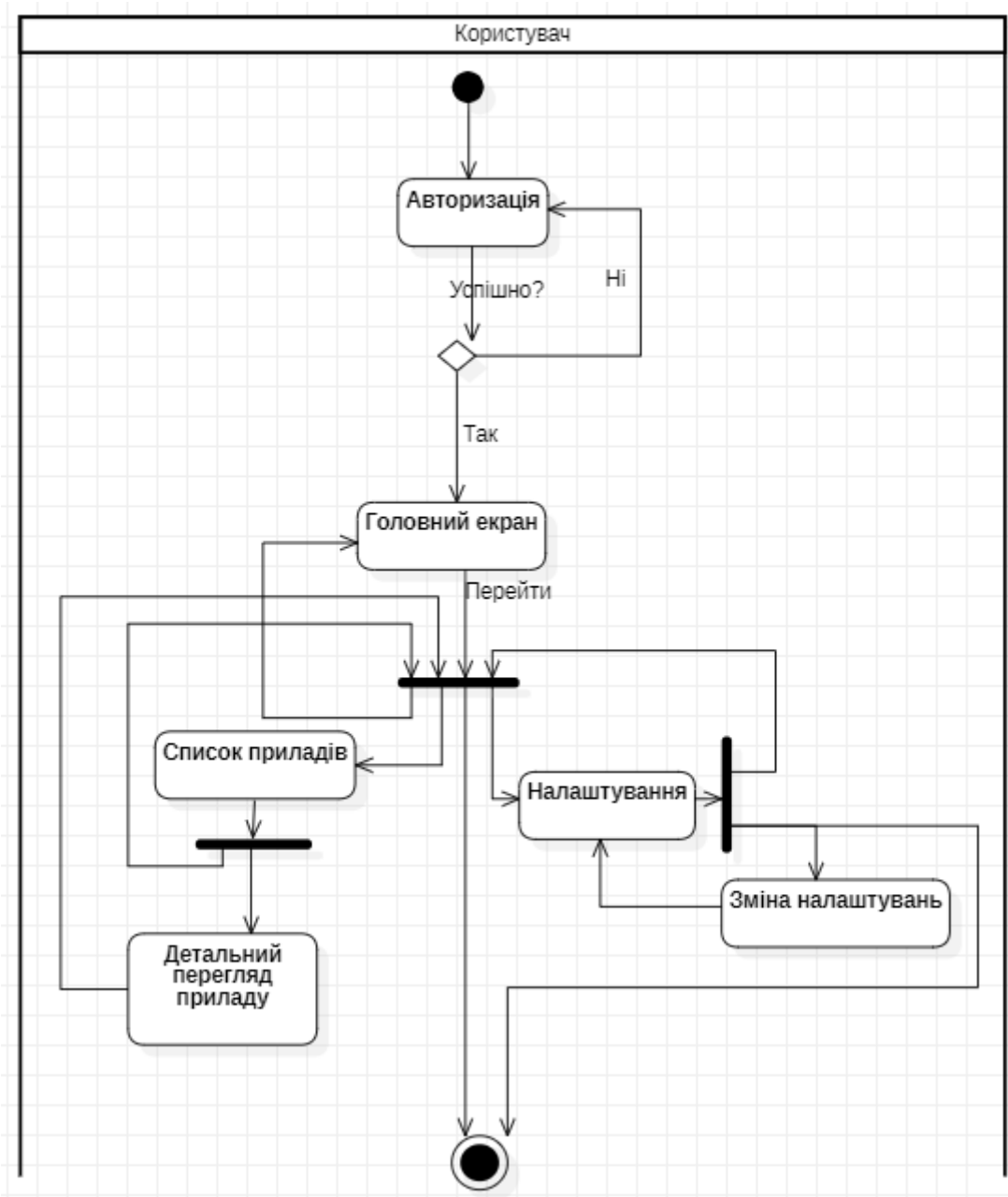
36. Tabs X - © Kartik [Електронний ресурс] – Режим доступу до ресурсу: <https://demos.krajee.com/tabs-x>

37. Popover X - © Kartik [Електронний ресурс] – Режим доступу до ресурсу: <https://demos.krajee.com/popover-x>

38. Alert - © Kartik [Електронний ресурс] – Режим доступу до ресурсу: <https://demos.krajee.com/widget-details/alert>

ДОДАТКИ

Алгоритм дій користувача



Послідовність відправки даних

