

## КВАЛІФІКАЦІЙНА РОБОТА

Програмно-технічний засіб контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КВРКІ 022047.22.02.03 ПЗ

Виконав здобувач IV курсу, група KI2-22-2

Керівник

доктор філософії

Науковий ступінь, учене звання

Нормоконтролер





канд. фіз.-мат. наук, доц.

Науковий ступінь, учене звання

До захисту допускаю:  
завідувач кафедри КІС

«07» червня 2026 р.

дата

  
Підпис  
  
Підпис  
  
Підпис  
  
Підпис

Ілля КУЧЕРЕНКО

Ініціали, прізвище

Юрій ВОЙЧУР

Ініціали, прізвище

Тетяна КИСІЛЬ

Ініціали, прізвище

Ольга ПАВЛОВА

Ініціали, прізвище

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІПС



Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Кучеренкові Іллі Ігоровичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase

Керівник проекту (роботи) Войчур Юрій Олексійович, ДФ

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та огляд відомих рішень

Проектування програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase

Симуляція програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Структурна схема програмно-технічного засобу

Схема монтажна

Схема організації програмного забезпечення ESP32

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання « 10 » 01 2026 р.

**КАЛЕНДАРНИЙ ПЛАН**

| №з/п | Назва етапів (розділів) дипломного проекту (роботи)   | Термін виконання етапів проекту (роботи) | Примітка |
|------|---|--|----------|
| 1    | Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником                                  | 10.01.2026                               | виконано |
| 2    | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 01.02.2026                               | виконано |
| 3    | Робота над розділом 1 – дослідження предметної області та постановка задачі   | 01.03.2026                               | виконано |
| 4    | Робота над розділом 2 – проектування структури програмно-технічного засобу  | 01.04.2026                               | виконано |
| 5    | Робота над розділом 3 – реалізація, симуляція та розгортання програмно-технічного засобу                              | 29.04.2026                               | виконано |
| 6    | Оформлення пояснювальної записки згідно вимог   | 25.05.2026                               | виконано |
| 7    | Попередній захист ВКР   | 25.05.2026                               | виконано |
| 8    | Захист ВКР на засіданні ЕК  | Червень 2026 року                        |          |

Здобувач

  
Підпис

Ілля КУЧЕРЕНКО

Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи

  
Підпис

Юрій ВОЙЧУР

Імя, ПРІЗВИЩЕ



## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase».

Автор роботи: Ілля КУЧЕРЕНКО.

Керівник роботи: Юрій ВОЙЧУР.


Пояснювальна записка: 60 с., 35 рис., 6 табл., 3 дод., 60 джерела.

Графічна частина: 3 креслення.

ІНТЕРНЕТ РЕЧЕЙ, FIREBASE, МІКРОКЛІМАТ, МІКРОКОНТРОЛЕР ESP32.

Стрімкий розвиток концепції Інтернету речей та здешевлення мікроконтролерних платформ відкривають нові можливості для створення доступних систем моніторингу та автоматичного керування мікрокліматом з хмарною інтеграцією. Незважаючи на наявність комерційних систем автоматизації будівель класу BAS, більшість із них є фінансово недоступними для державних навчальних закладів або не забезпечують необхідної гнучкості налаштування без залучення спеціалізованих підрядників. Це зумовлює актуальність розробки доступного, функціонально повного та відкритого програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на базі мікроконтролерної платформи ESP32 та хмарної інфраструктури Firebase.

Метою дипломної роботи є проектування та реалізація програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на платформі ESP32 з використанням хмарної бази даних Firebase Realtime Database, що забезпечує автоматизований моніторинг температури, вологості та атмосферного тиску у трьох аудиторіях одночасно, дистанційне керування виконавчими пристроями через веб-інтерфейс та автоматичне регулювання мікроклімату на основі налаштовуваних порогових правил.

  
\_\_\_\_\_



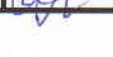

Підпис здобувача

02.06.2026  
\_\_\_\_\_

Дата

## ЗМІСТ

|   |    |
|---|----|
| Зміст .....   | 2  |
| Вступ.....  | 4  |
| 1 Дослідження предметної області та огляд відомих рішень .....  | 6  |
| 1.1 Значення мікроклімату в навчальних приміщеннях.....   | 6  |
| 1.2 Нормативні вимоги до мікроклімату навчальних приміщень.....   | 7  |
| 1.3. Огляд існуючих систем моніторингу та керування мікрокліматом.....  | 8  |
| 1.4. Концепція Інтернету речей.....   | 11 |
| 1.5. Огляд хмарних платформ для зберігання даних IoT-систем.....  | 15 |
| 2 Проектування програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на основі esp32 та firebase .....                         | 21 |
| 2.1 Вимоги до програмно-технічного засобу .....   | 21 |
| 2.2 Структура програмно-технічного засобу.....  | 24 |
| 2.3 Монтажна схема програмно-технічного засобу.....   | 27 |
| 2.4. Вибір та аналіз апаратних компонентів.....   | 29 |
| 2.5 Проектування структури бази даних.....  | 33 |
| 2.6. Проектування алгоритму автоматичного керування.....  | 35 |
| 2.7 Висновки до другого розділу .....   | 37 |
| 3 Реалізація, розгортання та тестування програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на основі esp32 та firebase..... | 38 |
| 3.1 Узагальнений підхід до реалізації, розгортання та тестування програмно-технічного засобу.....   | 38 |
| 3.2 Реалізація хмарної інфраструктури на основі Firebase .....  | 41 |
| 3.2. Реалізація програмного симулятора ESP32-клієнтів.....  | 48 |
| 3.3. Програмна реалізація ESP32-клієнтів .....  | 50 |
| 3.4. Розгортання веб-застосунку на Firebase Hosting.....  | 51 |
| 3.5. Тестування системи.....  | 54 |
| 3.6 Висновки до третього розділу .....  | 58 |

|  |                |                |   |             |
|--|----------------|----------------|---|-------------|
| КВРКІ. 022047.22.02.03 ПЗ  |                |                |   |             |
|  |                |                |   |             |
| <b>Зм.</b>   | <b>Арк.</b>    | <b>№докум.</b> | <b>Підпис</b>   | <b>Дата</b> |
| <b>Виконав</b>   | Ілля КУЧЕРЕНКО |                |  |             |
| <b>Перевір.</b>  | Юрій Войчур    |                |  |             |
| <b>Н.контр.</b>  | Тетяна КИСЛІЬ  |                |  |             |
| <b>Затвер.</b>   | Ольга ПАВЛОВА  |                |  |             |
| Програмно-технічний засіб контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase |                |                | Літера  | Арк.вш      |
|  |                |                | у   | 2           |
|  |                |                | Арк.внів  | 67          |
| <b>ХНУ КІ2-22-2</b>  |                |                |   |             |

|  |    |
|--|----|
| Висновки .....   | 59 |
| Перелік джерел посилань .....                                    | 61 |
| Додаток А Структурна схема програмно-технічного засобу .....     | 68 |
| Додаток Б Схема монтажна.....                                    | 69 |
| Додаток В Схема організації програмного забезпечення ESP32 ..... | 70 |

## ВСТУП

Актуальність дослідження. Забезпечення комфортних та безпечних умов перебування у навчальних приміщеннях є одним із пріоритетних завдань адміністрацій закладів освіти. Мікроклімат навчальних аудиторій це сукупність показників температури, відносної вологості та атмосферного тиску, які безпосередньо впливають на фізичний стан учасників навчального процесу, їх здатність до концентрації уваги та ефективність засвоєння матеріалу. Недотримання нормативних вимог ДСанПіН 5.5.2.008-01 щодо параметрів мікроклімату призводить до підвищення стомлюваності, зростання захворюваності та зниження якості освітнього процесу в цілому. Традиційні підходи до регулювання мікроклімату, такі як ручне відкривання вікон, нерегульоване опалення та автономні побутові прилади, не забезпечують в повній мірі точного підтримання параметрів у встановлених нормах меж, не ведуть автоматизованого обліку показників і не дозволяють дистанційно контролювати стан одразу кількох аудиторій. Стрімкий розвиток концепції Інтернету речей та здешевлення мікроконтролерних платформ відкривають нові можливості для створення доступних систем моніторингу та автоматичного керування мікрокліматом з хмарною інтеграцією. Незважаючи на наявність комерційних систем автоматизації будівель класу BAS, більшість із них є фінансово недоступними для державних навчальних закладів або не забезпечують необхідної гнучкості налаштування без залучення спеціалізованих підрядників. Це зумовлює актуальність розробки доступного, функціонально повного та відкритого програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на базі мікроконтролерної платформи ESP32 та хмарної інфраструктури Firebase.

Метою дипломної роботи є проєктування та реалізація програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на платформі ESP32 з використанням хмарної бази даних Firebase Realtime

|     |      |          |        |      |                           |           |
|-----|------|----------|--------|------|---------------------------|-----------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>4 |
| Зм. | Арк. | № док.м. | Підпис | Дата |                           |           |

Database, що забезпечує автоматизований моніторинг температури, вологості та атмосферного тиску у трьох аудиторіях одночасно, дистанційне керування виконавчими пристроями через веб-інтерфейс та автоматичне регулювання мікроклімату на основі налаштовуваних порогових правил.

Об'єктом дослідження є система автоматизованого моніторингу та керування параметрами мікроклімату навчальних приміщень на основі технологій Інтернету речей та хмарних обчислень.

Предметом дослідження є методи та засоби проєктування програмно-технічного засобу контролю та керування мікрокліматом із застосуванням мікроконтролера ESP32, сенсора BME280, хмарної платформи Firebase та веб-застосунку з підтримкою ручного та автоматичного режимів керування виконавчими пристроями.

Практична цінність роботи полягає в тому, що розроблений програмно-технічний засіб може бути впроваджений у навчальних закладах як доступна альтернатива дорогим комерційним системам автоматизації будівель, забезпечуючи моніторинг мікроклімату кількох аудиторій в єдиному веб-інтерфейсі, автоматичне керування кліматичним обладнанням на основі налаштовуваних правил, захищений дистанційний доступ через браузер із будь-якого пристрою та можливість подальшого масштабування на більшу кількість приміщень без зміни архітектури системи.

|     |      |          |        |      |                           |           |
|-----|------|----------|--------|------|---------------------------|-----------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>5 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |           |

# 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ВІДОМИХ РІШЕНЬ

## 1.1 Значення мікроклімату в навчальних приміщеннях

Мікроклімат навчальних приміщень є одним із ключових факторів, що визначають ефективність освітнього процесу, стан здоров'я учасників навчання та загальну якість перебування у закладі освіти. Під мікрокліматом розуміють сукупність фізичних параметрів повітряного середовища приміщення – температури, відносної вологості та атмосферного тиску, що безпосередньо впливають на теплообмін людського організму з навколишнім середовищем [1-12].

Численні педагогічні та медичні дослідження підтверджують пряму залежність між параметрами мікроклімату та продуктивністю розумової діяльності. Зниження температури повітря нижче 18°C призводить до погіршення концентрації уваги та сповільнення реакцій унаслідок активізації терморегуляторних механізмів організму. Підвищення температури понад 26°C викликає відчуття дискомфорту, підвищує стомлюваність та знижує здатність до тривалого зосередження [13]. Обидва відхилення від комфортного діапазону негативно позначаються на засвоєнні навчального матеріалу.

Відносна вологість повітря не менш важлива. При вологості нижче 30–35% слизові оболонки дихальних шляхів пересихають, що підвищує сприйнятливості до респіраторних інфекцій особливо актуально в умовах колективного перебування значної кількості людей в одному приміщенні. Надмірна вологість понад 70% сприяє розвитку плісняви та підвищує відчуття духоти. Комфортний діапазон вологості для навчальних приміщень становить 40–60%.

Атмосферний тиск, хоча й не є безпосередньо керованим параметром, важливий як діагностичний показник: різкі зміни тиску корелюють із метеочутливістю частини учнів та педагогів, що позначається на їх самопочутті.

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк. |
|     |      |          |        |      |                           | 6    |
| Зм. | Арк. | № докum. | Підпис | Дата |                           |      |

Моніторинг тиску дозволяє своєчасно прогнозувати такі ситуації та вживати відповідних заходів [14-21].

## 1.2 Нормативні вимоги до мікроклімату навчальних приміщень

В Україні параметри мікроклімату у навчальних закладах регламентуються державними санітарними нормами. Згідно з ДСанПіН 5.5.2.008-01 «Державні санітарні правила і норми влаштування, утримання загальноосвітніх навчальних закладів та організації навчально-виховного процесу», температура повітря у навчальних класах повинна підтримуватись у межах 17–20°C у холодний період і 22–25°C у теплий [61]. Рекомендована відносна вологість становить 40–60%.

Таблиця 1.1 – Нормативні діапазони параметрів мікроклімату навчальних приміщень

| Параметр                              | Нормативний діапазон | Нормативний документ |
|---------------------------------------|----------------------|----------------------|
| Температура повітря (холодний період) | 17–20°C              | ДСанПіН 5.5.2.008-01 |
| Температура повітря (теплий період)   | 22–25°C              | ДСанПіН 5.5.2.008-01 |
| Відносна вологість повітря            | 40–60%               | ДСанПіН 5.5.2.008-01 |
| Швидкість руху повітря                | Не більше 0,1 м/с    | ДСТУ ISO 7730:2006   |
| Атмосферний тиск (нормальний)         | 980–1030 гПа         | Довідковий діапазон  |

Норми ДСТУ ISO 7730:2006 «Ергономіка теплового середовища» визначають показники теплового комфорту для виробничих та адміністративних приміщень, принципи яких поширюються й на освітні заклади. Стандарт введено в дію на підставі рекомендацій ISO та враховує суб'єктивне сприйняття теплового комфорту різними категоріями людей.

Забезпечення відповідності цим нормам у реальних умовах навчального закладу ускладнюється динамічним характером теплових навантажень: кількість присутніх у класі людей, сонячна інсоляція, температура зовнішнього повітря та тепловіддача обладнання постійно змінюються протягом дня. Це обумовлює необхідність не лише пасивного вимірювання параметрів, але й активного автоматичного керування кліматичним обладнанням.

### 1.3. Огляд існуючих систем моніторингу та керування мікрокліматом

Ринок систем автоматизації мікроклімату в приміщеннях є широким і охоплює рішення від простих побутових термостатів до промислових систем автоматизації будівель класу BAS (Building Automation System). Для розуміння місця розробленого засобу в загальному ландшафті необхідно розглянути основні категорії існуючих рішень.

Найбільш поширеними комерційними рішеннями для моніторингу мікроклімату є системи класу «розумний будинок» від таких виробників як Honeywell, Siemens, KNX та Loxone. Ці системи забезпечують комплексну автоматизацію будівлі, включаючи керування опаленням, вентиляцією та кондиціонуванням повітря (системи HVAC). Вони відрізняються високою надійністю, наявністю сертифікатів відповідності стандартам і розгалуженою мережею авторизованих монтажних організацій.

Системи Honeywell орієнтовані переважно на автоматизацію HVAC-систем та енергоменеджмент (рис. 1.1). Компанія пропонує широкий спектр контролерів, термостатів, датчиків температури та вологості, а також

|     |      |          |        |      |                           |           |
|-----|------|----------|--------|------|---------------------------|-----------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>8 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |           |

програмних платформ для централізованого моніторингу. Однією з особливостей Honeywell є інтеграція з професійними системами диспетчеризації будівель. Їх обладнання підтримує промислові протоколи BACnet та Modbus, що дозволяє об'єднувати різні підсистеми в єдину мережу керування. Основними перевагами рішень Honeywell є висока надійність, стабільність роботи та підтримка масштабованих інфраструктур. Недоліком є висока вартість обладнання та складність налаштування, що часто потребує залучення сертифікованих спеціалістів.



Рисунок 1.1 – Монітор якості повітря Honeywell

Компанія Siemens є одним із найбільших виробників систем автоматизації будівель та промислових об'єктів. Її рішення використовуються для комплексного керування мікрокліматом, освітленням, безпекою та енергоспоживанням. Особливістю систем Siemens є використання контролерів серії Desigo та підтримка стандарту BACnet для інтеграції різних пристроїв. Системи дозволяють автоматично регулювати температуру та вентиляцію залежно від параметрів середовища та заданих сценаріїв роботи. Крім того, Siemens надає потужні програмні засоби для аналітики та диспетчеризації. Основним недоліком таких систем є їх висока складність та значна вартість

|     |      |          |        |      |                           |           |
|-----|------|----------|--------|------|---------------------------|-----------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>9 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |           |

впровадження, через що вони переважно використовуються у великих комерційних або промислових будівлях.

KNX є міжнародним стандартом автоматизації будівель, який підтримується великою кількістю виробників обладнання (рис. 1.2). На відміну від окремих компаній, KNX є відкритою технологією, що забезпечує сумісність між пристроями різних брендів. Система використовує спеціальну шину зв'язку для обміну даними між датчиками, контролерами та виконавчими механізмами. KNX дозволяє реалізовувати керування освітленням, опаленням, вентиляцією, кондиціонуванням, жалюзі та іншими елементами інфраструктури будівлі. Основною перевагою KNX є гнучкість та можливість масштабування системи. Однак налаштування KNX-систем потребує спеціалізованого програмного забезпечення ETS та відповідної підготовки персоналу. Крім того, вартість сертифікованого KNX-обладнання є достатньо високою.



Рисунок 1.2 – KNX Airokit2 Pro датчик якості повітря

Система Loxone орієнтована переважно на сегмент «розумного будинку» та невеликих комерційних об'єктів. Центральним елементом системи є контролер Miniserver, який забезпечує взаємодію між датчиками, виконавчими пристроями та користувацьким інтерфейсом. Loxone підтримує автоматизацію мікроклімату, освітлення, мультимедійних систем, безпеки та енергоменеджменту. Однією з ключових особливостей є зручний графічний інтерфейс конфігурації та можливість створення автоматичних сценаріїв без

необхідності глибоких знань програмування. Система також підтримує мобільний доступ та віддалений моніторинг через хмарні сервіси. Проте LoXone є закритою екосистемою, що обмежує використання обладнання сторонніх виробників та ускладнює глибоку модифікацію функціоналу.

Проте комерційні BAS-системи мають суттєві недоліки з точки зору їх застосування в навчальних закладах України. Вартість проектування, обладнання та монтажу такої системи для однієї школи може сягати сотень тисяч гривень, що робить їх практично недоступними для більшості державних навчальних закладів. Крім того, ці системи є закритими: модифікація логіки керування або розширення функціональності потребує участі виробника або авторизованого сервісного партнера і не може бути виконана силами персоналу закладу.

Альтернативою комерційним BAS є хмарні IoT-платформи, що надають інфраструктуру для збору, зберігання та візуалізації даних із підключених пристроїв. Серед найбільш відомих платформ – ThingSpeak, Blynk, AWS IoT Core, Microsoft Azure IoT Hub та Google Firebase. Ці платформи орієнтовані на розробників і надають API для підключення пристроїв, сховища даних та інструменти побудови дашбордів.

#### 1.4. Концепція Інтернету речей

Інтернет речей є сучасною концепцією побудови інформаційних систем, у межах якої фізичні пристрої отримують можливість взаємодіяти між собою через мережу Інтернет без необхідності постійного втручання людини. До таких пристроїв належать різноманітні сенсори, контролери, виконавчі механізми та обчислювальні модулі, здатні збирати, передавати, обробляти й аналізувати дані у режимі реального часу. Концепція IoT була запропонована Kevin Ashton наприкінці 1990-х років і сьогодні активно застосовується в системах

автоматизації будівель, промисловості, транспорту, медицині, сільському господарстві та освітній сфері.

Основною особливістю IoT-систем є поєднання фізичного обладнання та програмних сервісів у єдину мережеву інфраструктуру. Кожен пристрій може виступати джерелом даних або виконавчим елементом, який реагує на команди системи керування. Завдяки цьому формується кіберфізична система, здатна автоматично реагувати на зміни навколишнього середовища та підтримувати необхідні параметри роботи без участі оператора.

У сучасних навчальних закладах технології Інтернету речей дедалі частіше використовуються для автоматизації різних елементів інфраструктури. До таких рішень належать системи контролю доступу до приміщень, автоматичне керування освітленням, моніторинг енергоспоживання, охоронна та пожежна сигналізація, а також системи контролю мікроклімату аудиторій. Автоматизація мікроклімату є особливо актуальною задачею, оскільки температура, вологість і якість повітря безпосередньо впливають на самопочуття, концентрацію уваги та ефективність навчального процесу.

Підтримання оптимальних параметрів середовища в аудиторіях є важливим не лише з точки зору комфорту, але й з позиції енергоефективності. У багатьох навчальних закладах регулювання опалення та вентиляції здійснюється вручну, що призводить до нераціонального використання енергоресурсів. Використання IoT-систем дозволяє автоматизувати цей процес шляхом постійного аналізу показників сенсорів та автоматичного керування вентиляторами, системами опалення або кондиціонування.

Типова IoT-система складається з кількох основних рівнів. На нижньому рівні знаходяться сенсори та виконавчі пристрої, які взаємодіють із фізичним середовищем. Для вимірювання параметрів мікроклімату можуть використовуватись датчики температури, вологості, атмосферного тиску або концентрації вуглекислого газу. На рівні обробки даних використовуються мікроконтролери, зокрема ESP32, які виконують збір даних із сенсорів, первинну

обробку інформації та передачу даних до хмарного сервера через бездротову мережу Wi-Fi.

Мікроконтролер ESP32 набув значної популярності в IoT-проєктах завдяки поєднанню низької вартості, високої продуктивності та наявності вбудованих модулів Wi-Fi і Bluetooth (рис. 1.3). Контролер підтримує багатозадачність, має достатню кількість GPIO-выводів та може працювати з різними типами датчиків через інтерфейси I2C, SPI та UART. Це дозволяє створювати гнучкі системи моніторингу та керування без необхідності використання додаткових мережевих модулів.



Рисунок 1.3 – Мікроконтролер ESP32

Важливим елементом сучасних IoT-систем є використання хмарних платформ для централізованого зберігання та обробки даних. Хмарна архітектура забезпечує можливість віддаленого доступу до системи з будь-якого пристрою, підключеного до Інтернету. Адміністратор або викладач може в реальному часі переглядати параметри мікроклімату аудиторій через вебінтерфейс або мобільний застосунок, отримувати повідомлення про перевищення допустимих значень та дистанційно змінювати режими роботи системи.

Суттєвою перевагою IoT-підходу є можливість накопичення історичних даних у хмарному сховищі. Зібрані часові ряди температури, вологості та інших параметрів можуть використовуватись для подальшого аналізу ефективності роботи системи. На основі таких даних можливо визначати закономірності зміни мікроклімату, оцінювати енергоспоживання та формувати рекомендації щодо оптимізації режимів роботи обладнання. Перспективним напрямом розвитку IoT-систем у сфері освіти є інтеграція з аналітичними сервісами та елементами штучного інтелекту. У майбутньому це дозволить автоматично прогнозувати зміни мікроклімату залежно від часу доби, кількості людей у приміщенні або погодних умов. Крім того, накопичені дані можуть використовуватись для дослідження взаємозв'язку між параметрами середовища та показниками відвідуваності чи продуктивності навчального процесу.

Одним із ключових аспектів побудови IoT-систем є вибір протоколу обміну даними між пристроями та серверною частиною. Найбільш поширеними протоколами у таких системах є MQTT, HTTP/REST та WebSocket. Кожен із них має власні переваги та обмеження.

Протокол MQTT (Message Queuing Telemetry Transport) є легковаговим мережевим протоколом, спеціально розробленим для IoT-пристроїв із невеликими обчислювальними ресурсами. MQTT працює за моделлю «публікація-підписка», де всі повідомлення проходять через центральний брокер. Пристрої можуть публікувати дані в певні теми (topics), а інші пристрої – підписуватись на них для отримання інформації. Основною перевагою MQTT є мінімальний мережевий трафік та низькі затримки передачі даних. Завдяки компактному формату пакетів MQTT добре підходить для бездротових систем із нестабільним з'єднанням. Проте використання MQTT потребує окремого брокера, наприклад Mosquitto або EMQX, що збільшує складність адміністрування системи.

Протокол HTTP/REST є найбільш поширеним способом взаємодії клієнтів із вебсервісами та API. У такій моделі пристрій надсилає HTTP-запити типу GET,

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>14 |
| Зм. | Арк. | № докum. | Підпис | Дата |                           |            |

POST або PUT для отримання чи запису даних на сервері. Основною перевагою HTTP є універсальність та простота інтеграції з більшістю хмарних платформ. Однак для систем реального часу HTTP має певні недоліки. Через значний обсяг службових заголовків збільшується мережевий трафік, а для отримання оновлень необхідно періодично виконувати опитування сервера, що створює додаткове навантаження та затримки.

Технологія WebSocket забезпечує постійне двостороннє з'єднання між клієнтом і сервером. Після встановлення з'єднання сервер може надсилати нові дані клієнту без додаткових запитів. Це дозволяє реалізувати оновлення інформації в режимі реального часу та мінімізувати затримки передачі даних. Саме такий підхід використовується у Firebase Realtime Database, де WebSocket виступає транспортним механізмом для синхронізації даних між клієнтськими пристроями та хмарною базою даних.

#### 1.5. Огляд хмарних платформ для зберігання даних IoT-систем

Вибір хмарної платформи для зберігання та обробки даних є одним із ключових архітектурних рішень при розробці IoT-системи. Від правильного вибору залежить затримка передачі даних, вартість експлуатації, складність інтеграції з апаратними вузлами та можливості масштабування. Нижче розглянуто основні хмарні платформи та бази даних, що застосовуються в IoT-системах моніторингу.

Однією із відкритих IoT-платформ є ThingSpeak. Дана платформа орієнтована насамперед на збір і візуалізацію числових часових рядів. Пристрій надсилає дані через простий HTTP GET-запит, після чого вони відображаються у вигляді графіків на каналі. Платформа безкоштовна для некомерційного використання з обмеженням на частоту оновлень – не частіше одного разу на 15 секунд. ThingSpeak добре підходить для навчальних прототипів і швидкої візуалізації даних, проте має суттєві обмеження: відсутність механізму

реального часу (дані оновлюються лише при перезавантаженні сторінки), відсутність вбудованої автентифікації користувачів, мінімальні можливості побудови власного інтерфейсу та відсутність підтримки двостороннього зв'язку для керування виконавчими пристроями. Для системи, що вимагає не лише моніторингу, але й керування реле в реальному часі, ThingSpeak є функціонально недостатнім рішенням.

Ще однією платформою є Blynk. Головним акцентом даної хмарної платформи є мобільний інтерфейс та швидке прототипування. Вона надає готові віджети (кнопки, перемикачі, графіки), які розробник розташовує у мобільному застосунку без написання коду інтерфейсу. Blynk підтримує двосторонній зв'язок між пристроєм та додатком, що теоретично дозволяє керувати реле. Проте безкоштовний тариф Blynk Free суттєво обмежує кількість «енерджі» (внутрішня валюта для розблокування віджетів), кількість підключених пристроїв та частоту оновлень. Для підключення трьох незалежних вузлів із власним інтерфейсом на платному тарифі Blynk Plus вартість становить кілька десятків доларів на місяць. Крім того, Blynk орієнтований виключно на мобільні застосунки і не надає повноцінного веб-інтерфейсу, доступного з браузера без встановлення додатку.

Альтернативним рішенням є використання зв'язки InfluxDB та Grafana. InfluxDB це спеціалізована база даних часових рядів, оптимізована для зберігання та запиту телеметричних даних із мітками часу. У парі з інструментом візуалізації Grafana вона формує популярний стек для промислового моніторингу IoT-систем. InfluxDB забезпечує надзвичайно ефективне стиснення часових рядів та потужну мову запитів Flux для агрегації, фільтрації та побудови дашбордів. Хмарна версія InfluxDB Cloud має безкоштовний рівень, обмежений 30-денним терміном зберігання даних та 5 МБ запису на день. Основним недоліком цього стека є складність розгортання та налаштування: потрібно окремо конфігурувати InfluxDB, Grafana, джерело даних та дашборди. Крім того, відсутній вбудований механізм двостороннього зв'язку для керування

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>16 |
| Зм. | Арк. | № докum. | Підпис | Дата |                           |            |

пристроями InfluxDB є виключно сховищем даних без підтримки команд керування.

На сьогодні на ринку присутнє також рішення від AWS. Amazon Web Services надає комплексну екосистему для IoT: AWS IoT Core виступає брокером повідомлень на базі MQTT, DynamoDB або Timestream – сховищем даних, AWS Lambda є сервісом обробки подій, а AWS IoT Greengrass дозволяє виконувати логіку на крайніх вузлах мережі. Ця екосистема є найбільш повною і масштабованою з усіх розглянутих, забезпечуючи промисловий рівень надійності та безпеки. Проте складність налаштування та вартість робить її недоцільною для системи малого масштабу: AWS IoT Core тарифікується за кількістю повідомлень (0,08–1,00 USD за мільйон), а загальна вартість стека для трьох IoT-пристроїв із урахуванням Lambda, DynamoDB і передачі даних може становити кілька десятків доларів на місяць навіть при незначному навантаженні. Налаштування IAM-ролей, сертифікатів X.509, правил IoT та тригерів Lambda потребує значної кваліфікації в галузі хмарних обчислень.

Іншим варіантом є використання рішень від Microsoft Azure, зокрема сервіс Azure IoT Hub. Azure IoT Hub є аналогом AWS IoT Core від Microsoft і орієнтований на корпоративні IoT-рішення. Він підтримує протоколи MQTT, AMQP та HTTP, забезпечує двосторонній зв'язок через механізм хмара- пристрій та інтегрується з Azure Stream Analytics для обробки потоків даних. Безкоштовний рівень F1 дозволяє надсилати до 8000 повідомлень на день від максимум 500 пристроїв. Як і AWS, Azure IoT Hub потребує значних зусиль на початкове налаштування: реєстрація пристроїв, управління сертифікатами, налаштування маршрутизації повідомлень та інтеграція з сховищем даних є окремими задачами, кожна з яких вимагає знайомства з відповідними Azure-сервісами. Для прототипу або навчального проєкту такий рівень складності є надлишковим.

Ще одним варіантом є використання Firebase – платформи компанії Google, що надає набір хмарних сервісів за моделлю Backend as a Service:

Realtime Database, Authentication, Hosting, Cloud Firestore та інші. Firebase Realtime Database зберігає дані у вигляді JSON-дерева та синхронізує їх між усіма підключеними клієнтами через постійне WebSocket-з'єднання. Ключова відмінність від HTTP-орієнтованих платформ полягає в тому, що після підписки клієнта на певний вузол будь-яка зміна цього вузла миттєво доставляється всім підписникам без явного запиту. Така особливість забезпечує справжній режим реального часу без затримок, характерних для циклічного опитування.

З технічної точки зору Firebase Realtime Database забезпечує механізм оновлення даних у реальному часі через постійне WebSocket-з'єднання, що є критично важливим для системи моніторингу з відображенням актуальних показників у браузері. Альтернативні рішення на базі HTTP/REST або MQTT вимагали б або реалізації додаткового сервера для push-повідомлень, або значно більшого часового лагу між записом даних та їх відображенням.

З економічної точки зору безкоштовний тарифний план Spark (станом на 2026 рік) дозволяє одночасно підтримувати до 100 паралельних з'єднань, зберігати до 1 ГБ даних та виконувати до 360 МБ завантажень на день. Ці показники з великим запасом покривають потреби системи з трьома вузлами та кількома адміністраторами без жодних фінансових витрат.

З організаційної точки зору Firebase надає інтегроване середовище з єдиною консоллю керування для бази даних, автентифікації та хостингу. Це суттєво спрощує розгортання та підтримку системи: замість налаштування та інтеграції трьох окремих сервісів адміністратор керує всіма компонентами в одному інтерфейсі.

Firebase має добре задокументований Arduino/ESP32 SDK (бібліотека FirebaseClient), що дозволяє реалізувати повний цикл взаємодії, починаючи від автентифікації, закінчуючи записом та читанням даних на мікроконтролері без написання низькорівневого мережевого коду. Це скорочує час розробки та знижує ймовірність помилок у реалізації протоколу.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>18 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

## 1.6 Постановка задачі

Аналіз предметної області, огляд існуючих рішень та технологій дозволяють сформулювати конкретне технічне завдання на розробку програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase.

Метою роботи є розробка та практична реалізація системи, яка забезпечить автоматизований моніторинг параметрів мікроклімату у трьох навчальних аудиторіях та дистанційне керування виконавчими пристроями через веб-інтерфейс із хмарним бекендом. Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести аналіз нормативних вимог до параметрів мікроклімату навчальних приміщень та сформулювати технічні і функціональні вимоги до програмно-технічного засобу.

2. Розробити архітектуру системи, що охоплює апаратний рівень вузлів збору даних та керування, рівень хмарного зберігання і передачі інформації та рівень користувацького веб-інтерфейсу.

3. Обґрунтувати вибір апаратних компонентів системи: мікроконтролера ESP32 як обчислювального вузла з вбудованим Wi-Fi, сенсора BME280 для одночасного вимірювання температури, вологості та атмосферного тиску, а також релейних модулів для керування виконавчими пристроями.

4. Розробити монтажну схему апаратного вузла з визначенням електричних з'єднань між ESP32, сенсором BME280, модулем реле, OLED-дисплеєм та системою живлення.

5. Спроекувати структуру даних Firebase Realtime Database з ієрархічним розподілом вузлів для телеметрії трьох аудиторій, стану реле та порогових значень автоматичного керування, а також визначити правила безпеки для обмеження доступу за UID автентифікованого користувача.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>19 |
| Зм. | Арк. | № докum. | Підпис | Дата |                           |            |

6. Реалізувати хмарну інфраструктуру Firebase: створити проєкт, налаштувати сервіс автентифікації Email/Password, розгорнути екземпляр Realtime Database та зареєструвати веб-застосунок із підключеним хостингом.

7. Розробити прошивку для мікроконтролера ESP32, що реалізує циклічне зчитування показників сенсора BME280, їх публікацію до Firebase RTDB, перевірку стану реле та виконання алгоритму автоматичного керування з гістерезисом на основі порогових значень із бази даних.

8. Розробити програмний симулятор на мові Python, що відтворює роботу трьох незалежних ESP32-вузлів у вигляді паралельних потоків, для верифікації хмарної інфраструктури та тестування системи без фізичного обладнання.

9. Розробити та розгорнути веб-застосунок на Firebase Hosting, що забезпечує відображення показників мікроклімату трьох аудиторій у режимі реального часу, ручне керування реле та налаштування порогових значень автоматичного керування через браузерний інтерфейс.

10. Провести комплексне тестування системи зокрема перевірити коректність запису телеметрії до БД, функціонування ручного та автоматичного керування реле, а також відповідність відображуваних даних у веб-застосунку реальному стану системи.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>20 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

## 2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ КОНТРОЛЮ ТА КЕРУВАННЯ МІКРОКЛІМАТОМ НАВЧАЛЬНИХ АУДИТОРІЙ НА ОСНОВІ ESP32 ТА FIREBASE

### 2.1 Вимоги до програмно-технічного засобу

Проєктування програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій розпочинається з формулювання системи вимог, які визначають функціональні можливості, технічні характеристики та обмеження майбутнього рішення. Вимоги сформовано на основі аналізу умов експлуатації в навчальних закладах, нормативних документів щодо параметрів мікроклімату в навчальних приміщеннях та технічних можливостей сучасних засобів IoT.

Система має забезпечувати безперервний моніторинг трьох ключових параметрів мікроклімату – температури повітря, відносної вологості та атмосферного тиску одночасно у трьох навчальних аудиторіях. Вимірювання повинні виконуватись із заданим часовим інтервалом та передаватись до хмарного сховища даних у режимі реального часу.

Система керування має підтримувати два режими впливу на мікрокліматичні умови. Перший – ручний режим, за якого оператор через веб-інтерфейс може безпосередньо вмикати або вимикати виконавчі пристрої кожної аудиторії незалежно. Другий – автоматичний режим, за якого система самостійно приймає рішення про активацію виконавчих пристроїв на підставі порівняння поточних показників із налаштованими пороговими значеннями. Перехід між режимами має здійснюватись без переривання роботи системи.

Веб-застосунок має надавати оператору актуальну інформацію про стан усіх аудиторій на єдиній сторінці без необхідності перемикання між розділами. Інтерфейс повинен відображати поточні значення параметрів мікроклімату, стан кожного виконавчого пристрою та дозволяти налаштування порогових значень

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КвРКІ. 022047.22.02.03 ПЗ | Арк.<br>21 |
| Зм. | Арк. | № докum. | Підпис | Дата |                           |            |

авто-правил для кожної аудиторії окремо. Оновлення даних на сторінці має відбуватись автоматично без перезавантаження.

Доступ до системи має бути захищений автентифікацією. Несанкціоновані запити до бази даних та спроби керування виконавчими пристроями без авторизації повинні відхилятися на рівні правил безпеки хмарної платформи.

Щодо технічних вимог, то кожен вузол системи, що обслуговує одну аудиторію, має включати мікроконтролер з вбудованим модулем Wi-Fi, давач для одночасного вимірювання температури, вологості та тиску, а також три незалежних канали комутації для підключення виконавчих пристроїв. Живлення вузла має бути автономним або від мережі 220 В через відповідний адаптер.

Інтервал оновлення телеметричних даних у базі даних не повинен перевищувати 15 секунд. Час реакції системи на команду ручного керування реле з моменту натискання кнопки у веб-інтерфейсі до фактичного спрацювання виконавчого пристрою не повинен перевищувати 10 секунд.

Точність вимірювання температури повинна становити не гірше  $\pm 0,5^{\circ}\text{C}$  у діапазоні від  $0^{\circ}\text{C}$  до  $+50^{\circ}\text{C}$ . Точність вимірювання відносної вологості – не гірше  $\pm 3\%$  у діапазоні від 20% до 80% відносної вологості. Ці вимоги обумовлені нормами ДСанПіН 5.5.2.008-01 щодо гігієнічних вимог до навчальних приміщень.

Веб-застосунок має коректно відображатись у сучасних браузерях без встановлення додаткового програмного забезпечення. Застосунок має бути доступним за публічною HTTPS-адресою з будь-якого пристрою, підключеного до мережі Інтернет.

Щодо надійності та безпеки, то кожен вузол системи має функціонувати автономно в частині автоматичного керування: навіть за відсутності зв'язку з інтернетом або недоступності хмарної платформи мікроконтролер повинен продовжувати вимірювання та, за можливості, виконувати локально записані авто-правила. Відновлення зв'язку не повинно вимагати перезапуску пристрою.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>22 |
| Зм. | Арк. | № докum. | Підпис | Дата |                           |            |

Передача даних між вузлами системи та хмарною платформою має здійснюватись виключно по захищеному з'єднанню HTTPS/WSS. Зберігання облікових даних (API-ключів, паролів) у відкритому вигляді у вихідному кодї клієнтського застосунку не допускається — замість цього використовуються механізми автентифікації Firebase, що обмежують доступ до даних на рівні правил безпеки бази даних.

База даних має бути структурована таким чином, щоб дані різних користувачів були ізольовані одне від одного. Права на читання та запис мають надаватись виключно автентифікованому користувачеві стосовно власних вузлів даних. Таким чином, зведені вимоги до проєктованого програмно-технічного засобу наведені у таблиці 2.1.

Таблиця 2.1 – Зведена таблиця вимог до програмно-технічного засобу

| Категорія вимог | Вимога                          | Значення / умова                      |
|-----------------|---------------------------------|---------------------------------------|
| Моніторинг      | Кількість аудиторій             | 3                                     |
| Моніторинг      | Параметри вимірювання           | Температура, вологість, тиск          |
| Моніторинг      | Інтервал оновлення даних        | Не більше 15 секунд                   |
| Точність        | Температура                     | $\pm 0,5^{\circ}\text{C}$             |
| Точність        | Відносна вологість              | $\pm 3\%$                             |
| Керування       | Режими керування                | Ручний та автоматичний                |
| Керування       | Виконавчі пристрої на аудиторію | 3 (термостат, зволожувач, вентиляція) |
| Керування       | Час реакції на команду          | Не більше 10 секунд                   |
| Інтерфейс       | Платформа доступу               | Веб-браузер (будь-який пристрій)      |

Продовження таблиці 2.1

| Інтерфейс | Протокол       | HTTPS                             |
|-----------|----------------|-----------------------------------|
| Безпека   | Автентифікація | Email/Password (Firebase Auth)    |
| Безпека   | Ізоляція даних | За UID користувача                |
| Живлення  | Тип живлення   | Автономне (Li-Ion) або від мережі |

## 2.2 Структура програмно-технічного засобу

Програмно-технічний засіб контролю та керування мікрокліматом побудовано за трирівневою архітектурою, що включає рівень збору та виконання, рівень хмарного зберігання і передачі даних та рівень користувацького інтерфейсу. Така структура забезпечує чіткий розподіл відповідальності між компонентами системи та дозволяє масштабувати кожен рівень незалежно.

На структурній схемі, яка подана на рис. 2.1, показано взаємозв'язок між усіма компонентами системи. Лівий блок представляє апаратний рівень одного вузла – датчики та виконавчі пристрої, що підключені до мікроконтролера ESP32. Центральний канал зв'язку – вбудований Wi-Fi модуль мікроконтролера. Правий блок відображає хмарну і клієнтську частини системи.

Апаратний рівень системи представлений трьома ідентичними вузлами – по одному на кожному навчальну аудиторію. Кожен клієнтський вузол побудований навколо мікроконтролера ESP32, який є обчислювальним ядром і засобом бездротового зв'язку одночасно завдяки вбудованому модулю Wi-Fi стандарту 802.11 b/g/n.

До кожного вузла підключено давач BME280, який вимірює три параметри мікроклімату: температуру, відносну вологість та атмосферний тиск. Зв'язок між сенсором і мікроконтролером здійснюється за протоколом I2C, що дозволяє

підключити до тієї ж шини додаткові пристрої, зокрема OLED-дисплей для локальної індикації. Виконавча частина вузла реалізована через триканальний модуль реле, керований безпосередньо з GPIO-виходів ESP32. До реле підключено три виконавчі пристрої: термостат (Реле 1), зволожувач (Реле 2) та вентилятор (Реле 3). Живлення вузла забезпечується блоком живлення 5V DC, від якого отримують енергію як сам мікроконтролер, так і модуль реле.

Програмна частина вузла виконує два паралельних процеси. З інтервалом 15 секунд відбувається зчитування показників сенсора BME280 та їх публікація до Firebase Realtime Database.

Паралельно з інтервалом 5 секунд вузол зчитує поточний стан реле з бази даних і відповідно перемикає GPIO-виходи. Крім цього, вузол самостійно перевіряє виконання авто-правил та оновлює стан реле у базі даних при їх спрацюванні.

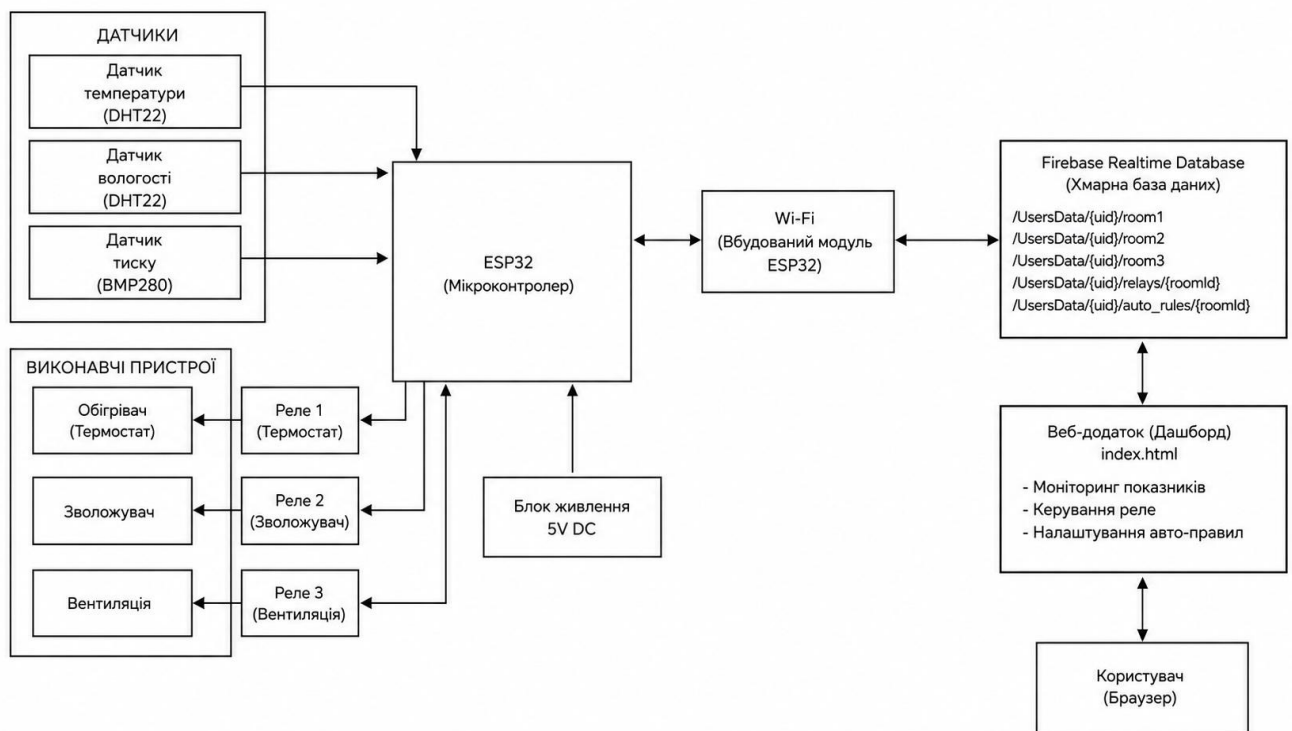


Рисунок 2.1 – Структурна схема програмно-технічного засобу контролю та керування мікрокліматом

Центральним елементом архітектури є Firebase Realtime Database – хмарна база даних реального часу з NoSQL-структурою на основі JSON-дерева. Дані організовано ієрархічно: кореневий вузол UsersData містить підвузли, ідентифіковані унікальним UID автентифікованого користувача. Під UID знаходяться вузли room1, room2, room3 із поточними показниками мікроклімату; вузол relays із підвузлами для кожної аудиторії та трьома булевими значеннями стану реле; вузол auto\_rules із пороговими значеннями temp\_min, hum\_min та temp\_max для кожної аудиторії окремо.

Ключовою перевагою Firebase Realtime Database є механізм підписки на події: клієнти, як ESP32-вузли, так і веб-застосунок отримують оновлення миттєво після зміни будь-якого вузла даних без необхідності виконання циклічних HTTP-запитів. Це забезпечує мінімальну затримку між записом нового значення одним клієнтом та його відображенням у всіх інших клієнтах.

Доступ до даних захищено правилами безпеки Firebase, які перевіряють автентифікацію кожного запиту та звіряють UID запитувача з UID вузла даних. Автентифікація реалізована через Firebase Authentication з провайдером Email/Password.

Веб-застосунок розгорнуто на Firebase Hosting і доступний за публічною адресою classroom-microclimate.web.app. Застосунок реалізовано як статичний односторінковий HTML-документ із вбудованими стилями та JavaScript-кодом, що взаємодіє з Firebase через офіційний JavaScript SDK.

Після успішної авторизації користувач отримує доступ до дашборду, на якому відображаються три картки аудиторій із поточними показниками мікроклімату, елементами керування реле та полями для налаштування авто-правил. Всі дані оновлюються в режимі реального часу завдяки функції onValue SDK, яка підписується на зміни у відповідних вузлах бази даних. Команди керування реле передаються до бази даних функцією set, після чого вузол ESP32 зчитує їх при наступній перевірці стану та виконує фізичне перемикання.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>26 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

## 2.3 Монтажна схема програмно-технічного засобу

Монтажна схема програмно-технічного засобу відображає електричні з'єднання між усіма компонентами вузла для однієї навчальної аудиторії. Центральним елементом схеми виступає мікроконтролер ESP32, до якого підключено всі периферійні пристрої, тобто давач вимірювання параметрів мікроклімату, модуль реле, дисплей та система живлення.

Дачач BME280 підключено до ESP32 за інтерфейсом I2C. Лінія даних SDA з'єднана з GPIO21, лінія тактування SCL з GPIO22. Живлення дачача здійснюється від виводу 3.3V мікроконтролера, спільна земля об'єднана з GND ESP32. Дачач забезпечує вимірювання трьох параметрів мікроклімату – температури, відносної вологості та атмосферного тиску в одному корпусі, що спрощує монтаж порівняно з використанням окремих датчиків.

Для керування виконавчими пристроями застосовано чотириканальний модуль реле, з яких задіяно три канали. Сигнальні входи IN1, IN2 та IN3 підключено відповідно до GPIO26, GPIO27 та GPIO14 мікроконтролера. До навантажувальних виходів реле підключено три виконавчі пристрої: термостат, зволожувач та вентилятор. Живлення котушок реле подається від лінії 5V, яку забезпечує перетворювач напруги. Четвертий канал модуля залишається незадіяним і може бути використаний для розширення функціональності в майбутньому.

OLED-дисплей підключено до тієї ж шини I2C, що й сенсор BME280, тобто до тих самих ліній SDA та SCL. Це можливо завдяки тому, що кожен пристрій на шині I2C має унікальну адресу. Дачач BME280 за замовчуванням використовує адресу 0x76, тоді як типовий OLED-дисплей – 0x3C. Дисплей призначено для локального відображення поточних показників мікроклімату безпосередньо на пристрої, незалежно від наявності з'єднання з Firebase.

Система живлення побудована на основі двох літій-іонних акумуляторів формату 18650, з'єднаних послідовно у конфігурації 2S. Захист акумуляторної

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КвРКІ. 022047.22.02.03 ПЗ | Арк.<br>27 |
| Зм. | Арк. | № докum. | Підпис | Дата |                           |            |

батареї та балансування заряду забезпечує модуль BMS (Battery Management System). Від акумуляторної батареї через вимикач живлення напруга подається на підвищувальний DC-DC перетворювач напруги, який стабілізує вихідну напругу до рівня 5V. Ця напруга живить модуль реле та через вбудований стабілізатор ESP32 – сам мікроконтролер і підключені до нього пристрої. Використання акумуляторного живлення забезпечує автономну роботу вузла незалежно від наявності мережевої розетки поблизу навчальної аудиторії.

Усі компоненти об'єднані спільною шиною живлення та землі, що забезпечує стабільну роботу схеми та виключає виникнення різниці потенціалів між окремими модулями.

Монтажну схему програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase наведено на рис. 2.2 (монтажну схему створено в Circuit Designer IDE [54]).

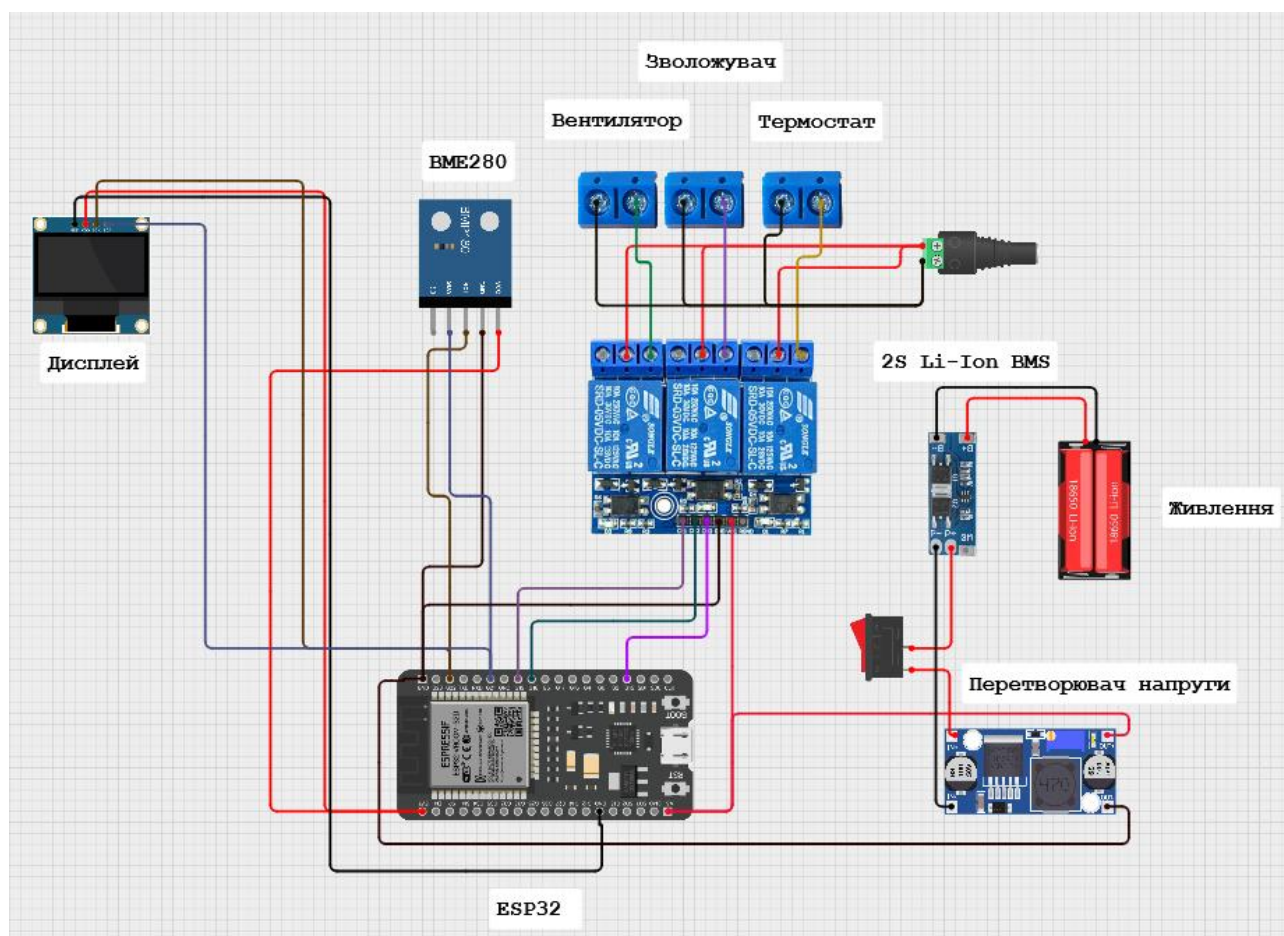


Рисунок 2.2 – Монтажна схема програмно-технічного засобу

## 2.4. Вибір та аналіз апаратних компонентів

Центральним апаратним компонентом кожного вузла системи обрано мікроконтролер ESP32 виробництва Espressif Systems. Вибір зумовлений сукупністю технічних характеристик, що відповідають вимогам системи моніторингу та керування мікрокліматом [22-24].

ESP32 побудовано на двоядерному 32-розрядному процесорі Xtensa LX6 з тактовою частотою до 240 МГц. Обсяг оперативної пам'яті становить 520 КБ, флеш-пам'яті 4 МБ, що достатньо для зберігання прошивки та бібліотек. Вбудований модуль Wi-Fi підтримує стандарти 802.11 b/g/n у діапазоні 2,4 ГГц, а також наявний модуль Bluetooth 4.2/BLE, хоча в даній системі він не використовується.

Мікроконтролер має 34 програмовані GPIO-виводи, підтримує інтерфейси I2C, SPI, UART, I2S, а також два 12-розрядних АЦП із 18 каналами. Наявність апаратної підтримки I2C дозволяє підключити сенсор BME280 та OLED-дисплей на одну шину без додаткових мікросхем. Живлення мікроконтролера — від 3,0 до 3,6 В, робоча температура від -40°C до +85°C. Споживана потужність у режимі активного Wi-Fi складає близько 80–170 мА.

Порівняно з попередником ESP8266, мікроконтролер ESP32 має суттєві переваги, зокрема подвійне ядро процесора дозволяє розподілити задачу публікації даних та задачу перевірки реле між різними потоками виконання; наявність апаратного шифрування AES, SHA та RSA підвищує безпеку з'єднання з Firebase; більший обсяг пам'яті забезпечує комфортне розміщення бібліотек FirebaseClient та Adafruit BME280 без ризику переповнення.

Для вимірювання параметрів мікроклімату обрано сенсор BME280 виробництва Bosch Sensortec (рис. 2.3). Ключовою перевагою цього сенсора є вимірювання трьох фізичних величин – температури, відносної вологості та атмосферного тиску в єдиному корпусі розмірами 2,5 × 2,5 × 0,93 мм. Це спрощує

монтаж схеми, зменшує кількість з'єднань і знижує можливість виникнення апаратних помилок порівняно з використанням трьох окремих датчиків.

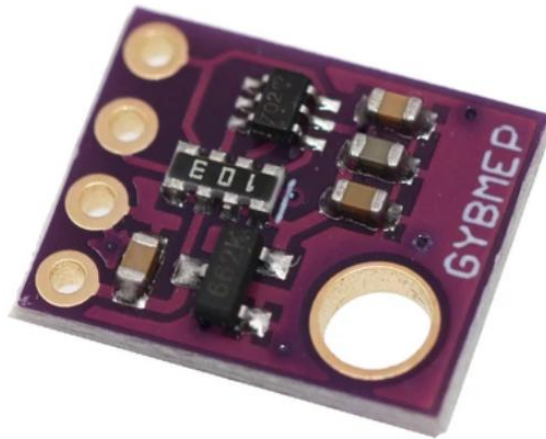


Рисунок 2.3 – Давач ВМЕ280

Таблиця 2.2 – Основні технічні характеристики мікроконтролера ESP32

| Характеристика              | Значення                       |
|-----------------------------|--------------------------------|
| Процесор                    | Xtensa LX6, 2 ядра, до 240 МГц |
| Оперативна пам'ять (SRAM)   | 520 КБ                         |
| Флеш-пам'ять                | 4 МБ                           |
| Wi-Fi                       | 802.11 b/g/n, 2,4 ГГц          |
| Bluetooth                   | 4.2 / BLE                      |
| GPIO                        | 34 програмовані виводи         |
| Інтерфейси                  | I2C, SPI, UART, I2S, ADC, DAC  |
| Напруга живлення            | 3,0–3,6 В                      |
| Споживання (Wi-Fi активний) | 80–170 мА                      |
| Робоча температура          | -40°C ... +85°C                |

Діапазон вимірювання температури становить від  $-40^{\circ}\text{C}$  до  $+85^{\circ}\text{C}$  з точністю  $\pm 1,0^{\circ}\text{C}$  (типове значення  $\pm 0,5^{\circ}\text{C}$  при  $25^{\circ}\text{C}$ ). Діапазон вимірювання відносної вологості від 0% до 100% із точністю  $\pm 3\%$ . Діапазон вимірювання атмосферного тиску від 300 до 1100 гПа з точністю  $\pm 1,0$  гПа. Сенсор підтримує інтерфейси I2C та SPI; у даній системі використовується I2C з адресою 0x76.

Споживана потужність сенсора в режимі нормального вимірювання становить менше 3,6 мкА, що робить його практично непомітним з точки зору енергоспоживання всього вузла. Час відгуку сенсора на зміну температури становить близько 1 с, що цілком задовольняє вимогу оновлення даних раз на 15 секунд.

Таблиця 2.3 – Технічні характеристики сенсора BME280

| Параметр                      | Значення   |
|-------------------------------|--|
| Вимірювані величини           | Температура, вологість, тиск                                 |
| Діапазон температури          | $-40^{\circ}\text{C} \dots +85^{\circ}\text{C}$              |
| Точність температури          | $\pm 0,5^{\circ}\text{C}$ (типова при $25^{\circ}\text{C}$ ) |
| Діапазон вологості            | 0% ... 100% RH   |
| Точність вологості            | $\pm 3\%$ RH   |
| Діапазон тиску                | 300 ... 1100 гПа   |
| Точність тиску                | $\pm 1,0$ гПа  |
| Інтерфейс                     | I2C (адреса 0x76) / SPI                                      |
| Напруга живлення              | 1,71–3,6 В   |
| Споживання (нормальний режим) | < 3,6 мкА  |
| Розміри корпусу               | $2,5 \times 2,5 \times 0,93$ мм                              |

Важливо розрізнити сенсор ВМЕ280 від зовні схожого ВМР280: останній вимірює лише температуру та тиск, не маючи каналу вимірювання вологості. У даному програмно-технічному засобі обрано саме ВМЕ280, оскільки вологість є одним із трьох ключових параметрів мікроклімату, що підлягають моніторингу.

Для комутації виконавчих пристроїв застосовано чотириканальний релейний модуль на базі електромеханічних реле типу SRD-05VDC-SL-C. Хоча в системі задіяно лише три канали, чотириканальний модуль обрано з міркувань резервування та можливого розширення функціональності в подальшому.

Котушка кожного реле живиться від напруги 5 В і керується логічним сигналом рівня 3,3 В з GPIO-виходу ESP32 через транзисторний ключ та оптронну розв'язку, що захищає мікроконтролер від перешкод силового кола. Контакти реле розраховані на комутацію змінного струму напругою до 250 В та силою струму до 10 А, або постійного струму до 30 В та 10 А. Такі параметри дозволяють підключати реальні побутові прилади, такі як електронагрівачі, зволожувачі та вентилятори без проміжних підсилювачів потужності.



Рисунок 2.4 – Чотирьохканальний модуль реле srd-05vdc-sl-c

Модуль є активним за низьким рівнем сигналу: подача LOW на вхід IN вмикає реле, HIGH – вимикає. Це враховано в прошивці ESP32: при ініціалізації

GPIO-виходи встановлюються у HIGH, щоб запобігти мимовільному спрацюванню реле при завантаженні мікроконтролера.

До кожного вузла підключено монохромний OLED-дисплей із діагоналлю 0,96 дюйма та роздільною здатністю  $128 \times 64$  пікселів на базі контролера SSD1306. Дисплей підключено до тієї ж шини I2C, що й сенсор BME280, за адресою 0x3C, що дозволяє обійтись двома провідниками даних для обох пристроїв одночасно.

Призначення дисплея – локальна індикація поточних показників мікроклімату безпосередньо на вузлі, без необхідності звертатись до веб-інтерфейсу. Це особливо корисно під час налагодження системи або у випадках тимчасової відсутності з'єднання з мережею. На дисплеї відображаються значення температури, вологості та тиску, а також стан Wi-Fi-з'єднання та стан реле.

Споживання OLED-дисплея SSD1306 становить близько 20 мА при повній яскравості. Напруга живлення становить 3,3–5 В, що дозволяє підключати його безпосередньо від виводу 3,3 В мікроконтролера ESP32.

## 2.5 Проектування структури бази даних

Структура Firebase Realtime Database спроектована з урахуванням двох ключових вимог, що включають ефективність читання та запису при частому оновленні телеметричних даних, а також чітка ізоляція даних за UID користувача для забезпечення безпеки.

Кореневий вузол бази даних містить єдиний підвузол UserData, під яким знаходяться вузли, ідентифіковані UID автентифікованих користувачів. Така структура дозволяє правилам безпеки перевіряти приналежність даних конкретному користувачу без складних запитів.

Під вузлом UID знаходяться чотири групи підвузлів. Перша група це room1, room2, room3 містить поточні показники мікроклімату кожної аудиторії.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КвРКІ. 022047.22.02.03 ПЗ | Арк.<br>33 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Кожен вузол аудиторії має поля temperature (значення float), humidity (значення float), pressure (значення float) та timestamp (Unix-час у секундах). Друга група relays містить підвузли room1/room2/room3, кожен із яких має три булеві поля relay1, relay2, relay3, що відповідають стану термостата, зволожувача та вентиляції. Третя група auto\_rules містить підвузли room1/room2/room3 із полями temp\_min, hum\_min та temp\_max, що визначають порогові значення для автоматичного керування. Четверта група зарезервована для розширення наприклад, зберігання часових рядів показників для побудови графіків.

Таблиця 2.4 – Структура вузлів Firebase Realtime Database

| Шлях у базі даних                  | Тип даних | Призначення                      |
|------------------------------------|-----------|----------------------------------|
| UserData/{uid}/room1/temperature   | float     | Поточна температура, аудиторія 1 |
| UserData/{uid}/room1/humidity      | float     | Поточна вологість, аудиторія 1   |
| UserData/{uid}/room1/pressure      | float     | Поточний тиск, аудиторія 1       |
| UserData/{uid}/room1/timestamp     | integer   | Unix-час останнього оновлення    |
| UserData/{uid}/relays/room1/relay1 | boolean   | Стан термостата, аудиторія 1     |
| UserData/{uid}/relays/room1/relay2 | boolean   | Стан зволожувача, аудиторія 1    |
| UserData/{uid}/relays/room1/relay3 | boolean   | Стан вентиляції, аудиторія 1     |

| Шлях у базі даних                        | Тип даних | Призначення                         |
|--|-----------|-------------------------------------|
| UserData/{uid}/auto_rules/room1/temp_min | float     | Мін. температура для авто-вмикання  |
| UserData/{uid}/auto_rules/room1/hum_min  | float     | Мін. вологість для авто-вмикання    |
| UserData/{uid}/auto_rules/room1/temp_max | float     | Макс. температура для авто-вмикання |

Аналогічна структура повторюється для room2 та room3. Така організація даних дозволяє ESP32-вузлу звертатись лише до власного підвузла, не зчитуючи зайвих даних. Веб-застосунок підписується на всі три вузли аудиторій одночасно, отримуючи оновлення в реальному часі незалежно від того, який із вузлів записав нові дані.

## 2.6. Проєктування алгоритму автоматичного керування

Алгоритм автоматичного керування мікрокліматом реалізований на рівні прошивки кожного ESP32-вузла і виконується локально без залежності від доступності веб-застосунку. Це забезпечує автономну роботу системи навіть у випадках, коли оператор не взаємодіє з інтерфейсом.

Логіка алгоритму базується на порівнянні поточних показників мікроклімату з пороговими значеннями, що зберігаються у вузлі auto\_rules відповідної аудиторії. Після кожного вимірювання сенсором BME280 мікроконтролер зчитує актуальні порогові значення з бази даних, що дозволяє оператору змінювати параметри авто-правил через веб-інтерфейс без перепрошивки пристрою.

Алгоритм керування термостатом (Реле 1) реалізує гістерезис для запобігання частому перемикачню на межі порогу. Реле вмикається, коли

температура опускається нижче значення  $temp\_min$ . Вимикання відбувається лише тоді, коли температура перевищує  $temp\_min$  на величину гістерезису ( $1^{\circ}C$ ), тобто досягає значення  $temp\_min + 1^{\circ}C$ . Аналогічно реалізовано алгоритм для зволожувача (Реле 2): вмикання при вологості нижче  $hum\_min$ , вимикання при досягненні  $hum\_min + 3\%$ . Реле вентиляції (Реле 3) вмикається при перевищенні температурою значення  $temp\_max$  і вимикається при зниженні до  $temp\_max - 1^{\circ}C$ .

При спрацюванні авто-правила мікроконтролер записує новий стан реле до Firebase RTDB, що забезпечує відображення змін у веб-застосунку в режимі реального часу. Таким чином, оператор завжди бачить актуальний стан виконавчих пристроїв незалежно від причини їх перемикавання – ручного чи автоматичного.

Таблиця 2.5 – Умови спрацювання авто-правил керування виконавчими пристроями

| Реле   | Виконавчий пристрій | Умова вмикання  | Умова вимикання                 |
|--------|---------------------|-----------------|---------------------------------|
| Реле 1 | Термостат           | $T < temp\_min$ | $T \geq temp\_min + 1^{\circ}C$ |
| Реле 2 | Зволожувач          | $H < hum\_min$  | $H \geq hum\_min + 3\%$         |
| Реле 3 | Вентиляція          | $T > temp\_max$ | $T \leq temp\_max - 1^{\circ}C$ |

Ручне керування з веб-застосунку має рівноправний пріоритет з автоматичним: якщо оператор вручну вимкнув реле, яке було ввімкнене автоматично, воно залишатиметься вимкненим до наступного циклу перевірки авто-правил. При наступному спрацюванні умови авто-правило знову ввімкне

реле. Для повного блокування автоматичного керування конкретним реле оператор має відповідно змінити порогові значення авто-правил через інтерфейс.

Значення гістерезису обрано емпірично: для температури – 1°C, для вологості – 3%. Ці значення є достатніми для усунення ефекту «брязкіту» при незначних коливаннях показників сенсора поблизу порогового значення і водночас не призводять до суттєвого відхилення від заданого рівня мікроклімату.

## 2.7 Висновки до другого розділу

У даному розділі сформульовано систему функціональних, технічних і безпекових вимог до програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій. Розроблено трирівневу архітектуру системи, що включає апаратні вузли на базі ESP32, хмарну платформу Firebase та веб-застосунок. Обґрунтовано вибір апаратних компонентів: мікроконтролер ESP32 обрано завдяки вбудованому Wi-Fi, достатній продуктивності та підтримці необхідних інтерфейсів; датчик BME280 завдяки одночасному вимірюванню трьох параметрів мікроклімату в одному корпусі; чотириканальний модуль реле завдяки оптронній розв'язці та достатньому навантажувальному струму для реальних виконавчих пристроїв. Спроектовано структуру бази даних Firebase RTDB з ізоляцією даних за UID та ієрархічним розподілом вузлів для телеметрії, стану реле та авто-правил.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>37 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

### 3 РЕАЛІЗАЦІЯ, РОЗГОРТАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ КОНТРОЛЮ ТА КЕРУВАННЯ МІКРОКЛІМАТОМ НАВЧАЛЬНИХ АУДИТОРІЙ НА ОСНОВІ ESP32 ТА FIREBASE

3.1 Узагальнений підхід до реалізації, розгортання та тестування програмно-технічного засобу

Реалізація програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій побудована на основі хмарної архітектури, у якій Firebase виступає централізованим бекендом без необхідності розгортання власного сервера. Система складається з трьох функціональних рівнів: рівня збору та публікації даних, рівня хмарного сховища і рівня користувацького інтерфейсу.

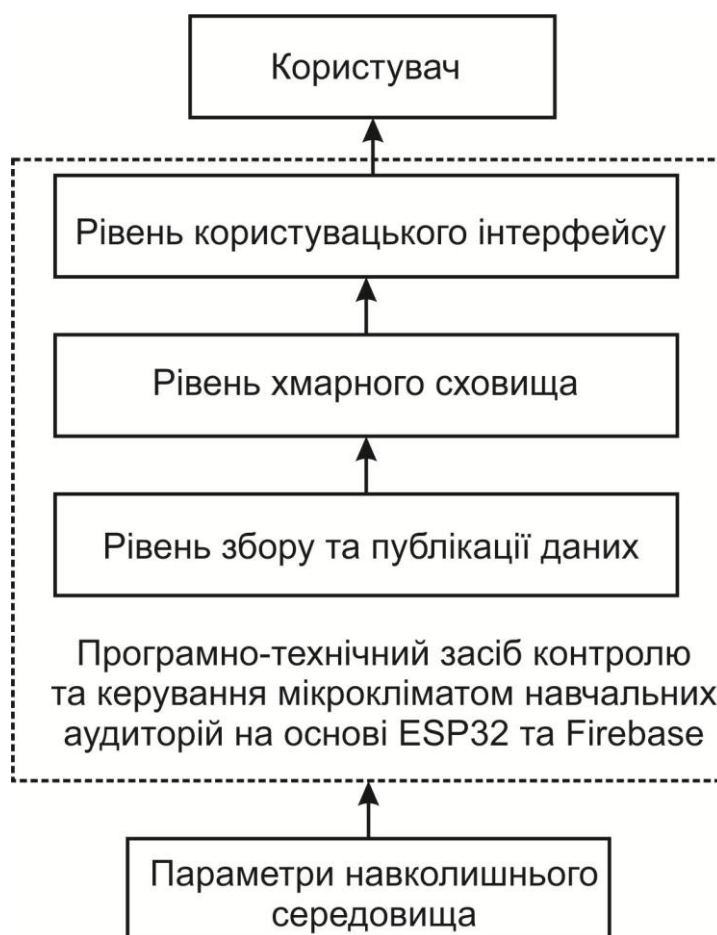


Рисунок 3.1 – Функціональні рівні програмно-технічного засобу

На рівні збору даних у реальному розгортанні системи передбачається використання мікроконтролерів ESP32 із підключеними сенсорами BME280, які вимірюють температуру, відносну вологість та атмосферний тиск у кожній із трьох навчальних аудиторій. До кожного мікроконтролера також підключено три релейних модулі, що виконують роль виконавчих пристроїв – термостата, зволожувача та вентиляції. З метою тестування реалізованого програмно-технічного засобу контролю та керування мікрокліматом, окрім фізичної реалізації, було також відтворено повністю функції через програмний симулятор, реалізований мовою Python, що дозволило перевірити коректність роботи всіх компонентів системи починаючи від публікації телеметричних даних до механізмів ручного та автоматичного керування реле без залучення фізичного обладнання, забезпечивши повноцінне наскрізне тестування хмарної інфраструктури, веб-застосунку та логіки авто-правил в умовах, максимально наближених до реального функціонування системи.

Симулятор побудовано на основі багатопотокової моделі: для кожної аудиторії створюється окремий потік виконання, що функціонує незалежно від інших. Таким чином, три потоки room1, room2 та room3 поведуться як три фізично розділені ESP32-клієнти. Кожен потік з інтервалом 15 секунд генерує реалістичні показники мікроклімату із застосуванням гаусівського шуму та синусоїдального дрейфу, публікує їх до Firebase Realtime Database та з інтервалом 5 секунд зчитує поточний стан трьох реле. Окрім цього, кожен потік локально перевіряє авто-правила: якщо температура опускається нижче встановленого порогу, він самостійно встановлює стан відповідного реле у значення «увімкнено», записуючи зміну до бази даних.

На рівні хмарного сховища налаштовано Firebase Realtime Database з ієрархічною структурою вузлів: дані кожної аудиторії зберігаються під унікальним UID автентифікованого користувача у підвузлах room1, room2, room3, стани реле у підвузлі relays, а порогові значення для автоматичного керування у підвузлі auto\_rules. Доступ до даних захищено правилами безпеки,

що дозволяють читання та запис виключно автентифікованому користувачу. Автентифікацію реалізовано через Firebase Authentication з провайдером Email/Password.

```
{
  "yN2WEkSiGhVcrpiUDZxb3WspCxt2": {
    "auto_rules": {
      "room1": {
        "hum_min": 40,
        "temp_max": 26,
        "temp_min": 19
      },
      "room2": {
        "hum_min": 40,
        "temp_max": 26,
        "temp_min": 19
      },
      "room3": {
        "hum_min": 40,
        "temp_max": 26,
        "temp_min": 19
      }
    }
  },
  "room1": {
    "humidity": 47.04,
    "pressure": 1013.01,
    "temperature": 21.04,
    "timestamp": 1778615313
  },
  "room2": {
    "humidity": 51.55,
    "pressure": 1012.64,
    "temperature": 19.52,
    "timestamp": 1778615313
  },
  "room3": {
    "humidity": 44.18,
    "pressure": 1013.94,
    "temperature": 23.19,
    "timestamp": 1778615314
  }
}
```

Рисунок 3.2 – Приклад формату даних у БД

На рівні користувацького інтерфейсу розроблено односторінковий веб-застосунок, розгорнутий на Firebase Hosting. Застосунок у режимі реального часу відображає показники мікроклімату трьох аудиторій, надає можливість ручного перемикання реле через браузер та дозволяє змінювати порогові значення авто-правил. Взаємодія з базою даних реалізована через Firebase JavaScript SDK з використанням функції onValue для підписки на живі оновлення без перезавантаження сторінки.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>40 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

### 3.2 Реалізація хмарної інфраструктури на основі Firebase

Реалізацію програмно-технічного засобу моніторингу та керування мікрокліматом розпочато зі створення хмарної інфраструктури на платформі Google Firebase, яка виконує роль централізованого серверного бекенду без необхідності розгортання власного сервера. На початковому екрані консолі Firebase (рис. 3.1) розробник отримує доступ до середовища та має можливість створити новий проєкт. Було створено проєкт «Classroom-Microclimate», якому автоматично відповідатиме унікальний ідентифікатор «classroom-microclimate». Саме цей ідентифікатор використовується надалі у всіх URL-адресах сервісів – бази даних та хостингу.

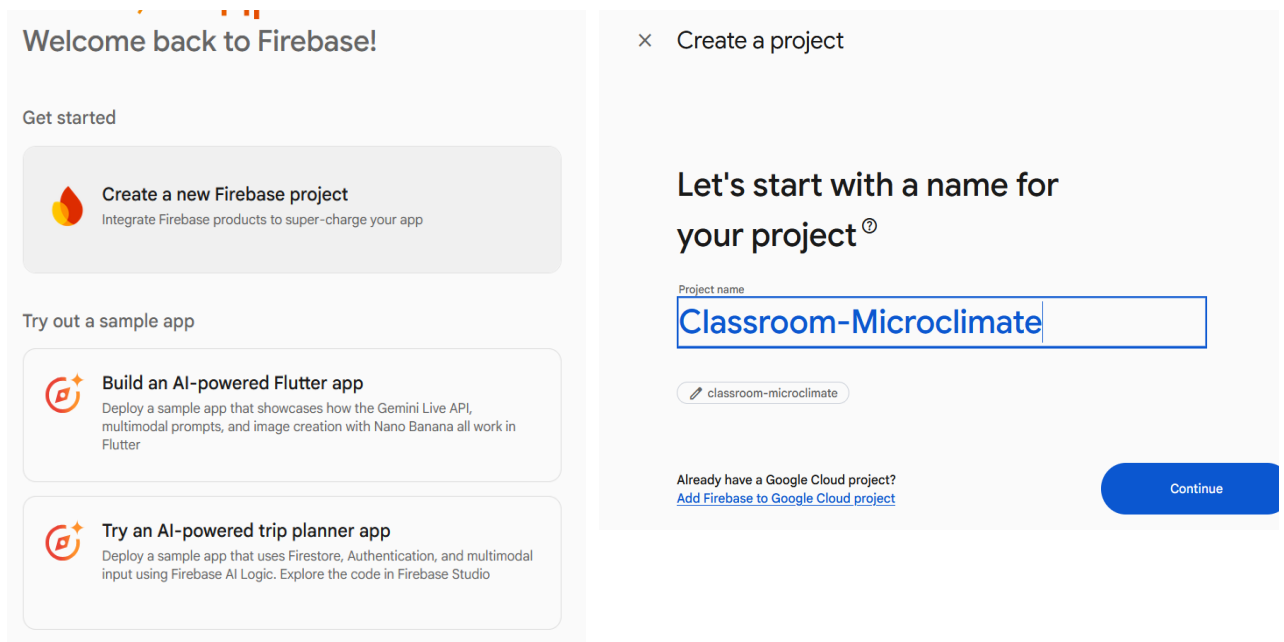


Рисунок 3.3 Початковий екран консолі Firebase та вікно створення проєкту

Після входу до консолі проєкту (рис. 3.4) відображається головна панель керування з переліком доступних категорій сервісів – баз даних, безпеки, хостингу та аналітики. Проєкт працює на безкоштовному тарифному плані Spark, якого достатньо для цілей даної кваліфікаційної роботи.

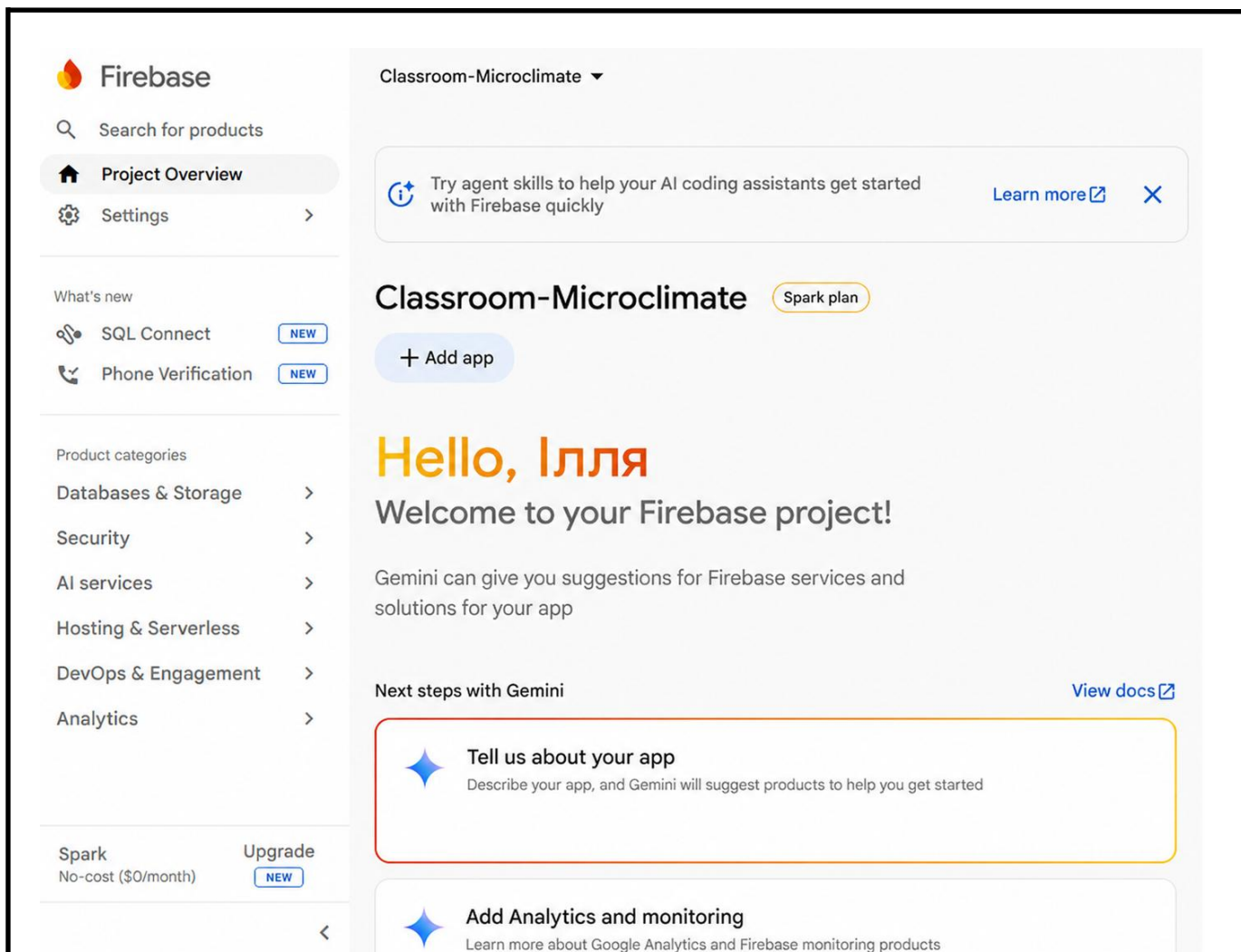


Рисунок 3.4 – Головна панель проєкту Classroom-Microclimate у консолі Firebase

Для забезпечення захищеного доступу до системи активовано сервіс Firebase Authentication. Автентифікація виступає обов'язковим компонентом: правила безпеки бази даних прив'язані до ідентифікатора автентифікованого користувача, тому жодний запит до даних не може бути виконаний без попередньої авторизації.

Далі на вкладці Sign-in method обрано провайдер Email/Password (рис. 3.5) як найбільш придатний для застосунку з фіксованим числом адміністраторів. Інші провайдери, такі як Google, Facebook, GitHub тощо були залишені вимкненими.

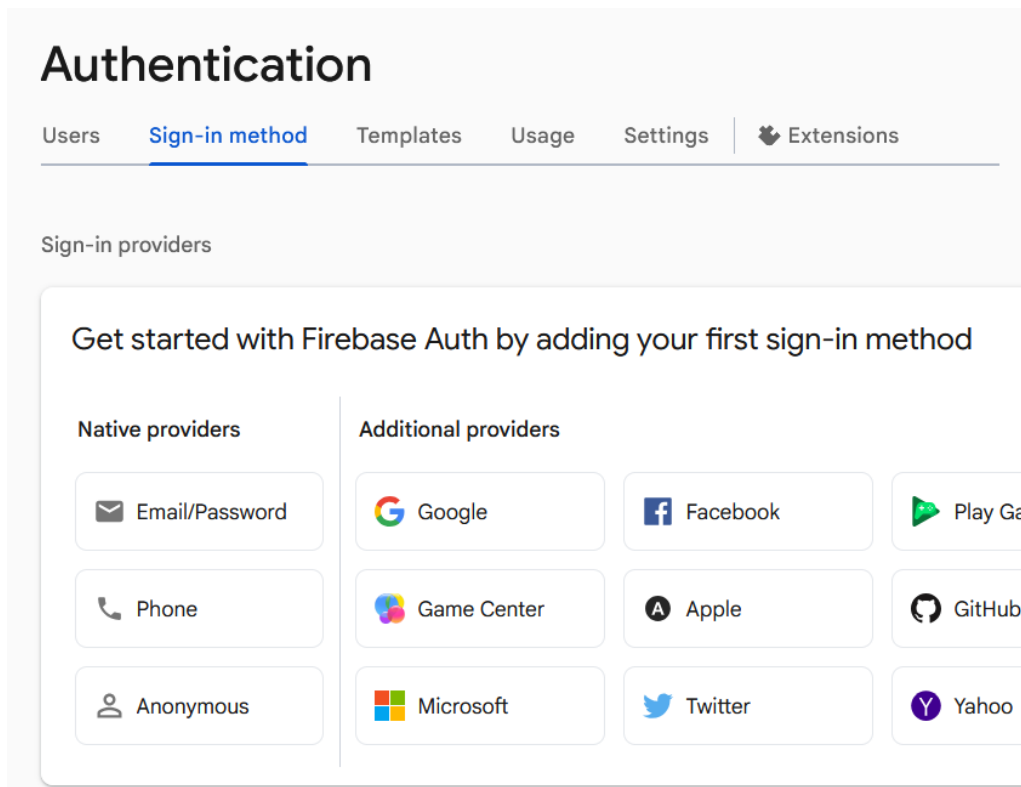


Рисунок 3.5 – Вибір методу автентифікації Email/Password серед доступних провайдерів

Провайдер Email/Password активовано перемиканням відповідного перемикача (рис. 3.6). Опцію passwordless sign-in (вхід лише за посиланням на пошту) залишено вимкненою, тобто система використовуватиме класичну пару логін/пароль.

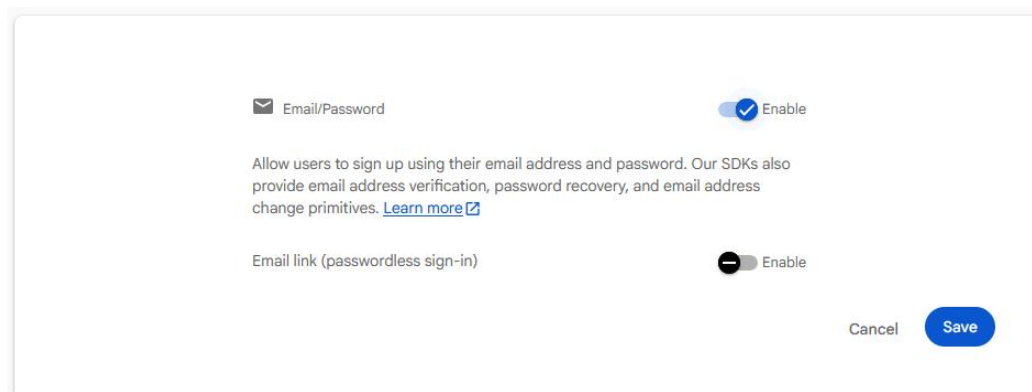


Рисунок 3.6 – Увімкнення провайдера Email/Password з підтвердженням збереження



Classroom-Microclimate, ідентифікатор classroom-microclimate та UID. Отримані дані мають важливе практичне значення, оскільки вони використовуються для подальшого налаштування конфігураційних файлів веб-застосунку та клієнтів ESP32, а також відповідного Python-скрипту.

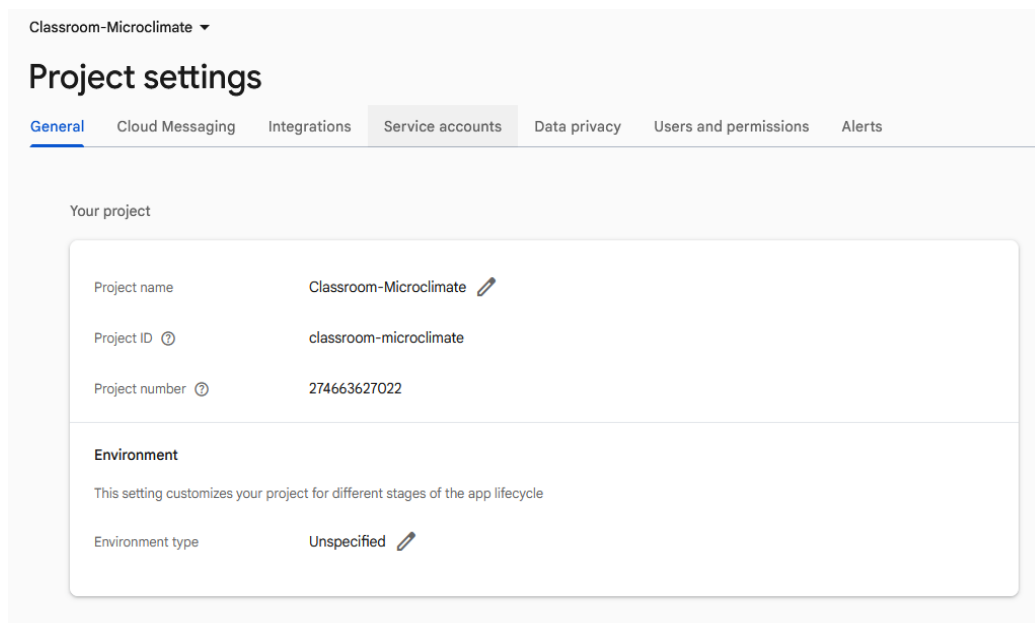


Рисунок 3.10 – Загальні налаштування проєкту

На тій самій сторінці у блоці Your apps ініційовано реєстрацію веб-застосунку (рис. 3.11). Для цього обрано платформу Web (значок «`</>`»), що запустило процес реєстрації та генерації конфігурації.

Після створення проєкту було виконано реєстрацію веб-застосунку у Firebase. У формі реєстрації вказано назву застосунку Classroom-Microclimate, увімкнено опцію Firebase Hosting та обрано відповідний сайт classroom-microclimate. Це дозволило одночасно отримати необхідну конфігурацію застосунку та підготувати інфраструктуру для хостингу.

На наступному етапі Firebase згенерував об'єкт конфігурації firebaseConfig, який містив ключові параметри (API key, authDomain, projectId, storageBucket, messagingSenderId та appId). Цей об'єкт було скопійовано та

інтегровано у файли index.html і login.html веб-застосунку для підключення Firebase SDK, а також у файли прошивки для мікроконтролерів ESP32.

Для подальшого розгортання застосунку було встановлено Firebase CLI за допомогою команди `npm install -g firebase-tools`. Після авторизації через `firebase login`, ініціалізації проєкту командою `firebase init` та публікації файлів за допомогою `firebase deploy` веб-застосунок став доступним за адресою <https://classroom-microclimate.web.app>.

У розділі Your apps Firebase відобразив зареєстрований веб-застосунок Classroom-Microclimate з відповідним App ID та прив'язаним сайтом хостингу.

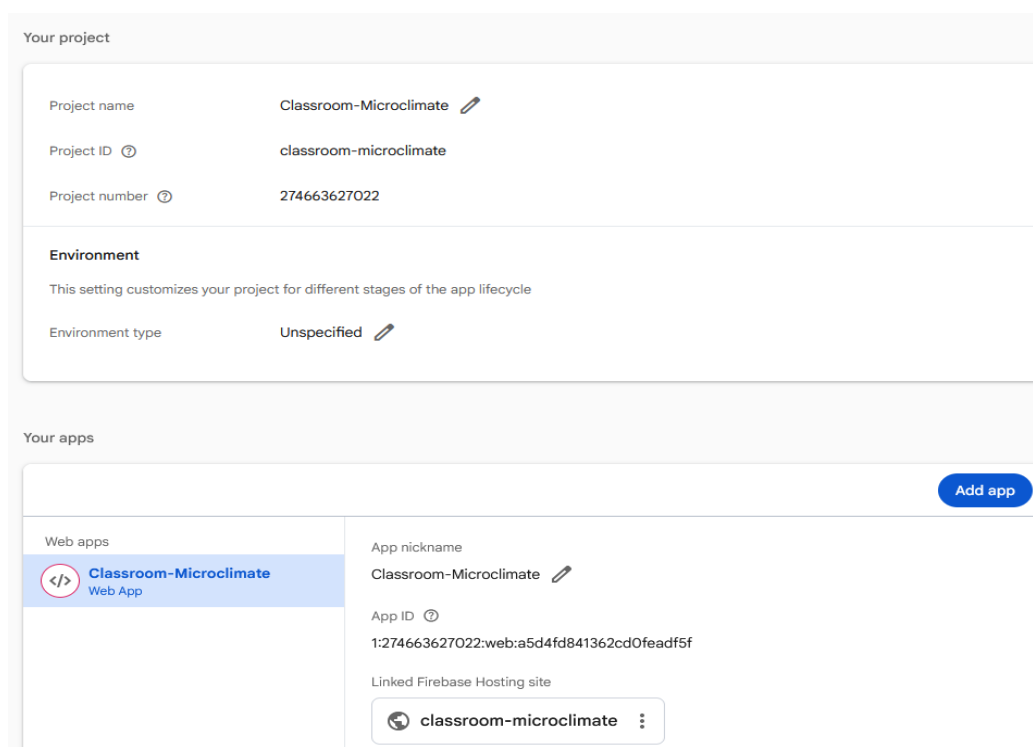


Рисунок 3.11 – Зареєстрований веб-застосунок у розділі налаштувань проєкту

Наступним кроком було налаштовано сховище даних Firebase Realtime Database. На стартовому екрані сервісу запропоновано створити базу даних. RTDB обрано замість Cloud Firestore з огляду на простішу структуру ключ-значення, яка добре підходить для потокового запису телеметрії з сенсорів.

У діалозі налаштування бази (рис. 3.12) обрано географічне розташування сервера Belgium (europe-west1). Це мінімізує затримки для кінцевих пристроїв, що знаходяться в Україні (у порівнянні із іншими варіантами розташування).

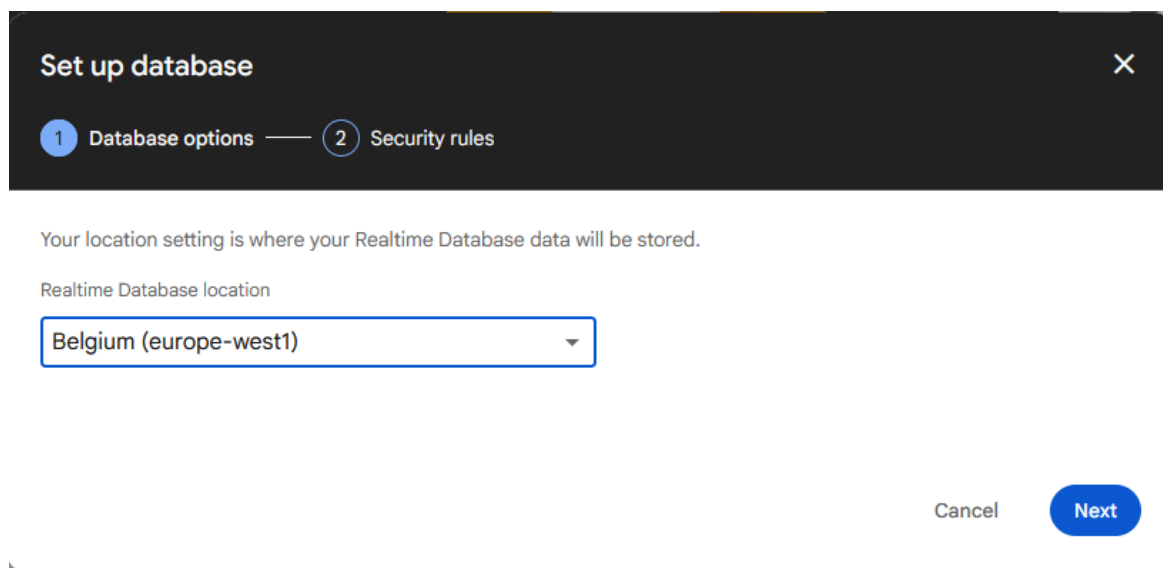


Рисунок 3.12 – Вибір регіону розташування бази даних

На другому кроці налаштування (рис. 3.13) обрано режим запуску Start in test mode, який на початковому етапі дозволяє читання і запис без автентифікації – це спрощує первинне налагодження. Надалі було змінено правила безпеки на обмежувальні, прив'язані до UID користувача.

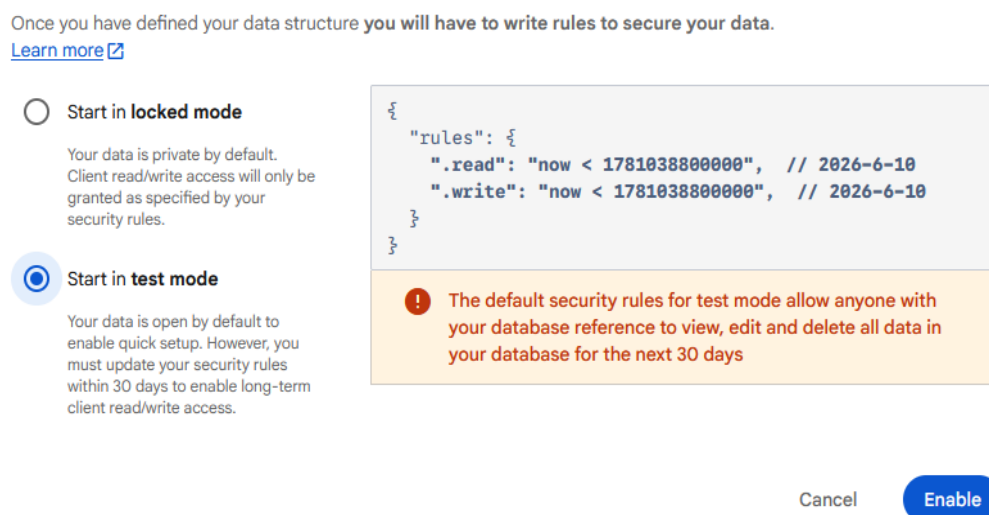


Рисунок 3.13 – Вибір початкового режиму безпеки бази даних (test mode)

Після створення бази даних у вкладці Data відображався її URL та початковий стан – значення null, що підтверджує успішне розгортання порожньої бази даних. URL <https://classroom-microclimate-default-rtdb.europe-west1.firebaseio.com> використовується у конфігурації Python-симулятора та веб-застосунку, а також кожного із трьох клієнтів на основі ESP32.

Завершальним кроком налаштування хмарної інфраструктури було встановлення правил безпеки бази даних (рис. 3.14). Реалізовано правило, згідно з яким читання та запис до вузла UsersData/{uid} дозволяється виключно автентифікованому користувачу з відповідним UID. Це гарантує ізоляцію даних: навіть якщо в системі існуюватиме кілька облікових записів, кожен матиме доступ лише до власних вузлів.

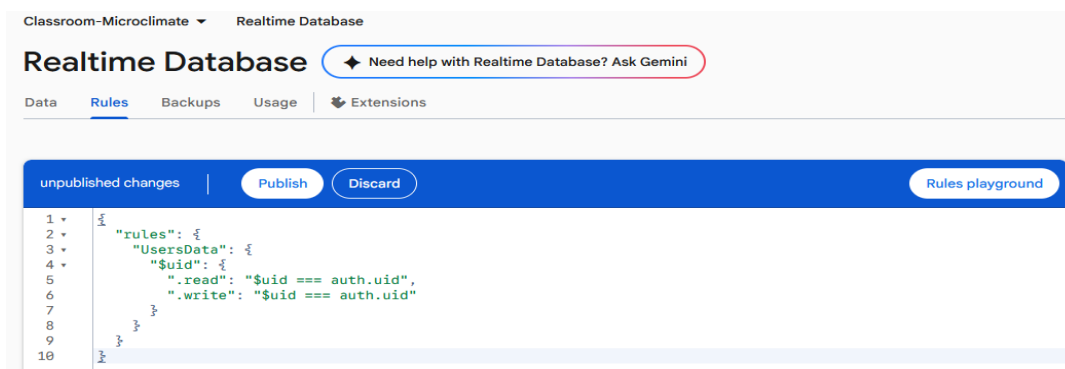


Рисунок 3.14 – Правила безпеки Realtime Database (обмеження доступу за UID автентифікованого користувача)

### 3.2. Реалізація програмного симулятора ESP32-клієнтів

У доповнення до фізичної реалізації, з метою верифікації та тестування програмно-технічного засобу, розроблено Python-симулятор, який відтворює роботу апаратної частини, зокрема мікроконтролерів ESP32, датчиків BME280 та DHT22, а також модулів реле в програмному середовищі. Програма функціонує в трьох паралельних потоках, кожен із яких є незалежним клієнтом і відповідає одній навчальній аудиторії.

Кожен потік-клієнт виконує дві основні функції. Перша це генерація та публікація телеметричних даних мікроклімату: температури, вологості та атмосферного тиску. Значення генеруються із застосуванням гаусівського шуму та синусоїдального дрейфу, що імітує реальну поведінку датчиків BME280 та DHT22 у приміщенні. Друга функція – зчитування стану реле з бази даних і реагування на зміни, ініційовані як веб-застосунком (ручне керування), так і автоматичними правилами ( $\text{if } T < T_{\text{min}} \rightarrow \text{relay1} = \text{true}$ ).

Перший запуск симулятора (рис. 3.15) демонструє успішне підключення до Firebase (повідомлення «[Firebase] Підключено ✓»), паралельний старт трьох потоків room1, room2, room3 та перше читання і запис авто-правил. Потоки розпочали публікацію даних з інтервалом 15 секунд.

```

=====
ESP32 Classroom Microclimate Simulator
=====

[Firebase] Підключено ✓

Аудиторії: room1 (cyan) room2 (green) room3 (yellow)
Сенсор: кожні 15 сек
Реле: кожні 5 сек

Ctrl+C для зупинки

-----
22:08:08 [room1] Запуск симулятора (Firebase)...
22:08:09 [room2] Запуск симулятора (Firebase)...
22:08:09 [room1] Авто-правила записано у Firebase: {'temp_min': 19.0, 'hum_min': 40.0, 'temp_max': 26.0}
22:08:09 [room1] 🌡️ 20.9°C 💧 47.5% ↓ 1013.0hPa | ● Термостат ● Зволожувач ● Вентиляція
22:08:09 [room2] Авто-правила записано у Firebase: {'temp_min': 19.0, 'hum_min': 40.0, 'temp_max': 26.0}
22:08:09 [room2] 🌡️ 19.4°C 💧 52.1% ↓ 1012.5hPa | ● Термостат ● Зволожувач ● Вентиляція
22:08:09 [room3] Запуск симулятора (Firebase)...
22:08:09 [room3] Авто-правила записано у Firebase: {'temp_min': 19.0, 'hum_min': 40.0, 'temp_max': 26.0}
22:08:10 [room3] 🌡️ 23.1°C 💧 44.5% ↓ 1013.8hPa | ● Термостат ● Зволожувач ● Вентиляція
22:08:24 [room1] 🌡️ 20.9°C 💧 47.3% ↓ 1013.1hPa | ● Термостат ● Зволожувач ● Вентиляція
22:08:25 [room2] 🌡️ 19.4°C 💧 52.1% ↓ 1012.5hPa | ● Термостат ● Зволожувач ● Вентиляція
22:08:25 [room3] 🌡️ 23.1°C 💧 45.0% ↓ 1013.8hPa | ● Термостат ● Зволожувач ● Вентиляція

```

Рисунок 3.15 – Перший запуск Python-симулятора (підключення до Firebase та старт трьох потоків)

У процесі подальшої роботи симулятор циклічно публікував оновлені показники для кожної аудиторії. Було відмічено, що значення температури та вологості незначно змінюються між ітераціями. Це підтверджує коректну роботу генератора реалістичних даних. Реле в цьому стані вимкнені (порожні кола поруч із назвами).

### 3.3. Програмна реалізація ESP32-клієнтів

Програмна реалізація мікроконтролерної частини системи здійснюється у середовищі Arduino IDE з використанням мови C++. Для кожної з трьох навчальних аудиторій передбачено окремий мікроконтролер ESP32, на який завантажується ідентична прошивка. Відмінність між екземплярами полягає лише у значенні константи ROOM\_ID, яка визначає, до якого вузла бази даних публікуватимуться дані конкретного пристрою.

Для роботи прошивки підключено такі бібліотеки: FirebaseClient авторства Mobizt для взаємодії з Firebase Realtime Database, Adafruit BME280 Library та Adafruit Unified Sensor для зчитування показників із сенсора. Зв'язок із сенсором BME280 здійснюється через інтерфейс I2C. Лінії SDA та SCL підключені до GPIO21 та GPIO22 відповідно. Три релейних модулі підключено до GPIO26, GPIO27 та GPIO14; оскільки більшість поширених модулів реле є активними за низьким рівнем сигналу, стан HIGH на виході мікроконтролера відповідає вимкненому реле, а LOW – увімкненому.

У функції setup() виконується ініціалізація послідовного порту, встановлення початкового стану виходів реле у HIGH (вимкнено), ініціалізація шини I2C та підключення до мережі Wi-Fi. Після встановлення з'єднання ініціалізується Firebase-застосунок через об'єкт UserAuth з параметрами API key, email та пароля користувача, після чого отримується посилання на екземпляр RealtimeDatabase.

Основний цикл loop() реалізує два незалежних таймери. Перший спрацьовує кожні 15 секунд і виконує зчитування показників із BME280 – температури, відносної вологості та атмосферного тиску – та їх запис до Firebase за шляхом UsersData/{UID}/{ROOM\_ID}. Паралельно зчитуються поточні автоправила з вузла auto\_rules/{ROOM\_ID} і перевіряється виконання порогових умов: якщо температура нижча за temp\_min, до бази записується relay1 = true; якщо вологість нижча за hum\_min – relay2 = true; якщо температура перевищує



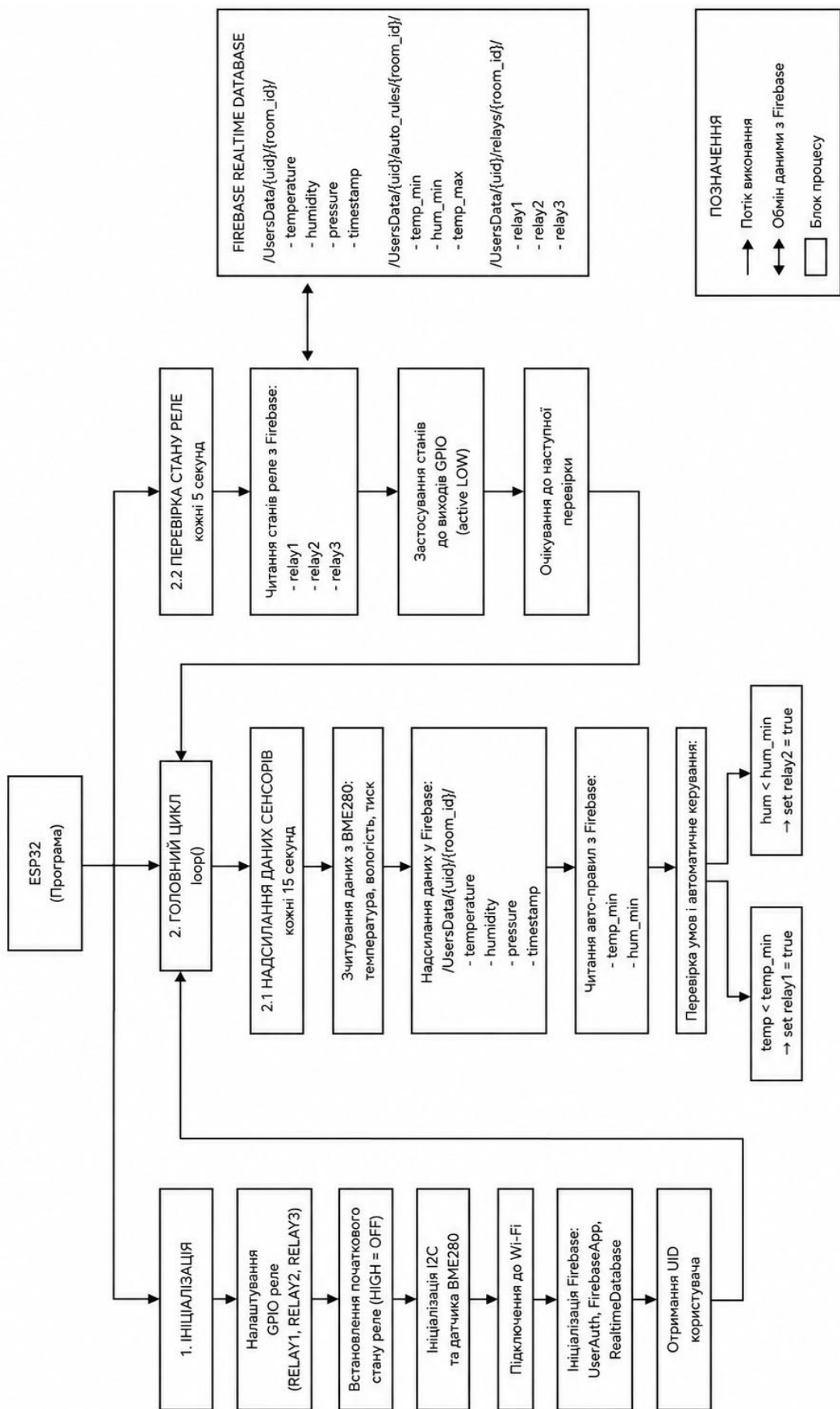


Рисунок 3.17 – Схема організації програмного забезпечення ESP32

В результаті було отримано повідомлення повідомленням «Firebase initialization complete!» і CLI автоматично створив файли `firebase.json`, `.firebaserc` та `database.rules.json`, завантаживши поточні правила безпеки бази даних із хмари.

```
/ Set up automatic builds and deploys with GitHub? No
+ Wrote public\404.html
+ Wrote public\index.html

=== Database Setup

Firebase Realtime Database Security Rules allow you to define how your data should be
structured and when your data can be read from and written to.

/ What file should be used for Realtime Database Security Rules? database.rules.json
i database: ensuring required API firebase.database.googleapis.com is enabled...
+ database: required API firebase.database.googleapis.com is enabled

i Downloaded the existing Realtime Database Security Rules of database classroom-microclimate-default-rtdb from the Fir
base console
+ Wrote database.rules.json
Future modifications to database.rules.json will update Realtime Database Security Rules when you run
firebase deploy.

=== Agent Skills Setup
If you are using an AI coding agent, Firebase Agent Skills make it an expert at Firebase.
/ Would you like to install agent skills for Firebase? No

+ Wrote configuration info to firebase.json
+ Wrote project information to .firebaserc
+ Wrote .gitignore

+ Firebase initialization complete!
```

Рисунок 3.18 – Успішне завершення ініціалізації Firebase CLI

Деплой виконано командою `firebase deploy` (рис. 3.19). CLI послідовно завантажив файли до сервісу Hosting, зафіналізував версію та опублікував її. Результатом є рядок «Deploy complete!» із посиланням на застосунок за адресою <https://classroom-microclimate.web.app>.

```
=== Deploying to 'classroom-microclimate'...

i deploying hosting
i hosting[classroom-microclimate]: beginning deploy...
i hosting[classroom-microclimate]: found 1 files in public
+ hosting[classroom-microclimate]: file upload complete
i hosting[classroom-microclimate]: finalizing version...
+ hosting[classroom-microclimate]: version finalized
i hosting[classroom-microclimate]: releasing new version...
+ hosting[classroom-microclimate]: release complete

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/classroom-microclimate/overview
Hosting URL: https://classroom-microclimate.web.app
```

Рисунок 3.19 – Успішне розгортання застосунку

### 3.5. Тестування системи

Тестування системи проводилось у три послідовних етапи: перевірка запису даних у Firebase RTDB, перевірка відображення даних у веб-застосунку та перевірка механізмів керування реле, зокрема як ручного, так і автоматичного.

Спочатку відбувалось тестування запису даних у БД Firebase RTDB. Після запуску Python-симулятора перевірено структуру даних у консолі Firebase RTDB. На першому скріншоті (рис. 3.20) видно появу вузла UsersData із UID зареєстрованого користувача. Це підтвердило, що симулятор успішно автентифікується та записує дані під правильним шляхом.

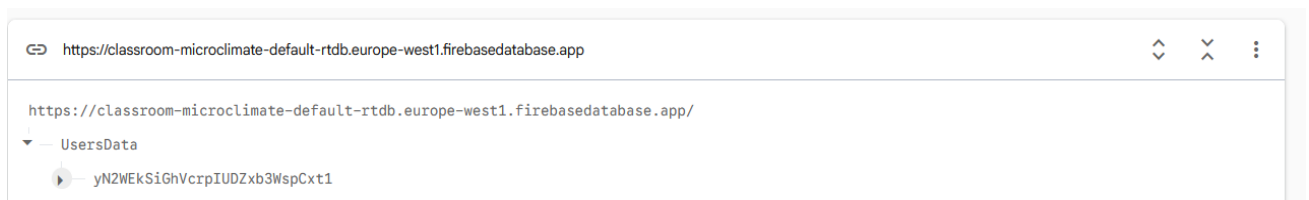


Рисунок 3.20 – Поява вузла UsersData з UID користувача у Firebase Realtime Database після запуску симулятора

Розгортання вузла UID (рис. 3.21) виявило коректну структуру даних. На рис. 3.21 можна побачити структуру даних, включаючи значення для авто правил, так і для кожної кімнати.

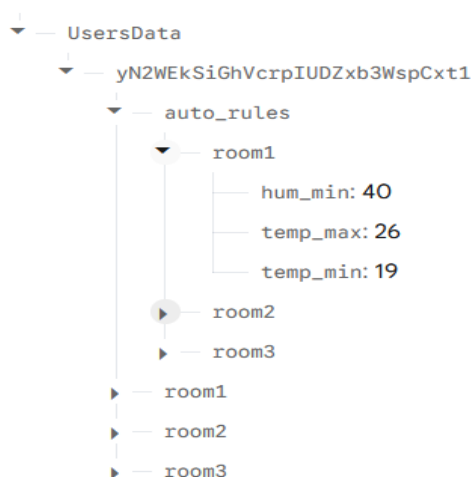


Рисунок 3.21 – Структура бази даних

Розгортання вузлів аудиторій (рис. 3.22) показало наявність актуальних показників: для room1 – humidity: 47.35, pressure: 1013.12, temperature: 20.99; для room2 – humidity: 51.78, pressure: 1012.52, temperature: 19.33. Поле timestamp відображає мітку часу публікації у форматі Unix. Дані оновлюються щоразу при черговій ітерації симулятора або клієнтів ESP32.



Рисунок 3.22 – Актуальні телеметричні дані вузлів у Firebase RTDB

Веб-застосунок відкрито за публічною адресою <https://classroom-microclimate.web.app>. Після авторизації дашборд відобразив картки трьох аудиторій (рис. 3.23) з актуальними значеннями температури, вологості та тиску, що в режимі реального часу оновлювались завдяки механізму onValue Firebase SDK. Також відображено блок правил автоматизації для кожної аудиторії.

Для перевірки ручного керування в консолі симулятора зафіксовано момент, коли веб-застосунок змінив стан реле (рис. 3.24). Записи [manual] relay2 → ON для room1 та [manual] relay1 → ON для room2 підтверджують, що команди перемикачів, надіслані з браузера, успішно записуються до Firebase і зчитуються симулятором у межах інтервалу перевірки реле (5 секунд).

На рисунку 3.37 показано стан дашборду після ручного вмикання реле Термостат у аудиторії 2 через веб-інтерфейс. Перемикач relay1 у картці «Кімната 2» активовано, тоді як в інших аудиторіях реле залишаються вимкненими.

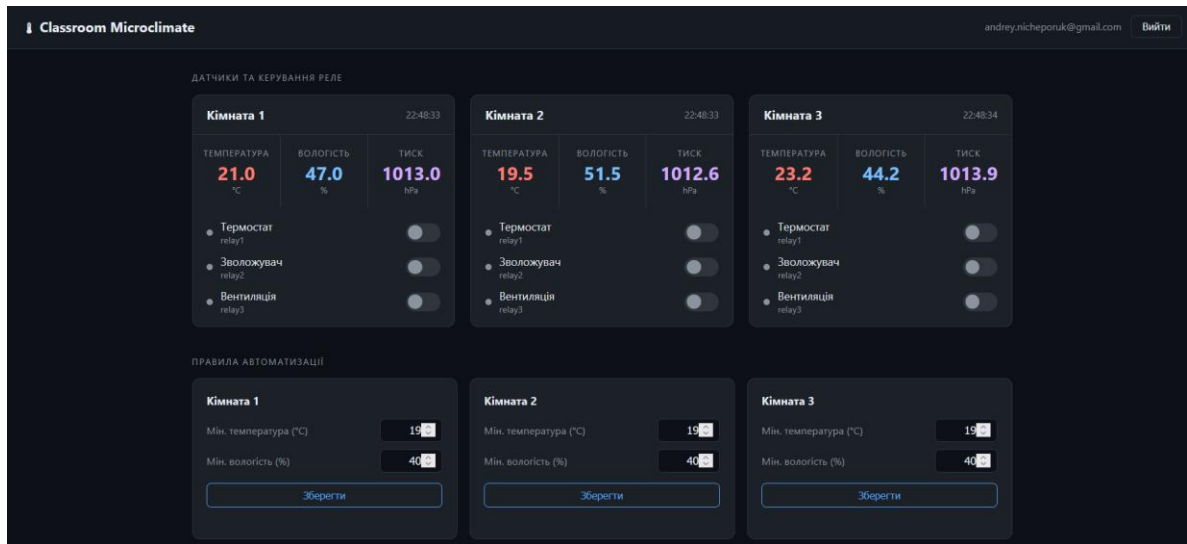


Рисунок 3.24 – Головна сторінка веб-застосунку з трьома картками

```

-----
07:07:05 [room1] Запуск симулятора (Firebase)...
07:07:06 [room2] Запуск симулятора (Firebase)...
07:07:06 [room1] Авто-правила завантажено: {'temp_min': 19.0, 'hum_min': 40.0, 'temp_max': 26.0}
07:07:06 [room2] Авто-правила завантажено: {'temp_min': 19.0, 'hum_min': 40.0, 'temp_max': 26.0}
07:07:06 [room1] 🌡️ 21.2°C 💧 48.1% ↓ 1013.0hPa | 🟡 Термостат 🟡 Зволожувач 🟡 Вентиляція
07:07:06 [room1] [manual] relay2 → 🟢 ON
07:07:06 [room2] 🌡️ 19.4°C 💧 51.6% ↓ 1012.4hPa | 🟡 Термостат 🟡 Зволожувач 🟡 Вентиляція
07:07:06 [room2] [manual] relay1 → 🟢 ON
07:07:06 [room3] Запуск симулятора (Firebase)...
07:07:06 [room3] Авто-правила завантажено: {'temp_min': 19.0, 'hum_min': 40.0, 'temp_max': 26.0}
07:07:07 [room3] 🌡️ 23.1°C 💧 44.1% ↓ 1013.8hPa | 🟡 Термостат 🟡 Зволожувач 🟡 Вентиляція

```

Рисунок 3.25 – Відображення в консолі симулятора команд ручного керування

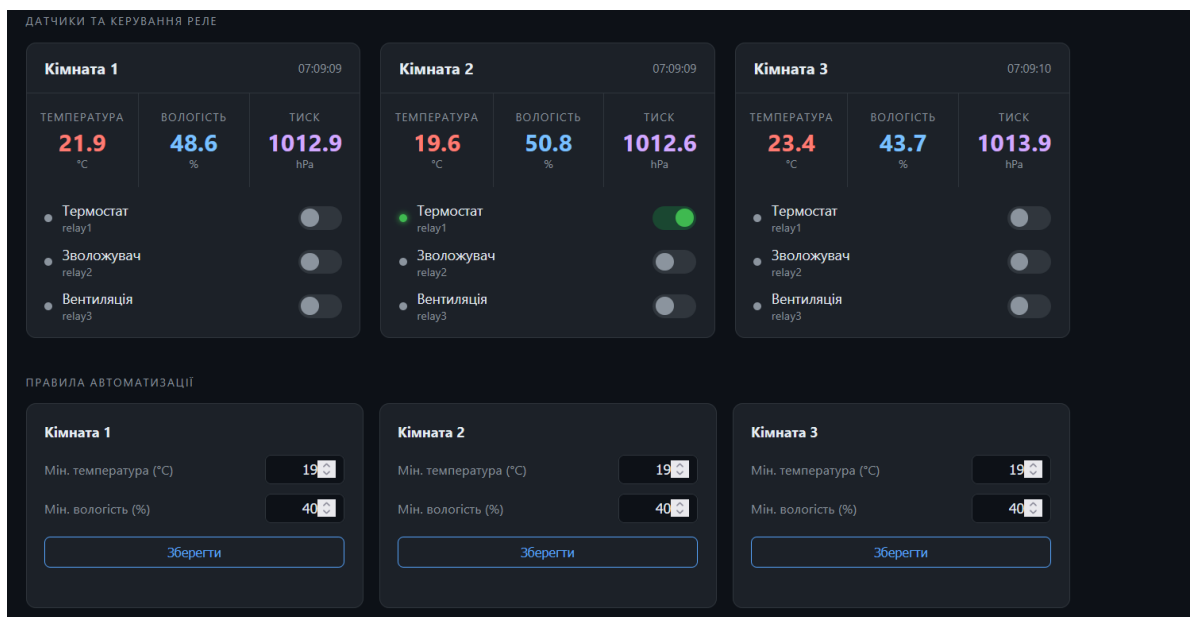


Рисунок 3.26 – Веб-застосунок після ручного вмикання реле Термостат в аудиторії 2

Перевірку автоматичного керування здійснено шляхом спостереження за реакцією симулятора на значення температури, що перетинає встановлений поріг. На рисунку 3.27 зафіксовано момент, коли симулятор зафіксував температуру 19.6°C для room2 – нижче порогу  $temp\_min = 19^\circ\text{C}$ . У відповідь система автоматично встановила  $relay1 = true$ . Водночас для room3 при температурі 23.3°C жодного автоматичного спрацювання не відбулось.

```
07:08:38 [room2] 19.6°C 51.5% ↓ 1012.6hPa | ● Термостат ● Зволожувач ● Вентиляція
07:08:39 [room3] 23.3°C 43.5% ↓ 1013.8hPa | ● Термостат ● Зволожувач ● Вентиляція
07:08:49 [room2] [manual] relay3 → ● ON
```

Рисунок 3.27 – Консоль симулятора (автоматичне вмикання relay1 при температурі нижче порогу)

На рисунку 3.28 зафіксовано зворотну ситуацію, тобто автоматичне вимкнення relay3 для room2 після того, як температура повернулась у допустимий діапазон. Запис  $[auto] relay3 \rightarrow OFF$  підтверджує коректну роботу гістерезису: реле вимикається лише тоді, коли значення перевищило поріг вимкнення, а не просто повернулось до граничного значення.

```
07:08:53 [room1] 21.9°C 48.7% ↓ 1012.9hPa | ● Термостат ● Зволожувач ● Вентиляція
07:08:54 [room2] [auto] relay3 → ○ OFF
```

Рисунок 3.28 – Консоль симулятор (автоматичне вимкнення relay3 при поверненні температури до норми)

Результати тестування підтвердили коректну роботу всіх компонентів системи: хмарна інфраструктура Firebase функціонує відповідно до правил безпеки; Python-симулятор виконує роль трьох незалежних ESP32-клієнтів з паралельною публікацією та зчитуванням даних; веб-застосунок відображає актуальні показники в режимі реального часу та дозволяє здійснювати як ручне, так і автоматичне керування виконавчими пристроями через механізм реле.

Також перевірено роботу одного клієнта ESP32 (тестування параметрів моніторингу та керування мікрокліматом для однієї аудиторії).

### 3.6 Висновки до третього розділу

За результатами виконання завдання по реалізації, розгортанню та тестування програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase було проведено ряд практичних робіт, що охопили всі ключові етапи створення системи.

На етапі реалізації хмарної інфраструктури створено та налаштовано проєкт Firebase Classroom-Microclimate. Спроектовано та реалізовано ієрархічну структуру бази даних із вузлами для телеметрії трьох аудиторій, стану реле та порогових значень авто-правил.

На етапі програмної реалізації клієнтської частини розроблено Python-симулятор, що відтворює роботу трьох незалежних ESP32-вузлів у вигляді паралельних потоків виконання. Кожен потік генерує реалістичні показники мікроклімату, публікує їх до Firebase та виконує локальну перевірку авто-правил. Розроблено односторінковий веб-застосунок із двома HTML-сторінками формою авторизації та дашбордом моніторингу з відображенням даних у режимі реального часу та елементами ручного і автоматичного керування реле. Також реалізовано апаратного клієнта ESP32, що так само як і програмний симулятор виконував публікацію одометричних даних.

На етапі розгортання виконано ініціалізацію локального проєкту засобами Firebase CLI та публікацію веб-застосунку на Firebase Hosting.

На етапі тестування підтверджено коректність запису телеметричних даних до RTDB та відповідність структури вузлів спроектованій схемі, перевірено роботу ручного керування реле через веб-інтерфейс із підтвердженням відображення команд у консолі симулятора, а також верифіковано роботу автоматичного керування.

## ВИСНОВКИ

У дипломній роботі вирішено актуальну задачу розробки програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на основі мікроконтролерної платформи ESP32 та хмарної інфраструктури Firebase, що забезпечує автоматизований моніторинг параметрів повітряного середовища та дистанційне керування виконавчими пристроями через веб-інтерфейс. У ході виконання роботи отримано такі результати.

Проведено аналіз нормативних вимог до мікроклімату навчальних приміщень відповідно до ДСанПіН 5.5.2.008-01 та ДСТУ ISO 7730:2006, що встановлюють допустимі діапазони температури та відносної вологості. Здійснено огляд існуючих рішень комерційних систем автоматизації будівель класу BAS, хмарних IoT-платформ та DIY-рішень на мікроконтролерах. Встановлено, що жодне з існуючих рішень не поєднує одночасно режим реального часу, підтримку кількох аудиторій, двостороннє керування, вбудовану автентифікацію та повністю безкоштовну інфраструктуру. На основі порівняльного аналізу обґрунтовано вибір Firebase як хмарної платформи.

Розроблено трирівневу архітектуру системи, що охоплює апаратний рівень вузлів на базі ESP32, рівень хмарного зберігання на Firebase та рівень веб-інтерфейсу користувача. Обґрунтовано вибір апаратних компонентів, зокрема мікроконтролера ESP32, давача BME280 як єдиного компонента для одночасного вимірювання температури, вологості та атмосферного тиску, чотиріканального релейного модуля для безпечної комутації виконавчих пристроїв. Розроблено монтажну схему апаратного вузла з визначенням усіх електричних з'єднань між ESP32, сенсором BME280, модулем реле, OLED-дисплеєм та автономною системою живлення на базі Li-Ion акумуляторів із BMS та DC-DC перетворювачем.

Спроектовано ієрархічну структуру Firebase Realtime Database з вузлами для телеметрії трьох аудиторій, стану реле та порогових значень автоматичного

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>59 |
| Зм. | Арк. | № докum. | Підпис | Дата |                           |            |

керування. Визначено правила безпеки бази даних, що обмежують читання та запис виключно автентифікованим користувачем відповідно до його UID, що унеможливорює несанкціонований доступ до даних.

Розроблено прошивку мікроконтролера ESP32 мовою C++ у середовищі Arduino IDE з використанням бібліотек FirebaseClient та Adafruit BME280. Також розроблено програмний Python-симулятор, що відтворює роботу трьох незалежних ESP32-вузлів у вигляді паралельних потоків виконання. Симулятор генерує реалістичні показники мікроклімату із застосуванням гаусівського шуму та синусоїдального дрейфу, публікує їх до бази даних та виконує локальну перевірку авто-правил, що дозволило повністю верифікувати хмарну інфраструктуру та логіку керування без фізичного обладнання.

Розроблено та розгорнуто веб-застосунок на Firebase Hosting, доступний за адресою `classroom-microclimate.web.app`. Застосунок реалізує відображення показників мікроклімату трьох аудиторій у режимі реального часу через механізм підписок, ручне перемикання реле через браузерний інтерфейс та налаштування порогових значень автоматичного керування для кожної аудиторії окремо.

Проведено тестування системи, яке підтвердило коректність запису телеметричних даних до RTDB та відповідність структури вузлів спроектованій схемі, функціонування ручного керування реле з підтвердженням відображення команд у консолі симулятора, а також коректність автоматичного керування, тобто вмикання реле при спрацюванні порогових умов та їх вимикання після повернення показників у норму з урахуванням гістерезису.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>60 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Lelieveld J., Evans J. S., Fnais M., Giannadaki D., Pozzer A. The contribution of outdoor air pollution sources to premature mortality on a global scale. *Nature*. 2015. Vol. 525. P. 367–371.
2. Kozłowski R., Szwed M., Kozłowska A., Przybylska J., Mach T. Quality Management System in Air Quality Measurements for Sustainable Development. *Sustainability*. 2024. Vol. 16. P. 7537.
3. Alsamrai O., Redel-Macias M. D., Dorado M. P. Real-Time Intelligent Monitoring of Outdoor Air Quality in an Urban Environment Using IoT and Machine Learning Algorithms. *Appl. Sci*. 2025. Vol. 15. P. 9088.
4. Alrashidi H., Sibai F. N., Abonamah A., Alrashidi M., Alsaber A. PM2.5: Air Quality Index Prediction Using Machine Learning: Evidence from Kuwait's Air Quality Monitoring Stations. *Sustainability*. 2025. Vol. 17. P. 9136.
5. Kortoçi P., Motlagh N. H., Zaidan M. A., Fung P. L., Varjonen S., Rebeiro-Hargrave A., Niemi J. V., Nurmi P., Hussein T., Petäjä T. et al. Air pollution exposure monitoring using portable low-cost air quality sensors. *Smart Health*. 2022. Vol. 23. P. 100241.
6. Banciu C., Florea A., Bogdan R. Monitoring and Predicting Air Quality with IoT Devices. *Processes*. 2024. Vol. 12. P. 1961.
7. Shahid S., Brown D. J., Wright P., Khasawneh A. M., Taylor B., Kaiwartya O. Innovations in Air Quality Monitoring: Sensors, IoT and Future Research. *Sensors*. 2025. Vol. 25. P. 2070.
8. Paithankar D. N., Pabale A. R., Kolhe R. V., William P., Yawalkar P. M. Framework for implementing air quality monitoring system using LPWA-based IoT technique. *Meas. Sens*. 2023. Vol. 26. P. 100709.
9. Budde M., Schwarz A. D., Müller T., Laquai B., Streibl N., Schindler G., Köpke M., Riedel T., Dittler A., Beigl M. Potential and limitations of the low-cost SDS011 particle sensor for monitoring urban air quality. *ProScience*. 2018. Vol. 5. P. 12.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>61 |
| Зм. | Арк. | № докum. | Підпис | Дата |                           |            |

10. Božilov A., Tasić V., Živković N., Lazović I., Blagojević M., Mišić N., Topalović D. Performance assessment of NOVA SDS011 low-cost PM sensor in various microenvironments. *Environ. Monit. Assess.* 2022. Vol. 194. P. 595.

11. Wardoyo A. Y. P., Dharmawan H. A., Nurhuda M., Adi E. T. P. Optimization of PM2.5 Measurement System Using NOVA SDS011 Sensor. *J. Phys. Conf. Ser.* 2020. Vol. 1428. P. 012053.

12. Easterline L. M., Putri A. A. Z. R., Atmaja P. S., Dewi A. L., Prasetyo A. Smart Air Monitoring with IoT-based MQ-2, MQ-7, MQ-8, and MQ-135 Sensors using NodeMCU ESP32. *Procedia Comput. Sci.* 2024. Vol. 245. P. 815–824.

13. Sai K. B. K., Subbareddy S. R., Luhach A. K. IOT based air quality monitoring system using MQ135 and MQ7 with machine learning analysis. *Scalable Comput. Pract. Exp.* 2019. Vol. 20. P. 599–606.

14. Sugianto M. A., Zulfikar Z. Design of a Carbon Monoxide Gas Measurement Tool Using Microcontroller-Based MQ-7 Gas Sensor in Households. *NEWTON Netw. Inf. Technol.* 2024. Vol. 3. P. 36–47.

15. Kobbekaduwa N., Oruthota P., De Mel W. R. Calibration and implementation of heat cycle requirement of MQ-7 semiconductor sensor for detection of carbon monoxide concentrations. *Adv. Technol.* 2021. Vol. 1. P. 377–392.

16. Petrică S. M., Făgărășan I., Arghira N., Stamatescu I., Neculoiu G., Flangea O. Energy efficient IoT air quality monitoring system. *Proceedings of the 2023 24th International Conference on Control Systems and Computer Science (CSCS)*, Bucharest, Romania, 24–26 May 2023. P. 508–513.

17. Liu S., Xia C., Zhao Z. A low-power real-time air quality monitoring system using LPWAN based on LoRa. \*Proceedings of the 2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)\*, Hangzhou, China, 25–28 October 2016. P. 379–381.

18. Rodríguez-Huerta R., Martínez-Castillo J., Morales-González E., Herrera-May A. L. Development of a Monitoring System for CO/CO<sub>2</sub> with Android.

*Proceedings of the 2019 IEEE International Conference on Engineering Veracruz (ICEV)*, Veracruz, Mexico, 14–17 October 2019. P. 1–6.

19. Jadhav R. J., Radhakrishnan P., Jadhav D. A., Ashreetha B., Divya J., Mukherjee S. Internet of Things Enabled Gas Leakage Detection Over Industrial Areas using Powerful MQ Series Sensor and Controller. *Proceedings of the 2024 International Conference on Inventive Computation Technologies (ICICT)*, Kathmandu, Nepal, 24–26 April 2024. P. 1679–1686.

20. Weber K., Kramer T., Fischer C., Pohl T. Testing and Inter-comparison of Low-Cost Sensors (SDS 011) and Mid-Cost Sensors (AN2 and AN3) for Particulate Matter. \*Mid-Term and 1st International Networking Workshop of the SmartAQnet Project\*. Munich: Karlsruhe Institute of Technology, 2018. P. 17.

21. Snyder M. Closed System Evaluation of VerdeTerra’s Indoor Air Purification System in Relation to CO2 Scrubbing. Atlanta: VerdeTerra, 2024.

22. Посібник з мікроконтролерів ESP32. *Ariat Tech*. URL: <https://ua.ariat-tech.com/blog/ESP32-Microcontroller-Guide.html> (дата звернення: 14.05.2026).

23. ESP32 Marauder: повний гід по всіх інструментах для Wi-Fi та Bluetooth тестування. *HackYourMom*. URL: <https://hackyourmom.com/osvita/esp32-marauder-povnyj-gid-po-vsih-instrumentah-dlya-wi-fi-ta-bluetooth-testuvannya/> (дата звернення: 14.05.2026).

24. ESP32. Охорона. URL: <https://охорона.com/esp32/> (дата звернення: 14.05.2026).

25. Як інтернет речей (IoT) трансформує українську промисловість. *UGIC*. URL: <https://ugic.com.ua/yak-internet-rechej-iot-transformuye-ukrayinsku-promyslovist/> (дата звернення: 14.05.2026).

26. Що таке IoT (інтернет речей) простими словами? *Atiko*. URL: <https://www.atiko.com.ua/articles-ua/chto-takoe-iot-prostymi-slovami/> (дата звернення: 14.05.2026).

|     |      |           |        |      |                           |            |
|-----|------|-----------|--------|------|---------------------------|------------|
|     |      |           |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>63 |
| Зм. | Арк. | № док.ум. | Підпис | Дата |                           |            |

27. Розкриваємо поняття - "Інтернет речей". *Infotel*. URL: <https://infotel.ua/news/rozkrivaemo-ponyattya---internet-rechej> (дата звернення: 14.05.2026).

28. Барометр/термометр/датчик вологості на чіпі BME280. *Mini-Tech*. URL: <https://www.mini-tech.com.ua/ua/barometr-termometr-gigrometr-bme280> (дата звернення: 14.05.2026).

29. Все, що вам потрібно знати про ємність акумуляторів 18650. *Large Battery*. URL: <https://www.large-battery.com/uk/blog/18650-battery-capacity-guide/> (дата звернення: 14.05.2026).

30. OLED I2C/ІІС LCD 128x64 0.96" графічний дисплей. *Mini-Tech*. URL: <https://www.mini-tech.com.ua/ua/oled-i2c-0-96-128x64-graficheskij-displej> (дата звернення: 14.05.2026).

31. Понижуючий регульований перетворювач на LM2596 зі світлодіодним вольтметром. *Arduino.ua*. URL: <https://arduino.ua/prod1275-ponijaushhii-regyliryemii-preobrazovatel-na-lm2596-so-svetodiodnim-voltmetrom> (дата звернення: 14.05.2026).

32. Postolache O. A., et al. Smart sensors network for air quality monitoring applications. *IEEE Trans. Instrum. Meas.* 2009. Vol. 58. P. 3253–3262.

33. Leung D. Y. C. Outdoor-indoor air pollution in urban environment: Challenges and opportunity. *Front. Environ. Sci.* 2015. Vol. 2. P. 69.

34. Kang J., Hwang K.-I. A comprehensive real-time indoor air-quality level indicator. *Sustainability*. 2016. Vol. 8. P. 881.

35. Bartolomeo J. Detecting CO in the home. *Home Autom. Build. Cont.* 1995. Vol. 51. P. 55.

36. Pandey Y., et al. Tabrez An Environment Friendly Energy-Saving Dispatch Using Mixed Integer Linear Programming Relaxation in the Smart Grid with Renewable Energy Sources. *Distrib. Gener. Altern. Energy J.* 2022. Vol. 37.

37. Kuo H.-W., Shen H.-Y. Environment. Indoor and outdoor PM<sub>2.5</sub> and PM<sub>10</sub> concentrations in the air during a dust storm. *Build. Environ.* 2010. Vol. 45. P. 610–614.

38. Zanella A., Bui N., Castellani A., Vangelista L., Zorzi M. Internet of things for smart cities. *IEEE Internet Things J.* 2014. Vol. 1. P. 22–32.

39. Meadow J. F., et al. Indoor airborne bacterial communities are influenced by ventilation, occupancy, and outdoor air source. *Indoor Air.* 2014. Vol. 24. P. 41–48.

40. Tran T. V., et al. Battery-free smart-sensor system for real-time indoor air quality monitoring. *Sens. Actuators B Chem.* 2017. Vol. 248. P. 930–939.

41. Apte J. S., et al. PM<sub>2.5</sub> reduces global and regional life expectancy. *Environ. Sci. Technol. Lett.* 2018. Vol. 5. P. 546–551.

42. Lee C.-L., Lee J.-S. Development of indoor air quality supervision systems using zigbee wireless networks. *Proceedings of the 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Wuhan, China, 31 May – 2 June 2018. P. 95–98.

43. Wang E., Xiao L., Han X., Tan B., Luo L. Design of an Agile Training System Based on Wireless Mesh Network. *IEEE Access.* 2022. Vol. 10. P. 84302–84316.

44. Glória A. F., Sebastião P. J. Autonomous configuration of communication systems for IoT smart nodes supported by machine learning. *IEEE Access.* 2021. Vol. 9. P. 75021–75034.

45. Abraham S., Li X. A cost-effective wireless sensor network system for indoor air quality monitoring applications. *Procedia Comput. Sci.* 2014. Vol. 34. P. 165–171.

46. Bamodu O., Xia L., Tang L. J. An indoor environment monitoring system using low-cost sensor network. *Energy Procedia.* 2017. Vol. 141. P. 660–666.

47. Alhmiedat T., Samara G. A low cost ZigBee sensor network architecture for indoor air quality monitoring. *Netw. Internet Archit.* 2017. Vol. 15. P. 140–144.

48. Pantazis N. A., Vergados D. J., Vergados D. D., Douligieris C. J. A. H. N. Energy efficiency in wireless sensor networks using sleep mode TDMA scheduling. *Ad Hoc Netw.* 2009. Vol. 7. P. 322–343.

49. Fung C.-C., Yang P., Zhu Y. Infiltration of Diesel Exhaust into a Mechanically Ventilated Building. *Indoor Air 2014—13th International Conference on Indoor Air Quality and Climate*. 1 January 2014. P. 556–558.

50. Kumar A., Singh I., Sud S. Indoor environment gas monitoring system based on the digital signal processor. *Proceedings of the 2009 International Multimedia, Signal Processing and Communication Technologies*, Aligarh, India, 14–16 March 2009. P. 245–249.

51. Ahmed M. M., Banu S., Paul B. Real-time air quality monitoring system for Bangladesh's perspective based on Internet of Things. *Proceedings of the 2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, Khulna, Bangladesh, 7–9 December 2017. P. 1–5.

52. Firdhous M., Sudantha B., Karunaratne P. IoT enabled proactive indoor air quality monitoring system for sustainable health management. *Proceedings of the 2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*, Chennai, India, 23–24 February 2017. P. 216–221.

53. Dorcea D., Hnatiuc M., Lazar I. Acquisition and calibration interface for gas sensors. *Proceedings of the 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, Iasi, Romania, 25–28 October 2018. P. 120–123.

54. Cirkuit Designer. *Cirkuit Designer*. URL: <https://app.cirkitdesigner.com> (дата звернення: 14.05.2026).

55. Small A. Introducing Cirkuit Designer: an AI-powered IDE for building Arduino and circuit projects. *Medium*. URL: <https://medium.com/@cirkit-designer/introducing-cirkit-designer-an-ai-powered-ide-for-building-arduino-and-circuit-projects-3fafb519b9bd> (дата звернення: 14.05.2026).

56. Cirkuit Designer a Tool for Designing and Diagramming Circuit Projects. *Instructables*. URL: <https://www.instructables.com/Cirkit-Designer-a-Tutorial/> (дата звернення: 14.05.2026).

57. Розбір умовних позначень та основних принципів: як читати електричні схеми? *Powercom*. URL: <https://powercom.ua/dbzh-pcm-yak-chytaty-elektroskhemu> (дата звернення: 14.05.2026).

58. Мелешко В. А., Швець А. О., Краснополський Н. О., Касперович О. О. Основи проектування електронних систем: лабораторний практикум, К.: НАУ, 2014. 102 с.

59. Firebase. *Firestore*. URL: <https://firebase.google.com> (дата звернення: 14.05.2026).

60. What is Firebase? The complete story, abridged. *Medium*. URL: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0> (дата звернення: 14.05.2026).

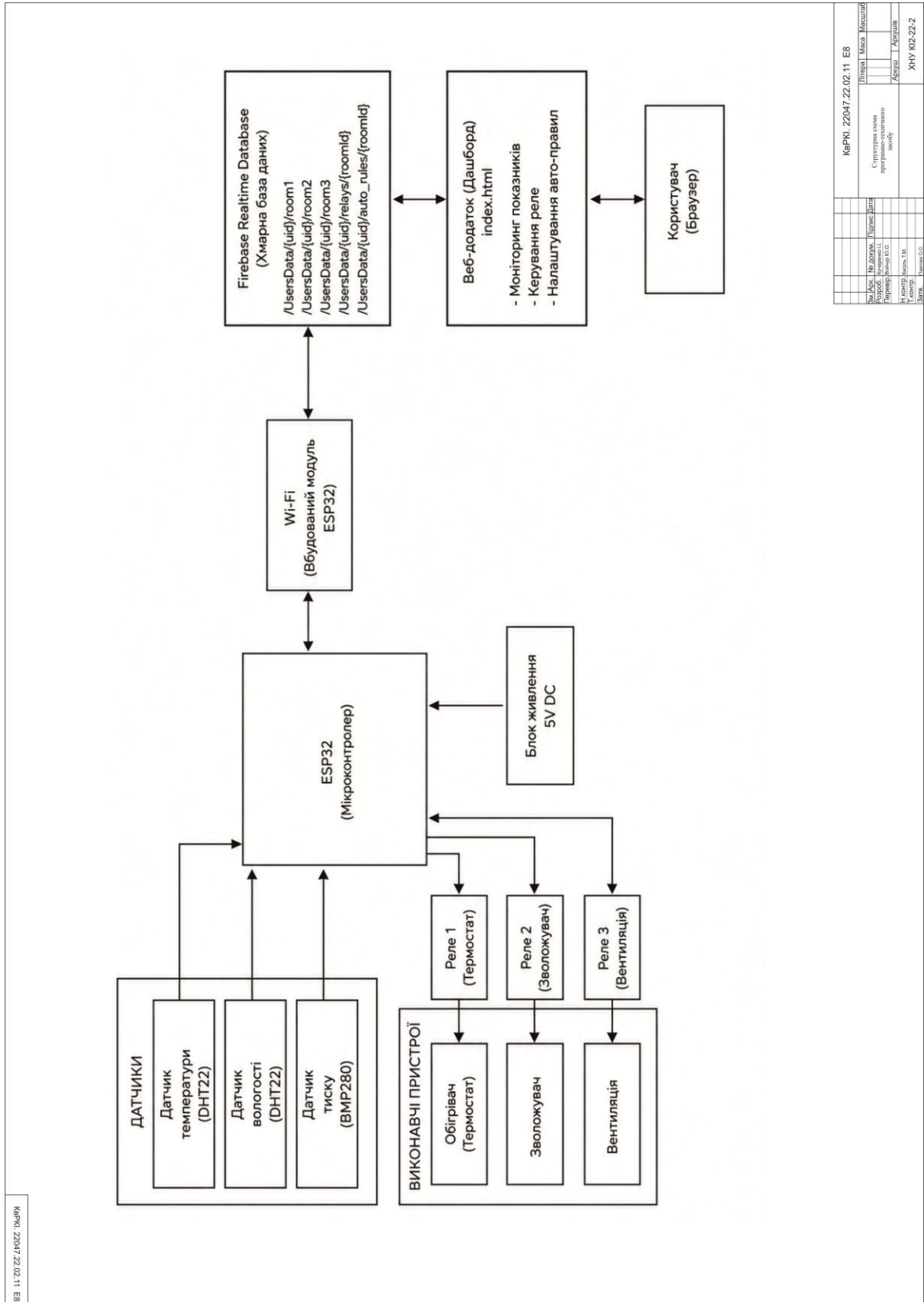
61. ДСанПіН 5.5.2.008-01 Державні санітарні правила і норми влаштування, утримання загальноосвітніх навчальних закладів та організації навчально-виховного процесу. URL: [https://online.budstandart.com/ua/catalog/doc-page.html?id\\_doc=1524](https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=1524) (дата звернення: 14.05.2026).

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 022047.22.02.03 ПЗ | Арк.<br>67 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

# ДОДАТОК А

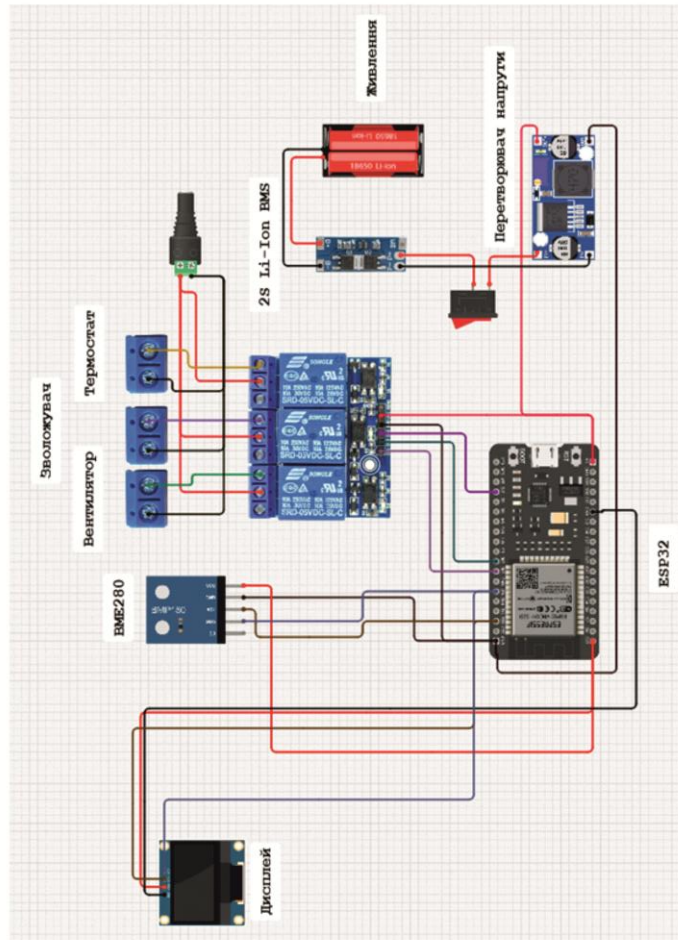
(обов'язковий)

Копія креслення «Структура програмно-технічного засобу»



# ДОДАТОК Б (обов'язковий)

## Копія креслення «Схема монтажна»

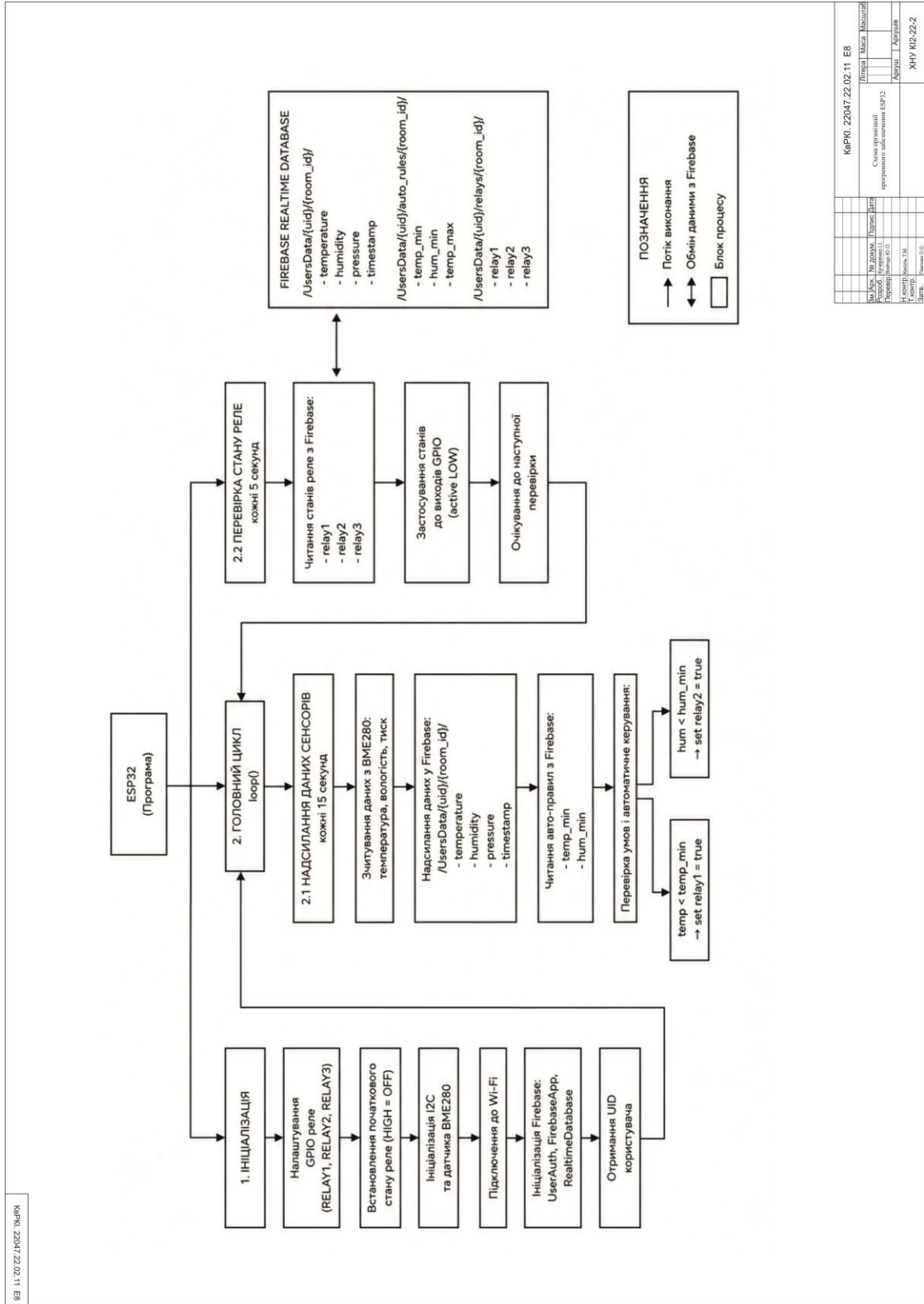


КВРК. 22047.22.02.11 Е4

|                         |          |           |              |        |
|-------------------------|----------|-----------|--------------|--------|
| КВРК. 22047.22.02.11 Е4 |          | Листопад  | Місяць       | Підпис |
| Від: Акт.               | На: Акт. | Від: Акт. | Підпис: Акт. |        |
| Схема монтажна          |          | Акт.      | Акт.         |        |
| ХНУ КП-22-2             |          | Акт.      | Акт.         |        |

# ДОДАТОК В (обов'язковий)

Копія креслення «Схема організації програмного забезпечення ESP32»



Зав. кафедри КІС  
д-р. філософії Ользі ПАВЛОВІЙ

Ілля КУЧЕРЕНКО

---

ІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-2

### ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року

Tue May 26 08:57:43 EEST 2026, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

## Anti-Plagiarism (<http://ap.km.ua>) v-15.701

**Максимальне співпадіння з одним документом 1.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. **Помилки в документах: 15%**

|  |          |         |                             |         |
|--|----------|---------|-----------------------------|---------|
| ID: 272268<br>Назва: БКР Програмно-технічний засіб контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase<br>Додано в БД: 2026-05-26<br>Автора: Ілля КУЧЕРЕНКО<br>Керівники: Юрій ВОЙЧУР<br>Консультанти:<br>Опоненти: | Документ |         | Сумарний збіг по Базі Даних |         |
|  | Символи  | Лексеми | Символи                     | Лексеми |
|  | 82180    | 644     | 2064 (3%)                   | 24 (4%) |

### Джерело плагіату

| ID | Опис | Наявність плагіату в документі |         |
|----|------|--------------------------------|---------|
|    |      | Символи                        | Лексеми |
|    |      |                                |         |

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Ілля КУЧЕРЕНКО

**Співавтор:**

**Назва:** Програмно-технічний засіб контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase

**Експерт:** Юрій ВОЙЧУР

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 6.96%

**Коефіцієнт подібності 2:** 3.09%

**Мікропробіли:** 3

**Заміна букв:** 1

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2026-05-23 16:22:48.0

**Після аналізу Звіту подібності констатую наступне:**

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

**Обґрунтування:**

2026-05-23

Дата



Доцент Андрій Нічепорук

експерт

## РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Ілля КУЧЕРЕНКО

Тема: Програмно-технічний засіб контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   60  

1. Короткий зміст роботи та прийнятих рішень: Метою дипломної роботи є проектування та реалізація програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на платформі ESP32 з використанням хмарної бази даних Firebase Realtime Database, що забезпечує автоматизований моніторинг температури, вологості та атмосферного тиску у трьох аудиторіях одночасно, дистанційне керування виконавчими пристроями через веб-інтерфейс та автоматичне регулювання мікроклімату на основі налаштовуваних порогових правил.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено дослідження предметної області та огляд відомих рішень у сфері контролю та керування мікрокліматом навчальних приміщень, а саме: розглянуто значення мікроклімату в навчальних приміщеннях; проаналізовано нормативні вимоги до мікроклімату; виконано огляд існуючих систем моніторингу та керування мікрокліматом; досліджено концепцію Інтернету речей; проведено огляд хмарних платформ для зберігання даних IoT-систем.

У другому розділі кваліфікаційної роботи виконано проектування програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Google Firebase, а саме: визначено вимоги до програмно-технічного

засобу; розроблено структуру системи; створено монтажну схему; проведено вибір та аналіз апаратних компонентів; спроектовано структуру бази даних; розроблено алгоритм автоматичного керування мікрокліматом; сформовано висновки до другого розділу.

У третьому розділі кваліфікаційної роботи виконано реалізацію, розгортання та тестування програмно-технічного засобу контролю та керування мікрокліматом навчальних аудиторій, а саме: розроблено узагальнений підхід до реалізації та тестування системи; реалізовано хмарну інфраструктуру на основі Firebase; створено програмний симулятор клієнтів на базі ESP32; виконано програмну реалізацію ESP32-клієнтів; здійснено розгортання веб-застосунку на Firebase Hosting; проведено тестування системи та аналіз результатів її функціонування.

5. Негативні сторони роботи: До недоліків роботи можна віднести обмежену масштабованість проведених експериментів та недостатній обсяг даних для повноцінної оцінки роботи системи в різних умовах експлуатації.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на достатньому технічному рівні.

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: добре С (75)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

Басіголь Олександр Анатолійович  
доцент кафедри КН, ХНУ

“28” 05 2026 р.

 (підпис)

## РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

### КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-технічний засіб контролю та керування мікрокліматом навчальних аудиторій на основі ESP32 та Firebase

Автор Ілля КУЧЕРЕНКО

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: ДФ, Юрій ВОЙЧУР

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

| №   | Висновок  | Позначка про відповідність |
|-----|---|----------------------------|
| 1   | Ознаки академічного плагіату  |                            |
| 1.1 | Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.   | відповідає                 |
| 1.2 | Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.   |                            |
| 1.3 | Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат. |                            |
| 1.4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.  |                            |
| 2   | Інші види порушень академічної доброчесності  |                            |

#### Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.


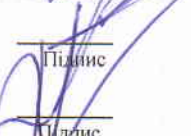

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 6,96% і адресується до 37 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

27.05.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

  
Підпис  
  
Підпис  
  
Підпис

Ольга ПАВЛОВА  
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК  
Ім'я, ПРІЗВИЩЕ

Юрій ВОЙЧУР  
Ім'я, ПРІЗВИЩЕ