Міністерство освіти і науки України

Ніжинський державний університет імені Миколи Гоголя

Навчально-науковий інститут природничо-математичних,   
 медико-біологічних наук та інформаційних технологій

Кафедра інформаційних технологій, фізико-математичних та економічних наук

Освітня програма: Комп’ютерні науки  
Спеціальність: 122 Комп’ютерні науки

**КВАЛІФІКАЦІЙНА РОБОТА** на здобуття освітнього ступеня магістр

**Автоматизована система вимірювання та аналізу рівня шкідливих речовин в довкіллі**

 студента **Горошка Геннадія Олександровича**

**Науковий керівник:**

Казачков Іван Васильович

доктор технічних наук, професор

Рецензенти:

Фетісов В.С.,

кандидат економічних наук, доцент,

Нестеренко О.В.,

доктор технічних наук, професор

Допущено до захисту: \_\_\_ \_\_\_\_\_\_\_\_ 2024р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ніжин – 2024

АНОТАЦІЯ

Магістерська робота присвячена розробці автоматизованої системи для вимірювання та аналізу рівня шкідливих речовин у повітрі. У роботі запропоновано використання сучасних мікроконтролерів і сенсорів, зокрема модулів ENS160, BMP280, та AHT20, для збору екологічних даних. Інформація з сенсорів обробляється мікроконтролером ESP8266, передається на сервер для збереження у базі даних MySQL та візуалізується через веб-інтерфейс. Розроблена система надає можливість відображати дані у реальному часі, фільтрувати їх за часовими проміжками та проводити аналіз для оцінки стану довкілля.

Система реалізована з урахуванням адаптивності для різних пристроїв та включає інструменти інтерактивного графічного відображення. Це сприяє своєчасному виявленню змін у якості повітря та підтримує прийняття рішень щодо зменшення забруднення. Результати роботи демонструють можливість застосування розробленої системи в екологічному моніторингу та для підвищення екологічної обізнаності.

**Ключові слова:** автоматизована система, якість повітря, сенсори, ESP8266, моніторинг довкілля.

ABSTRACT

The master's thesis focuses on the development of an automated system for measuring and analyzing the levels of harmful substances in the air. The study proposes the use of modern microcontrollers and sensors, including ENS160, BMP280, and AHT20 modules, for collecting environmental data. The information from the sensors is processed by an ESP8266 microcontroller, transferred to a server for storage in a MySQL database, and visualized through a web interface. The developed system enables real-time data display, filtering by time intervals, and analysis for assessing environmental conditions.

The system is implemented with adaptability for various devices and includes tools for interactive graphical data representation. This approach facilitates the timely detection of air quality changes and supports decision-making to reduce pollution. The findings demonstrate the applicability of the developed system in environmental monitoring and raising ecological awareness.

**Keywords:** automated system, air quality, sensors, ESP8266, environmental monitoring.

ЗМІСТ

[ВСТУП 4](#_Toc184236151)

[1. ТЕОРЕТИЧНІ ОСНОВИ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ 5](#_Toc184236152)

[1.1. Характеристики шкідливих речовин в атмосфері та їх вплив на здоров'я людини та довкілля 5](#_Toc184236153)

[1.2. Вибір сенсорного обладнання для моніторингу якості повітря 7](#_Toc184236154)

[2. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ 9](#_Toc184236155)

[2.1. Основні компоненти системи моніторингу 9](#_Toc184236156)

[2.2. Принципова електрична схема підключення та налаштування сенсорів 16](#_Toc184236157)

[2.3. Налаштування бездротового з'єднання та передача даних 19](#_Toc184236158)

[3. РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ 22](#_Toc184236159)

[3.1. Створення серверного середовища для обробки даних 22](#_Toc184236160)

[3.2. Обробка даних з сенсорів та їх зберігання 25](#_Toc184236161)

[3.3. Створення таблиці для збереження показників сенсорів 26](#_Toc184236162)

[3.4. Розробка веб-сайту для візуалізації та аналізу даних 29](#_Toc184236163)

[4. РЕАЛІЗАЦІЯ ВЕБ-ІНТЕРФЕЙСУ ДЛЯ ВІДОБРАЖЕННЯ ДАНИХ СЕНСОРІВ 33](#_Toc184236164)

[4.1. Реєстрація доменного імені 35](#_Toc184236165)

[4.2. Огляд функціональності сайту 37](#_Toc184236166)

[ВИСНОВКИ 41](#_Toc184236167)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 42](#_Toc184236168)

[ДОДАТОКИ 44](#_Toc184236169)

# ВСТУП

Сучасні технології активно розвиваються в напрямку автоматизації процесів моніторингу довкілля, що відкриває нові можливості для збору, зберігання, обробки та аналізу великих обсягів даних у реальному часі. Одним із актуальних завдань сьогодення є моніторинг якості атмосферного повітря, який має безпосередній вплив на здоров'я людей та екосистеми.

Ключову роль у розробці рішень для збору та обробки таких даних відіграють сучасні мікроконтролери, сенсори та серверні технології. Вони дозволяють реалізувати ефективні системи, які здатні реагувати на зміни в навколишньому середовищі та надавати користувачам актуальну інформацію.

Мета проекту — розробити автоматизовану систему на базі мікроконтролерів та сенсорів, яка забезпечить вимірювання рівнів шкідливих речовин у повітрі в режимі реального часу, передавання даних на сервер для подальшого аналізу та візуалізації. Особлива увага приділяється оптимізації апаратного та програмного забезпечення для стабільної роботи системи в реальних умовах.

Робота охоплює не лише створення технічного рішення, але й його практичну реалізацію, що дозволить отримати важливі дані при моніторингу екологічної ситуації та прийняття обґрунтованих рішень щодо покращення якості повітря.

# ТЕОРЕТИЧНІ ОСНОВИ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ

Автоматизований моніторинг якості повітря за допомогою сучасних інформаційних технологій дозволяє відображати характеристики шкідливих речовин у повітрі в реальному часі, що дає змогу оперативно оцінювати їхній вплив на довкілля та здоров’я. Це забезпечує можливість своєчасного реагування на небезпечні зміни, прийняття відповідних рішень, а також прогнозування потенційних ризиків. Завдяки високій точності вимірювань і великій кількості зібраних даних можна отримати глибокий аналіз екологічної ситуації та ефективніше впроваджувати заходи для зменшення забруднення.

## Характеристики шкідливих речовин в атмосфері та їх вплив на здоров'я людини та довкілля

Основні забруднювачі атмосферного повітря включають леткі органічні сполуки (ЛОС), вуглекислий газ (CO₂), оксиди азоту (NOₓ), діоксид сірки (SO₂), озон (O₃) та тверді частинки (PM2.5 і PM10). Кожен з цих забруднювачів має специфічні властивості, а їх вплив на довкілля та здоров’я людини добре задокументовано.

Леткі органічні сполуки (ЛОС) – це група хімічних речовин, які легко випаровуються в атмосферу. Вони входять до складу вихлопних газів транспорту, промислових викидів, а також містяться у фарбах, розчинниках та побутових засобах. ЛОС взаємодіють з оксидами азоту (NOₓ) і сприяють утворенню фотохімічного смогу. За даними Всесвітньої організації охорони здоров'я (ВООЗ), ЛОС можуть викликати подразнення дихальних шляхів, головний біль та втомлюваність.

Вуглекислий газ (CO₂) – парниковий газ, що утворюється в результаті спалювання викопного палива (нафти, газу, вугілля). Хоча CO₂ не вважається токсичним у низьких концентраціях, його надмірна кількість у повітрі є основним фактором глобального потепління, що змінює кліматичні умови, впливаючи на всі екосистеми.

Оксиди азоту (NOₓ) – включають сполуки NO та NO₂, які переважно походять із вихлопних газів транспортних засобів і промислових викидів. Вдихання NOₓ може спричиняти подразнення легенів, загострення астми, а також сприяти утворенню кислотних дощів. Дані агентства з охорони навколишнього середовища EPA (англ. *Environmental Protection Agency*) свідчать, що зменшення NOₓ сприяє покращенню якості повітря, особливо в міських районах.

Діоксид сірки (SO₂) – це газ, який виділяється при спалюванні сірковмісного вугілля та нафти, особливо на електростанціях та заводах. SO₂ є одним з каталізаторів кислотних дощів, що спричиняють деградацію ґрунтів і водойм. За даними ВООЗ, високі рівні SO₂ у повітрі можуть викликати подразнення очей і дихальних шляхів, особливо в людей з респіраторними захворюваннями.

Озон (O₃) – це вторинний забруднювач, який утворюється в атмосфері в результаті хімічних реакцій між ЛОС і NOₓ під дією сонячного світла. Озон на рівні ґрунту є небезпечним для здоров'я, вдихання може призвести до кашлю, болю в грудях і подразнення дихальних шляхів. Агентства, як-от EPA, попереджають, що високі концентрації озону можуть також негативно впливати на рослинність, знижуючи врожайність.

Тверді частинки (PM2.5 і PM10) – мікроскопічні частинки пилу, попелу, сажі та інших речовин, які можуть проникати в легені. PM2.5 мають розмір до 2.5 мкм, а PM10 – до 10 мкм. Ці частинки здатні проникати в глибокі частини легенів, що підвищує ризик серцево-судинних і респіраторних захворювань. ВООЗ рекомендує підтримувати середньорічні рівні PM2.5 нижче 15 мкг/м³, оскільки перевищення цього порогу асоціюється зі значним підвищенням смертності.

Індекс якості повітря AQI (англ. Air Quality Index) використовується для оцінки рівня забруднення повітря та надання простого числового значення, яке вказує на ступінь ризику для здоров’я. Наприклад, значення AQI від 0 до 50 вказує на «хороший» рівень якості повітря, тоді як значення понад 300 свідчать про небезпечний рівень забруднення. AQI дозволяє швидко оцінювати ризики для різних груп населення і сприяє прийняттю рішень щодо зменшення забруднення.

Аналізуючи рівень шкідливих речовин, автоматизована система може сприяти своєчасному виявленню небезпечних умов і мінімізації їхнього впливу на здоров’я та довкілля.

## Вибір сенсорного обладнання для моніторингу якості повітря

Для автоматизованих систем моніторингу повітря використовуються різноманітні сенсори, що забезпечують точне вимірювання параметрів довкілля.

Сенсори можна виокремити на:

1. Температурні та вологості (наприклад, AHT20), які дозволяють оцінювати основні кліматичні умови.
2. Сенсори атмосферного тиску (наприклад, BMP280), що вимірюють тиск і є важливими для загальної оцінки стану атмосфери.
3. Газові сенсори, зокрема для вимірювання рівня ЛОС і CO₂ (наприклад, ENS160), які дозволяють визначити рівні летких органічних сполук та інших забруднювачів.
4. Сенсори для розрахунку індексу якості повітря (AQI), що допомагає швидко оцінювати екологічні ризики.

Сенсори зазвичай взаємодіють із мікроконтролерами (наприклад, ESP8266), які збирають і передають дані на сервер для їх обробки та збереження. Для забезпечення стабільної роботи та точного збору даних важливо враховувати технічні характеристики сенсорів: точність вимірювання, чутливість, час відгуку та енергоспоживання.

Таким чином, автоматизована система моніторингу, що поєднує в собі кілька типів сенсорів і алгоритми обробки даних, може стати важливим інструментом для підтримки та підвищення якості повітря.

# АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

Вибір апаратного забезпечення є важливим етапом у розробці автоматизованої системи моніторингу якості повітря, оскільки від цього залежить не лише ефективність і надійність роботи системи, але й точність збору даних та їх обробки. Важливо правильно підібрати компоненти, які будуть взаємодіяти між собою, а також забезпечувати стабільну роботу в різних умовах навколишнього середовища. У процесі вибору необхідно враховувати технічні характеристики сенсорів, мікроконтролерів, джерел живлення та інших складових.

Розглянемо основні компоненти, що входять до складу таких систем і визначають їх ефективність, надійність та точність роботи.

## 2.1. Основні компоненти системи моніторингу

Мікроконтролер ESP8266.

ESP8266 — це популярний мікроконтролер з підтримкою Wi-Fi, розроблений китайською компанією Espressif Systems, який зазвичай використовується у проектах, пов’язаних з Інтернетом речей (IoT).

Мікроконтролер ESP8266 не має вбудованої пам’яті для збереження програм, тому зовнішня флеш-пам'ять підключається через SPI-інтерфейс і використовується для виконання користувацьких програм. У 2016 році Espressif випустила оновлену версію — ESP8285, яка поєднує мікроконтролер з флеш-пам'яттю обсягом 1 МБ, що ще більше розширило можливості для розробників. Наступним кроком у розвитку лінійки став випуск ESP32, який має розширені характеристики.

Мікроконтролер ESP8266 базується на 32-бітовому процесорі Tensilica Xtensa L106 з частотою 80 МГц, який можна розігнати до 160 МГц. Підтримує стандарт Wi-Fi IEEE 802.11 b/g/n, забезпечуючи підключення до локальної мережі та Інтернету. Для безпеки передбачена підтримка протоколів WEP і WPA/WPA2. Працює ESP8266 при напрузі живлення від 2,2 до 3,6 В, а споживання електроенергії варіюється в залежності від режиму роботи, 215 мА при передачі даних, 100 мА при прийомі та 70 мА в режимі очікування. Є три режими енергозбереження: Modem sleep (15 мА), Light sleep (0,4 мА) і Deep sleep (15 мкА), які допомагають зменшити енергоспоживання, що є важливим для автономних пристроїв.

Мікроконтролер ESP8266 знайшов широке застосування у IoT-пристроях, які можуть підключатися до домашньої або офісної мережі через Wi-Fi. Він здатен функціонувати як точка доступу або як клієнт мережі. У режимі точки доступу користувач може підключитися до мікроконтролера зі смартфона чи комп'ютера і налаштувати параметри мережі через веб-інтерфейс. У клієнтському режимі ESP8266 підключається до існуючої мережі Wi-Fi, дозволяючи віддалено контролювати пристрій через Інтернет. Цей мікроконтролер активно використовується для побудови систем керування побутовими приладами, моніторингу параметрів середовища та автоматизації. Серед популярних IoT-протоколів - HTTP для веб-контролю та MQTT для передачі повідомлень між пристроями.

На ринку існує кілька аналогів ESP8266, випущених різними виробниками. Наприклад, MediaTek пропонує мікроконтролери MT7681 і MT7687 з ядром Cortex M4, а Texas Instruments - CC3200 з аналогічним ядром. Компанія Realtek також представила серію мікроконтролерів RTL8195 і RTL8710 з розширеним набором можливостей. ESP8266, однак, залишається одним із найпопулярніших рішень завдяки підтримці широкого набору бібліотек, зручності налаштування та потужній спільноті розробників, які активно підтримують проекти, що базуються на цьому мікроконтролері.

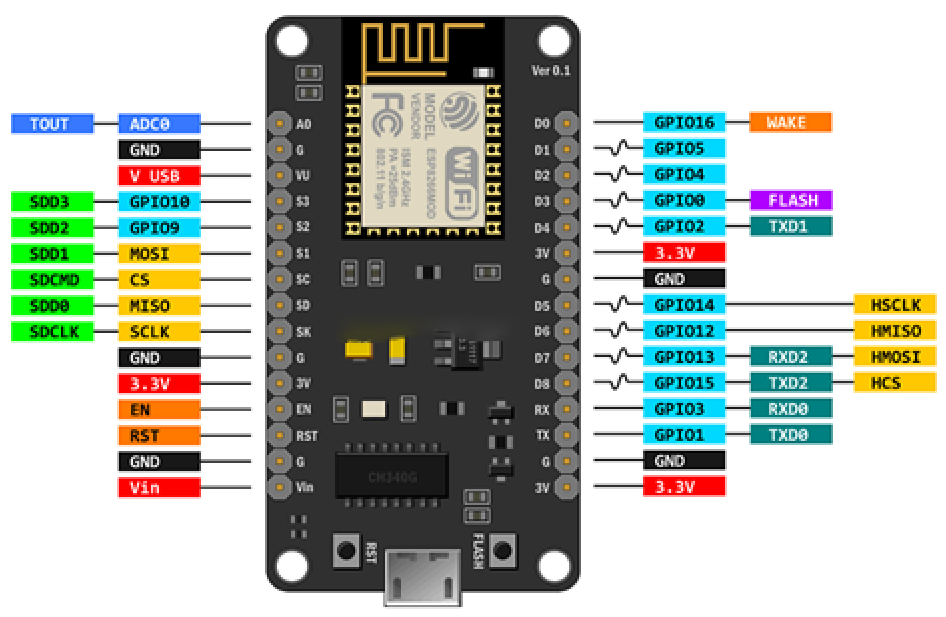


Рисунок 1.1 - мікроконтролер ESP8266

Модуль NodeMCU V3 ESP8266 (Рис.1.1) має кілька основних виводів, які забезпечують гнучкість і зручність підключення різноманітних периферійних пристроїв. Для коректної роботи модуля важливо розуміти функції кожного виводу.

Вивід GND (земля) служить загальним проводом для з’єднання з іншими компонентами, що є стандартним для всіх електронних схем. Контакти живлення включають Vin, 3.3V і VUSB. Контакт Vin призначений для підключення зовнішнього джерела живлення (5V), а вбудований стабілізатор AMS1117-3.3 забезпечує роботу в широкому діапазоні напруг (5-10 V). Проте через можливість перегріву стабілізатора не слід подавати напругу понад 10. Контакт 3.3V призначений для живлення периферійних пристроїв, але максимальне сумарне навантаження не має перевищувати 300 мА. VUSB під’єднаний до USB-роз’єму, забезпечуючи напругу 5V під час підключення до комп’ютера або іншого джерела живлення через USB.

Контакти GPIO (General Purpose Input/Output) дозволяють налаштовувати їх як входи або виходи та задавати програмні функції для керування пристроями або отримання сигналів з датчиків. Контакти управління, такі як RST (Reset), використовуються для скидання мікроконтролера до початкового стану, а EN (Chip Enable) визначає режим роботи пристрою: сигнал високого рівня активує мікроконтролер, а низького — переводить його у режим енергозбереження. Контакт WAKE застосовується для виходу з режиму глибокого сну (deep-sleep mode), що важливо для енергозберігаючих додатків.

Вивід ADC0/TOUT надає доступ до 10-бітного аналого-цифрового перетворювача (АЦП) модуля, що дозволяє вимірювати аналогові сигнали в діапазоні від 0 до 3.3 V завдяки внутрішньому дільнику напруги. Для підключення до інших пристроїв використовується UART — асинхронний послідовний інтерфейс, що підтримує швидкість передачі даних до 4,5 Мбіт/с. Крім того, NodeMCU V3 має SPI для підключення периферійних пристроїв, SDIO для роботи з SD-картами та зовнішньою пам'яттю, а також можливість роботи з інтерфейсом I2C для підключення сенсорів, хоч його потрібно реалізувати програмно через GPIO-контакти (зазвичай GPIO5 для SCL і GPIO4 для SDA).

Модуль також підтримує PWM (широтно-імпульсну модуляцію), яка дозволяє керувати потужністю підключених пристроїв, таких як світлодіоди або двигуни. На платі передбачено вбудований світлодіод, підключений до GPIO2, який може бути використаний для індикації стану модуля або як звичайний вихід для сигнальних цілей. Кнопка Flash дозволяє підключити GPIO0 до землі, що важливо для режиму перепрошивки модуля, а також може використовуватись як стандартна кнопка управління.



Рисунок 1.2 - Wi-Fi модуль NodeMCU V3 ESP8266

На рис. 1.2 зображено NodeMCU V3 ESP8266 - універсальний модуль для проектів IoT, який поєднує широкий набір функцій з можливістю програмного налаштування.

Барометр BMP280 (датчик атмосферного тиску).

Модуль для вимірювання атмосферного тиску, заснований на датчику BMP280 від компанії BOSCH (Рис. 1.3), є сучасним і вдосконаленим рішенням для точного моніторингу тиску та температури. Цей датчик, що є оновленою версією популярного BMP180, відрізняється компактнішими розмірами, зниженим енергоспоживанням та високою точністю вимірювань. Також BMP280 має заводське калібрування і підтримує два послідовні інтерфейси для з’єднання з іншими пристроями: I2C і SPI, що забезпечує гнучкість при підключенні до різноманітних контролерів.

Датчик BMP280 підтримує три режими роботи, що дозволяють адаптувати його функціонування до різних завдань. SLEEP - це режим зниженого енергоспоживання, в якому датчик працює з мінімальним використанням ресурсів. FORCED - аналогічний режиму роботи попередніх моделей (BMP085 та BMP180), де датчик активується командою з контролера, проводить вимірювання, передає результати і повертається у режим енергозбереження. Унікальним для цього датчика є NORMAL-режим, в якому BMP280 автоматично прокидається, проводить вимірювання тиску і температури та знову засинає. Тимчасові параметри цього режиму можна налаштовувати, що дозволяє зчитувати дані в будь-який момент без додаткових команд.

Для підвищення точності результатів BMP280 використовує фільтрацію, яка забезпечує можливість налаштування параметрів. Наприклад, можна задати OVERSAMPLING для температури і тиску з роздільною здатністю 16-20 біт, що підвищує точність вимірювань. Крім того, TSB (час між вимірами) може бути налаштований з затримкою від 0,5 до 4000 мс для забезпечення оптимальної частоти відліків, а FILTER\_COEFFICIENT визначає рівень фільтрації, що допомагає зменшити вплив короткочасних змін.

Серед технічних характеристик датчика — діапазон робочої напруги від 1,71 В до 3,6 В, що робить його сумісним з більшістю мікроконтролерів. Він підтримує швидкість передачі даних до 3,4 МГц для інтерфейсу I2C і працює з низьким струмом (2,7 мкА при частоті відліків 1 Гц), що є важливим для автономних пристроїв. Діапазон вимірювання тиску варіюється від 300 до 1100 гПа дозволяє використовувати датчик на висотах від 9000 м над рівнем моря до -500 м. Завдяки низькому рівню шуму (до 0,2 Па для тиску і 0,01°C для температури), датчик забезпечує високу точність та стабільність вимірювань, що робить його ідеальним для точного моніторингу метеорологічних та кліматичних умов.

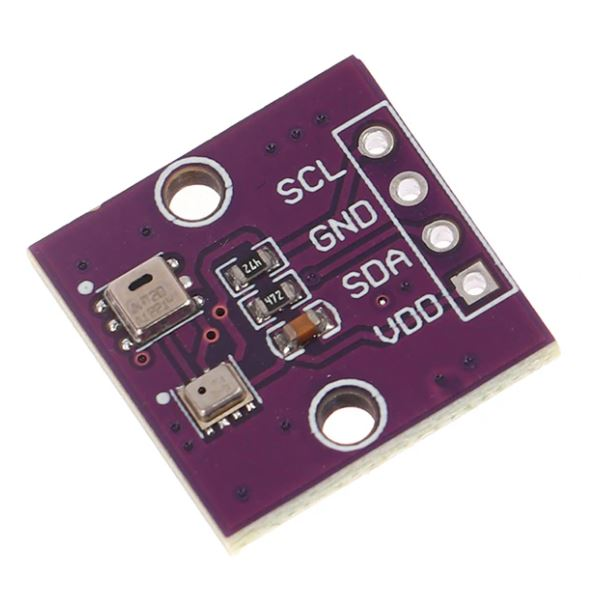


Рисунок 1.3 Модуль прецизійних датчика тиску BMP280

Модуль датчик якості повітря, температури та вологості.

Модуль для вимірювання якості повітря, температури та вологості на основі датчика ENS160 (Рис.1.4) є потужним інструментом для моніторингу параметрів повітряного середовища. ENS160 — це мультигазовий датчик, що здатен виявляти широкий спектр газів та летких органічних сполук (ЛОС), таких як толуол, водень, етанол, діоксид азоту (NO₂) та озон, що робить його універсальним рішенням для моніторингу повітря. Датчик також використовує інтегрований алгоритм попереднього калібрування та функцію автоматичної корекції базової лінії, завдяки чому він автоматично адаптується до змін умов навколишнього середовища і зберігає точність вимірювань.

Модуль ENS160 підтримує інтерфейси I2C і SPI, що забезпечує зручність інтеграції з різноманітними мікроконтролерами та іншими електронними пристроями. На виході модуль може надавати кілька важливих показників, що дозволяють оцінити якість повітря. Серед них — TVOC (загальні леткі органічні сполуки), eCO₂ (еквівалентний рівень вуглекислого газу), а також AQL (індекс якості повітря, який узагальнює параметри забрудненості повітря). Ці показники корисні для виявлення загальних умов забруднення повітря, що може бути застосовано як у побутових, так і в промислових умовах.

Модуль ENS160 працює в широкому діапазоні температур і вологості, забезпечуючи точність вимірювань у температурному діапазоні від -40°C до +85°C і відносній вологості від 5% до 95%. Це дозволяє використовувати датчик у різних кліматичних умовах. Модуль живиться від напруги 3,3 В постійного струму, що робить його сумісним з багатьма сучасними мікроконтролерами, такими як ESP8266. Габарити модуля складають 20x25,3 мм, що дозволяє легко інтегрувати його в компактні системи для автоматизованого контролю якості повітря.

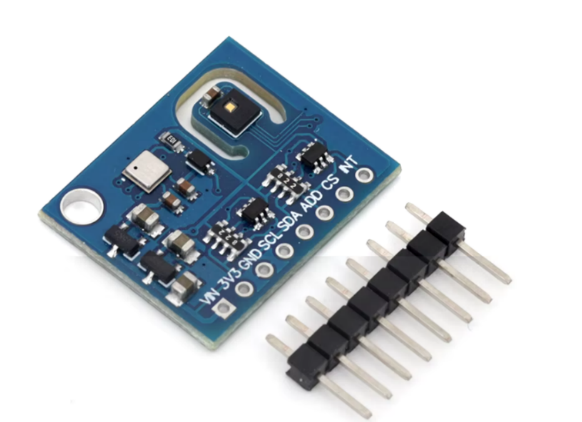


Рисунок 1.4 - Модуль датчик якості повітря, температури та вологості ENS160+AHT21

## 2.2. Принципова електрична схема підключення та налаштування сенсорів

Принципова електрична схема описує з’єднання компонентів системи моніторингу якості повітря. Основою конструкції є мікроконтролер ESP8266 (Рис.2.1), який забезпечує збирання та передачу даних з підключених сенсорів до віддаленого сервера через Wi-Fi. До ESP8266 підключено кілька сенсорів для збору екологічних показників - датчик температури та вологості AHT20, датчик атмосферного тиску BMP280 та сенсор якості повітря ENS160 для вимірювання летких органічних сполук (TVOC) та еквіваленту CO2 (eCO2). Кожен сенсор живиться від ESP8266 через стабілізатор напруги, який забезпечує стабільність роботи приладів та захищає їх від коливань напруги. Дані передаються по шині I2C, що забезпечує зручність у підключенні та дозволяє зберегти стабільність з'єднання між мікроконтролером та сенсорами.

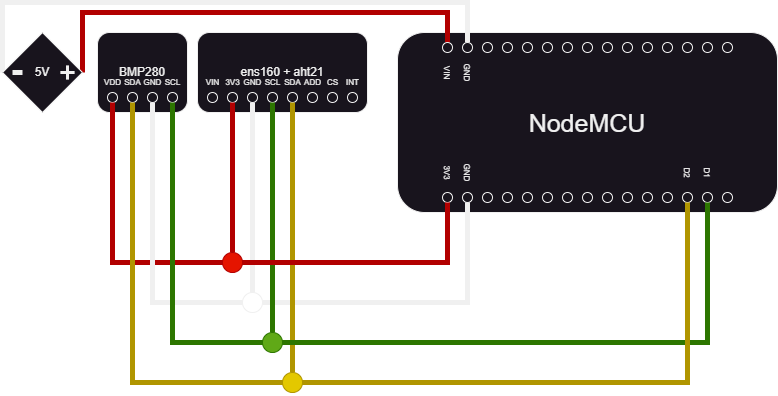


Рисунок 2.1 - електрична схема підключення модулів

Сенсорна система зібрана на макетній платі (Рис.2.2), що забезпечує компактність та зручність у налаштуванні і тестуванні. На платі розташовані компоненти системи - ESP8266, AHT20, BMP280 та ENS160, підключені відповідно до принципової електричної схеми. Така збірка на макетній платі дозволяє швидко протестувати систему, відстежуючи її роботу та проводити корекцію при потребі. Всі з'єднання виконані за допомогою з’єднувальних проводів на макетній платі, що надає можливість оперативно змінювати конфігурацію, зокрема для тестування нових сенсорів або модулів.

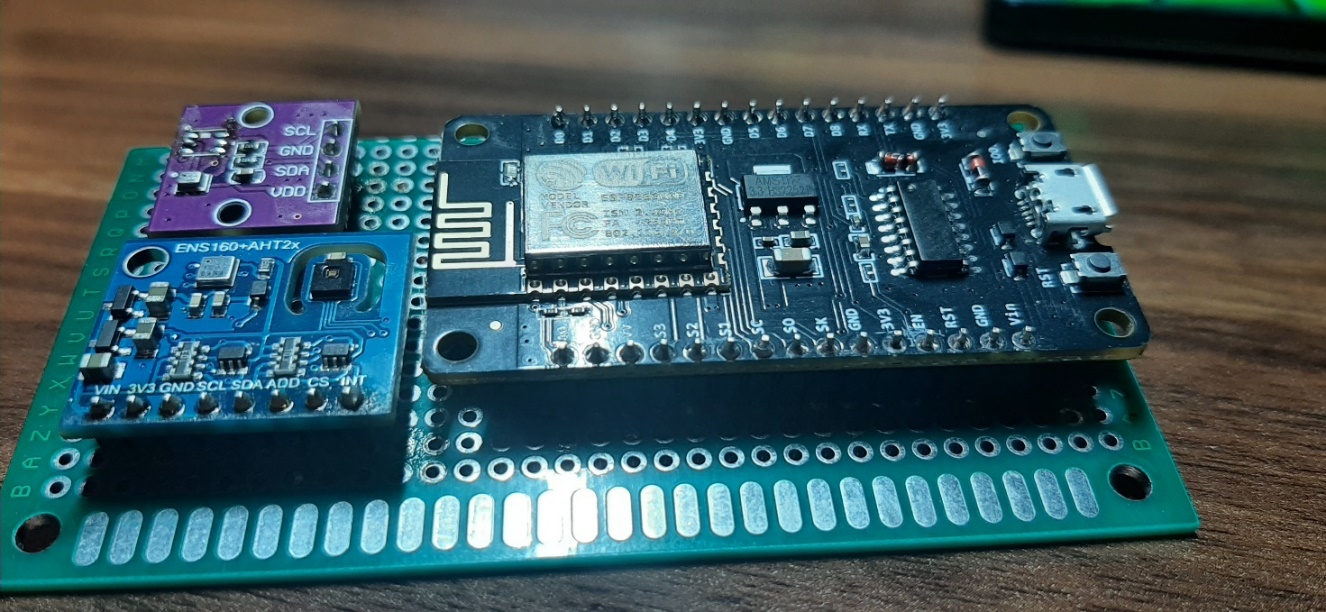


Рисунок 2.2 - макетна плата з сенсорами та мікроконтролером

Система моніторингу базується на мікроконтролері ESP8266 та використанні сенсорів AHT20 для вимірювання температури та вологості, BMP280 для вимірювання тиску та ENS160 для визначення індексу якості повітря (AQI), концентрації летких органічних речовин (TVOC) та рівня eCO2.

Система використовує три основні сенсори для збору даних про стан довкілля:

1. AHT20 — сенсор для вимірювання температури та вологості.
2. BMP280 — сенсор для вимірювання атмосферного тиску.
3. ENS160 — сенсор для вимірювання індексу якості повітря, TVOC та eCO2.

Код ініціалізації цих сенсорів (Рис.2.3-2.5) міститься у функціях *initAHT20()*, *initBMP280()* та *initENS160()*. У випадку, якщо сенсор не підключено або виникла проблема, система виводить відповідне повідомлення через послідовний порт.

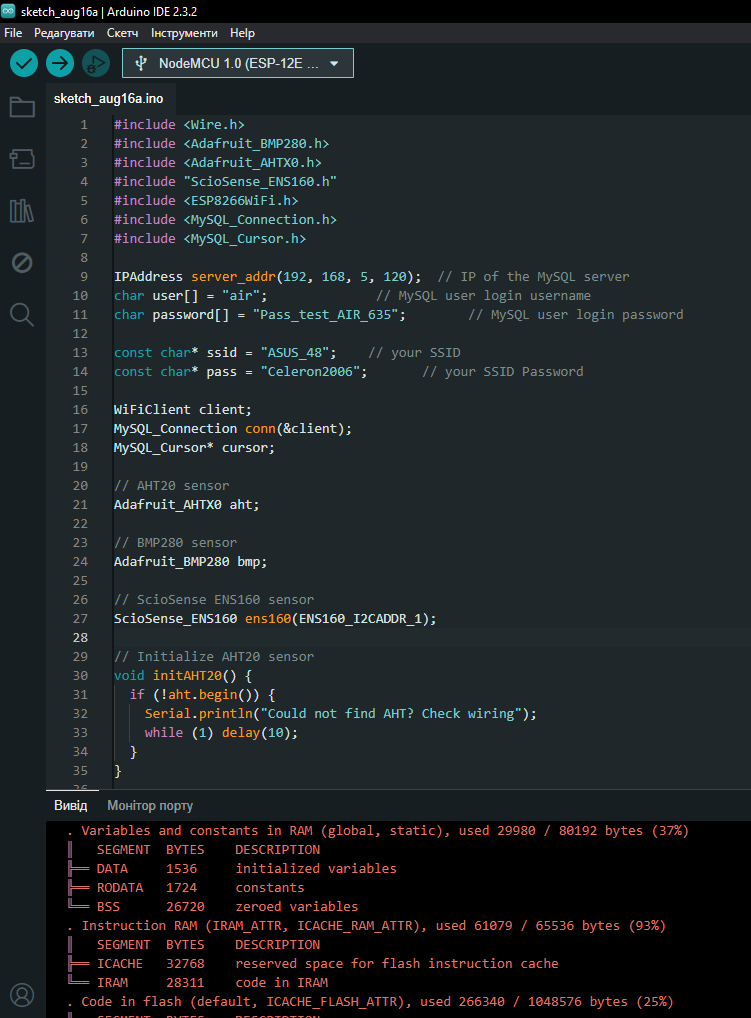


Рисунок 2.3- код ініціалізації сенсора AHT20

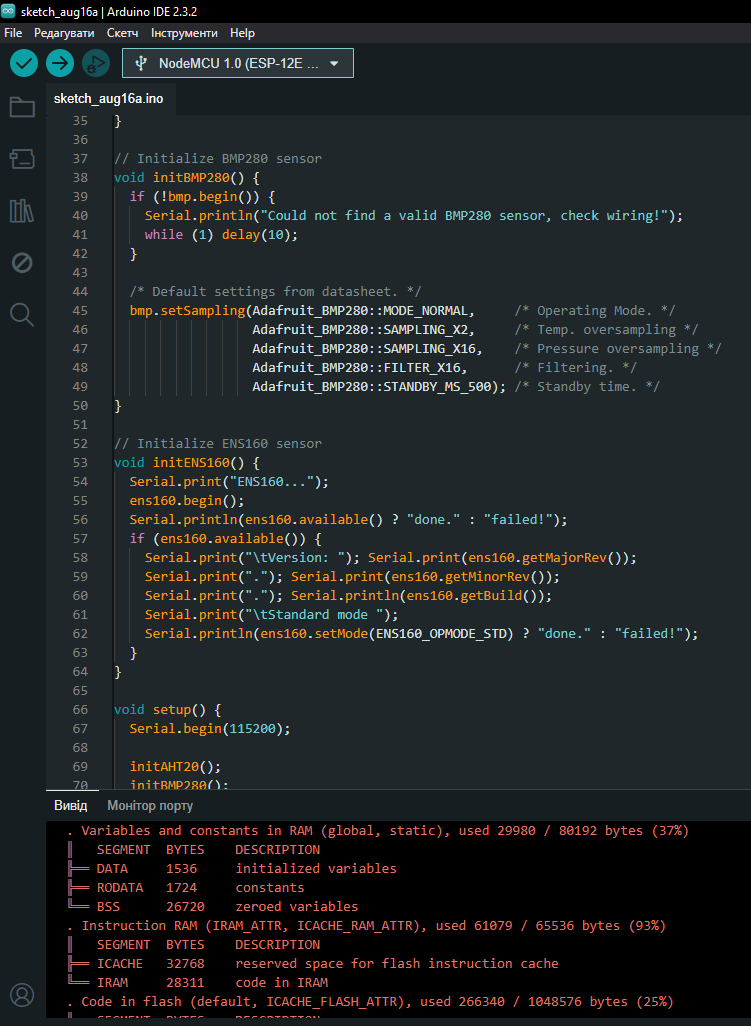


Рисунок 2.4- код ініціалізації сенсора BMP280

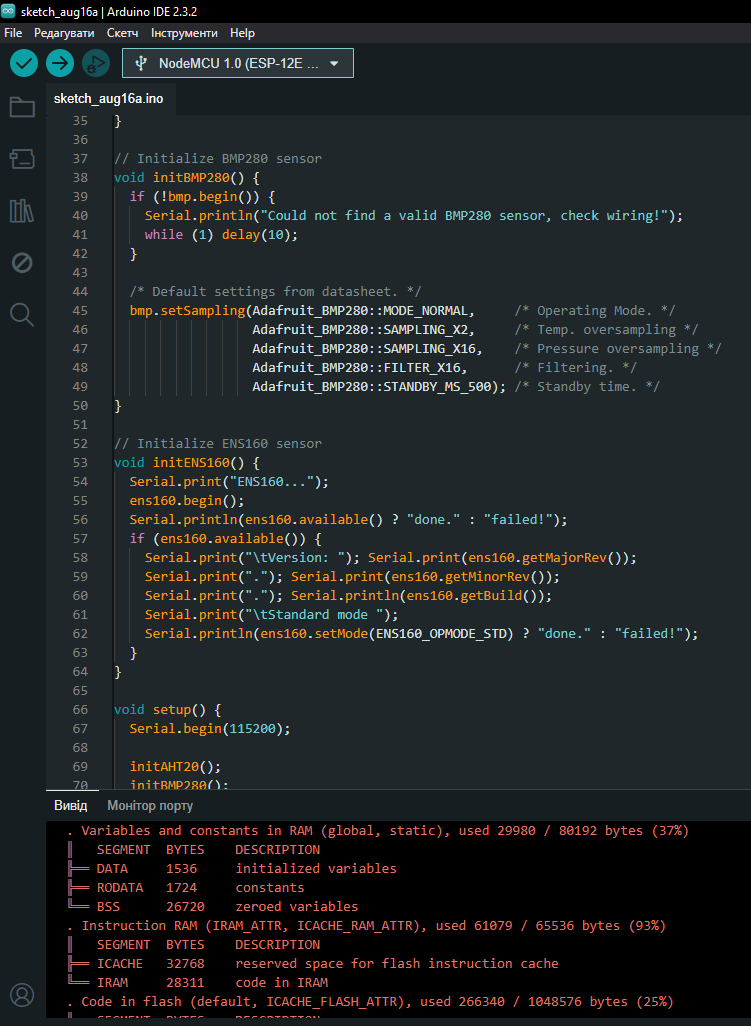


Рисунок 2.5 - код ініціалізації сенсора ENS160

## 2.3. Налаштування бездротового з'єднання та передача даних

Мікроконтролер ESP8266 з'єднується з локальною Wi-Fi мережею для передачі даних. У разі успішного підключення виводиться інформація про IP-адресу пристрою (Рис.2.6)

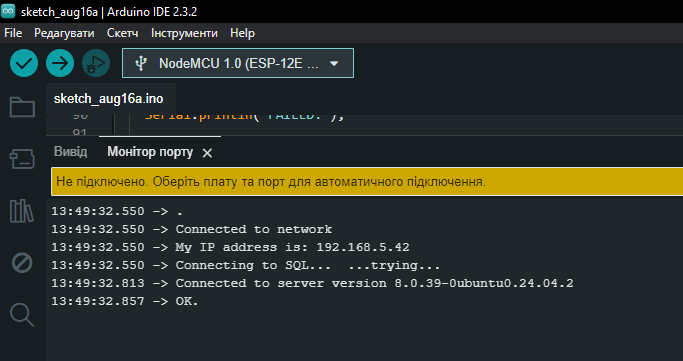


Рисунок 2.6 - процес підключення до мережі WiFi

Основним етапом роботи системи є збирання даних з різних сенсорів, після чого вони передаються для збереження в базу даних MySQL (Рис.2.7). Пристрій підключається до цієї бази через Wi-Fi мережу. Цикл збору даних виглядає наступним чином:

* зчитування даних з сенсора AHT20. Зчитуються показники температури та вологості, які зберігаються у змінних temp (температура) та humidity1 (вологість). Ці дані необхідні для моніторингу кліматичних умов в атмосфері;
* зчитування даних з сенсора BMP280. Вимірюється атмосферний тиск, який конвертується в одиниці гектопаскалів (hPa). Цей параметр використовується для оцінки погодних умов та передбачення змін в атмосфері;
* зчитування даних з сенсора ENS160. Вимірюються показники якості повітря, зокрема AQI (індекс якості повітря), TVOC (загальна кількість летких органічних сполук) та eCO2 (еквівалент вуглекислого газу). Ці дані необхідні для визначення рівня забруднення повітря та його впливу на здоров'я;

Після того, як дані зібрані, вони передаються у вигляді SQL-запиту до бази даних MySQL, де зберігаються для подальшого аналізу, візуалізації або автоматичного реагування на зміни параметрів якості повітря.



Рисунок 2.7 - фрагмент коду передачі даних до сервера бази даних MySQL

# РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ

Для створення середовище розглянемо рішення від Proxmox, яке буде використовуватись як платформа для віртуалізації (Рис.3.1).

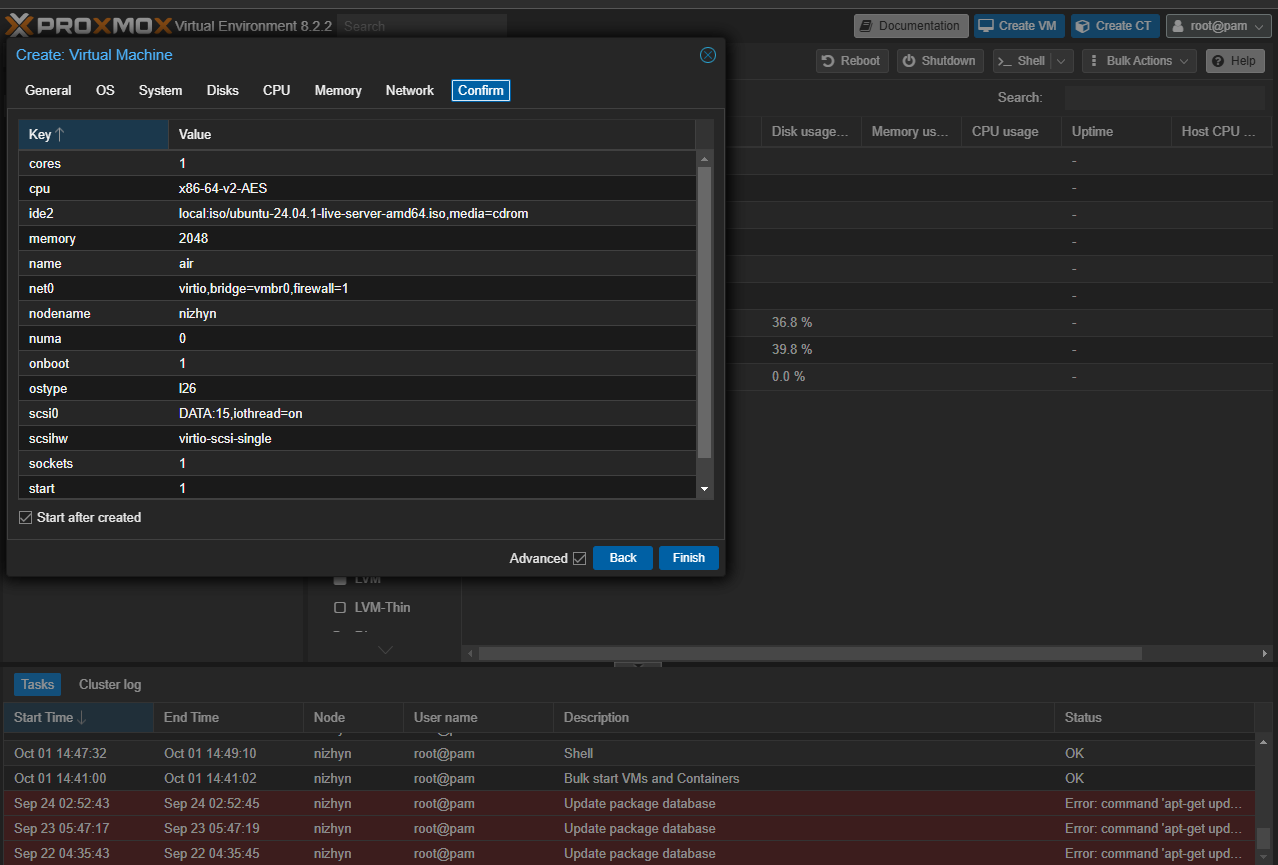


Рисунок 3.1 - оболонка Proxmox Virtual Environment

Proxmox Virtual Environment (Proxmox VE) — це потужна платформа для віртуалізації з відкритим вихідним кодом, яка дозволяє керувати віртуальними машинами (VM).

Після інсталяції, за допомогою веб-інтерфейсу Proxmox, створимо віртуальну машину. У налаштуваннях віртуальної машини задамо параметри, а саме тип ОС, в нашому випадку Linux, обсяг оперативної пам’яті, розмір диску та кількість ядер CPU. Вкажемо мережевий інтерфейс для підключення до локальної мережі. Після завершення цих налаштувань встановлюємо Ubuntu Server і розпочинаємо його налагодження.

## Створення серверного середовища для обробки даних

Інсталяція та налаштування Ubuntu Server не потребує специфічних налаштувань. Під час встановлення задаємо основні параметри, такі як мова, часова зона, розкладка клавіатури, облікові дані адміністратора та оновимо систему до актуальної версії за допомогою команди *sudo apt update* *&&* *sudo apt upgrade* (Рис.3.2).

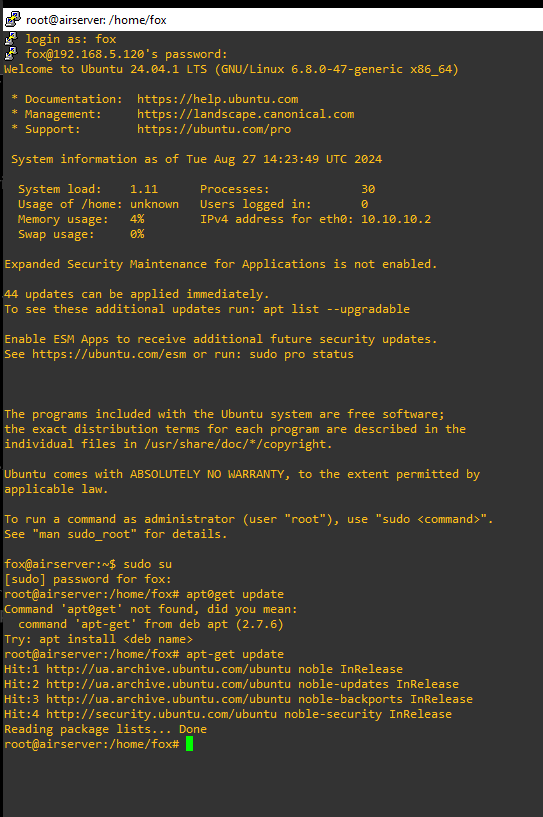


Рисунок 5.2 - оновлення пакетів Ubuntu Server

Встановлюємо відповідну часову зону командою *timedatectl set-timezone Europe/Kyiv*, що необхідно для коректного відображання дати та часу (Рис.3.3).

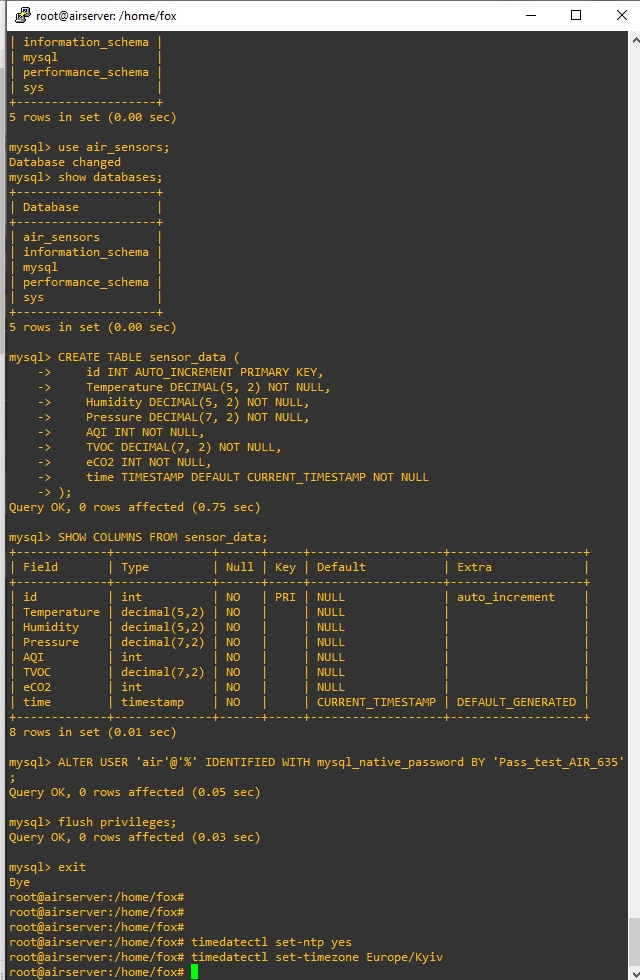


Рисунок 3.3 - встановлення часової зони Europe/Kyiv

Після завершення налаштування операційної системи на базі Ubuntu, наступним етапом є встановлення MySQL Server, який слугуватиме для зберігання даних, зібраних із сенсорів.

Початково інсталяція MySQL здійснюється за допомогою команди *sudo apt install mysql-server.* Цей крок дозволяє налаштувати сервер баз даних на нашій системі. Після встановлення, для підвищення рівня безпеки, запускаємо утиліту *mysql\_secure\_installation*, яка пропонує серію кроків для налаштування безпеки. Це включає в себе встановлення пароля для користувача *root*, видалення анонімних користувачів та обмеження доступу до *MySQL*.

Наступним кроком є підключення до MySQL і створення нової бази даних, зокрема *air\_sensors*, яка буде зберігати інформацію про дані сенсорів. Крім того, створюється новий користувач, якому надаються права доступу до цієї бази.

В базі даних використаємо таблицю *sensor\_data*, для зберігання показників з датчиків. Таблиця включає поля для зберігання даних про температуру, вологість, атмосферний тиск, індекс якості повітря (AQI), концентрацію летких органічних сполук (TVOC) та еквівалент вуглекислого газу (eCO2).

Для застосування всіх змін, які були внесені під час налаштування користувачів та прав доступу, виконується команда - *FLUSH PRIVILEGES;*

Цей етап завершується налаштуванням бази даних та підготовкою до подальшої роботи з нею для збереження й обробки даних від сенсорів.

Для налаштування доступу до MySQL із зовнішніх джерел потрібно внести зміни в конфігураційний файл *mysqld.cnf*. Цей файл визначає параметри, за якими MySQL приймає підключення, тому для того, щоб сервер міг приймати запити з інших пристроїв, знадобиться відкоригувати параметр, який контролює доступ.

Під корегуємо файл *mysqld.cnf* за допомогою текстового редактора *nano*, виконавши команду *sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf*. У відкритому файлі параметр *bind-address* зазвичай встановлений на значення *127.0.0.1*, що обмежує доступ до MySQL лише з локального комп’ютера. Щоб дозволити доступ з будь-якої IP-адреси, змінемо значення на *0.0.0.0*. Така зміна дозволить підключатися до MySQL із зовнішніх серверів або пристроїв, що взаємодіють через інтернет (Рис.3.4).

Після внесення змін збережемо файл. Перезавантажимо MySQL застосуваши нову конфігурацію за допомогою команди *sudo systemctl restart mysql.* Ця дія активує зміни, дозволяючи підключення з будь-якої IP-адреси.

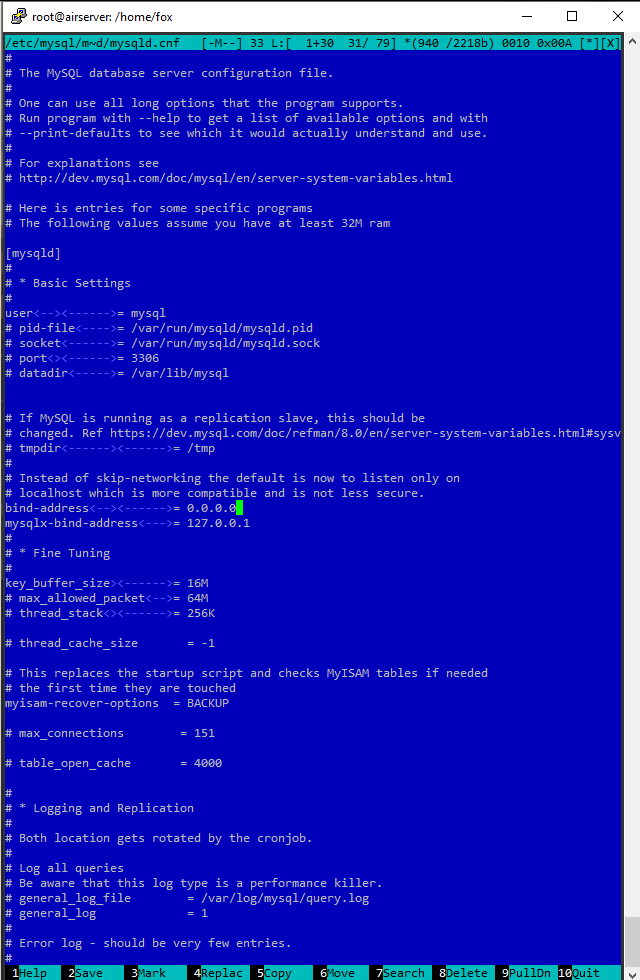


Рисунок 3.4 - налагодження mysqld.cnf

* 1. Обробка даних з сенсорів та їх зберігання

Для організації зберігання даних, що надходять від сенсорів, необхідно налаштувати базу даних MySQL. На першому етапі створюється користувач, який буде мати права доступу до бази даних. Користувач, наприклад, з іменем air, буде мати можливість підключатися до серверу MySQL з будь-якої IP-адреси. Це досягається за допомогою наступної команди - *CREATE USER 'air'@'%' IDENTIFIED BY 'Pass\_test\_AIR\_635';*

У даному випадку:

*air* — ім’я користувача, який має доступ до бази даних;

*%* — вказує на можливість підключення з будь-якої IP-адреси;

*Pass\_test\_AIR\_635* — пароль, що встановлюється для користувача air.

Наступним кроком є створення бази даних, в якій зберігатимуться всі зібрані дані. Для цього виконується команда - *CREATE DATABASE air\_sensors*;

Ця команда створює базу даних з ім'ям *air\_sensors*, що є основною для зберігання даних сенсорів.

Для забезпечення належного рівня доступу до бази даних для користувача *air*, необхідно надати йому права на виконання всіх необхідних операцій. Для цього використовується команда - *GRANT ALL PRIVILEGES ON air\_sensors.\* TO 'air'@'%'*;

Ця команда дозволяє користувачу *air* виконувати всі операції в межах бази даних *air\_sensors*, включаючи читання, запис і модифікацію даних (Рис.3.5).

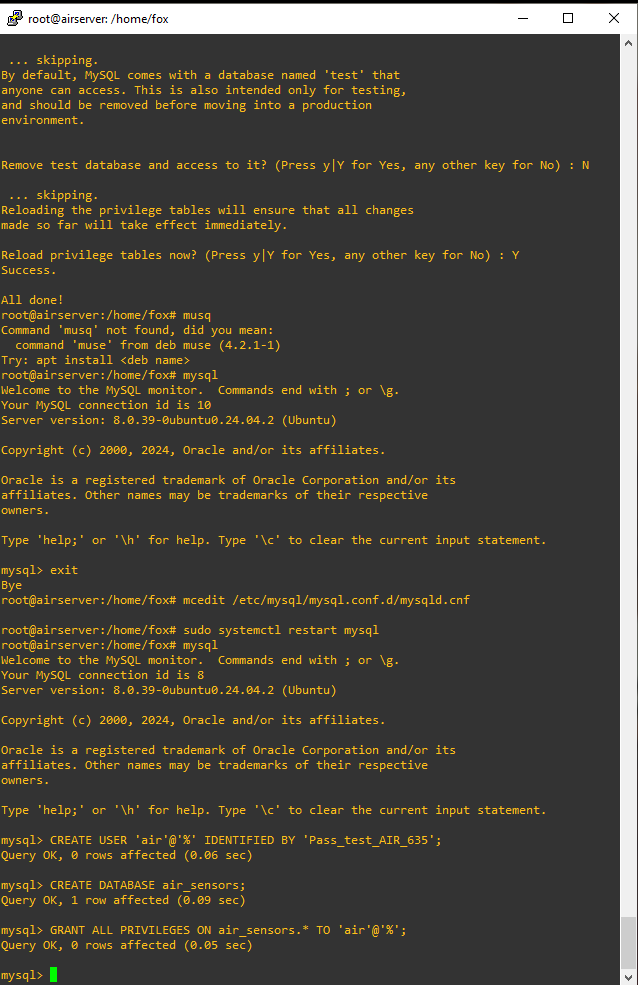


Рисунок 3.5 - налагодження бази даних

Після налаштування користувача та прав доступу можна переходити до створення таблиці, яка безпосередньо зберігатиме дані сенсорів, зокрема показники температури, вологості, тиску та інших екологічних параметрів.

## Створення таблиці для збереження показників сенсорів

Після створення користувача та бази даних, наступним етапом є створення таблиці *sensor\_data*, яка буде зберігати дані з сенсорів. Для цього спочатку потрібно вибрати базу даних *air\_sensors*, використовуючи команду - *USE air\_sensors*;

Ця команда забезпечує роботу в межах обраної бази даних, що дозволяє зберігати та обробляти дані саме в ній. Після вибору бази даних, створюється таблиця *sensor\_data* (Рис.3.6).

Структура таблиці БД передбачає наступні поля:

1. *id* - унікальний ідентифікатор кожного запису, який автоматично збільшується при додаванні нових даних.
2. *Temperature* - температура повітря, що зберігається у форматі десяткового числа з точністю до двох знаків. Це дозволяє забезпечити достатню точність вимірювання температури.
3. *Humidity* - відносна вологість повітря, також зберігається з точністю до двох знаків, що забезпечує необхідну точність даних.
4. *Pressure* - атмосферний тиск, який вимірюється в гПа (гектопаскалях).
5. *AQI* - індекс якості повітря, який дозволяє оцінити рівень забруднення повітря в конкретному регіоні.
6. *TVOC* - рівень летких органічних сполук (Total Volatile Organic Compounds), який є важливим показником забруднення повітря.
7. *eCO2* - еквівалент вуглекислого газу (CO2), що дозволяє оцінити рівень забруднення повітря на основі концентрації CO2.
8. *time* - мітка часу, яка автоматично встановлюється при кожному записі і вказує на момент збору даних.

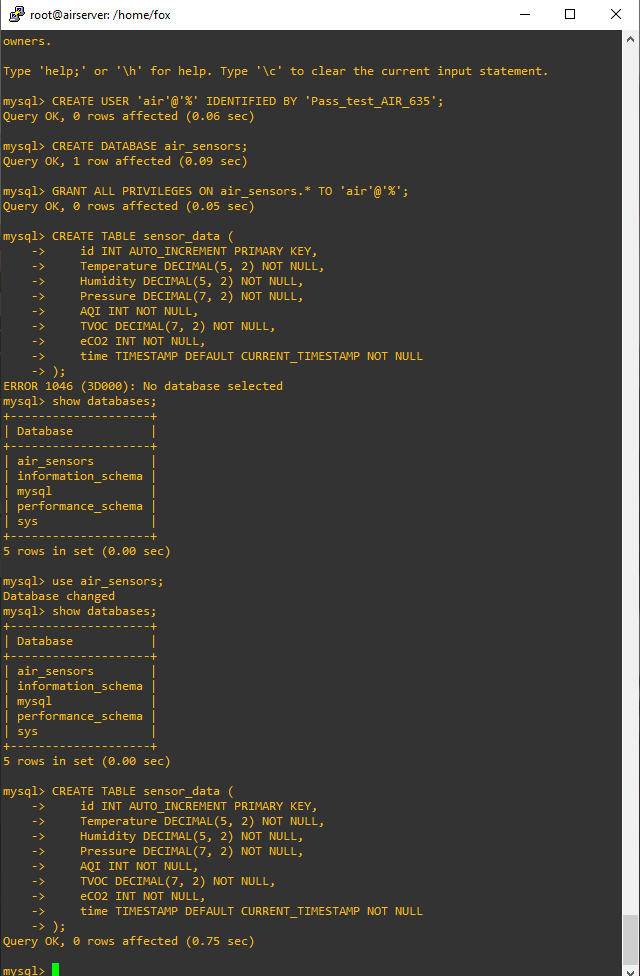


Рисунок 3.6 - створення таблиці sensor\_data

Застосована структура таблиці забезпечить ефективне зберігання даних з під’єднаних сенсорів і дозволяє подальше їх оброблення та аналіз.

Після створення таблиці *sensor\_data* необхідно перевірити її структуру, щоб упевнитись у коректності визначених полів та типів даних. Для цього застосовується SQL-команда - *SHOW COLUMNS FROM sensor\_data*; , що виводить всі стовпці таблиці разом з їхніми типами даних та додатковими характеристиками, що дає змогу перевірити відповідність структури таблиці вимогам проекту.

Для підвищення рівня безпеки необхідно завершити налаштування, попередньо використовуючи оновлений метод автентифікації для користувача *air*. Застосовуємо команду - *ALTER USER 'air'@'%' IDENTIFIED WITH mysql\_native\_password BY 'Pass\_test\_AIR\_635'*; Вказана команда змінює метод автентифікації для користувача *air* на *mysql\_native\_password*, що є більш безпечним варіантом.

Заключним етапом налаштування є виконання команди - *FLUSH PRIVILEGES*; , що застосовує всі внесені зміни в налаштування прав доступу та автентифікації, забезпечуючи їх негайне застосування без необхідності перезапуску сервера MySQL (Рис.3.7).

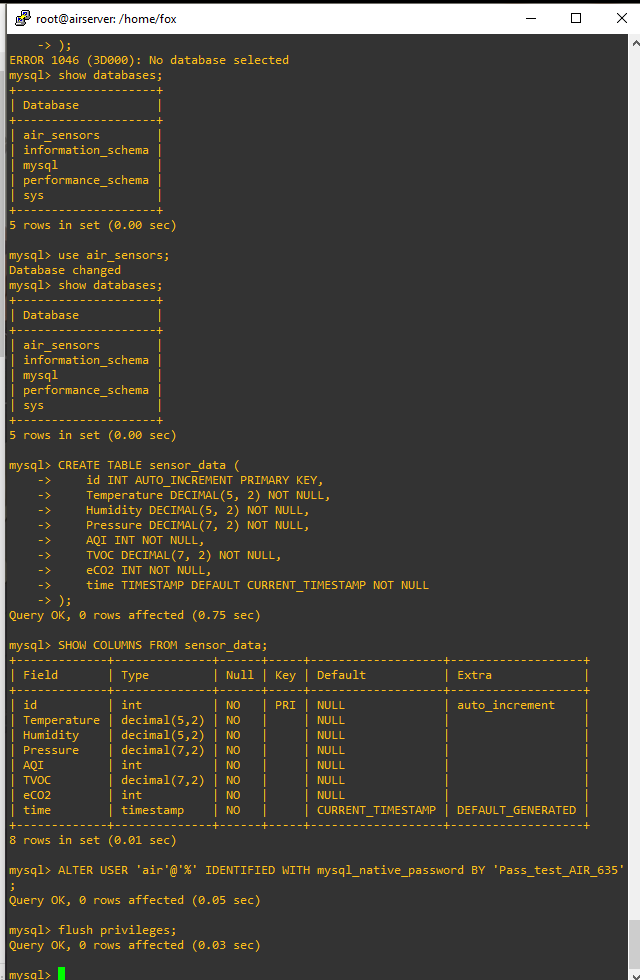


Рисунок 3.7 - створення таблиці та налаштування безпеки в MySQL

## Розробка веб-сайту для візуалізації та аналізу даних

Для налаштування веб-сервера Apache та створення веб-сайту, що відображатиме дані з сенсорів, необхідно встановити Apache та підтримку PHP, а також налаштувати зв’язок із базою даних MySQL.

Встановлення Apache, забезпечить роботу веб-сайту. Для цього виконуємо команду - *sudo apt install apache2,* що дозволить завантажити та встановити сервер Apache, який надасть можливість обробляти запити HTTP та обслуговувати веб-сторінки. Наступним кроком буде встановлення PHP для підтримки серверної логіки на веб-сайті, а також підтримку MySQL для взаємодії з базою даних. Використовуємо наступні команди - *sudo apt install php php-mysql,* це дозволить серверу виконувати PHP-скрипти та підключатися до бази даних MySQL, щоб отримувати і відображати дані сенсорів.

Після виконання вказаних кроків веб-сервер Apache буде готовий до обробки запитів, а PHP забезпечить динамічне відображення даних на веб-сайті.

Сервіс Apache запустимо за допомогою команди - *sudo systemctl start apache2,* для забезпечення автоматичного запуску Apache при завантаженні системи виконаємо команду - *sudo systemctl enable apache2*, що дозволить сервісу автоматично запускатися при кожному перезавантаженні сервера. Для перевірки статусу служби Apache використаємо команду - *service apache2 status.* Перевіримо, що веб-сервер Apache працює коректно, і його статус знаходиться в активному (*running*) стані.

Використаємо директорію /*var/www/air\_quality* для розміщення файлів веб-сайту. Для того щоб налаштувати правильні права доступу до цієї папки, потрібно виконати наступні кроки. За допомогою команди - *sudo chown -R $USER:$USER /var/www/air\_quality* встановлено власника директорії для поточного користувача та групу, забезпечуючи йому повний контроль над файлами в цій папці. Для надання прав доступу до файлів веб-сайту застосовано команду - *sudo chmod -R 755 /var/www/air\_quality,* це дозволяє користувачеві виконувати, читати та змінювати файли, а також забезпечує доступ для інших користувачів на читання та виконання, але без можливості змінювати ці файли (Рис.3.8). Таким чином, директорія була підготовлена для розміщення веб-сайтів із належними правами доступу.

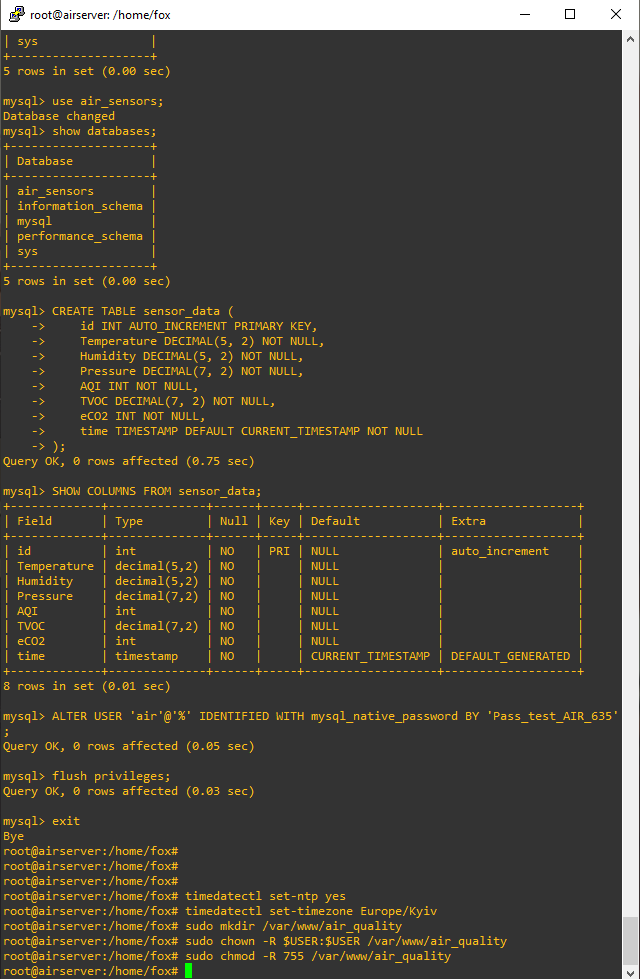


Рисунок 3.8 - надання прав доступу до файлів веб-сайту

У папці /*var/www/air\_quality* створемо просту HTML-сторінку з використанням текстового редактора mcedit, вміст сторінки див.рис.3.9.

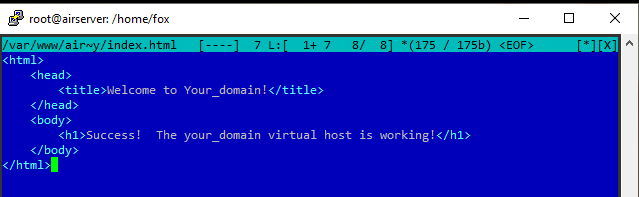


Рисунок 3.9 - тестова HTML-сторінка

Базова сторінка допоможе переконатися, що Apache працює належним чином.

Для того щоб веб-сервер Apache розпізнавав новий сайт, розміщений у директорії */var/www/air\_quality*, необхідно створити відповідний конфігураційний файл. Це здійснюється шляхом створення файлу *air\_quality.conf* у директорії налаштувань Apache за допомогою наступної команди - *sudo mcedit /etc/apache2/sites-available/air\_quality.conf*

Цей файл міститиме параметри конфігурації, що вказують веб-серверу, де знаходиться коренева директорія сайту, а також інші налаштування, необхідні для правильного обслуговування веб-сторінок.

Вміст конфігураційного файлу *air\_quality.conf* включає важливі налаштування для веб-сервера Apache, зокрема вказівку на використання директорії */var/www/air\_quality* як кореневої для домену *air\_quality*.

Для активації нового віртуального хосту виконаємо команду *sudo a2ensite air\_quality.conf*, яка дозволяє Apache використовувати конфігураційний файл для веб-сайту *air\_quality*. Після цього стандартний віртуальний хост деактивуємо за допомогою *sudo a2dissite 000-default.conf*, що забезпечить правильну маршрутизацію запитів до нового сайту. Перед перезапуском Apache проведемо тест конфігурації за допомогою *sudo apache2ctl configtest*, щоб переконатися у відсутності помилок у налаштуваннях. Для застосування змін перезавантажимо сервіс Apache командою *sudo systemctl restart apache2* (Рис.3.10). Ці кроки забезпечать коректну активацію та запуск веб-сайту.

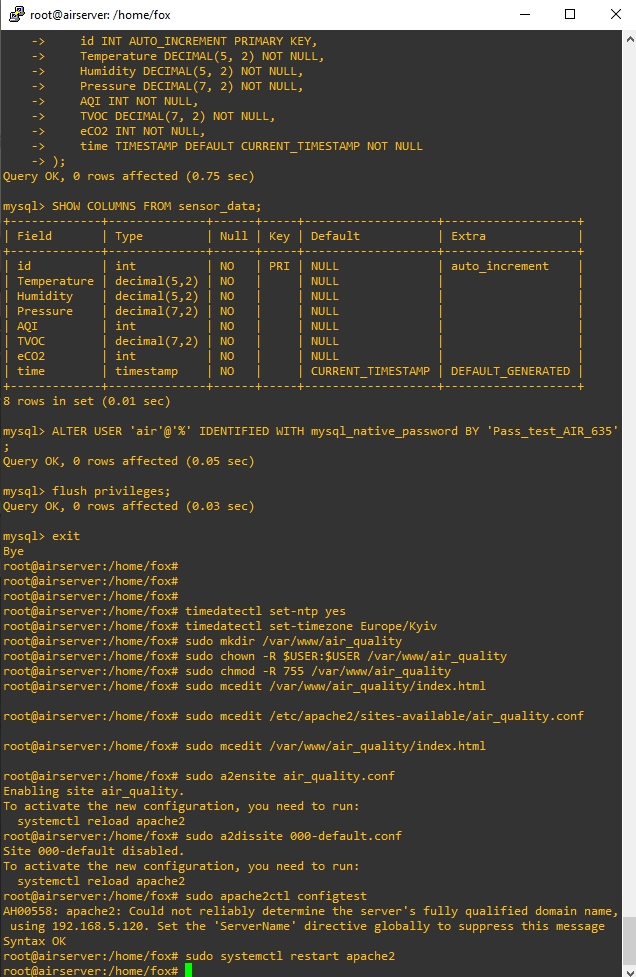


Рисунок 3.10 - активація віртуального хосту

Після завершення налаштувань, веб-сервер Apache готовий обслуговувати новий сайт, і тепер можна відкрити браузер, щоб перевірити доступність тестової сторінки. Відкривши сторінку (Рис.3.11) побачимо повідомлення, яке підтверджує успішну роботу налаштованого віртуального хоста, свідчачи про правильне налаштування серверного середовища.

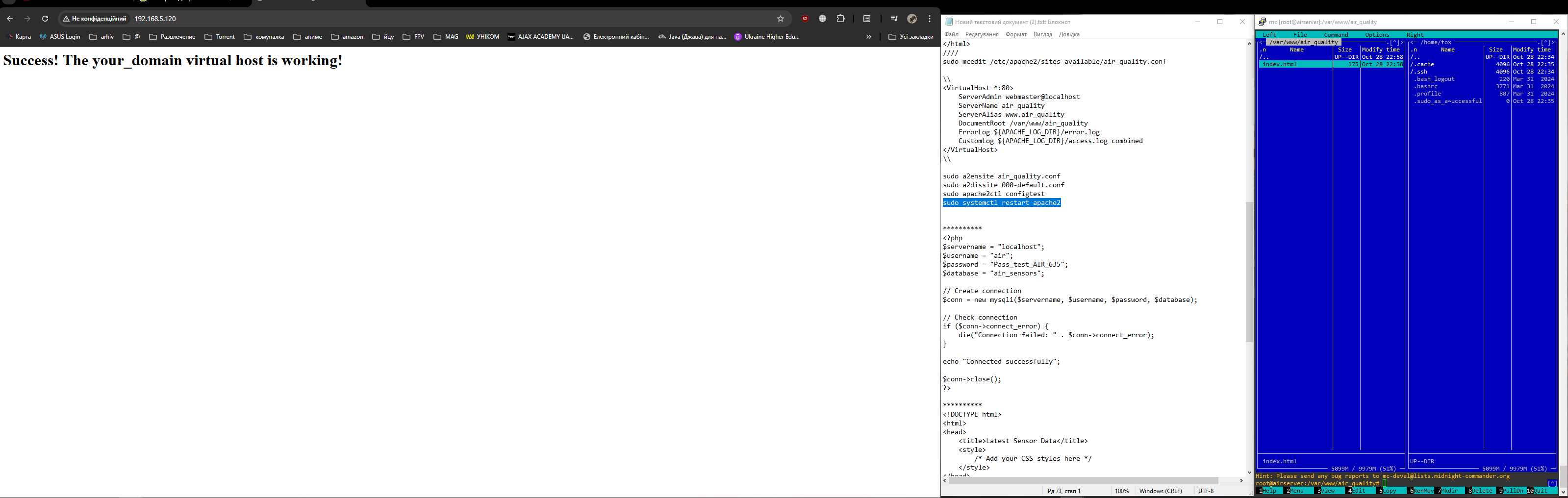


Рисунок 3.11 - робота віртуального хоста

# РЕАЛІЗАЦІЯ ВЕБ-ІНТЕРФЕЙСУ ДЛЯ ВІДОБРАЖЕННЯ ДАНИХ СЕНСОРІВ

Розробка веб-інтерфейсу для відображення та фільтрації даних сенсорів розпочинається з налаштування надійного з'єднання між веб-додатком та базою даних MySQL. Для цього було створено PHP-скрипт, який встановлює підключення до бази даних *air\_sensors*. Параметри підключення включають використання *localhost* як сервера бази даних, ім'я користувача *air*, пароль *Pass\_test\_AIR\_635* та назву бази даних. Підключення реалізується через об'єкт mysqli, що забезпечує зручний інтерфейс для виконання запитів у PHP. У разі успішного підключення скрипт виводить повідомлення "*Connected successfully*", а при виникненні помилки – повідомлення з текстом самої помилки для швидкої ідентифікації проблеми.

Наступним кроком створимо HTML-документ, який забезпечить графічний інтерфейс для відображення даних. Інтерфейс містить кілька основних компонентів, що підвищують зручність використання та ефективність взаємодії з даними. Основний контейнер *.container* включає заголовок *Sensor Data*, панель керування та область для відображення графіка. Панель керування містить два елементи для вибору початкової та кінцевої дати *(<input type="date">)* і кнопку для застосування фільтрації. Таке рішення дозволяє користувачам вибирати часовий діапазон для перегляду даних, що надає можливість аналізувати зміни показників протягом певного періоду.

Для графічного відображення даних використовується бібліотека *Chart.js*, яка дозволяє створювати динамічні та інтерактивні графіки з різними типами візуалізації. Завдяки гнучким налаштуванням та підтримці різних видів графіків було обрано лінійний графік для відображення зібраних сенсорами параметрів, таких як температура, вологість, тиск, індекс якості повітря (AQI), концентрація летких органічних сполук (TVOC) і рівень eCO2. На графіку кожен параметр представлено у вигляді окремої лінії різного кольору, що полегшує візуальне відстеження змін та аналіз взаємозв’язків між показниками.

На сторінці використовується JavaScript-скрипт, що містить кілька важливих функцій для інтерактивного відображення даних. Функція *fetchData()* відповідає за запит до серверного файлу з використанням AJAX-запиту та отримання даних без необхідності перезавантаження сторінки. Після успішного отримання відповідь парсується у формат JSON, і дані розподіляються по масивах для кожного параметра, таких як температура, вологість, тиск та інші показники якості повітря. Далі функція *updateChart()* використовує ці масиви для створення або оновлення графіка. Завдяки такому підходу відображення даних на графіку стає динамічним, що значно покращує досвід користувача, дозволяючи відстежувати зміни в реальному часі.

Для зручності користувачів розроблено функцію фільтрації даних за вибраний часовий період. Коли користувач задає діапазон дат через панель керування, активується функція *filterDataByDate()*, яка перевіряє, чи відповідає кожен запис сенсорних даних обраному інтервалу. Якщо значення потрапляють у межі вказаного діапазону, вони відбираються для подальшого відображення. Далі функція *updateChart()* оновлює графік, показуючи лише ті дані, що відповідають встановленим критеріям.

Інтерфейс розроблено з урахуванням принципів адаптивного дизайну для забезпечення оптимальної зручності доступу до даних на різних пристроях, включаючи комп'ютери, планшети та мобільні телефони. Застосування запитів дозволяє динамічно змінювати розташування елементів керування відповідно до ширини екрана - при вузькій ширині панель керування переходить у вертикальне розташування, що підвищує зручність користування на мобільних пристроях.

Таким чином, створений веб-інтерфейс надає користувачам можливість не лише переглядати поточні значення показників сенсорів, але й здійснювати аналіз даних у графічному вигляді. Це забезпечує комплексний підхід до моніторингу якості повітря, дозволяючи користувачам отримувати глибоке розуміння екологічної ситуації та приймати обґрунтовані рішення для її поліпшення.

## 4.1. Реєстрація доменного імені

Реєстрацію доменного імені проведемо на хостингу Freehost, яка починається зі створення облікового запису на цій платформі. Після авторизації користувач має можливість вибрати бажане доменне ім’я та перевірити його доступність у потрібній доменній зоні — у даному випадку, це *nizhyn.digital* (Рис.4.1). Якщо вибране ім’я доступне, його можна зареєструвати шляхом додавання домену до кошика та продовження його реєстрації.

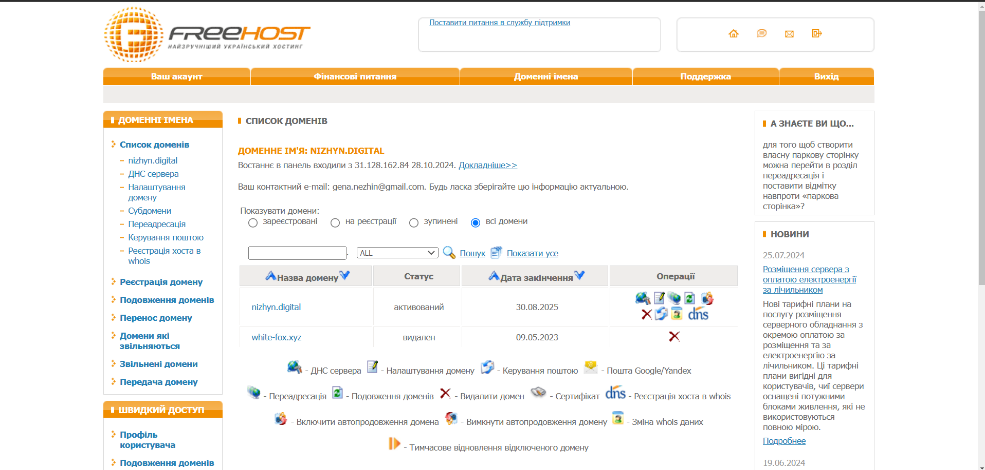


Рисунок 4.1 - реєстрація на хостингу Freehost

Щоб прив’язати домен до IP-адреси сервера, потрібно у розділі керування доменом на платформі хостингу, в налаштуваннях DNS додати A-запис (Рис.4.2), ввівши доменне ім'я у полі «Ім'я», вибравши тип запису «A» та вказавши IP-адресу сервера в полі «Значення». Це дозволить домену направлятися на потрібний сервер, забезпечуючи коректне відображення сайту.

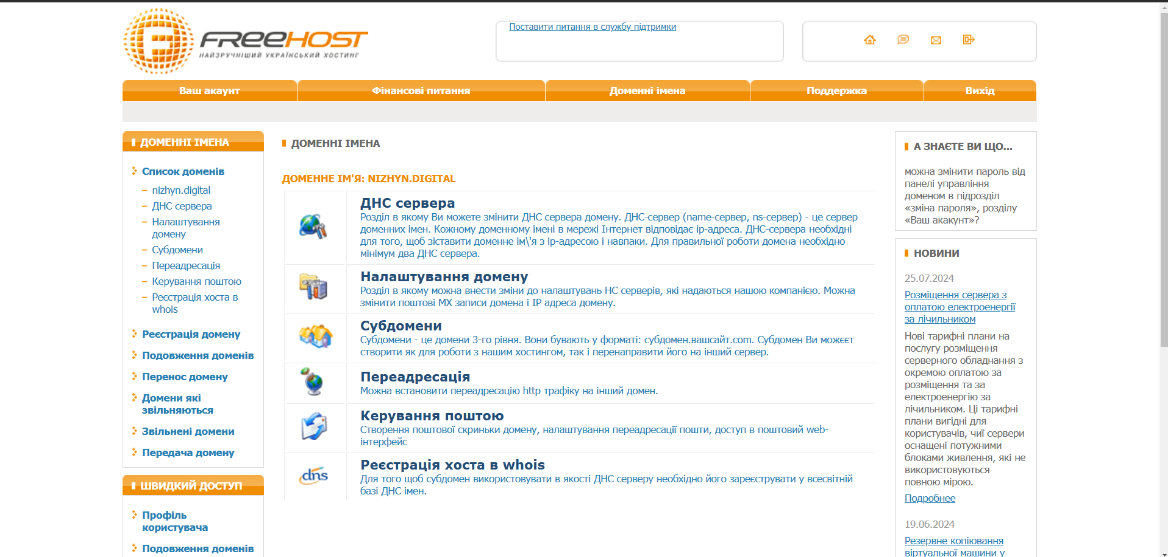


Рисунок 4.2 - налаштування DNS запису

Після внесення змін в налаштування DNS, може знадобитися від кількох до 48 годин для оновлення DMS записів. Для перевірки коректності прив'язки домену, введемо його у браузері та перевіримо, чи сайт доступний. Для додаткової перевірки можна скористатися онлайн-сервісом, таким як [*whatsmydns.net*](https://whatsmydns.net) *(Рис.4.3)*, який дозволяє відстежувати поширення DNS-записів по різних географічних регіонах і підтвердити правильність налаштувань..

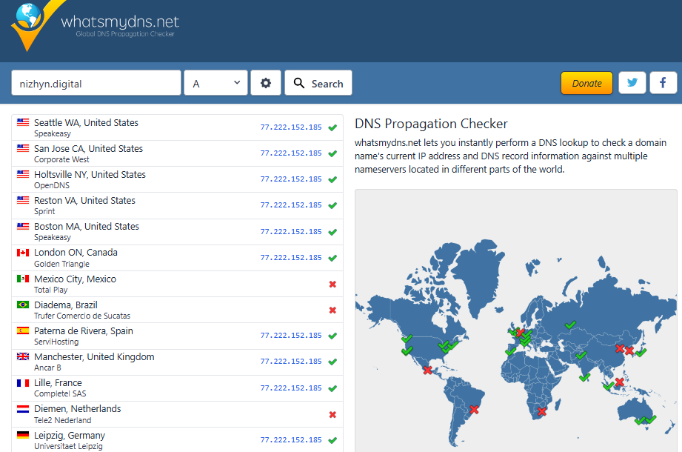


Рисунок 4.3 - перевірка доступності сервера

## 4.2. Огляд функціональності сайту

Сайт для відображення даних сенсорів оснащений рядом корисних функцій, що сприяють візуалізації та аналізу зібраної інформації про шкідливі речовини в довкіллі. Для налаштування сервера було обрано порт *8181,* тому для доступу до веб-сторінки нашого сайту користувачам необхідно перейти за посиланням: [*http://nizhyn.digital:8181/*](http://nizhyn.digital:8181/) (Рис.4.4)



Рисунок 4.4 - сторінка сайту моніторингу шкідливих речовин

Інтерфейс веб-сайту для відображення даних сенсорів включає панель для вибору діапазону дат, що дає користувачам можливість зручно фільтрувати дані за певний період, вибираючи початкову та кінцеву дати (Рис.4.5). Ця функція є особливо корисною для відстеження змін концентрації шкідливих речовин протягом часу, що допомагає виявляти тенденції або аномальні значення.



Рисунок 4.5 - діапазон відображання показників

Графічне представлення даних, яке було реалізовано за допомогою бібліотеки *Chart.js*, наглядно дозволяє створювати динамічні та інтерактивні графіки відображення показань. На графіку відображаються кілька важливих параметрів, таких як температура, вологість, тиск, індекс якості повітря (AQI), рівень летких органічних сполук (TVOC) та концентрація eCO2. Кожен параметр представлений окремими осями, що дозволяє користувачам вибирати, які показники відображати на графіку, і вмикати або вимикати їх для легшої оцінки змін та взаємозв’язку між параметрами (Рис.4.6).



Рисунок 4.6 - відображення потрібних показників

Інтерфейс сайту є зручним та адаптивним, що дозволяє працювати з графіками на різних пристроях, включаючи комп'ютери та мобільні телефони, забезпечуючи комфортний досвід користування.

Функціональність сайту забезпечує зручне оновлення графіка на основі фільтрованих даних. Інструменти підказок на графіку допомагають користувачам бачити значення параметрів у зрозумілому та доступному форматі (Рис.4.7-4.8).

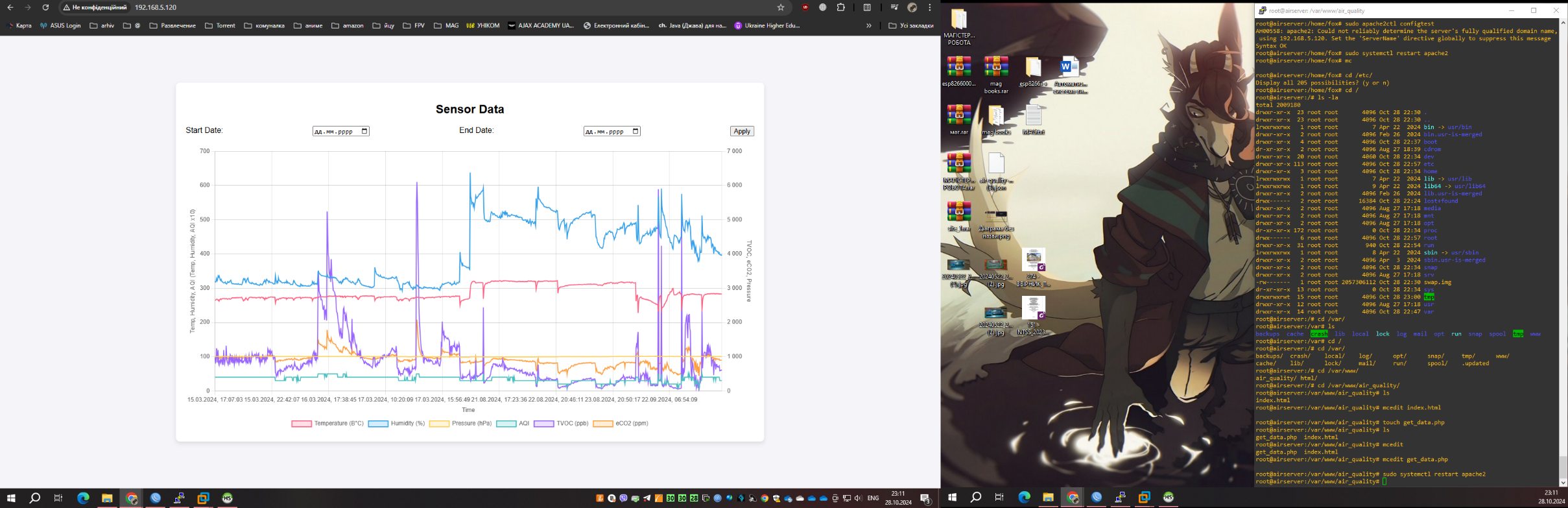


Рисунок 4.7 - відображення показників датчиків

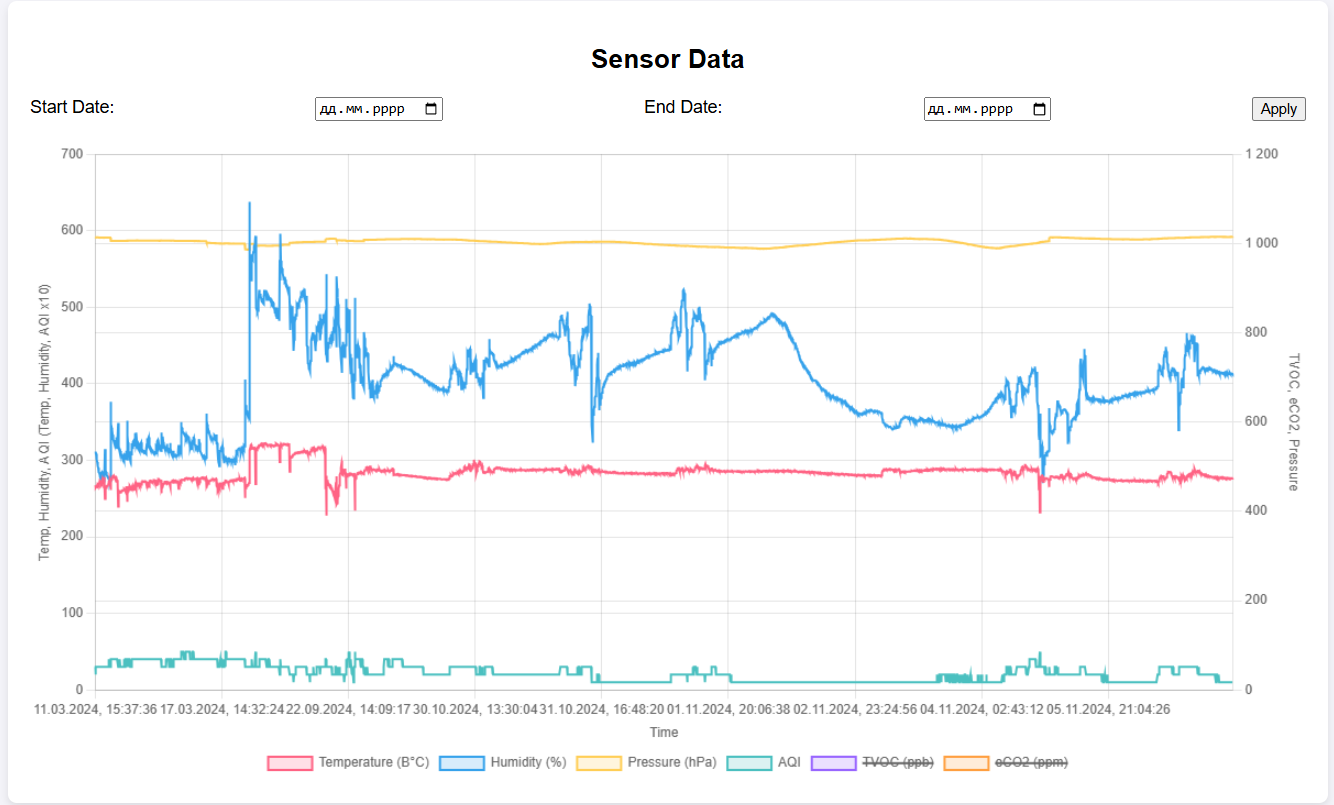


Рисунок 4.8 - вибіркове відображення показників датчиків

Незважаючи на те, що система моніторингу вже надає базові функції для візуалізації та аналізу даних, існують різні шляхи для вдосконалення цієї системи. Наприклад, розширивши географію використання датчиків якості повітря та об'єднання їх в єдину систему моніторингу, можна досягти більш масштабованого та комплексного підходу до аналізу екологічної ситуації. Збір даних з різних регіонів дозволить створити інтегровану карту, що відображатиме загальний стан забруднення повітря по всій території, що буде корисним не лише для громадян, а й для органів влади, екологічних центрів та науковців.

Ще однією важливою функцією є впровадження системи попереджень та оповіщень. За допомогою такої системи користувачі отримуватимуть сповіщення про перевищення встановлених порогових значень для різних параметрів якості повітря, таких як AQI, концентрація вуглекислого газу (eCO2) чи рівень токсичних газів. Це дозволить людям своєчасно реагувати на потенційно небезпечні ситуації та вживати необхідних заходів для збереження здоров'я.

Подальший розвиток системи також передбачає забезпечення інтеграції з іншими екологічними платформами. Взаємодія з державними чи міжнародними мережами моніторингу дозволить обмінюватися даними в режимі реального часу, що дасть змогу отримувати більш точну та актуальну інформацію про екологічний стан на різних рівнях, що сприятиме покращенню прогнозів.

Розробка мобільних додатків дозволить забезпечити користувачів інформацією про поточні показники якості повітря та при необхідності забезпечити оперативне оповіщення.

Ці та інші вдосконалення допоможуть перетворити систему моніторингу якості повітря на потужний інструмент, який наддасть користувачам можливість своєчасно реагувати на екологічні зміни та приймати рішення для покращення стану довкілля.

# ВИСНОВКИ

У рамках роботи була розроблена автоматизована система вимірювання та аналізу рівня шкідливих речовин в довкіллі, яка сприяє ефективному моніторингу стану навколишнього середовища та оперативному отриманню інформації про екологічні зміни. Система базується на використанні сучасних сенсорів якості повітря, що дозволяють здійснювати постійний збір даних щодо кількості шкідливих компонентів у атмосфері, таких як рівень вуглекислого газу (eCO2), летких органічних сполук (TVOC), індекс якості повітря (AQI), температура, вологість та тиск.

Веб-інтерфейс, розроблений для системи, забезпечує зручний доступ до даних та інтерактивне відображення результатів у графічному форматі. За допомогою панелі фільтрації користувачі можуть переглядати інформацію за конкретний часовий період, що дозволяє відстежувати зміни у якості повітря та виявляти тенденції чи аномальні ситуації.

Успішна реалізація функцій, таких як динамічне оновлення графіків і зручні інструменти для аналізу даних, сприяє більш точному моніторингу та покращенню взаємодії користувачів з системою. Подальший розвиток, зокрема розширення географії сенсорів, інтеграція з іншими екологічними платформами та розробка мобільних додатків, дозволить зробити систему більш масштабованою, доступною і корисною для широкого кола користувачів, від громадян до органів влади.

Завдяки вдосконаленню цієї системи в майбутньому буде можливість не лише покращити якість моніторингу екологічної ситуації, але й створити інструменти для оперативного реагування на шкідливі зміни в довкіллі. Впровадження таких рішень має важливе значення для підтримки здоров’я населення та збереження природних ресурсів.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Богдан Г. А., Глущенко М. О. Загальні тенденції побудови автоматизованих систем моніторингу якості повітря на промислових підприємствах. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» Вчені записки, 2023.
2. Горошко Г. О. Індикатори якості повітря : збірник тез доповідей Всеукраїнської мультидисциплінарної науково-практичної інтернет-конференції «Молодь і сучасні тренди наукової думки» (01 березня 2024 року) / Горошко Г. О. – Ніжин : НДУ ім. М. Гоголя, 2024. – С. 274-277.
3. Горошко Г. О. Інтерактивні технології в екологічній освіті. Новітні технології сучасного суспільства (НТСС-2023) : ІV Міжнародна науково-практична конференція (м. Чернігів, 19 грудня 2023 р.) : тези доповідей / Горошко Г. О. – Чернігів : НУ «Чернігівська політехніка», 2023. – С. 151-152.
4. ГОРОШКО О. Л. Розвиток інформаційної компетентності майбутніх вчителів математики засобами комп'ютерної візуалізації. Науковий журнал «Дистанційна освіта в Україні: інноваційні, нормативно-правові, педагогічні аспекти» №4, 2024. -С.247-255.
5. Данилюк Д. М. Автоматизована система очистки повітря від шкідливих домішок : кваліфікаційна робота бакалавра : 151 Автоматизація та комп’ютерно-інтегровані технології / Д. М. Данилюк ; Хмельниц. нац. ун-т. – Хмельницький, 2023. – С. 75.
6. Єніна І. І. Автоматизована система вимірювання показників повітря в побутовому приміщенні / Єніна І. І., Дяченко М. М. // Наукові записки : зб. наук. пр. - Кропивницький : ЦНТУ, 2017. - Вип. 21. - С. 77-81.
7. Лесковець Б. В. Автоматизована система дистанційного моніторингу стану навколишнього середовища : магістерська дис. : 151 Автоматизація та комп'ютерно-інтегровані технології / Київ, 2024. – С. 114.
8. Про моніторинг якості повітря в системі SaveEcoBot / SaveDnipro. URL: <https://www.savednipro.org/pro-monitoring-yakosti-povitrya-v-sistemi-saveecobot/> (дата звернення: 10.01.2024)
9. Станція моніторингу якості повітря / SaveDnipro. URL: <https://www.savednipro.org/product/stanciya-monitoringu-yakosti-povitrya/> (дата звернення: 15.11.2024)
10. ESP32 Node32 Lite PM2.5, Temperature, Humidity Sensors Node . Ton's Blog. URL: <https://ton.packetlove.com/blog/iot/esp32-node32-lite-pm25-temperature-humidity-sensors-node.html> (дата звернення: 17.08.2024)
11. George Smart. An ESP8266 Based Air Quality Sensor. URL: <https://www.george-smart.co.uk/2021/02/an-esp8266-based-air-quality-sensor/> (дата звернення: 15.11.2024)
12. IoT Air Pollution Monitoring with ESP8266. How2Electronics. URL: <https://how2electronics.com/iot-air-pollution-monitoring-esp8266/> (дата звернення: 14.10.2024)
13. Software IDE Documentation / Arduino. URL: <https://docs.arduino.cc/software/ide/> (дата звернення: 25.06.2024)
14. Weather and Air Quality Monitoring Station with ESP8266 and Home Assistant. Part of the Thing, 2021. URL: <https://partofthething.com/thoughts/weather-and-air-quality-monitoring-station-with-esp8266-and-home-assistant/> (дата звернення: 15.11.2024)

# ДОДАТОКИ

Додаток А

**Код для підключення ESP8266 до сенсорів та MySQL**

#include <Wire.h>

#include <Adafruit\_BMP280.h>

#include <Adafruit\_AHTX0.h>

#include "ScioSense\_ENS160.h"

#include <ESP8266WiFi.h>

#include <MySQL\_Connection.h>

#include <MySQL\_Cursor.h>

IPAddress server\_addr(192, 168, 5, 120); // IP of the MySQL server

char user[] = "air"; // MySQL user login username

char password[] = "Pass\_test\_AIR\_635"; // MySQL user login password

const char\* ssid = "ASUS\_48"; // your SSID

const char\* pass = "Celeron2006"; // your SSID Password

WiFiClient client;

MySQL\_Connection conn(&client);

MySQL\_Cursor\* cursor;

// AHT20 sensor

Adafruit\_AHTX0 aht;

// BMP280 sensor

Adafruit\_BMP280 bmp;

// ScioSense ENS160 sensor

ScioSense\_ENS160 ens160(ENS160\_I2CADDR\_1);

// Initialize AHT20 sensor

void initAHT20() {

if (!aht.begin()) {

Serial.println("Could not find AHT? Check wiring");

while (1) delay(10);

}

}

// Initialize BMP280 sensor

void initBMP280() {

if (!bmp.begin()) {

Serial.println("Could not find a valid BMP280 sensor, check wiring!");

while (1) delay(10);

}

/\* Default settings from datasheet. \*/

bmp.setSampling(Adafruit\_BMP280::MODE\_NORMAL, /\* Operating Mode. \*/

Adafruit\_BMP280::SAMPLING\_X2, /\* Temp. oversampling \*/

Adafruit\_BMP280::SAMPLING\_X16, /\* Pressure oversampling \*/

Adafruit\_BMP280::FILTER\_X16, /\* Filtering. \*/

Adafruit\_BMP280::STANDBY\_MS\_500); /\* Standby time. \*/

}

// Initialize ENS160 sensor

void initENS160() {

Serial.print("ENS160...");

ens160.begin();

Serial.println(ens160.available() ? "done." : "failed!");

if (ens160.available()) {

Serial.print("\tVersion: "); Serial.print(ens160.getMajorRev());

Serial.print("."); Serial.print(ens160.getMinorRev());

Serial.print("."); Serial.println(ens160.getBuild());

Serial.print("\tStandard mode ");

Serial.println(ens160.setMode(ENS160\_OPMODE\_STD) ? "done." : "failed!");

}

}

void setup() {

Serial.begin(115200);

initAHT20();

initBMP280();

initENS160();

// Begin WiFi section

Serial.printf("\nConnecting to %s", ssid);

WiFi.begin(ssid, pass);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

// print out info about the connection:

Serial.println("\nConnected to network");

Serial.print("My IP address is: ");

Serial.println(WiFi.localIP());

Serial.print("Connecting to SQL... ");

if (conn.connect(server\_addr, 3306, user, password))

Serial.println("OK.");

else

Serial.println("FAILED.");

// create MySQL cursor object

cursor = new MySQL\_Cursor(&conn);

}

void loop() {

// Read temperature and humidity from AHT20

sensors\_event\_t humidity1, temp;

aht.getEvent(&humidity1, &temp);

// Read pressure from BMP280

float pressure = bmp.readPressure() / 100.0F; // Convert to hPa

// Read AQI, TVOC, and eCO2 from ENS160

int AQI = 0, TVOC = 0, eCO2 = 0;

if (ens160.available()) {

ens160.set\_envdata(temp.temperature, humidity1.relative\_humidity);

ens160.measure(true);

ens160.measureRaw(true);

AQI = ens160.getAQI();

TVOC = ens160.getTVOC();

eCO2 = ens160.geteCO2();

}

// Insert data into MySQL database

if (conn.connected()) {

char INSERT\_SQL[200];

snprintf(INSERT\_SQL, sizeof(INSERT\_SQL), "INSERT INTO air\_sensors.sensor\_data (Temperature, Humidity, Pressure, AQI, TVOC, eCO2) VALUES (%.2f, %.2f, %.2f, %d, %d, %d)",

temp.temperature, humidity1.relative\_humidity, pressure, AQI, TVOC, eCO2);

cursor->execute(INSERT\_SQL);

}

delay(180000); // Delay for 3 minutes

}

Додаток Б

**Конфігураційний файл віртуального хосту Apache «*air\_quality.conf»***

<VirtualHost \*:80>

ServerAdmin webmaster@localhost

ServerName air\_quality

ServerAlias www.air\_quality

DocumentRoot /var/www/air\_quality

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Додаток В

**SQL скрипт для створення бази даних та таблиці «*Air.sql»***

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!50503 SET NAMES utf8mb4 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

-- Dumping database structure for air\_sensors

CREATE DATABASE IF NOT EXISTS `air\_sensors` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `air\_sensors`;

-- Dumping structure for таблиця air\_sensors.sensor\_data

CREATE TABLE IF NOT EXISTS `sensor\_data` (

`id` int NOT NULL AUTO\_INCREMENT,

`Temperature` decimal(5,2) NOT NULL,

`Humidity` decimal(5,2) NOT NULL,

`Pressure` decimal(7,2) NOT NULL,

`AQI` int NOT NULL,

`TVOC` decimal(7,2) NOT NULL,

`eCO2` int NOT NULL,

`time` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=734 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

-- Dumping data for table air\_sensors.sensor\_data: ~594 rows (приблизно)

INSERT INTO `sensor\_data` (`id`, `Temperature`, `Humidity`, `Pressure`, `AQI`, `TVOC`, `eCO2`, `time`) VALUES

(58, 26.74, 30.88, 1013.90, 3, 225.00, 723, '2024-03-11 13:37:36'),

(59, 26.50, 31.16, 1013.83, 2, 215.00, 712, '2024-03-11 13:45:10'),

(60, 26.43, 30.85, 1013.85, 3, 293.00, 796, '2024-03-11 13:48:10'),

(61, 26.44, 31.09, 1013.78, 3, 296.00, 800, '2024-03-11 13:51:10'),

(62, 26.54, 30.87, 1013.76, 3, 288.00, 791, '2024-03-11 13:54:11'),

/\*!40103 SET TIME\_ZONE=IFNULL(@OLD\_TIME\_ZONE, 'system') \*/;

/\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, '') \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=IFNULL(@OLD\_FOREIGN\_KEY\_CHECKS, 1) \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40111 SET SQL\_NOTES=IFNULL(@OLD\_SQL\_NOTES, 1) \*/;

Додаток Г

**Послідовність команд для налаштування веб-сервера Apache,   
бази даних MySQL та PHP**

sudo apt install apache2

sudo systemctl start apache2

sudo systemctl enable apache2

service apache2 status

hostname -I

sudo apt install php

sudo apt install php-mysql

sudo apt install mysql-server

sudo mysql\_secure\_installation

Додаток Д

**Налаштування користувача та бази даних в MySQL**

CREATE USER 'air'@'%' IDENTIFIED BY 'Pass\_test\_AIR\_635';

mcedit /etc/mysql/mysql.conf.d/mysqld.cnf || bind-address = 0.0.0.0

sudo systemctl restart mysql

mysql

CREATE DATABASE air\_sensors;

GRANT ALL PRIVILEGES ON air\_sensors.\* TO 'air'@'%';

CREATE TABLE sensor\_data (

id INT AUTO\_INCREMENT PRIMARY KEY,

Temperature DECIMAL(5, 2) NOT NULL,

Humidity DECIMAL(5, 2) NOT NULL,

Pressure DECIMAL(7, 2) NOT NULL,

AQI INT NOT NULL,

TVOC DECIMAL(7, 2) NOT NULL,

eCO2 INT NOT NULL,

time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP NOT NULL

);

SHOW COLUMNS FROM sensor\_data;

ALTER USER 'air'@'%' IDENTIFIED WITH mysql\_native\_password BY 'Pass\_test\_AIR\_635';

flush privileges;

Додаток Е

**Налаштування часу та часової зони в системі Linux**

timedatectl set-ntp yes

timedatectl set-timezone Europe/Kyiv

Додаток Ж

**Налаштування віртуального хоста Apache для веб-сайту *«air\_quality»***

sudo mkdir /var/www/air\_quality

sudo chown -R $USER:$USER /var/www/air\_quality

sudo chmod -R 755 /var/www/air\_quality

sudo mcedit /var/www/air\_quality/index.html

////

<html>

<head>

<title>Welcome to Your\_domain!</title>

</head>

<body>

<h1>Success! The your\_domain virtual host is working!</h1>

</body>

</html>

////

sudo mcedit /etc/apache2/sites-available/air\_quality.conf

\\

<VirtualHost \*:80>

ServerAdmin webmaster@localhost

ServerName air\_quality

ServerAlias www.air\_quality

DocumentRoot /var/www/air\_quality

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

\\

Додаток И

**Активування та деактивування віртуальних хостів Apache   
та перезапуск сервера**

sudo a2ensite air\_quality.conf

sudo a2dissite 000-default.conf

sudo apache2ctl configtest

sudo systemctl restart apache2

Додаток К

**PHP-скрипт для підключення до MySQL бази даних**

<?php

$servername = "localhost";

$username = "air";

$password = "Pass\_test\_AIR\_635";

$database = "air\_sensors";

// Create connection

$conn = new mysqli($servername, $username, $password, $database);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

echo "Connected successfully";

$conn->close();

?>

Додаток Л

**HTML-документ для відображення даних сенсорів з графіками**

<!DOCTYPE html>

<html>

<head>

<title>Sensor Data</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<style>

body, html {

margin: 0;

padding: 0;

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100%;

background-color: #f4f4f9;

box-sizing: border-box;

}

.container {

max-width: 1200px;

width: 100%;

padding: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

background-color: #fff;

border-radius: 8px;

box-sizing: border-box;

}

h2 {

text-align: center;

}

.controls {

display: flex;

justify-content: space-between;

margin-bottom: 20px;

}

#myChart {

width: 100%;

max-height: 1000px; /\* Ліміт по висоті \*/

height: 1000px; /\* Фіксована висота для кращого контролю \*/

}

@media screen and (max-width: 768px) {

.controls {

flex-direction: column;

align-items: center;

}

.controls label, .controls input, .controls button {

margin-bottom: 10px;

width: 100%;

text-align: center;

}

}

</style>

</head>

<body>

<div class="container">

<h2>Sensor Data</h2>

<div class="controls">

<label for="startDate">Start Date:</label>

<input type="date" id="startDate">

<label for="endDate">End Date:</label>

<input type="date" id="endDate">

<button onclick="filterDataByDate()">Apply</button>

</div>

<canvas id="myChart"></canvas>

</div>

<script>

var allData = [];

var chart;

function fetchData() {

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

try {

allData = JSON.parse(this.responseText);

} catch (e) {

console.error("Error parsing JSON data:", e);

return;

}

updateChart(allData);

} else if (this.readyState == 4) {

console.error("Failed to fetch data: Status", this.status);

}

};

xhttp.open("GET", "get\_data.php", true);

xhttp.send();

}

function updateChart(filteredData) {

var labels = [];

var temperatures = [];

var humidities = [];

var pressures = [];

var aqi = [];

var tvoc = [];

var eco2 = [];

var originalTemperatures = [];

var originalHumidities = [];

var originalAQI = [];

for (var i = 0; i < filteredData.length; i++) {

labels.push(new Date(filteredData[i].time).toLocaleString());

temperatures.push(filteredData[i].Temperature \* 10);

humidities.push(filteredData[i].Humidity \* 10);

pressures.push(filteredData[i].Pressure);

aqi.push(filteredData[i].AQI \* 10);

tvoc.push(filteredData[i].TVOC);

eco2.push(filteredData[i].eCO2);

originalTemperatures.push(filteredData[i].Temperature);

originalHumidities.push(filteredData[i].Humidity);

originalAQI.push(filteredData[i].AQI);

}

if (chart) {

chart.destroy();

}

var ctx = document.getElementById('myChart').getContext('2d');

chart = new Chart(ctx, {

type: 'line',

data: {

labels: labels.reverse(),

datasets: [{

label: 'Temperature (°C)',

data: temperatures.reverse(),

borderColor: 'rgba(255, 99, 132, 1)',

backgroundColor: 'rgba(255, 99, 132, 0.2)',

borderWidth: 2,

pointRadius: 0,

yAxisID: 'y1',

originalData: originalTemperatures.reverse()

}, {

label: 'Humidity (%)',

data: humidities.reverse(),

borderColor: 'rgba(54, 162, 235, 1)',

backgroundColor: 'rgba(54, 162, 235, 0.2)',

borderWidth: 2,

pointRadius: 0,

yAxisID: 'y1',

originalData: originalHumidities.reverse()

}, {

label: 'Pressure (hPa)',

data: pressures.reverse(),

borderColor: 'rgba(255, 206, 86, 1)',

backgroundColor: 'rgba(255, 206, 86, 0.2)',

borderWidth: 2,

pointRadius: 0,

yAxisID: 'y2'

}, {

label: 'AQI',

data: aqi.reverse(),

borderColor: 'rgba(75, 192, 192, 1)',

backgroundColor: 'rgba(75, 192, 192, 0.2)',

borderWidth: 2,

pointRadius: 0,

yAxisID: 'y1',

originalData: originalAQI.reverse()

}, {

label: 'TVOC (ppb)',

data: tvoc.reverse(),

borderColor: 'rgba(153, 102, 255, 1)',

backgroundColor: 'rgba(153, 102, 255, 0.2)',

borderWidth: 2,

pointRadius: 0,

yAxisID: 'y2'

}, {

label: 'eCO2 (ppm)',

data: eco2.reverse(),

borderColor: 'rgba(255, 159, 64, 1)',

backgroundColor: 'rgba(255, 159, 64, 0.2)',

borderWidth: 2,

pointRadius: 0,

yAxisID: 'y2'

}]

},

options: {

responsive: true,

maintainAspectRatio: true,

scales: {

x: {

ticks: {

maxRotation: 0,

autoSkip: true,

maxTicksLimit: 10

},

title: {

display: true,

text: 'Time'

}

},

y1: {

beginAtZero: true,

title: {

display: true,

text: 'Temp, Humidity, AQI (Temp, Humidity, AQI x10)'

}

},

y2: {

beginAtZero: true,

position: 'right',

title: {

display: true,

text: 'TVOC, eCO2, Pressure'

}

}

},

plugins: {

legend: {

position: 'bottom'

},

tooltip: {

mode: 'index',

intersect: false,

callbacks: {

label: function(context) {

var label = context.dataset.label || '';

if (label) {

label += ': ';

}

if (context.dataset.label === 'Temperature (°C)') {

label += context.dataset.originalData[context.dataIndex] + ' °C';

} else if (context.dataset.label === 'Humidity (%)') {

label += context.dataset.originalData[context.dataIndex] + ' %';

} else if (context.dataset.label === 'AQI') {

label += context.dataset.originalData[context.dataIndex];

} else {

label += context.parsed.y;

}

return label;

}

}

}

}

}

});

}

function filterDataByDate() {

var startDate = new Date(document.getElementById('startDate').value);

var endDate = new Date(document.getElementById('endDate').value);

endDate.setHours(23, 59, 59, 999);

var filteredData = allData.filter(function(item) {

var itemDate = new Date(item.time);

return itemDate >= startDate && itemDate <= endDate;

});

updateChart(filteredData);

}

fetchData();

</script>

</body>

</html>

Додаток М

**PHP скрипт для отримання останніх 1000 записів з бази даних та їх виведення в форматі JSON**

<?php

$servername = "localhost";

$username = "air";

$password = "Pass\_test\_AIR\_635";

$dbname = "air\_sensors";

// Створення з'єднання

$conn = new mysqli($servername, $username, $password, $dbname);

// Перевірка з'єднання

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// SQL Запит для отримання останніх 10 значень

$sql = "SELECT \* FROM sensor\_data ORDER BY id DESC LIMIT 1000";

$result = $conn->query($sql);

$rows = array();

while($row = $result->fetch\_assoc()) {

$rows[] = $row;

}

// Виведення даних у форматі JSON

echo json\_encode($rows);

$conn->close();

?>