

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

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

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-інтерфейсу інформаційної системи спостереження  
внутрішнього середовища «розумних приміщень»

Виконав: студент IV курсу, групи СН-41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Котлінський О.О.

(прізвище та ініціали)

Керівник

(підпис)

Станько А.А.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Марценко С.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Голотенко О.С.

(прізвище та ініціали)

Тернопіль  
2024

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

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

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ  
Завідувач кафедри

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

«26» червня 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

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

Студенту Котлінському Олександрю Олексійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-інтерфейсу інформаційної системи спостереження  
внутрішнього середовища «розумних приміщень».

Керівник роботи Станько Андрій Андрійович, доктор філософії, асистент кафедри КТ  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» квітня 2024 року № 4/7-470

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи Наукові публікації про технології систем спостереження  
внутрішнього середовища «розумних приміщень»

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

Вступ. 1 Аналіз предметної області та постановка завдання. Інформаційні системи «розумних міст». Поняття «розумного міста». Компоненти інформаційних систем «розумних міст». Роль внутрішнього середовища в концепції «розумного міста». Внутрішнє середовище «розумних приміщень». Параметри внутрішнього середовища. Веб-інтерфейс для інформаційних систем. Дослідження існуючих систем спостереження внутрішнього середовища «розумних приміщень». 2 Проектна частина. Розробка архітектури веб-інтерфейсу. Розробка бази даних веб-інтерфейсу. Діаграми прецедентів ролей в системі. Аналітичне опрацювання даних системи. 3 Практична частина. Розробка веб-інтерфейсу. Реалізація бази даних. Автентифікація та реєстрація користувачів. 4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел. Додаток А Перелік рисунків сторінок веб-інтерфейсу. Додаток Б React код створення компонентів веб-сторінок. Додаток В SQL запити створення всіх таблиць бази даних в системі.

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

1. Титульний слайд. 2. Актуальність 3-4. Мета, завдання, об'єкт дослідження, предмет дослідження 5. Внутрішнє середовище «розумних приміщень». 6. Веб-інтерфейс для інформаційних систем. 7. Розробка архітектури веб-інтерфейсу. 8. Розробка структури бази даних. 9. Діаграми прецедентів ролей в системі. 10. Розробка веб-інтерфейсу. 11. Автентифікація та реєстрація користувачів. 12. Висновки 13. Завершальний слайд



## АНОТАЦІЯ

Розробка веб-інтерфейсу інформаційної системи спостереження внутрішнього середовища «розумних приміщень» // Кваліфікаційна робота освітнього рівня «Бакалавр» // Котлінський Олександр Олексійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2024 // С. 70, рис. – 14, кресл – 13, додат. – 3, бібліогр. – 40.

**Ключові слова:** IoT, веб-інтерфейс, датчик, внутрішнє середовище, розумне місто, розумні приміщення, react, node.

Кваліфікаційна робота присвячена розробці веб-інтерфейсу інформаційної системи спостереження внутрішнього середовища «розумних приміщень».

В першому розділі кваліфікаційної роботи описано основні поняття та компоненти інформаційних систем «розумних міст». Висвітлено роль внутрішнього середовища в концепції «розумного міста» та розглянуто параметри, які його характеризують. Проаналізовано існуючі системи спостереження внутрішнього середовища «розумних приміщень» і зроблено висновки щодо їх недоліків.

В другому розділі кваліфікаційної роботи розроблено архітектуру веб-інтерфейсу, базу даних для нього, та діаграми прецедентів ролей у системі. Досліджено методи аналітичного опрацювання даних системи та подано результати проектування.

В третьому розділі кваліфікаційної роботи описано процес розробки веб-інтерфейсу, реалізації бази даних, а також автентифікацію та реєстрацію користувачів. Проаналізовано результати розробки та проведено їх тестування.

Об'єкт дослідження: інформаційні системи спостереження внутрішнього середовища «розумних приміщень». Предмет дослідження: веб-інтерфейс для керування датчиками у «розумному місті» та системи керування цими сенсорами.

## ANNOTATION

Development of the Information System Web Interface for the Internal Environment Monitoring of "Smart Premises" // Qualification work of the educational level "Bachelor" // Kotlinskyi Oleksandr Oleksiyovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-41 // Ternopil, 2024 // P. 70, fig. – 14, annexes. – 3, chair. -13, references - 40.

**Keywords:** IoT, web interface, sensor, indoor environment, smart city, smart premises, react, node.

The qualification work is dedicated to the development of the web interface of the information system for monitoring the internal environment of "smart premises".

The first section of the qualification paper considered the main concepts and components of information systems of smart cities. The role of the internal environment in the concept of a smart city is highlighted and the parameters characterizing it are considered. The existing systems for monitoring the internal environment of "smart premises" were analyzed and conclusions were drawn regarding their shortcomings.

In the second section of the qualification work, it is considered the architecture of the web interface, the database for it, and the role precedent diagrams in the system. Methods of analytical processing of system data were studied and design results were presented.

The third section of the qualification work describes the process of web interface development, database implementation, as well as user authentication and registration. The development results were analyzed and tested.

Object of research: information systems for monitoring the internal environment of smart premises. The subject of research: a web interface for managing sensors in a smart city and a system for managing these sensors.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

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

OAuth (англ. Open Authorization) –це протокол аутентифікації, який надає одним веб-додаткам можливість отримати доступ до ресурсів іншого додатка.

БД – База даних

ІС – Інформаційна система.

ВС – Внутрішнє середовище.

ІКТ – Інформаційно-комунікаційні технології

РМ – «Розумне місто»

## ЗМІСТ

Вступ.....	8
РОЗДІЛ 1. Аналіз предметної області та постановка завдання .....	10
1.1. Інформаційні системи «розумних міст» .....	10
1.1.1. Поняття «розумного міста» .....	12
1.1.2. Компоненти інформаційних систем «розумних міст» .....	15
1.1.3. Роль внутрішнього середовища в концепції «розумного міста» .....	17
1.2. Внутрішнє середовище «розумних приміщень».....	18
1.2.1. Параметри внутрішнього середовища .....	19
1.3. Веб-інтерфейс для інформаційних систем .....	21
1.3.1. Дослідження існуючих систем спостереження внутрішнього середовища «розумних приміщень».....	23
1.4. Висновок до першого розділу .....	24
РОЗДІЛ 2. Проектна частина .....	25
2.1. Розробка архітектури веб-інтерфейсу .....	25
2.2. Розробка структури бази даних веб-інтерфейсу .....	28
2.3. Діаграми прецедентів ролей в системі .....	31
2.4. Аналітичне опрацювання даних системи.....	37
2.5. Висновок до другого розділу .....	39
РОЗДІЛ 3. Практична частина .....	40
3.1. Розробка веб-інтерфейсу .....	40
3.2. Реалізація бази даних .....	47
3.3. Автентифікація та реєстрація користувачів .....	52
3.4. Висновок до третього розділу .....	56
РОЗДІЛ 4. Безпека життєдіяльності, основи хорони праці.....	58

	7
4.1. Вимоги безпеки праці під час експлуатації систем вентиляції, опалення та кондиціонування повітря.....	58
4.2. Розробка захисту від пожеж та вибухів в системах опалення, вентиляції, освітлення та кондиціонування повітря.....	61
Висновки.....	65
Перелік джерел.....	67
Додатки .....	71



## ВСТУП

**Актуальність теми.** Внаслідок глобалізації та швидкого розвитку технологій, міста по всьому світу стикаються з новими викликами в управлінні ресурсами та забезпеченні високого рівня життя для своїх мешканців. «Розумні міста», що використовують передові інформаційні системи для спостереження та управління різними аспектами міського середовища, стали актуальним напрямком сучасних досліджень у галузі інформаційних технологій та міського планування. Одним із ключових аспектів «розумних міст» є забезпечення оптимального внутрішнього середовища приміщень, що включає спостереження якості повітря, температури, вологості та інших параметрів. Тому розробка веб-інтерфейсу для інформаційних систем спостереження внутрішнього середовища «розумних приміщень» є надзвичайно актуальною темою, яка сприятиме підвищенню якості життя та ефективності використання ресурсів.

**Мета і задачі дослідження.** Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є підвищення якості послуг з спостереження та управління внутрішнім середовищем «розумних приміщень» через розробку ефективного веб-інтерфейсу. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Проаналізувати стан досліджень в області ІС «розумних міст».
- Дослідити компоненти ІС «розумних міст» та їх роль у забезпеченні оптимального внутрішнього середовища.
- Визначити ключові параметри внутрішнього середовища, які мають бути відстежені.
- Розробити архітектуру веб-інтерфейсу для інформаційної системи спостереження внутрішнього середовища.
- Створити базу даних для зберігання та управління даними системи.
- Розробити діаграми прецедентів ролей у системі.
- Провести аналітичне опрацювання даних системи та оцінити ефективність запропонованого рішення.

**Практичне значення одержаних результатів.** Практичне значення одержаних результатів полягає в розробці веб-інтерфейсу, який дозволить ефективно спостерігати та управляти параметрами внутрішнього середовища «розумних приміщень». Це сприятиме підвищенню якості повітря, температурного режиму та інших умов у приміщеннях, що, у свою чергу, підвищить комфорт і здоров'я мешканців. Запропоноване рішення може бути використане в різних сферах, включаючи житлові будинки, офіси, промислові приміщення та громадські будівлі.

## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

У сучасному світі концепція «розумного міста» набуває все більшої популярності, впроваджуючи інноваційні технології для покращення якості життя мешканців та підвищення ефективності управління міськими ресурсами [1]. Одним з ключових елементів РМ є інформаційні системи, які забезпечують збирання, обробку та аналіз даних з різних джерел, включаючи ВС приміщень. Враховуючи зростаючий інтерес до здорового та комфортного проживання, особлива увага приділяється спостереженню таких параметрів, як температура, вологість, якість повітря тощо.

ВС «розумних приміщень» відіграє важливу роль у загальній екосистемі РМ. Ефективне управління цими параметрами потребує впровадження складних систем спостереження та управління, які можуть бути реалізовані за допомогою різноманітних апаратних платформ, таких як Arduino або Raspberry Pi [2]. Ці платформи дозволяють інтегрувати різні датчики для збору даних, які можуть бути проаналізовані і використані для прийняття рішень в режимі реального часу.

Одним з важливих аспектів створення інформаційної системи для спостереження внутрішнього середовища є розробка зручного та функціонального веб-інтерфейсу. Веб-інтерфейс повинен забезпечувати користувачам легкий доступ до даних, зручне управління параметрами середовища та можливість оперативного реагування на зміни [3]. Технології, такі як React, Node.js та Express, надають розробникам потужні інструменти для створення ефективних та інтерактивних веб-інтерфейсів.

### **1.1. Інформаційні системи «розумних міст»**

Сучасні міста постійно розвиваються, і разом із цим зростають вимоги до управління міською інфраструктурою та забезпечення комфортних умов для мешканців. Концепція «розумного міста» передбачає використання новітніх

технологій для покращення якості життя, підвищення ефективності використання ресурсів і зменшення негативного впливу на навколишнє середовище [4]. Основою цієї концепції є інформаційні системи, які забезпечують збирання, обробку, зберігання та аналіз великої кількості даних з різноманітних джерел.

Інформаційні системи РМ охоплюють широкий спектр веб-інтерфейсів, які можуть бути класифіковані за їх призначенням. До них належать системи управління транспортом, енергозабезпеченням, водопостачанням, відходами, системами безпеки та інше [5]. Всі ці системи взаємопов'язані і працюють разом для створення єдиної інформаційної мережі, що дозволяє місту функціонувати більш ефективно.

Основні компоненти ІС «розумних міст»:

1. Давачі: Збір даних є першим і одним з найважливіших етапів. Давачі встановлюються в різних частинах міста і збирають інформацію про рух транспорту, якість повітря, рівень шуму, споживання енергії та багато іншого.

2. Мережі передачі даних: Зібрані дані передаються через мережі, які можуть включати дротові та бездротові технології, такі як оптоволоконні кабелі, Wi-Fi, 5G тощо.

3. Централізовані платформи обробки даних: Дані, що надходять, збираються і зберігаються на централізованих платформах. Ці платформи використовують потужні обчислювальні ресурси для обробки великих обсягів інформації в реальному часі.

4. Аналітичні інструменти: Після обробки даних використовуються різноманітні аналітичні інструменти для виявлення тенденцій, прогнозування подій та прийняття обґрунтованих рішень. Ці інструменти можуть базуватися на алгоритмах машинного навчання та штучного інтелекту.

5. Інтерфейси користувача: Результати аналізу даних представляються у зручній для користувачів формі через веб-інтерфейси, мобільні застосунки та інші платформи. Це дозволяє різним групам користувачів (мешканцям,

адміністраторам, службам реагування) швидко отримувати необхідну інформацію та діяти відповідно [6].

Важливою складовою частиною РМ є забезпечення комфортних умов всередині будівель та приміщень, де люди проводять більшу частину свого часу. Контроль і управління внутрішнім середовищем включають спостереження таких параметрів, як температура, вологість, рівень шкідливих газів та якість повітря. Це не тільки підвищує комфорт проживання, але й позитивно впливає на здоров'я мешканців та продуктивність праці [7].

Системи спостереження внутрішнього середовища інтегруються з іншими компонентами РМ, що дозволяє автоматично регулювати параметри на основі зовнішніх умов, наприклад, температури та рівня забруднення повітря на вулиці. Використання сучасних технологій, таких як IoT, дозволяє створювати «розумні приміщення», які самостійно адаптуються до змін і забезпечують максимальний комфорт та безпеку [8].

Отже, ІС «розумних міст» є комплексними та багатофункціональними структурами, які забезпечують збір та аналіз даних, прийняття рішень та автоматизацію процесів для покращення якості життя мешканців та ефективності управління міськими ресурсами. Далі розглянемо детальніше кожен з цих аспектів, починаючи з поняття РМ.

### **1.1.1. Поняття «розумного міста»**

«Розумне місто» (Smart City) – це концепція, що включає впровадження інформаційно-комунікаційних технологій (ІКТ) та IoT для управління міськими ресурсами та інфраструктурою більш ефективно і стійко [9]. Метою «розумного міста» є підвищення якості життя мешканців, оптимізація використання міських ресурсів, зменшення негативного впливу на навколишнє середовище та створення безпечного і комфортного середовища для проживання [10].

Концепція «розумного міста» виникла як відповідь на зростаючі виклики урбанізації та необхідність підвищення ефективності використання міських

ресурсів. У сучасних умовах швидкого розвитку технологій, впровадження ІКТ стало ключовим фактором у створенні інноваційних рішень для міського середовища [11]. РМ спрямоване на використання цих технологій для покращення якості життя мешканців, зменшення негативного впливу на навколишнє середовище, а також для забезпечення економічної та соціальної стійкості.

РМ характеризується широким впровадженням ІoT, який дозволяє з'єднати фізичні об'єкти з мережею Інтернет для автоматичного збору та передачі даних [12]. Це включає встановлення датчиків по всьому місту, які можуть спостерігати різні параметри, такі як рівень забруднення повітря, рух транспорту, стан інфраструктури та споживання енергії. Зібрані дані передаються до централізованих платформ обробки, де вони аналізуються за допомогою потужних обчислювальних ресурсів та аналітичних інструментів.

Однією з ключових особливостей «розумного міста» є інтеграція даних з різних джерел для забезпечення комплексного управління міськими системами (див. рисунок 1.1). Наприклад, дані про транспорт можуть використовуватися для оптимізації руху громадського транспорту, зменшення заторів та покращення доступності для мешканців. Дані про споживання енергії можуть допомогти зменшити витрати та впровадити енергоефективні рішення [13].



Рисунок 1.1 – Складові «розумного міста»

Іншою важливою складовою «розумного міста» є інтерфейси користувача, які забезпечують мешканцям доступ до інформації про стан міського середовища та дозволяють взаємодіяти з міськими службами [14]. Це можуть бути мобільні застосунки, веб-портали, інформаційні табло та інші засоби, що роблять дані доступними для широкого загалу. Такі сервіси сприяють залученню громадськості до процесів прийняття рішень та підвищують прозорість роботи міських адміністрацій.

Енергоефективність та стійкість також є важливими аспектами «розумного міста». Використання відновлюваних джерел енергії, «розумних систем управління енергоспоживанням» та технологій зменшення викидів парникових газів сприяють зменшенню негативного впливу на навколишнє середовище. Наприклад, «розумні мережі» можуть автоматично регулювати споживання енергії в залежності від потреби, а «розумні будівлі» – оптимізувати використання ресурсів для забезпечення комфортних умов проживання [15].

Система «розумного міста» також включає управління водопостачанням та водовідведенням, яке здійснюється за допомогою датчиків, що перевіряють якість води та виявляють витіки. Це дозволяє зменшити витрати води та забезпечити її ефективне використання. У сфері управління відходами, «розумні контейнери» для сміття повідомляють про свою наповненість, що дозволяє оптимізувати маршрути збору відходів та стимулювати сортування сміття [16]. «Розумна охорона здоров'я» є ще одним напрямом, де впроваджуються інноваційні технології. Телемедицина, електронні медичні картки та датчики для спостереження здоров'я пацієнтів дозволяють покращити надання медичних послуг та забезпечити своєчасну допомогу [17]. Таким чином, РМ є комплексною екосистемою, що базується на використанні новітніх технологій для забезпечення сталого розвитку та підвищення якості життя мешканців. Реалізація цієї концепції вимагає інтеграції різних компонентів та тісної взаємодії між державними установами, приватним сектором та громадськістю. Це сприяє створенню інноваційного міського середовища, яке здатне швидко адаптуватися до змін та відповідати на виклики сучасності.

### 1.1.2. Компоненти інформаційних систем «розумних міст»

Інформаційні системи «розумних міст» складаються з різних компонентів, які працюють разом для забезпечення ефективного управління міськими ресурсами та підвищення якості життя мешканців. Основними компонентами є датчики, мережі передачі даних, централізовані платформи обробки даних, аналітичні інструменти та інтерфейси користувача [18].

Датчики є основою для збору даних у «розумному місті». Вони встановлюються у різних частинах міста та збирають інформацію про різні параметри навколишнього середовища, інфраструктури та поведінки мешканців. Це можуть бути датчики якості повітря, які вимірюють рівні забруднення, датчики руху для спостереження транспортних потоків, датчики вологості та температури для контролю кліматичних умов, а також датчики споживання енергії та води [19]. Зібрані дані є основою для аналізу та прийняття обґрунтованих рішень щодо управління міськими системами.

Зібрані датчиками дані передаються через мережі передачі даних до централізованих платформ обробки. Ці мережі можуть включати дротові та бездротові технології, такі як оптоволоконні кабелі, Wi-Fi, 4G, 5G, LoRaWAN та інші [20]. Надійність і швидкість передачі даних є критично важливими для забезпечення своєчасного та точного аналізу інформації. Безперебійна робота мереж передачі даних забезпечує ефективну взаємодію між різними компонентами інформаційної системи.

Централізовані платформи обробки даних є серцем інформаційної системи «розумного міста». Вони отримують дані з мереж передачі, зберігають їх та здійснюють обробку. Використання потужних обчислювальних ресурсів дозволяє здійснювати аналіз великих обсягів даних у реальному часі. Це забезпечує можливість швидко реагувати на зміни в міському середовищі та приймати обґрунтовані рішення. Централізовані платформи можуть бути побудовані на базі хмарних технологій, що дозволяє забезпечити високу масштабованість та доступність сервісів [21].



Після збору та обробки даних використовуються аналітичні інструменти для виявлення тенденцій, прогнозування подій та прийняття рішень. Ці інструменти можуть включати методи машинного навчання, штучного інтелекту, аналізу великих даних (Big Data) та інші сучасні технології. Аналітичні інструменти дозволяють не тільки аналізувати поточні дані, але й прогнозувати майбутні події, що є критично важливим для управління міськими ресурсами та інфраструктурою. Наприклад, за допомогою прогнозування можна передбачити затори на дорогах та оптимізувати маршрути громадського транспорту [22].

Інтерфейси користувача забезпечують доступ до інформаційної системи для різних груп користувачів. Це можуть бути мобільні застосунки, веб-портали, інформаційні табло, інтерфейси для диспетчерських служб та інші засоби комунікації. Інтерфейси користувача дозволяють мешканцям міста отримувати інформацію про стан навколишнього середовища, транспортні потоки, споживання ресурсів, а також взаємодіяти з міськими службами. Для міських адміністрацій та служб реагування інтерфейси користувача забезпечують інструменти для спостереження та управління міськими системами, прийняття оперативних рішень та реагування на надзвичайні ситуації [23].

Ключовим аспектом ефективної роботи ІС «розумного міста» є тісна взаємодія між усіма компонентами. Давачі збирають дані, які передаються через мережі до централізованих платформ обробки. Після аналізу дані стають доступними через інтерфейси користувача, що дозволяє приймати обґрунтовані рішення та забезпечувати зворотний зв'язок. Взаємодія між компонентами створює єдину екосистему, яка забезпечує ефективне управління міськими ресурсами та підвищення якості життя мешканців.

Таким чином, компоненти ІС «розумних міст» є взаємопов'язаними елементами, які працюють разом для створення інноваційного та ефективного міського середовища. Вони забезпечують збір, передачу, обробку та аналіз даних, а також надають інструменти для взаємодії з користувачами, що дозволяє оптимізувати використання ресурсів, підвищити якість життя та забезпечити стійкий розвиток міста.

### 1.1.3. Роль внутрішнього середовища в концепції «розумного міста»

ВС відіграє важливу роль у концепції «розумного міста», оскільки від нього залежить комфорт, здоров'я та продуктивність мешканців. Система спостереження та управління внутрішнім середовищем включає в себе контроль за якістю повітря, температурою, вологістю та іншими параметрами, які впливають на загальний стан приміщень та благополуччя людей.

Якість повітря є одним із найважливіших факторів, що впливають на здоров'я та комфорт мешканців. В умовах «розумного міста» використовуються датчики для спостереження рівнів забруднення, таких як дрібнодисперсні частинки (PM2.5 і PM10), вуглекислий газ (CO<sub>2</sub>), оксиди азоту (NO<sub>x</sub>) та інші шкідливі речовини. Дані з датчиків дозволяють автоматично регулювати системи вентиляції та очищення повітря для забезпечення оптимальних умов у приміщеннях. Крім того, інформація про якість повітря може бути доступна мешканцям через мобільні застосунки або інші інтерфейси користувача, що дозволяє їм приймати обґрунтовані рішення щодо свого перебування в тих чи інших місцях [24].

Комфортна температура і вологість є критичними для створення здорових умов проживання та роботи. Системи управління кліматом, такі як опалення, вентиляція та кондиціонування повітря, можуть автоматично регулюватися на основі даних датчиків для підтримання комфортних умов. Це не тільки підвищує рівень комфорту, але й сприяє енергоефективності, оскільки системи працюють лише тоді, коли це необхідно, і на оптимальному рівні.

ВС «розумного міста» інтегрується з іншими міськими системами для забезпечення комплексного підходу до управління ресурсами та створення комфортних умов. Наприклад, дані про якість повітря можуть використовуватися для оптимізації роботи транспортних систем, зменшуючи забруднення в найбільш завантажених районах. Взаємодія з системами охорони здоров'я може забезпечити своєчасне виявлення та реагування на проблеми, пов'язані зі здоров'ям мешканців. А також датчики якості повітря та інші пристрої дозволяють стежити за викидами від промислових джерел, таких як електростанції та заводи,

щоб переконатися, що вони відповідають стандартам, встановленим державними органами, і відповідно скоригувати свою діяльність на свіжому повітрі. Однією з головних переваг спостереження якості повітря є те, що він допомагає нам переконатися, що повітря, яким ми дихаємо, є безпечним [25].

Забезпечення комфортного та здорового внутрішнього середовища має численні переваги для мешканців та міста в цілому. По-перше, це сприяє підвищенню рівня здоров'я та добробуту населення, зменшуючи ризик виникнення хронічних захворювань та покращуючи загальне самопочуття. По-друге, комфортні умови проживання та роботи підвищують продуктивність, що позитивно впливає на економічний розвиток міста. По-третє, ефективне управління ресурсами сприяє зменшенню енергоспоживання та впливу на навколишнє середовище, що є важливим для сталого розвитку міста.

Також зміни погодних умов, такі як збільшення частоти хвиль спеки та лісових пожеж, можуть впливати на рівень забруднюючих речовин у повітрі. Відстежуючи ці зміни, спостереження якості повітря може допомогти виявити вплив зміни клімату на якість повітря та вжити заходів для його пом'якшення.

## **1.2. Внутрішнє середовище «розумних приміщень»**

ВС «розумних приміщень» є важливою складовою концепції «розумного міста», зосередженою на створенні комфортних, здорових та енергоефективних умов для мешканців. У сучасних умовах зростання урбанізації та підвищення вимог до житлових і робочих умов, забезпечення оптимальних параметрів внутрішнього середовища стає надзвичайно важливим. Це включає контроль за вологою, температурою, домішками, твердими частинками, радіацією та газами, які можуть суттєво впливати на здоров'я та благополуччя людей [11].

Якість повітря є одним з найважливіших факторів, що впливають на здоров'я та комфорт мешканців. У «розумних приміщеннях» встановлюються давачі, які вимірюють рівні забруднення, такі як дрібнодисперсні частинки (PM2.5 і PM10), вуглекислий газ (CO<sub>2</sub>), леткі органічні сполуки (VOC) та інші

шкідливі речовини. Дані з датчиків передаються до центральної системи, яка автоматично регулює роботу систем вентиляції та очищення повітря. Це забезпечує підтримання оптимальних умов у приміщеннях, зменшує ризик захворювань, пов'язаних з поганою якістю повітря, та підвищує загальне самопочуття мешканців.

### **1.2.1. Параметри внутрішнього середовища**

Вологість є одним з ключових параметрів, що визначає комфорт та здоров'я в приміщенні. Занадто низька вологість може призвести до сухості шкіри, подразнення слизових оболонок та проблем з диханням. Висока вологість, з іншого боку, може сприяти розвитку цвілі, грибків та шкідливих бактерій, що також негативно впливає на здоров'я. В «розумних приміщеннях» використовуються датчики для постійного спостереження рівня вологості. Дані з датчиків передаються до системи управління кліматом, яка автоматично регулює роботу зволожувачів або осушувачів повітря для підтримання оптимального рівня вологості [26].

Температура є ще одним важливим параметром, який впливає на комфорт та здоров'я мешканців. Оптимальна температура сприяє продуктивності та загальному благополуччю. В «розумних приміщеннях» датчики температури дозволяють постійно спостерігати цей параметр, а системи опалення, вентиляції та кондиціонування повітря (HVAC) автоматично регулюють температуру для підтримання комфортних умов. Системи можуть адаптуватися до індивідуальних потреб мешканців, забезпечуючи персоналізовані рішення для кожного приміщення [26].

Домішки у повітрі, такі як пилок, шерсть тварин, аерозолі та інші алергени, можуть значно погіршити якість повітря в приміщенні та викликати алергічні реакції або проблеми з диханням. «Розумні системи» спостереження якості повітря використовують датчики для виявлення та вимірювання рівнів домішок. Дані з датчиків дозволяють системам очищення повітря автоматично активуватися

та фільтрувати повітря, видаляючи шкідливі домішки і забезпечуючи чисте повітря у приміщенні [27].

Тверді частинки, такі як дрібнодисперсні частинки (PM2.5 і PM10), можуть бути особливо шкідливими для здоров'я, проникаючи глибоко в легені та викликаючи респіраторні захворювання. У «розумних приміщеннях» встановлюються датчики для спостереження рівня твердих частинок. Ці дані використовуються для автоматичного регулювання систем вентиляції та очищення повітря, що дозволяє зменшити концентрацію шкідливих частинок та забезпечити безпечне середовище для мешканців [27].

Радіаційний фон також може впливати на здоров'я людей, хоча цей фактор рідше зустрічається у звичайних умовах. В «розумних приміщеннях» можуть бути встановлені датчики радіації, які контролюють рівень радіації у приміщенні. У разі виявлення підвищених рівнів, система може вжити відповідних заходів, таких як активація системи вентиляції або інформування мешканців про необхідність вжити запобіжних заходів.

Важливим аспектом якості повітря є концентрація шкідливих газів, таких як вуглекислий газ (CO<sub>2</sub>), чадний газ (CO), оксиди азоту (NO<sub>x</sub>), формальдегід та інші леткі органічні сполуки. Високі рівні цих газів можуть бути шкідливими для здоров'я та викликати різноманітні симптоми, від головного болю до серйозних респіраторних захворювань. Датчики для виявлення цих газів є невід'ємною частиною системи спостереження повітря у «розумних приміщеннях». Вони дозволяють вчасно виявляти небезпечні концентрації газів і автоматично активувати системи вентиляції або очищення повітря [27].

Всі ці параметри внутрішнього середовища інтегруються у єдину систему управління, що дозволяє здійснювати комплексний контроль за якістю повітря. Дані з різних датчиків збираються і аналізуються у реальному часі, що дозволяє системі автоматично регулювати роботу різних пристроїв для підтримання оптимальних умов. Це включає автоматичне включення та вимкнення систем вентиляції, кондиціонування, зволоження або осушення повітря, а також систем очищення повітря. Інтеграція з іншими міськими системами, такими як

енергетичні та транспортні мережі, дозволяє підвищити ефективність використання ресурсів та забезпечити стійкий розвиток.

Контроль за параметрами внутрішнього середовища «розумних приміщень» має численні переваги. По-перше, це забезпечує комфортні умови для проживання та роботи, що підвищує якість життя мешканців. По-друге, це сприяє покращенню здоров'я, зменшуючи ризик виникнення хронічних захворювань та поліпшуючи загальне самопочуття. По-третє, ефективне управління ресурсами сприяє енергоефективності, знижуючи витрати на енергоспоживання та зменшуючи вплив на навколишнє середовище.

Таким чином, ВС «розумних приміщень» відіграє ключову роль у забезпеченні комфортних, здорових та ефективних умов для мешканців. Використання сучасних технологій для спостереження та управління параметрами повітря сприяє підвищенню якості життя, економічній ефективності та сталому розвитку.

### **1.3. Веб-інтерфейс для інформаційних систем**

Веб-інтерфейс є критичним компонентом ІС, забезпечуючи ефективну та зручну взаємодію користувачів з системою. У контексті «розумних приміщень» веб-інтерфейс дозволяє користувачам контролювати різні аспекти внутрішнього середовища, отримувати дані в реальному часі, аналізувати історичні дані та приймати обґрунтовані рішення для забезпечення оптимальних умов проживання. Веб-інтерфейси є потужним інструментом, який робить складні системи доступними та зрозумілими для широкого кола користувачів.

Компоненти веб-інтерфейсу включають панель управління, сторінки спостереження, управління, аналізу та сповіщень. Панель управління є головним екраном, що відображає основну інформацію про стан внутрішнього середовища. Вона може містити графіки, таблиці та інші візуальні елементи, що дозволяють швидко оцінити поточний стан системи. Сторінка спостереження дозволяє детально відслідковувати всі параметри внутрішнього середовища в

реальному часі, включаючи карти приміщень з відображенням розташування давачів та їх показань. Сторінка управління надає інструменти для налаштування та управління системами клімат-контролю, де користувачі можуть задавати бажані параметри температури, вологості та інших характеристик. Сторінка аналізу містить інструменти для аналізу історичних даних, де користувачі можуть вибирати періоди часу, переглядати графіки змін параметрів та порівнювати дані для виявлення тенденцій. Сторінка сповіщень відображає повідомлення про події та аномалії, що виникають у системі, дозволяючи користувачам переглядати історію сповіщень та налаштовувати параметри оповіщення.

Для створення сучасного веб-інтерфейсу використовуються передові технології та інструменти розробки. Фронтенд розробляється за допомогою HTML, CSS та JavaScript. Популярні фреймворки, такі як React, Angular або Vue.js, дозволяють створювати інтерактивні та динамічні веб-інтерфейси. Бекенд забезпечує обробку запитів користувачів та взаємодію з БД за допомогою серверних технологій, таких як Express.js або Django. Бази даних, реляційні або нереляційні, використовуються для зберігання та управління даними. Інтерфейси програмування застосунків (API), такі як RESTful API або GraphQL, забезпечують зв'язок між фронтендом і бекендом [28].

Веб-інтерфейс забезпечує кілька ключових переваг для ІС «розумних приміщень». По-перше, він забезпечує доступність, дозволяючи користувачам отримувати доступ до системи з будь-якого пристрою, що має доступ до Інтернету. По-друге, інтуїтивно зрозумілий та зручний інтерфейс підвищує ефективність взаємодії користувачів з системою, дозволяючи швидко отримувати необхідну інформацію та здійснювати управління. По-третє, використання сучасних технологій дозволяє створювати швидкі та ефективні веб-інтерфейси, що забезпечують високу продуктивність та надійність. По-четверте, веб-інтерфейс дозволяє легко масштабувати систему, додаючи нові функції та можливості без значних витрат. По-п'яте, веб-інтерфейс може легко

інтегруватися з іншими системами та сервісами, забезпечуючи комплексний підхід до управління внутрішнім середовищем.

Таким чином, веб-інтерфейс є невід'ємною частиною сучасних ІС, включаючи системи спостереження та управління внутрішнім середовищем «розумних приміщень». Він забезпечує зручний доступ до інформації, ефективно управління та аналіз даних, сприяючи підвищенню якості життя мешканців та ефективності використання ресурсів.

### **1.3.1. Дослідження існуючих систем спостереження внутрішнього середовища «розумних приміщень»**

Різні існуючі системи для спостереження внутрішнього середовища мають вагомні недоліки, такі як висока вартість, складне налаштування, непереносимість інформації тощо. У цих роботах системи часто мають обмеження, такі як обмежений обсяг, зосереджений головним чином на загальному спостереженні навколишнього середовища без особливого акценту на параметрах якості повітря [29] та недостатні методології для перевірочних вимірювань [29]. Крім того, інтегровані системи «розумних приміщень» на основі ІоТ більше зосереджені на сповіщеннях безпеки, а не на всебічному аналізі якості повітря [30], а проблеми калібрування датчиків, управління даними та складності реалізації не були повністю вирішені в рішеннях спостереження «розумних міст» [31]. У кваліфікаційній роботі пропонується система спостереження забруднення повітря в «розумних приміщеннях», яка подолає більшість цих проблем.

Найголовнішим нововведенням є те, що системою пристроїв може керувати уповноважений персонал, який має відповідний доступ. Розроблена система вводить ролі з певними привілеями на основі рівня доступу, гарантуючи, що користувачі можуть виконувати відповідні завдання ефективно та безпечно. Управління системою здійснюється за допомогою веб-інтерфейсу, через який можна отримувати актуальні оновлення рівня забруднення приміщення з відповідним рівнем доступу.



У системі використовується IoT, що дозволяє застосовувати новітні технології. Система здатна виявляти основні забруднювачі повітря, такі як CO, CO<sub>2</sub>, SO<sub>2</sub> та інші. Також, IoT дозволяє отримувати доступ до даних з віддаленого місця та зберігати їх у базі даних, яку можна отримати в будь-який час для подальшої обробки. Оброблена інформація подається у формі різних діаграм, які легко зчитуються та надають інформацію в повному обсязі.

Цей інноваційний підхід забезпечує високу точність, надійне керування даними та гнучкість у різних середовищах, таким чином усуваючи обмеження попередніх розробок [29-31].

#### **1.4. Висновок до першого розділу**

У першому розділі кваліфікаційної роботи освітнього рівня “Бакалавр” проаналізовано предметну область. Сформовано перелік вимог до веб-інтерфейсу інформаційної системи спостереження внутрішнього середовища “розумних приміщень”. Виконано детальний аналіз концепції «розумного міста», його складових та ролі внутрішнього середовища в цій концепції. Описано основні характеристики якості повітря, які спостерігаються в «розумних приміщеннях», такі як вологість, температура, домішки, тверді частинки, радіація та гази. Визначено основні функції, компоненти та технології, необхідні для створення ефективного та зручного веб-інтерфейсу, що забезпечить користувачам можливість спостереження, управління, аналізу даних та отримання сповіщень у реальному часі.

## РОЗДІЛ 2. ПРОЕКТНА ЧАСТИНА

### 2.1. Розробка архітектури веб-інтерфейсу

Архітектура даного веб-інтерфейсу для спостереження якості повітряного середовища, яка представлена на рисунку 2.1, відображає повний цикл обробки даних від збору з датчиків до відображення користувачеві. Ця архітектура побудована на стеку MERN, який включає MySQL, Express, React і Node.js, Redux, який використовується для зберігання стану всередині веб-інтерфейсу, також використовує TailwindCSS для швидкого та ефективного стилювання інтерфейсу користувача.

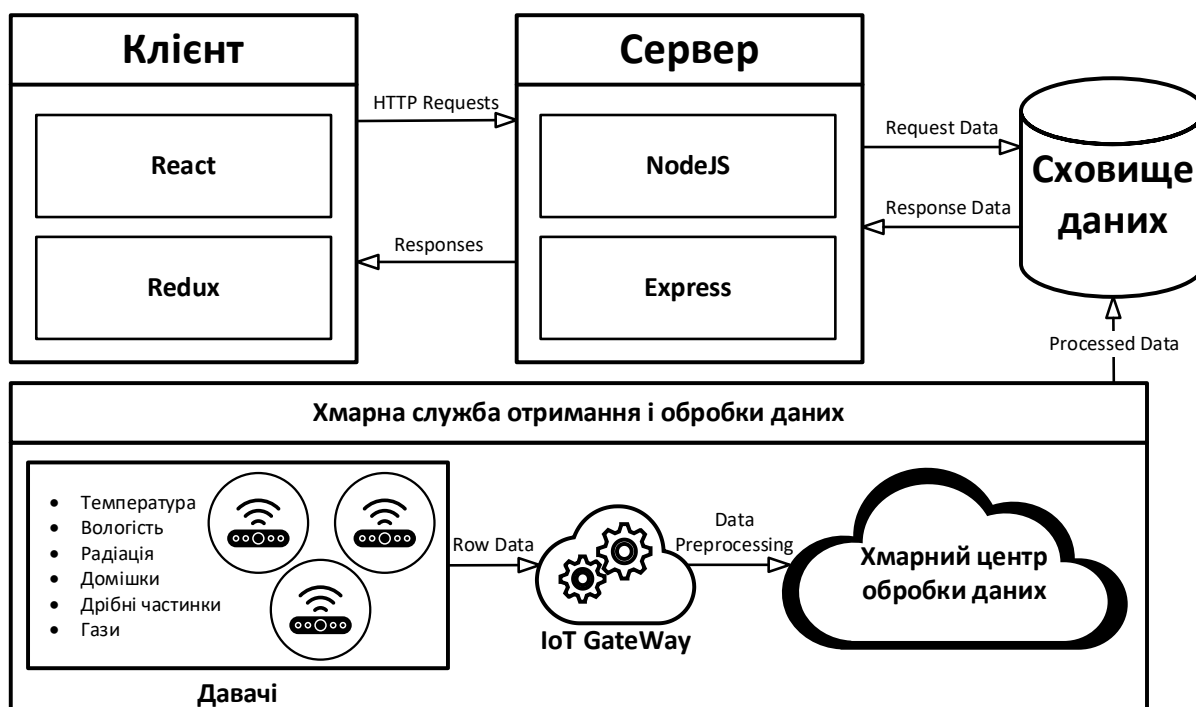


Рисунок 2.1 – Діаграма архітектури веб-інтерфейсу інформаційної системи спостереження внутрішнього середовища «розумних приміщень»

На клієнтському боці веб-інтерфейс побудований за допомогою React і Redux. React є популярною бібліотекою для створення інтерфейсів користувача, яка дозволяє створювати динамічні і інтерактивні компоненти. Кожен компонент React відповідає за відображення певної частини інтерфейсу і може

оновлюватися в реальному часі при зміні стану. Redux, в свою чергу, служить для управління станом веб-інтерфейсу. Він дозволяє централізовано зберігати стан всіх компонентів і забезпечує узгодженість даних між ними. Це особливо важливо в веб-інтерфейсах, де стан може змінюватися в різних частинах інтерфейсу і потребує синхронізації.

Клієнтська частина взаємодіє з сервером через HTTP-запити. Коли користувач виконує певні дії, такі як перегляд даних про якість повітря або налаштування параметрів пристроїв, клієнт надсилає відповідні запити на сервер. Серверна частина веб-інтерфейсу реалізована на Node.js з використанням фреймворку Express. Node.js забезпечує асинхронну обробку запитів, що дозволяє ефективно обробляти велику кількість одночасних підключень. Express є легким і гнучким фреймворком для створення серверних застосунків на базі Node.js, який дозволяє легко обробляти маршрутизацію запитів і керування middleware. Сервер обробляє HTTP-запити від клієнта і взаємодіє зі сховищем даних для отримання або збереження необхідної інформації.

Сервер приймає HTTP-запити від клієнта, обробляє їх і взаємодіє з БД для отримання або збереження необхідної інформації. У цій архітектурі використовується реляційна БД MySQL. MySQL є потужною системою керування базами даних, яка підтримує складні запити, транзакції і забезпечує надійне зберігання даних. Сервер надсилає SQL-запити до БД для отримання потрібних даних і формує відповіді, які відправляє назад клієнту.

Дані в базу даних надходять від давачів, які розташовані в різних приміщеннях і збирають інформацію про параметри повітряного середовища, такі як температура, вологість, рівень забруднення і концентрація газів. Ці давачі підключені до IoT шлюзу (IoT Gateway), який виконує попередню обробку зібраних даних. Попередня обробка може включати фільтрацію, нормалізацію і агрегацію даних, що дозволяє зменшити обсяг переданих даних і підвищити ефективність їх подальшої обробки. Алгоритм роботи хмарної служби отримання і обробки даних зображений на рисунку 2.2.

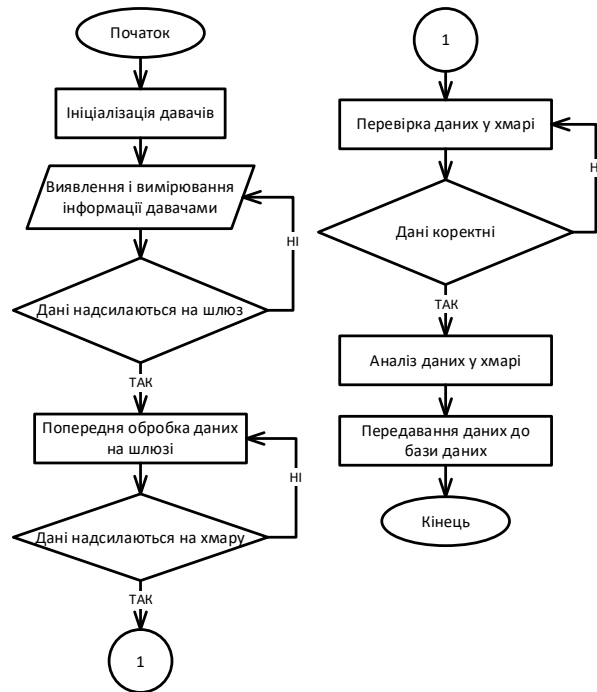


Рисунок 2.2 – Алгоритм хмарної служби отримання і обробки даних

Після попередньої обробки дані надсилаються з IoT шлюзу до хмарного центру обробки даних. Хмарний центр обробки даних виконує кілька ключових функцій. По-перше, він здійснює валідацію даних, перевіряючи їх на коректність і повноту. Це дозволяє виявляти і відкидати некоректні або неповні дані, забезпечуючи високу якість аналізу. По-друге, хмарний центр проводить аналіз даних, включаючи розрахунки, статистичний аналіз і виявлення аномалій. Результати аналізу можуть бути використані для генерування звітів, відправки сповіщень або інших дій.

Після завершення обробки хмарний центр надсилає оброблені дані до бази даних, де вони зберігаються і стають доступними для запитів з боку серверної частини веб-інтерфейсу. Сервер взаємодіє з БД, отримує необхідні дані і формує відповіді на запити клієнта. Це дозволяє користувачам отримувати актуальну інформацію про якість повітря в реальному часі, а також переглядати історичні дані і налаштовувати параметри пристроїв.

Інтерфейс користувача, створений за допомогою React і стилізований за допомогою TailwindCSS, забезпечує зручне і інтуїтивно зрозуміле відображення даних. TailwindCSS є утилітарним CSS-фреймворком, який дозволяє швидко і

ефективно створювати стилі для компонентів інтерфейсу. Це значно прискорює розробку і забезпечує узгоджений вигляд всього веб-інтерфейсу.

Таким чином, архітектура веб-інтерфейсу для спостереження якості повітряного середовища забезпечує повний цикл обробки даних від збору і попередньої обробки на давачах і IoT шлюзі, через валідацію та аналіз у хмарному центрі обробки, до зберігання в базі даних і відображення користувачеві через інтерфейс, побудований на стеку MERN. Кожен компонент цієї архітектури виконує важливу роль у забезпеченні ефективної та надійної роботи всього веб-інтерфейсу, що дозволяє користувачам отримувати точну і своєчасну інформацію про якість повітряного середовища.

## **2.2. Розробка структури бази даних веб-інтерфейсу**

БД, яка була створена для роботи веб-інтерфейсу інформаційної системи спостереження внутрішнього середовища «розумних приміщень» призначена для обробки даних, які збираються різноманітними давачами в рамках веб-інтерфейсу спостереження якості повітряного середовища. Вона містить інформацію про користувачів системи, їхні ролі, місця розташування давачів, типи платформ, до яких підключені давачі, а також дані давачів та їх історію. Це дозволяє відстежувати якість повітря в режимі реального часу та аналізувати історичні дані для виявлення трендів та аномалій.

БД містить декілька основних таблиць, що відповідають за різні аспекти системи. Діаграма структури БД зображена на рисунку 2.3. Таблиця "User" зберігає інформацію про користувачів системи. Кожен користувач має унікальний ідентифікатор (userID), ім'я користувача (username), пароль (password), електронну адресу (email) та роль (roleID). Роль користувача визначається зовнішнім ключем, який посилається на таблицю "Role". У таблиці "Role" зберігається інформація про різні ролі користувачів, зокрема унікальний ідентифікатор ролі (roleID) та назва ролі (roleName).

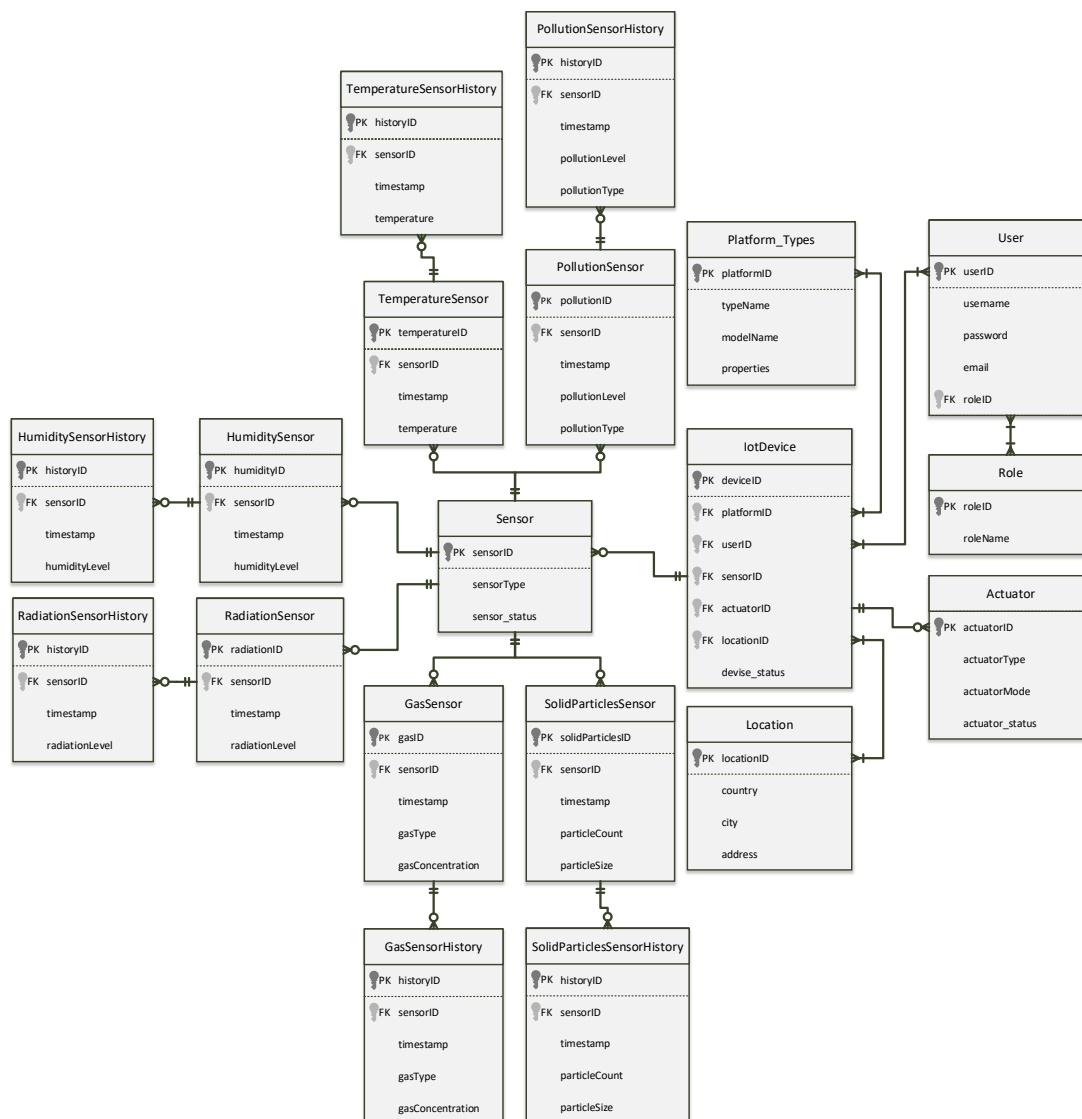


Рисунок 2.3 – Діаграма структури бази даних

Таблиця "Location" зберігає дані про місця розташування датчиків та пристроїв. Вона містить унікальний ідентифікатор місця розташування (locationID), назву країни (country), міста (city) та адресу (address). Ця інформація дозволяє точно визначити, де саме розташований кожен датчик або пристрій.

Таблиця "Platform\_Types" містить дані про різні типи платформ, до яких підключені датчики та пристрої. Вона містить унікальний ідентифікатор платформи (platformID), назву типу платформи (typeName), назву моделі платформи (modelName) та додаткові властивості у форматі JSON (properties). Ця інформація важлива для розуміння, які платформи використовуються та які їх характеристики.

Таблиця "IoTDevice" зберігає інформацію про IoT-пристрої, які підключені до системи. Вона містить унікальний ідентифікатор пристрою (deviceID), тип пристрою (deviceType), ідентифікатор користувача, який використовує пристрій (userID), ідентифікатор датчика, підключеного до пристрою (sensorID), ідентифікатор актуатора, підключеного до пристрою (actuatorID), ідентифікатор місця розташування пристрою (locationID), ідентифікатор платформи, до якої підключений пристрій (platformID), та статус пристрою (device\_status). Кожен IoT-пристрій асоціюється з конкретним користувачем, датчиком, актуатором, місцем розташування та платформою.

Таблиця "Actuator" містить інформацію про актуатори, які використовуються в системі. Вона містить унікальний ідентифікатор актуатора (actuatorID), тип актуатора (actuatorType) та статус актуатора (actuator\_status). Ця інформація допомагає визначити, які актуатори доступні та в якому вони стані.

Таблиця "Sensor" зберігає дані про датчики, які використовуються для збору інформації про якість повітря. Вона містить унікальний ідентифікатор датчика (sensorID), тип датчика (sensorType) та статус датчика (sensor\_status). Кожен датчик може мати один або кілька типів характеристик, які зберігаються в окремих таблицях.

Для кожного типу датчика створена окрема таблиця характеристик. Наприклад, таблиця "HumiditySensor" містить дані про датчики вологості. Вона містить унікальний ідентифікатор датчика вологості (humidityID), ідентифікатор датчика (sensorID), рівень вологості (humidityLevel) та час вимірювання (timestamp). Аналогічно, таблиця "TemperatureSensor" зберігає дані про датчики температури, таблиця "PollutionSensor" - про датчики домішок, таблиця "SolidParticlesSensor" - про датчики твердих частинок, таблиця "RadiationSensor" - про датчики радіації, а таблиця "GasSensor" - про датчики газу.

Крім основних таблиць датчиків, БД містить також історичні таблиці для зберігання даних вимірювань з датчиків у різний час. Наприклад, таблиця "HumiditySensorHistory" зберігає історичні дані датчиків вологості. Вона містить унікальний ідентифікатор запису історії (historyID), ідентифікатор датчика

вологості (sensorID), час вимірювання (timestamp) та рівень вологості (humidityLevel). Подібні таблиці існують для кожного типу давача, що дозволяє зберігати історичні дані для подальшого аналізу.

Відносини між суб'єктами БД реалізовані за допомогою зовнішніх ключів. Наприклад, відношення між таблицями "User" та "Role" є відношенням багато до багатьох, де кожен користувач має одну або багато ролей, і кожна роль може бути присвоєна багатьом користувачам. Відношення між таблицями "IoTDevice" та "User" також є відношенням багато до багатьох, де кожен IoT-пристрій асоціюється з одним або декількома користувачами. Аналогічно, кожен IoT-пристрій може мати один або декілька давачів, один актуаторів, але розташованих в одному місці та підключений до однієї платформи.

Кожен давач може мати одну або кілька характеристик, які зберігаються в окремих таблицях, таких як "HumiditySensor", "TemperatureSensor" та інші. Історичні таблиці зберігають дані вимірювань для кожного типу давача, що дозволяє зберігати та аналізувати історичні дані про якість повітря в разі заміни старих давачів.

Ця БД забезпечує ефективне збирання, зберігання та аналіз даних про якість повітря, що дозволяє відстежувати стан навколишнього середовища, виявляти тренди та аномалії, а також приймати обґрунтовані рішення для покращення якості повітря.

### **2.3. Діаграми прецедентів ролей в системі**

Діаграма прецедентів (use case diagram) в веб-інтерфейсі для спостереження якості повітряного середовища демонструє різні можливості та дії, які можуть виконувати користувачі в залежності від їхніх ролей. Кожна роль має свій набір можливостей, що забезпечують певні функціональні можливості в системі.

У веб-інтерфейсі для спостереження якості повітряного середовища передбачені різні ролі користувачів, кожна з яких має свої унікальні можливості



та привілеї. Ці ролі включають: Guest (гостя), User (користувача), Admin (адміністратора), Moder (технічного працівника) та Owner (власника девайсів). Нижче наведено детальний опис кожної ролі та їх можливостей.

Ролі в системі мають ієрархічну структуру (див. рисунок 2.4). Гість є базовою роллю, яку наслідує користувач. Роль користувача, в свою чергу, унаслідуються трьома іншими ролями: адміністратором, технічним працівником та власником. Це означає, що адміністратори, технічні працівники та власники мають всі можливості користувача, а також додаткові привілеї, специфічні для кожної з цих ролей.

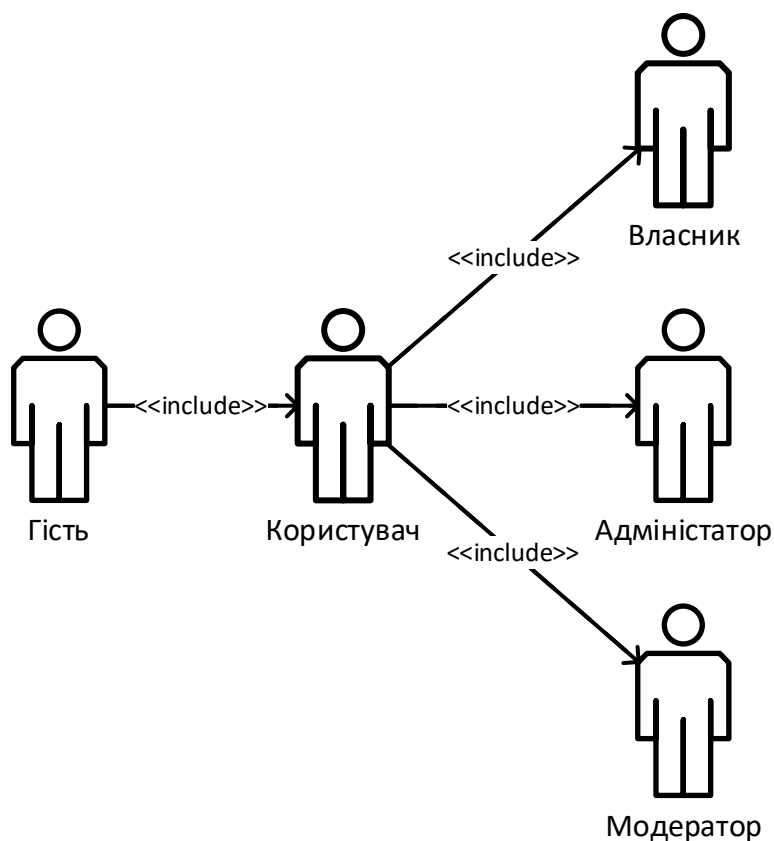


Рисунок 2.4 – Ієрархічна структура ролей в системі

Систему побудовано таким чином, що кожна наступна роль наслідує права та привілеї попередньої. Почнемо з базової ролі – гість. Роль гостя є відправною точкою для всіх користувачів системи. Незважаючи на обмежені можливості, ця роль дозволяє отримати базове уявлення про функціонал системи та доступні дані (див. рисунок 2.5).



Рисунок 2.5 – Діаграма прецедентів ролі гість інформаційної системи спостереження внутрішнього середовища «розумних приміщень»

Має найменший набір прав та привілеїв у системі. Він може ознайомитися з інформацією про веб-інтерфейс за допомогою головного меню, переглянути список приміщень які є у відкритому доступі, наприклад лекційні аудиторії загальнодоступні приміщення чи локації. Цей список відкриває можливість перегляду інформацію про окреме приміщення, дає характеристики повітряного середовища з показниками датчиків, також історичні показники якості повітряного середовища, які дають можливість відслідковувати і спостерігати зміну цих параметрів та інформацію про встановлені датчики показників якості повітряного середовища. Відповідно в цій інформації є список встановлених датчиків та перелік їхніх параметрів. Історичні дані показників дають змогу вибрати певний набір історичних даних відповідно до вибраного часового діапазону.

Наступна роль у ієрархії – це користувач. Користувачі мають додаткові можливості в порівнянні з гостями, що дозволяє їм більш активно взаємодіяти із системою. Користувачі можуть не тільки переглядати дані, але й отримувати доступ до більш детальної інформації та виконувати певні дії, які недоступні для гостей (див. рисунок 2.6).

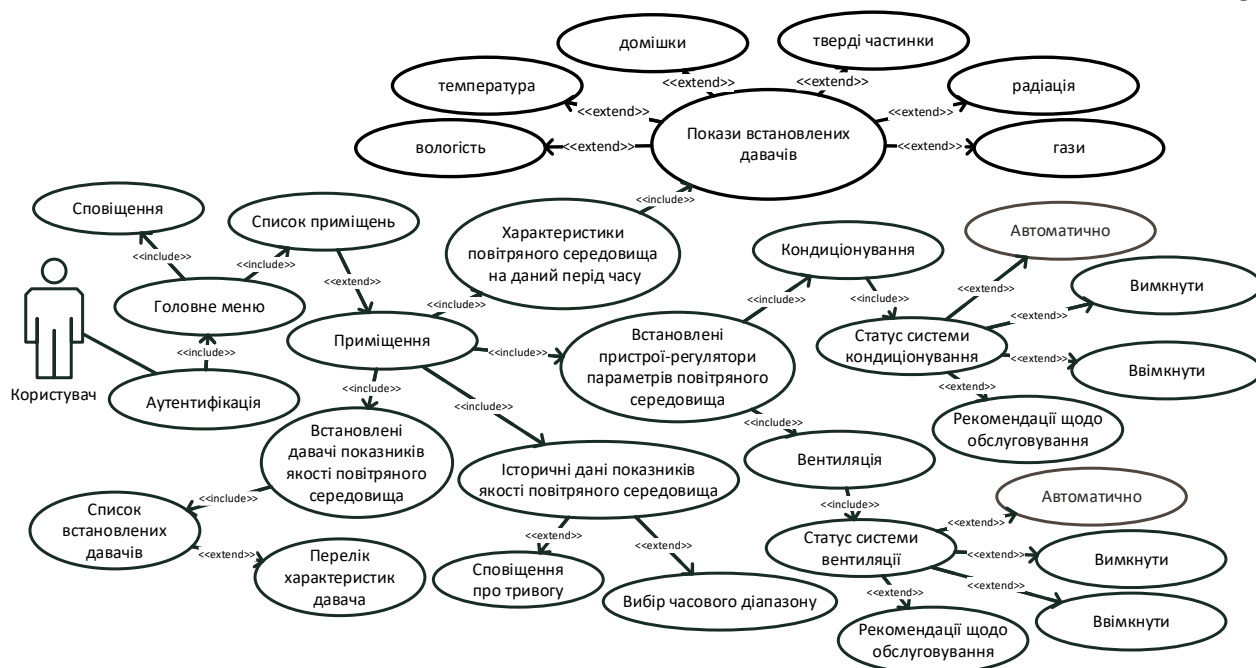


Рисунок 2.6 – Діаграма прецедентів ролі користувач інформаційної системи спостереження внутрішнього середовища «розумних приміщень»

Користувач інформаційної системи для спостереження та регулювання показників якості внутрішнього середовища «розумних приміщень» має більше можливостей, ніж гість. Він може виконувати всі дії, доступні гостю, а також додатково пройшовши процес автентифікації може переглянути перелік сповіщень або скористатися головним меню. В головному меню доступна інформація про веб-інтерфейс інформаційної системи та список приміщень які доступні для спостереження.

Список приміщень містить характеристики повітряного середовища, встановлені в них пристрої-регулятори параметрів повітряного середовища, історичні дані показників якості повітряного середовища для кожного приміщення. В даному випадку використовується посилання на історичні дані показників, які дають змогу вивести сповіщення про тривогу або здійснити вибір потрібного часового діапазону. Також список приміщень містить інформацію про встановленні датчі якості повітряного середовища. Це посилання містить список встановлених датчиків і перелік їх характеристик окремого датчика. Також характеристики повітряного середовища дають змогу переглянути покази встановлених датчиків, відповідно можна ознайомитися з показами вологості,

температури, домішок, твердих частинок, радіації або газів які можуть бути в повітряному середовищі.

Окремо для регулювання повітряного середовища «розумних приміщень» використовуються системи вентиляції та кондиціонування. Їхній статус можна переглянути за допомогою відповідних вкладень і перемкнути в автоматичний або ручний режим роботи. Системи вентиляції і кондиціонування мають подібні функціональні набори керування, тобто автоматичне або ручне вимкнення і ввімкнення, а також рекомендації щодо обслуговування. Дана система може використовуватися як рекомендаційна, коли згідно сигналів та показів систем кондиціонування або вентиляції можуть бути спродуковані рекомендації щодо звернень у сервісні центри для подальшого обслуговування цих систем. Далі у ієрархії – роль адміністратора, яка включає всі привілеї користувача та додає адміністративні можливості (див. рисунок 2.7).

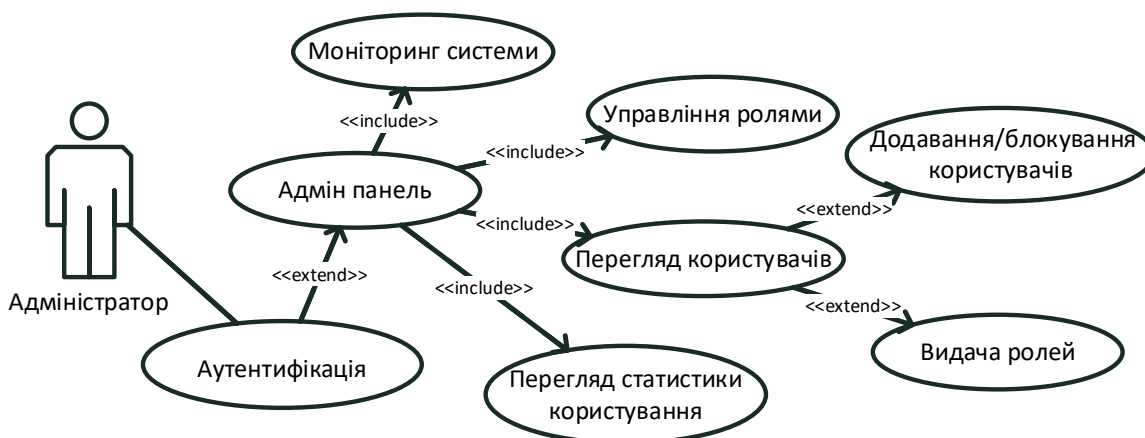


Рисунок 2.7 – Діаграма прецедентів ролі адміністратор інформаційної системи спостереження внутрішнього середовища «розумних приміщень»

Адміністратор інформаційної системи для супроводу внутрішнього середовища «розумних приміщень» має найбільше повноважень у системі. Він може виконувати всі дії, доступні користувачам, і додатково після автентифікації має змогу за допомогою головного меню переглянути інформацію про веб-інтерфейс, переглянути список приміщень, переглянути перелік сповіщень, або скористатися адміністративною панеллю.

Адміністративна панель дає змогу ознайомитися з статистикою використання веб-інтерфейсу і переглянути користувачів даної системи. Додатково може видавати ролі іншим користувачам, додавати або блокувати користувачів, переглядати статистику користування системою, налаштовувати давачі та актуатори, управляти ролями, включаючи створення, зміну та видалення ролей, а також здійснювати спостереження системи, включаючи мережеві налаштування та продуктивність.

Наступною роллю є модератор, який поєднує деякі адміністративні та технічні обов'язки (див. рисунок 2.8).



Рисунок 2.8 – Діаграма прецедентів ролі модератор інформаційної системи спостереження внутрішнього середовища «розумних приміщень»

Модератор має деякі адміністративні повноваження, а також виконує технічні обов'язки. Він може виконувати всі дії, доступні користувачам, і додатково додавати або блокувати користувачів, переглядати статистику користування, налаштовувати давачі та актуатори, надавати технічну підтримку користувачам, здійснювати моніторинг технічного стану системи, виконувати технічне обслуговування системи та девайсів, а також відладжувати систему.

Наступною роллю є власник девайсів. Це роль, орієнтована на адміністраторське управління лише власними пристроями (див. рисунок 2.9).

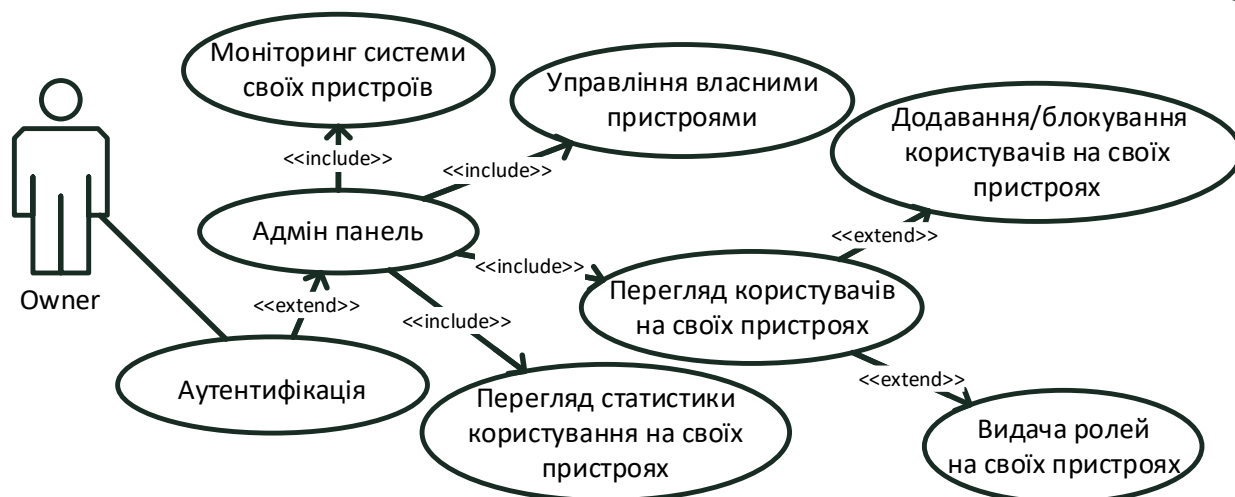


Рисунок 2.9 – Діаграма власник ролі модератор інформаційної системи спостереження внутрішнього середовища «розумних приміщень»

Власник девайсів має можливості, орієнтовані на управління власними пристроями. Вони можуть виконувати всі дії, доступні користувачам, і додатково додавати або блокувати користувачів для своїх пристроїв, переглядати статистику користування лише для своїх пристроїв, налаштовувати датчики та актуатори для своїх пристроїв, управляти власними пристроями (додавати, налаштовувати та видаляти) та видавати ролі для користувачів своїх пристроїв.

Таким чином, додаток забезпечує багатий функціонал для різних категорій користувачів, дозволяючи їм ефективно управляти та спостерігати якість повітряного середовища у приміщеннях. Ролі користувачів розподілені таким чином, щоб кожен мав доступ до необхідних інструментів та інформації відповідно до своїх повноважень і відповідальностей.

#### 2.4. Аналітичне опрацювання даних системи

Аналітичне опрацювання відіграє ключову роль у забезпеченні ефективного функціонування системи спостереження за якістю повітряного середовища. Модератор має доступ до розширених функцій, які дозволяють не лише слідкувати за станом системи, але й аналізувати її роботу, забезпечуючи підтримку користувачів і технічне обслуговування. Це надає можливість

оперативно виявляти проблеми, підвищувати ефективність роботи пристроїв та оптимізувати використання ресурсів.

Основне завдання модератора полягає у відслідковуванні стану пристроїв, аналізі ефективності їх використання та виявленні можливостей для оптимізації. Система, яка використовується для цих цілей, надає широкий спектр корисних функцій, що допомагають забезпечити ефективне управління та експлуатацію всіх компонентів «розумного приміщення».

Система дозволяє зібрати й опрацювати велику кількість даних про кожен підключений пристрій, включаючи такі аспекти, як навантаженість інтерфейсів, ефективність використання ресурсів, частота використання та інші параметри. Наприклад, кожен пристрій, такий як Arduino UNO, має певну кількість цифрових, аналогових та інших інтерфейсів. Ці дані зберігаються в системі за замовчуванням і автоматично підтягуються, коли модератор додає новий пристрій. Під час додавання пристрою модератор вносить інформацію про кількість інтерфейсів, що використовуються, що дозволяє системі точно фіксувати рівень навантаження на кожен з інтерфейсів.

Завдяки цим даним, система може формувати аналітичні звіти, які надають модератору детальну інформацію про використання ресурсів кожного пристрою. Ці звіти включають в себе дані про навантаженість інтерфейсів, їх використання в реальному часі, а також історичні дані, що дозволяють оцінити динаміку використання пристроїв. Модератор може використовувати ці звіти для прийняття обґрунтованих рішень щодо оптимізації системи, наприклад, для перерозподілу навантаження між пристроями або для виявлення та усунення вузьких місць у використанні інтерфейсів.

Крім того, система дозволяє модератору відстежувати ефективність використання пристроїв на основі кількості інтерфейсів, що використовуються, та їх завантаженості. Це дозволяє визначити, наскільки доцільним є використання того чи іншого пристрою в конкретних умовах, і чи може бути доцільно замінити або доповнити його іншим пристроєм.

Додатково, система дозволяє зберігати та аналізувати дані про ефективність використання пристроїв у різних сценаріях. Це допомагає модератору не тільки зрозуміти поточний стан системи, але й прогнозувати майбутні потреби та планувати відповідні заходи для забезпечення її ефективної роботи. Наприклад, якщо аналіз даних показує, що з певним типом пристроїв виникають постійні проблеми, модератор може вирішити замінити ці пристрої на більш ефективні аналоги або внести корективи у конфігурацію системи.

Таким чином, система надає модератору всі необхідні інструменти для комплексного аналізу та оптимізації роботи «розумних приміщень». Це включає можливість збору й аналізу детальних даних про кожен пристрій, формування аналітичних звітів, відстеження навантаженості інтерфейсів та ефективності використання пристроїв. Завдяки цим можливостям, модератор може забезпечити стабільну та ефективну роботу системи, а також приймати обґрунтовані рішення щодо її розвитку та оптимізації.

## **2.5. Висновок до другого розділу**

В другому розділі кваліфікаційної роботи освітнього рівня “Бакалавр” виконано проектування та розробку ключових компонентів веб-інтерфейсу інформаційної системи спостереження внутрішнього середовища «розумних приміщень». Спроектовано архітектуру системи з використанням стеку MERN, що забезпечує ефективну взаємодію між клієнтом та сервером, а також надійне зберігання даних. Детально розроблено структуру БД, що підтримує зберігання і обробку даних з різноманітних дачив у режимі реального часу. Діаграми прецедентів ролей у системі дозволяють зрозуміти функціональні можливості та права доступу різних користувачів. Виконано аналітичне опрацювання даних системи, яке забезпечує аналіз ефективності використання пристроїв та дозволяє отримувати дані щодо навантаженості системи. Усі ці аспекти інтегровані у єдиний веб-інтерфейс, що забезпечує зручний та інтуїтивно зрозумілий доступ користувачів до функціональності системи.



## РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

Веб-інтерфейс є основним інструментом взаємодії користувачів з інформаційними системами, що дозволяє здійснювати спостереження та управління різними параметрами внутрішнього середовища «розумних приміщень». Завдяки веб-інтерфейсу користувачі можуть отримувати доступ до даних з будь-якого пристрою, що має підключення до Інтернету, а також взаємодіяти з системою в режимі реального часу. Це особливо важливо для підтримки оптимальних умов проживання та роботи у приміщеннях, де якість повітря, температура, вологість та інші показники мають безпосередній вплив на комфорт і здоров'я мешканців.

Основні функції веб-інтерфейсу включають відображення актуальної інформації про стан внутрішнього середовища, управління системами вентиляції та кондиціонування, аналіз історичних даних та сповіщення про перевищення критичних показників. Веб-інтерфейс повинен бути інтуїтивно зрозумілим, зручним у використанні та забезпечувати швидкий доступ до необхідної інформації. З усіма розробленими сторінками веб-інтерфейсу можна ознайомитися в додатку А.

### 3.1. Розробка веб-інтерфейсу

Користувацька панель є важливою складовою будь-якої інформаційної системи, що дозволяє керувати користувачами та їхніми ролями. На прикладі зображення адміністраторської панелі (див рисунок 3.1) можна побачити, що ця панель включає такі дані:

- Імена користувачів
- Дати створення користувачів
- Ролі користувачів
- Дати останнього оновлення даних користувачів
- Опцію зміни ролі користувача

## Основні функції адміністративної панелі:

Відображення списку користувачів: Перш за все, адміністративна панель повинна забезпечувати зручне відображення списку всіх користувачів системи. Це включає їхні імена, дати створення акаунтів, поточні ролі та дати останнього оновлення. Така інформація дозволяє адміністраторам швидко орієнтуватися у системі та приймати необхідні управлінські рішення.

Редагування ролей користувачів: Однією з ключових функцій панелі є можливість зміни ролей користувачів. Адміністратори можуть змінювати ролі користувачів відповідно до їхніх обов'язків та повноважень. Це забезпечує гнучкість у керуванні доступом до різних функцій системи.

Фільтрація та пошук: Для зручності роботи з великою кількістю користувачів, панель повинна мати можливість фільтрації та пошуку. Це дозволяє швидко знаходити потрібного користувача за ім'ям або іншими параметрами.

Безпека та автентифікація: Важливою складовою є забезпечення безпеки даних користувачів. Адміністративна панель повинна бути захищена від несанкціонованого доступу, а всі операції повинні бути записані у лог-файли для аудиту.

The screenshot displays the 'Користувачі' (Users) management page in the SmartCity system. The interface includes a sidebar with navigation options (СИСТЕМА, Dashboard, Message, Users, Statistic, Report, Settings) and a top navigation bar with 'Home', 'Rooms', and 'Notification'. The main content area shows a table of users with the following data:

Ім'я	Дата Створення	Ролі	Дата Оновлення	Видача ролі
Cameron Williamson	01 March, 2024	Модератор	01 March, 2024	Виберіть роль
Savannah Nguyen	02 March, 2024	Користувач	02 March, 2024	Виберіть роль
Darlene Robertson	03 March, 2024	Власник	03 March, 2024	Модератор
Leslie Alexander	04 March, 2024	Власник	04 March, 2024	Адміністратор
Albert Flores	05 March, 2024	Користувач	05 March, 2024	Користувач
Dianne Russell	06 March, 2024	Користувач	06 March, 2024	Виберіть роль
Robert Fox	07 March, 2024	Користувач	07 March, 2024	Виберіть роль
Leslie Alexander	08 March, 2024	Модератор	08 March, 2024	Виберіть роль
Darrell Steward	09 March, 2024	Адміністратор	09 March, 2024	Виберіть роль
Arlene McCoy	10 March, 2024	Користувач	10 March, 2024	Виберіть роль

The table also includes a 'Загальна кількість' (Total count) of 20 users and a 'Підтвердити' (Confirm) button at the bottom right.

Рисунок 3.1 –Сторінка налаштування користувачів в панелі керування ролі адміністратор

На рисунку 3.1, адміністративної панелі видно, що інтерфейс включає таблицю з даними користувачів. Кожен рядок таблиці містить інформацію про окремого користувача: ім'я, дату створення, роль, дату останнього оновлення та випадаючий список для зміни ролі. Адміністратор може змінювати роль користувача, вибираючи нову роль з випадаючого списку. Після вибору нової ролі, відправляється запит до серверу для оновлення даних. Після успішного оновлення даних, нова роль відображається у таблиці. З розробленими компонентами веб-інтерфейсу для відображення списку користувачів можна ознайомитися в додатку Б.

На рисунку 3.2, сторінці реалізації інтерактивних діаграм даних датчиків користувачі можуть переглядати дані з різних датчиків, до яких він має доступ, які вимірюють параметри якості повітря, такі як температура, вологість, радіаційний фон, концентрація дрібних частинок та різних газів.

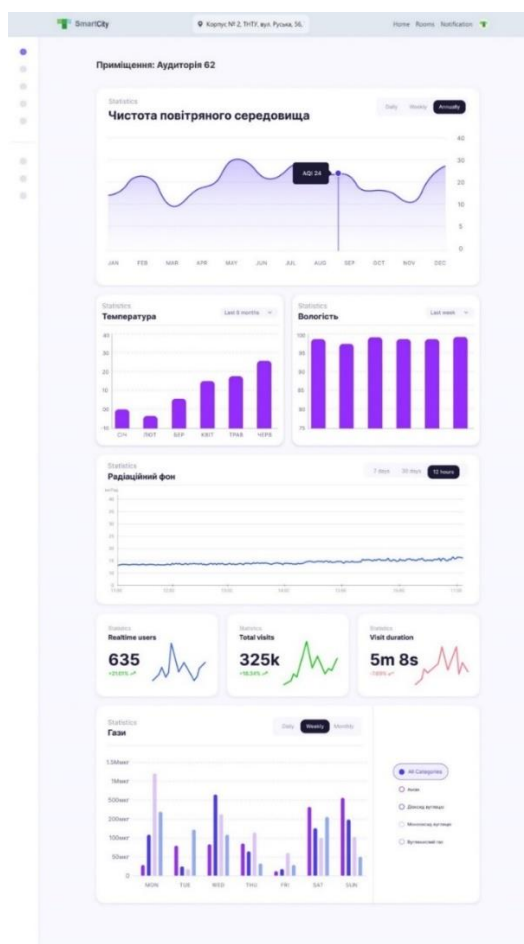


Рисунок 3.2 –Сторінка реалізації інтерактивних діаграм даних датчиків інформаційної системи спостереження внутрішнього середовища

Також, можна побачити, що сторінка містить інтерактивні діаграми, які відображають дані з давачів у різних часових інтервалах. Користувачі мають можливість сортувати історичні дані та змінювати часовий інтервал для перегляду. Кожен параметр якості повітря відображається на окремій діаграмі. Діаграми інтерактивні, що дозволяє користувачам отримувати додаткову інформацію при наведенні курсору на певні точки діаграми. Це забезпечує зручний спосіб аналізу даних в реальному часі.

Користувачі можуть вибирати різні часові інтервали для перегляду даних: добовий, тижневий, місячний та річний. Це дозволяє детально аналізувати тенденції та зміни параметрів якості повітря за різні періоди.

Додаткова можливість фільтрації даних дозволяє користувачам вибирати конкретні параметри для перегляду. Наприклад, можна відфільтрувати дані за типом газу або переглядати лише певні категорії даних.

На сторінці кожен компонент відповідає за відображення даних з різних давачів системи спостереження внутрішнього середовища. Компонент `AirQualityDashboard` є головним контейнером для всіх діаграм (див. лістинг 3.1). Він відповідає за завантаження даних з БД та передачу їх окремим компонентам діаграм. У цьому компоненті використовується `useEffect` для виконання дії `fetchAirQualityData` при першому завантаженні сторінки. Дія викликається за допомогою `useDispatch`, а дані отримуються за допомогою `useSelector`.

### Лістинг 3.1 – Компонент головний контейнер всіх діаграм на сторінці

```
const AirQualityDashboard = () => {
  const dispatch = useDispatch();
  const { data, loading } = useSelector(state =>
state.airQuality);

  useEffect(() => {
    dispatch(fetchAirQualityData());
  }, [dispatch]);

  if (loading) {
    return <p>Loading data...</p>;
  }

  return (
    <div>
```

```

    <TemperatureChart data={data.temperature} />
    <HumidityChart data={data.humidity} />
    <RadiationChart data={data.radiation} />
    <ParticlesChart data={data.particles} />
    <GasesChart data={data.gases} />
  </div>
);
};

export default AirQualityDashboard;

```

Компоненти на сторінці відповідають за відображення даних з різних давачів системи спостереження внутрішнього середовища. Вони використовують бібліотеку Recharts для побудови діаграм і підключаються до Redux Store для управління станом додатка та завантаження даних. Основна структура роботи компонентів включає завантаження даних з БД, відображення даних на діаграмах та інтерактивність.

Компонент TemperatureChart відповідає за відображення даних про температуру (див. лістинг 3.2). Він отримує дані у вигляді пропсів і використовує бібліотеку Recharts для побудови лінійної діаграми. LineChart відображає основну структуру діаграми, CartesianGrid додає сітку на фоні, XAxis і YAxis показують осі координат, Tooltip відображає додаткову інформацію при наведенні курсору, а Line малює саму лінію діаграми.

### Лістинг 3.2 – Компонент відображення даних температури приміщення

```

const TemperatureChart = ({ data }) => {
  return (
    <ResponsiveContainer width="100%" height={300}>
      <LineChart data={data}>
        <CartesianGrid strokeDasharray="3 3" />
        <XAxis dataKey="time" />
        <YAxis />
        <Tooltip />
        <Legend />
        <Line type="monotone" dataKey="value" stroke="#8884d8" />
      </LineChart>
    </ResponsiveContainer>
  );
};

export default TemperatureChart;

```

Всі інші компоненти побудовані аналогічним чином, але відображають різні типи даних. Компонент HumidityChart відповідає за відображення даних про вологість у вигляді гістограми. RadiationChart відображає дані про радіаційний фон у вигляді лінійної діаграми. ParticlesChart відображає дані про концентрацію частинок у повітрі у вигляді лінійної діаграми. GasesChart відповідає за відображення даних про концентрацію газів у повітрі у вигляді гістограми. Всі вони використовують ті ж самі конструкції для побудови діаграми, як і TemperatureChart, але з іншими даними.

Таким чином, кожен компонент відображає дані з різних датчиків і використовує бібліотеку Recharts для побудови діаграм з різними типами відображення, такими як лінійні діаграми і гістограми, забезпечуючи інтерактивність та можливість аналізу даних у різних часових інтервалах.

Далі розглянемо процес створення сторінок аутентифікації користувача. На рисунку 3.3 зображено сторінку реалізації аутентифікації користувача

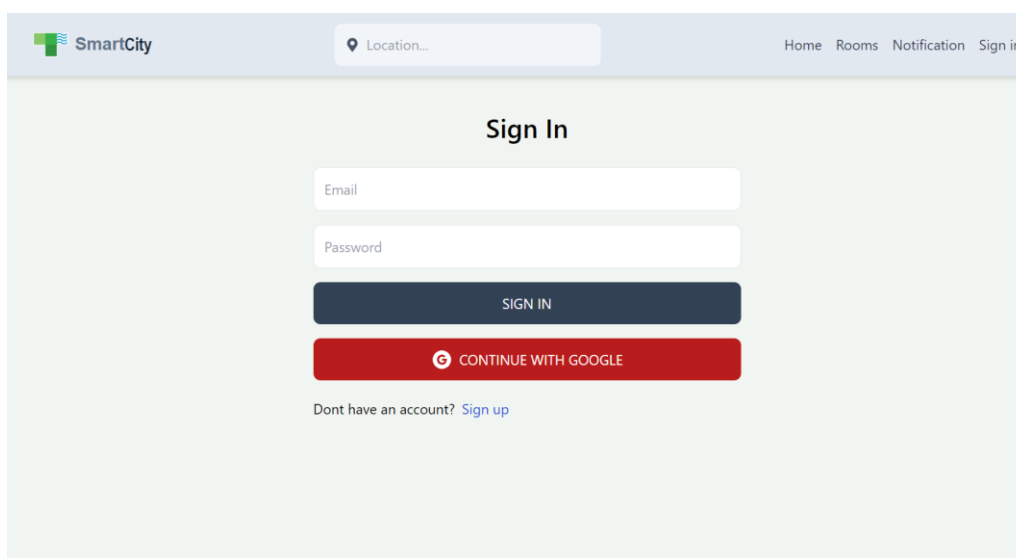


Рисунок 3.3 – Сторінка реалізації аутентифікації користувача

Процес створення сторінок аутентифікації користувача включає кілька ключових етапів:

1. Створення інтерфейсу для аутентифікації: включає в себе як звичайну форму для вводу даних (ім'я користувача, електронна пошта, пароль), так і кнопку для аутентифікації через сторонні сервіси (наприклад, Google).

2. Реалізація логіки для обробки аутентифікації на стороні клієнта: обробка подій вводу даних, перевірка правильності введених даних, взаємодія з бекендом для аутентифікації.

3. Реалізація логіки для обробки аутентифікації на стороні сервера: обробка запитів на реєстрацію та вхід, взаємодія з БД для збереження та перевірки даних користувачів, інтеграція з сторонніми сервісами для аутентифікації через OAuth.

Перший крок – створення інтерфейсу, який буде включати в себе форми для вводу даних та кнопки для аутентифікації через сторонні сервіси. Компоненти форми мають включати поля для введення імені користувача, електронної пошти та пароля. Кнопка для аутентифікації через Google буде викликати функцію, яка обробляє цей процес.

Для обробки подій на стороні клієнта використовуємо React. Код включає функції для обробки вводу даних користувача та відправки цих даних на сервер для подальшої обробки. Для цього використовуємо хуки `useState` для зберігання стану форми та `useEffect` для відправки запиту на сервер при поданні форми.

Для реалізації аутентифікації через Google використовуємо `Firebase Authentication`. Після кліку на кнопку для аутентифікації через Google викликається функція, яка відкриває вікно для аутентифікації через Google. Після успішної аутентифікації отримані дані передаються на сервер для збереження в базу даних та створення користувацької сесії. На стороні сервера використовуємо `Node.js` з `Express` для обробки запитів. Реалізуємо контролери для обробки реєстрації, входу та аутентифікації через Google. При реєстрації користувача дані зберігаються в базі даних після хешування пароля. При вході перевіряється відповідність введених даних з даними в базі. При аутентифікації через Google перевіряється, чи існує користувач з такою електронною поштою, якщо ні – створюється новий запис. Для налаштування `Firebase` створюємо конфігураційний файл, де вказуємо ключові параметри для інтеграції з `Firebase Authentication`. Цей файл використовується на стороні клієнта для ініціалізації `Firebase` та аутентифікації через Google.

### 3.2. Реалізація бази даних

БД є важливим компонентом цієї системи, оскільки вона зберігає всі дані, необхідні для її функціонування, включаючи інформацію про користувачів, пристрої, датчики та їх показники. Ми використовуємо базу даних MySQL через її надійність, масштабованість та підтримку широкого спектру функцій SQL.

Процес створення БД починається зі створення схеми, яка включає визначення всіх таблиць та зв'язків між ними. Основні таблиці БД включають таблиці користувачів, ролей, локацій, типів платформ, IoT пристроїв, датчиків та актуаторів, а також спеціалізовані таблиці для різних типів датчиків і їх історичних даних (див. Додаток В).

Після створення схеми необхідно додати початкові дані в таблиці. Наприклад, додамо декілька датчиків і початкові дані для них (див. лістинги 3.2, 3.3).

#### Лістинг 3.2 – Додавання декількох видів датчиків в таблицю sensors

```
INSERT INTO Sensor (sensorID, sensorType, sensor_status) VALUES
(1, Humidity, 'active');
INSERT INTO Sensor (sensorID, sensorType, sensor_status) VALUES
(2, Temperature, 'active');
INSERT INTO Sensor (sensorID, sensorType, sensor_status) VALUES
(3, 'Multi', 'active');
```

#### Лістинг 3.3 – Додавання даних в базу

```
INSERT INTO TemperatureSensor (sensorID, timestamp, temperature)
VALUES (1, NOW(), 23.5);
INSERT INTO HumiditySensor (sensorID, timestamp, humidityLevel)
VALUES (1, NOW(), 45.2);
INSERT INTO TemperatureSensor (sensorID, timestamp, temperature)
VALUES (2, NOW(), 26.1);
INSERT INTO PollutionSensor (sensorID, timestamp, pollutionLevel,
pollutionType) VALUES (2, NOW(), 55, 'PM2.5');
INSERT INTO PollutionSensor (sensorID, timestamp, pollutionLevel,
pollutionType) VALUES (2, NOW(), 30, 'PM10');
INSERT INTO HumiditySensor (sensorID, timestamp, humidityLevel)
VALUES (3, NOW(), 50.7);
INSERT INTO GasSensor (sensorID, timestamp, gasType,
gasConcentration) VALUES (3, NOW(), 'CO2', 400);
```



У системі спостереження внутрішнього середовища «розумних приміщень» важливою частиною є процес ефективного витягу та обробки даних з БД. Цей процес забезпечує актуальність відображуваної інформації, а також зберігання історичних даних для подальшого аналізу. БД складається з кількох таблиць, кожна з яких відповідає за збереження даних певного типу дачача.

Для отримання поточних даних з дачачів використовуються SQL-запити, які звертаються до відповідних таблиць БД. Кожна таблиця зберігає дані певного типу дачача, наприклад, таблиця TemperatureSensor містить дані про температуру, HumiditySensor – про вологість тощо. Коли дачач надсилає нові дані, ці таблиці оновлюються, що дозволяє системі завжди мати актуальну інформацію.

Ефективна робота з БД, особливо в контексті IoT, де задіяні дачачі, вимагає глибокого розуміння того, як отримувати дані з різних таблиць. Отримання даних дає змогу детально аналізувати показання дачачів, відстежувати зміни з часом і підтримувати повний історичний запис.

Щоб почати отримувати дані, потрібно зрозуміти структуру основної таблиці для дачачів, яку часто називають Sensor. Ця таблиця містить загальну інформацію про кожен дачач, наприклад його унікальний ідентифікатор (sensorID), тип (sensorType) і статус (sensor\_status). Отримання основних відомостей про дачач передбачає виконання SELECT запиту. Наприклад, наступний запит отримує sensorID, sensorType і sensor\_status для дачача з ID 1:

```
SELECT sensorID, sensorType, sensor_status  
FROM Sensor  
WHERE sensorID = 1;
```

Крім основної Sensor таблиці, кожен тип дачача, наприклад температури, вологості або забруднення, має свою окрему таблицю. У цих таблицях зберігаються докладні дані, що стосуються кожного типу дачача. Наприклад, щоб отримати докладні дані з цих таблиць, ви можете виконувати JOIN операції між Sensor таблицею та конкретними таблицями даних дачача. Наступний запит, що продемонстровано в лістингу 3.4, отримує дані про температуру та вологість для дачача з ID 1.

## Лістинг 3.4 – Отримання даних про температуру та вологість датчика з ID 1

```

SELECT
    s.sensorID,
    s.sensorType,
    ts.temperature,
    hs.humidityLevel
FROM
    Sensor s
LEFT JOIN
    TemperatureSensor ts ON s.sensorID = ts.sensorID
LEFT JOIN
    HumiditySensor hs ON s.sensorID = hs.sensorID
WHERE
    s.sensorID = 1;

```

Коли датчик потрібно замінити, дуже важливо зберегти старі дані датчика для історичного аналізу. Цей процес передбачає запис поточних даних датчика в таблиці історії, а потім оновлення основних таблиць датчика новими даними від датчика на заміну. Перед заміною, модератор запускає запис поточних даних у відповідні таблиці історії за допомогою кнопки в інтерфейсі клієнта. Це робиться за допомогою INSERT INTO ... SELECT операторів для перенесення поточних показань до таблиць історії. В лістингу 3.5 наведено приклад, як записано дані про поточну температуру в TemperatureSensorHistory таблицю.

## Лістинг 3.5 – Запис даних про поточну температуру в таблицю історії

```

INSERT INTO TemperatureSensorHistory (sensorID, timestamp,
temperature)
SELECT sensorID, NOW(), temperature
FROM TemperatureSensor
WHERE sensorID = 1;

```

Подібним чином, щоб записати поточні дані про вологість, команда буде схожою (див. лістинг 3.6)

## Лістинг 3.6 – Запис даних про поточну вологість в таблицю історії

```

INSERT INTO HumiditySensorHistory (sensorID, timestamp,
humidityLevel)
SELECT sensorID, NOW(), humidityLevel
FROM HumiditySensor
WHERE sensorID = 1;

```

Після запису даних давач замінюється, а нові записи даних додаються до основних таблиць давачів, щоб відобразити показання з нового давача. Наприклад, щоб вставити нові дані про температуру, вологість і радіацію, використовуються вже знайомі оператори (див. лістинг 3.7).

### Лістинг 3.7 – Вставка нових даних після заміни давача

```
INSERT INTO TemperatureSensor (sensorID, timestamp, temperature)
VALUES (1, NOW(), 24.0);
INSERT INTO HumiditySensor (sensorID, timestamp, humidityLevel)
VALUES (1, NOW(), 50.0);
INSERT INTO RadiationSensor (sensorID, timestamp, radiationLevel)
VALUES (1, NOW(), 0.02);
```

Під час аналізу даних давачі важливо перевірити, чи існують історичні дані для давача, особливо якщо він був замінений, і включити їх в аналіз. Щоб визначити, чи існують дані історії, можна виконати запити до таблиць історії. Наприклад, щоб перевірити історичні дані температури для давача з ID 1, можна використати такий запит:

```
SELECT COUNT(*)
FROM TemperatureSensorHistory
WHERE sensorID = 1;
```

Якщо історичні дані знайдено, їх можна отримати та поєднати з поточними даними давачів. Це забезпечує всебічне уявлення про продуктивність давача з часом. Щоб отримати історичні дані про температуру для давача з ідентифікатором 1, ви повинні використовувати:

```
SELECT sensorID, timestamp, temperature
FROM TemperatureSensorHistory
WHERE sensorID = 1;
```

Для поєднання поточних і історичних даних можна використовувати UNION операції. Цей підхід гарантує, що всі точки даних, як поточні, так і історичні, включені в набір даних. Наступний запит, що продемонстровано в лістингу 3.8, об'єднує поточні та історичні дані температури для давача з ID 1.

### Лістинг 3.8 – Об'єднання поточних та історичних даних температури

```
SELECT sensorID, timestamp, temperature
FROM TemperatureSensor
WHERE sensorID = 1
UNION ALL
SELECT sensorID, timestamp, temperature
FROM TemperatureSensorHistory
WHERE sensorID = 1;
```

Послідовно записуючи та зберігаючи дані в поточних та історичних таблицях, забезпечуючи точні часові позначки для відстеження, коли було зроблено зчитування даних, і впроваджуючи механізми реєстрації дій модераторів, таких як заміна датчиків, можна ефективно керувати та отримувати дані з БД датчиків. Цей комплексний підхід допомагає підтримувати цілісність даних, підтримувати надійний аналіз і приймати обґрунтовані рішення.

Наступний запит, наведений у лістингу 3.9, дозволяє отримати всі дані для датчика з ідентифікатором 1, об'єднуючи інформацію з кількох таблиць.

### Лістинг 3.9 – Отримання всіх поточних даних датчика з ідентифікатором 1

```
SELECT
    s.sensorID,
    s.sensorType,
    ts.timestamp AS tempTimestamp,
    ts.temperature,
    hs.timestamp AS humTimestamp,
    hs.humidityLevel,
    rs.timestamp AS radTimestamp,
    rs.radiationLevel
FROM
    Sensor s
LEFT JOIN
    TemperatureSensor ts ON s.sensorID = ts.sensorID
LEFT JOIN
    HumiditySensor hs ON s.sensorID = hs.sensorID
LEFT JOIN
    RadiationSensor rs ON s.sensorID = rs.sensorID
WHERE
    s.sensorID = 1;
```

Підсумовуючи, робота з БД передбачає:

- Вилучення даних з основної Sensor таблиці та окремих таблиць даних датчиків за допомогою SELECT та JOIN операцій.

- Обробка заміни давачів шляхом запису поточних даних у таблиці історії перед оновленням новими даними давачів.
- Перевірка та отримання історичних даних для поєднання з поточними даними для комплексного аналізу.

Отримання даних з БД у системі спостереження «розумних приміщень» є складним, але важливим процесом, який забезпечує доступність як поточних, так і історичних даних давачів. Завдяки правильній організації БД та ефективним SQL-запитам, система може надавати точні та актуальні дані для аналізу та прийняття рішень. Обробка історичних даних дозволяє зберігати важливу інформацію про минулі показники, що є критично важливим для довгострокового спостереження та аналізу стану внутрішнього середовища. Завдяки цьому підходу, користувачі можуть отримувати повну інформацію про стан своїх приміщень, незалежно від змін у давачах або їх налаштуваннях.

### **3.3. Автентифікація та реєстрація користувачів**

Автентифікація користувачів — важливий аспект веб-інтерфейсів, який гарантує, що лише ідентифіковані користувачі можуть отримати доступ до певних ресурсів і виконувати певні дії. У рамках інформаційної системи спостереження внутрішнього середовища «розумного приміщення» було реалізовано подвійний підхід до автентифікації користувачів. Це передбачає традиційне введення даних за допомогою електронної пошти та пароля, а також OAuth з використанням автентифікації Google. Надаючи ці параметри, ми підвищуємо зручність і безпеку для користувачів, задовольняючи різні вподобання та потреби користувачів.

Традиційний процес автентифікації починається з розробки елементів інтерфейсу користувача (UI) для реєстрації та сторінок входу. Ці сторінки створені так, щоб бути інтуїтивно зрозумілими та зручними для користувача, містять поля для введення імені користувача, електронної пошти та пароля. Форми містять кнопку надсилання, яка ініціює процес автентифікації (див.

рисунок 3.4). Крім того, щоб забезпечити гнучкість, користувальницький інтерфейс містить опцію OAuth, яка дозволяє користувачам входити за допомогою Google. Ця інтеграція не тільки спрощує процес входу, але й додає додатковий рівень безпеки.

Рисунок 3.4 – Інтерфейс автентифікації користувачів

Управління введенням даних користувачами в React важливо забезпечити динамічну обробку форми. Використовуючи `useState` хук React, ми можемо зручно керувати станом даних форми (див. лістинг 3.10). Прослуховувачі подій дозволяють фіксувати зміни у полях введення, що дозволяє оновлювати дані в реальному часі та перевіряти їх на коректність перед надсиланням.

#### Лістинг 3.10 – Управління вводу даних за допомогою `useState` у React

```
const [formData, setFormData] = useState({});
const handleChange = (e) => {
  setFormData({
    ...formData,
    [e.target.id]: e.target.value,
  });
};
```

Коли користувач надсилає форму реєстрації, програма надсилає POST-запит на сервер із введеними даними користувача. Ця дія включає встановлення

стану завантаження для надання зворотного зв'язку користувачеві та витончену обробку можливих помилок. Процес надсилання форми інкапсульований у `async` функції, яка використовується `fetch` для зв'язку із сервером. Якщо відповідь сервера вказує на помилку, стан завантаження скидається, а користувачеві відображається відповідне повідомлення про помилку (див. лістинг 3.11).

### Лістинг 3.11 – Надсилання форми реєстрації з використанням `async` функції

```
const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    setLoading(true);
    const res = await fetch("/api/auth/signup", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(formData),
    });
    const data = await res.json();
    if (data.success === false) {
      setLoading(false);
      setError(data.message);
      return;
    }
    setLoading(false);
    setError(null);
    navigate("/sign-in");
  } catch (error) {
    setLoading(false);
    setError(error.message);
  }
};
```

На стороні сервера контролер автентифікації відповідає за обробку запитів на реєстрацію та вхід. Для процесу реєстрації контролер хешує пароль користувача `bcryptjs` перед тим, як зберегти його в базі даних. Цей крок є критично важливим, оскільки хешування пароля забезпечує додатковий рівень захисту, навіть навіть у випадку компрометації бази, фактичні паролі залишаються в безпеці. Нові дані користувача потім зберігаються в базі даних, а повідомлення про успіх надсилається назад клієнту (див. лістинг 3.12). Таким чином, контролер автентифікації забезпечує безпеку та цілісність даних.

## Лістинг 3.12 – Автентифікація та обробка запитів на сервері

```

export const signup = async (req, res, next) => {
  const { username, email, password } = req.body;
  const hashedPassword = bcryptjs.hashSync(password, 10);
  const newUser = new User({ username, email, password:
hashedPassword });
  try {
    await newUser.save();
    res.status(201).json("User created successfully!");
  } catch (error) {
    next(error);
  }
};

export const signin = async (req, res, next) => {
  const { email, password } = req.body;
  try {
    const validUser = await User.findOne({ email });
    if (!validUser) return next(errorHandler(404, "User not
found!"));
    const validPassword = bcryptjs.compareSync(password,
validUser.password);
    if (!validPassword) return next(errorHandler(401, "Wrong
credentials!"));
    const token = jwt.sign({ id: validUser._id },
process.env.JWT_SECRET);
    const { password: pass, ...rest } = validUser._doc;
    res
      .cookie("access_token", token, { httpOnly: true })
      .status(200)
      .json(rest);
  } catch (error) {
    next(error);}}};

```

Для входу сервер перевіряє надані облікові дані зі збереженими даними користувача. Якщо облікові дані дійсні, сервер генерує JSON Web Token (JWT) і надсилає його назад клієнту. JWT використовуються для підтримки стану сеансу без збереження стану, що дозволяє клієнту безпечно автентифікувати наступні запити. Маркер встановлюється як файл cookie з httpOnly прапорцем, що підвищує безпеку, запобігаючи доступ до нього сценаріїв на стороні клієнта.

Процес автентифікації OAuth, зокрема в Google, пропонує спрощену та безпечну альтернативу традиційним входам. Процес починається зі створення компонента OAuth у інтерфейсі. Цей компонент використовує Firebase для керування автентифікацією Google. Коли користувачі натискають кнопку входу в



Google, відкривається спливаюче вікно, яке дозволяє їм пройти автентифікацію за допомогою свого облікового запису Google.

Інтеграція з Firebase передбачає імпорт необхідних модулів, таких як `GoogleAuthProvider`, `getAuth` і `signInWithPopup`. Коли користувач успішно входить через Google, Firebase повертає об'єкт результату, що містить відомості про користувача. Потім ця інформація надсилається на сервер для подальшої автентифікації.

На стороні сервера контролер OAuth перевіряє, чи існує користувач у базі даних. Якщо користувач існує, сервер генерує JWT і надсилає його назад клієнту, як у традиційному процесі входу. Якщо користувач не існує, сервер створює новий запис користувача. Для підвищення безпеки створюється довільно згенерований пароль, який хешується “bcryptjs” модулем перед збереженням. Дані нового користувача, зокрема дані, надані Google, як-от ім'я, електронна адреса та зображення профілю, зберігаються в базі даних. Потім для нового користувача створюється JWT і надсилається назад клієнту. Цей процес OAuth не лише спрощує вхід для користувачів, але також використовує надійну інфраструктуру безпеки Google.

Реалізація автентифікації користувачів у нашій інформаційній системі «розумних приміщень» ефективно збалансовує безпеку та зручність користувача. Забезпечуючи як традиційну автентифікацію електронною поштою/паролем, так і OAuth з Google, ми задовольняємо різноманітні вподобання користувачів. Цей подвійний підхід гарантує, що користувачі мають безпечні, гнучкі варіанти доступу до системи. Використовуючи сучасні методи автентифікації та найкращі практики, ми покращуємо загальну безпеку та зручність використання системи, пропонуючи безперебійну та безпечну роботу для користувачів.

### **3.4. Висновок до третього розділу**

В третьому розділі кваліфікаційної роботи освітнього рівня “Бакалавр” здійснено розробку та впровадження ключових компонентів інформаційної

системи спостереження внутрішнього середовища «розумних приміщень». Зокрема, представлено процес створення веб-інтерфейсу, який забезпечує користувачам зручний та ефективний доступ до даних датчиків у реальному часі. Реалізовано інтерактивні компоненти для відображення параметрів якості повітря.

Описано структуру та функціонування БД, яка включає всі необхідні таблиці та взаємозв'язки для зберігання та обробки інформації з датчиків. Було реалізовано механізми збереження історичних даних та забезпечено їх доступність для аналізу.

Детально зображено функціональні особливості автентифікації та реєстрації користувачів. Ретельно описано процеси захисту даних, що дозволяє контролювати доступ користувачів до різних функцій системи. Реалізовано безпечну реєстрацію нових користувачів та автентифікацію існуючих.

## **РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ**

Оскільки кваліфікаційна робота освітнього рівня бакалавр присвячена розробці веб-інтерфейсу інформаційної системи спостереження внутрішнього середовища «розумних приміщень», а робота користувачів персональних комп'ютерів (ПК) пов'язана з рядом небезпек, то в розділі безпеки життєдіяльності доцільно розглянути вимоги безпеки праці під час експлуатації систем вентиляції, опалення та кондиціонування повітря та розробку захисту від пожеж та вибухів в системах опалення, вентиляції, освітлення та кондиціонування повітря.

### **4.1. Вимоги безпеки праці під час експлуатації систем вентиляції, опалення та кондиціонування повітря.**

Згідно ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування» в проектах опалення, вентиляції та кондиціонування слід передбачати технічні рішення, що забезпечують різні параметри, ось деякі з них:

- нормовані параметри мікроклімату та концентрацію шкідливих речовин у повітрі зони обслуговування приміщень житлових будинків, громадських будівель і споруд, будівель адміністративного та побутового призначення згідно з санітарно-епідеміологічними вимогами;
- нормовані параметри мікроклімату та концентрацію шкідливих речовин у повітрі робочої зони виробничих, лабораторних та складських (далі - виробничих) приміщень у будівлях будь-якого призначення згідно з ГОСТ 12.1.005 і санітарно-гігієнічними вимогами до мікроклімату виробничих приміщень згідно з ДСН 3.3.6.042 та відповідно до положень розділу 5 "Параметри внутрішнього та зовнішнього повітря" цих Норм;
- охорону атмосферного повітря від вентиляційних викидів шкідливих речовин;

- механічну безпеку, електробезпеку, виконання вимог охорони праці під час монтажу, налагодження, випробувань та експлуатації опалювально-вентиляційного обладнання;
- вибухопожежобезпечність опалювально-вентиляційного обладнання [37].

У створенні сприятливих умов праці ефективним засобом є вентиляція – сукупність заходів та засобів призначених для забезпечення на постійних робочих місцях та зонах обслуговування виробничих приміщень метеорологічних умов та чистоти повітряного середовища, що відповідають гігієнічним та технічним вимогам. Залежно від способу переміщення повітря вентиляція буває природна і механічна (штучна) [38].

Природна вентиляція здійснюється за рахунок сили вітру і природними (гравітаційними) силами. Вітер, обдуваючи споруду, попереду неї створює зону підвищеного тиску, а з протилежного боку виникає зона певного розрідження. Під дією напору вітер через фрамуги, кватирки, створи й інші отвори проникає у приміщення, а під дією розрідження забруднене повітря виходить назовні. Перевагою природної вентиляції є її дешевизна та простота експлуатації. Основний її недолік у тому, що повітря надходить у приміщення без попереднього очищення, а виділене відпрацьоване повітря також не очищається і забруднює довкілля.

Механічна вентиляція – це примусове видалення з приміщень забрудненого повітря і заміна його на свіже за допомогою вентиляційних агрегатів. Сукупність вентиляційного агрегату, повітроводів, регулювальних, пускових та інших пристроїв складає вентиляційну систему для конкретного виробничого приміщення. Штучна вентиляція може бути загальнообмінною, місцевою та комбінованою.

Природна та штучна вентиляції повинні відповідати наступним санітарногігієнічним вимогам: створювати в робочій зоні приміщень нормовані параметри повітряного середовища; не вносити в приміщення забруднене повітря ззовні або шляхом засмокткування забрудненого повітря з суміжних

приміщень; не створювати на робочих місцях протягів чи різкого охолодження; бути доступними для управління та ремонту під час експлуатації; не створювати під час експлуатації додаткових незручностей, бути економічними, вибухопожежобезпечними, не заважати використовувати технологічні операції, не створювати перешкоди внутрицеховому транспорту, не впливати на якість продукції.

При роботі припливно-витяжної вентиляції необхідно, щоб кількість повітря, що надходить ззовні, не перевищувала або була на 10-15% меншою від кількості повітря, що видаляється витяжними пристроями.

Потребу в чистому повітрі на одного працівника на годину можна визначити за формулою:

$$a = k / p - g, \quad (4.1)$$

де:  $a$  – необхідний вентиляційний об'єм повітря;  $k$  – кількість літрів вуглекислого газу, що виділяє людина за 1 годину;  $p$  – допустима концентрація вмісту вуглекислого газу в приміщенні (0,1 %) – 1,0 л в 1 м<sup>3</sup> повітря;  $g$  – вміст вуглекислого газу в повітрі (0,03%) – 0,03 л в 1 м<sup>3</sup> повітря.

Швидкість руху повітря має становити:

- для видалення газів з холодних приміщень – 0,5-1 м/с, з теплих – 1,0-1,5;
- для видалення пилу – відповідно 0,8-1,5 м/с і 1,5-2,5 м/с.

Місцева вентиляція забезпечує нормалізацію повітряного середовища на робочих місцях. Вона може бути припливною (повітряні душі, повітряні та повітряно-теплові завіси) і витяжною (вловлювання шкідливих речовин безпосередньо біля місць їх утворення).

Для створення та автоматичного підтримування в приміщенні заданих або таких, що змінюються за певною програмою умов мікроклімату використовують кондиціонування.

Кондиціонування повітря — це створення автоматичного підтримування в приміщенні, незалежно від зовнішніх умов (постійних чи таких, що змінюються),

по визначеній програмі температури, вологості, чистоти і швидкості руху повітря. У відповідності з вимогами для конкретних приміщень повітря нагрівають або охолоджують, зволожують або висушують, очищають від забруднюючих речовин або піддають дезінфекції, дезодорації, озонуванню.

Для кондиціонування повітря у виробничих приміщеннях використовують такі кондиціонери:

- а) центральні, що встановлюються за межами робочих приміщень;
- б) місцеві, розміщені безпосередньо у приміщенні.

Будова, склад і експлуатація систем опалення мають відповідати вимогам ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування».

Для нагрівання окремих приміщень дозволяється використовувати електричні прилади з закритими спіралями і з такою потужністю споживання, яка б не призводила до підвищення сили струму понад допустиму для даної електромережі.

Опалювальні прилади розміщують у місцях, доступних для огляду, ремонту, очищення на відстані 0,1 м від поверхні стін. Не дозволяється розміщувати опалювальні прилади в нішах стін. Опалювальні прилади на сходових клітках розміщуються в нижніх поверхах, а також у відсіках тамбурів, які не мають зовнішніх дверей.

#### **4.2. Розробка захисту від пожеж та вибухів в системах опалення, вентиляції, освітлення та кондиціонування повітря.**

Перед початком опалювального сезону теплові мережі, які розміщені у приміщеннях, калориферні й теплогенераторні установки, котельні, печі та інші опалювальні прилади повинні бути перевірені й відремонтовані. Несправні опалювальні прилади не повинні допускатися до експлуатації.

Гарячі поверхні тепломереж, що розташовані у приміщеннях, у яких вони можуть створити небезпеку спалахування парів, газів, пилу або аерозолів, потрібно ізолювати таким способом, щоб температура на поверхні

теплоізолюваної конструкції була не менш ніж на 20% нижчою за температуру самоспалахування речовин [39].

Усі гарячі ділянки поверхонь трубопроводів і обладнання, яке розташоване в зоні можливого потрапляння на них вибухонебезпечних, горючих або легкозаймистих речовин, необхідно покрити металевою обшивкою. Не дозволяється експлуатація теплових мереж з просоченою вибухонебезпечними, горючими або легкозаймистими речовинами теплоізоляцією. Очищення печей та димоходів від сажі необхідно проводити перед початком, а також впродовж усього опалювального сезону, а саме [40]:

- кухонних кип'ятильників та плит – один раз на місяць;
- печей безперервної дії – не рідше одного разу на два місяці;
- опалювальних печей періодичної дії на рідкому та твердому паливі – не рідше одного разу на три місяці.

У приміщеннях котелень та інших теплогенеруючих установок населених пунктів і підприємств забороняється:

- сушити взуття, спецодяг та інші матеріали на паропроводах та котлах;
- працювати при відключених або зіпсованих приладах регулювання і контролю, а також за їх відсутності;
- розпалювати установки без їх попередньої продувки; подавати паливо, коли газові пальники або форсунки згасли;
- експлуатувати установки у випадку витікання газу із системи паливоподачі або підтікання рідкого палива;
- допускати до роботи працівників, які не пройшли навчання з пожежнотехнічного мінімуму, не отримали відповідних посвідчень кваліфікації.

Забороняється вносити зміни до елементів системи кондиціонування, вентиляції повітря і опалення, які перешкоджають поширенню пожежі. Не дозволяється робота технологічного обладнання у пожежонебезпечних та вибухопожежонебезпечних приміщеннях при відключених або несправних сухих фільтрах, гідрофільтрах, пиловловлювальних, пиловсмоктувальних та інших приладах вентиляційних систем.

При розміщенні вибухозахищених вентиляторів за межами приміщень для них необхідно влаштувати спеціальне укриття з негорючих матеріалів. Під час експлуатації вентиляційних систем забороняється [40]:

- експлуатувати переповнені циклони;
- видаляти за допомогою однієї системи відсосів пил, пару, різні гази та інші речовини, які при змішуванні можуть викликати вибух, горіння або спалахи;
- складувати впритул (на відстані менше половини метра) до устаткування і повітроводів негорючі матеріали в горючій упаковці або горючі матеріали;
- застосовувати припливно-витяжні повітроводи й канали для відведення газів від газових колонок, приладів опалення та інших нагрівальних приладів;
- залишати двері вентиляційних камер відчиненими, зберігати в камерах різне устаткування та матеріали;
- закривати решітки, отвори і витяжні канали.

Автономні моноблочні кондиціонери, а також автономні кондиціонери роздільного типу можна розміщувати у будівлях та приміщеннях різного призначення, крім приміщень, у яких не допускається рециркуляція. Зовнішні блоки автономних кондиціонерів роздільного типу потужністю по холоду до 12 кВт допускається розміщувати у критих переходах, відкритих сходових клітках, на незаскленних лоджіях.

Холодильне обладнання з аміаковмісним холодоагентом можна використовувати при реконструкції для холодопостачання систем кондиціонування виробничих приміщень, розміщуючи обладнання в окремих прибудовах, будинках або окремих приміщеннях одноповерхових виробничих будинків.

Під час експлуатації калориферів потрібно дотримуватися таких вимог [40]:

- слідкувати за тим, щоб транзитні канали, через які подається нагріте в калорифері повітря, не мали отворів, крім каналів, призначених для подавання повітря у приміщення;



- систематично проводити гідравлічним або пневматичним способом очищення калориферів від забруднень;

- не допускати виникнення зазорів між калориферами, а також між будівельними і калориферними конструкціями камер, а виявлені зазори зашпаровувати негорючими матеріалами;

- тримати в справному стані контрольно-вимірвальні прилади;

- відстань між конструкціями з важкогорючих та горючих матеріалів і калориферами повинна бути не меншою за півтора метра за наявності електричного або вогневого підігріву і не меншою за 0,1 метр, коли теплоносієм є пара або вода.

Прокладання, підключення, монтаж мереж, влаштування електричного захисту на лініях, які живлять побутові кондиціонери, повинні виконуватись відповідно до вимог інструкції виробника. Лінії живлення до кожного побутового кондиціонера чи групи кондиціонерів потрібно забезпечувати автономним пристроєм електричного захисту незалежно від наявності захисту на загальній лінії, яка відповідає за живлення групи кондиціонерів.

Під час експлуатації побутових кондиціонерів заборонено [40]:

- перетинати протипожежні перешкоди інженерними системами кондиціонера без влаштування проходок, які відповідають нормованій межі вогнестійкості протипожежної перешкоди;

- замінювати наявні триполюсні штепсельні роз'єднувачі на двополюсні;

- вносити в конструкцію кондиціонерів зміни, не передбачені заводом виробником;

Було розглянуто вимоги безпеки праці під час експлуатації систем вентиляції, опалення та кондиціонування повітря, а також розробку заходів захисту від пожеж та вибухів в цих системах. Оскільки в контексті виконання кваліфікаційної роботи, присвяченої розробці веб-інтерфейсу інформаційної системи спостереження внутрішнього середовища «розумних приміщень», це важливо тому, що забезпечення безпечних умов праці є критичним для ефективного та безперебійного функціонування таких систем.

## ВИСНОВКИ

Кваліфікаційна робота бакалавра присвячена розробці веб-інтерфейсу для інформаційної системи спостереження внутрішнього середовища «розумних приміщень». У вступі підкреслено актуальність теми, визначено мету та завдання дослідження, обґрунтовано вибір методів.

Перший розділ роботи охоплює аналіз предметної області. Здійснено детальне дослідження концепції «розумного міста», визначено його складові та роль внутрішнього середовища. Проаналізовано параметри якості повітря, такі як вологість, температура, домішки, тверді частинки, радіація та гази. Проведено огляд існуючих систем спостереження внутрішнього середовища «розумних приміщень» та сформовано вимоги до веб-інтерфейсу для забезпечення ефективного спостереження та управління параметрами внутрішнього середовища.

Другий розділ присвячено проектуванню та розробці ключових компонентів веб-інтерфейсу. Здійснено проектування архітектури системи з використанням стеку MERN, що забезпечує надійну взаємодію між клієнтом та сервером, а також зберігання даних. Детально розроблено структуру бази даних, яка підтримує зберігання і обробку даних з датчиків у режимі реального часу. Діаграми прецедентів ролей допомагають зрозуміти функціональні можливості та права доступу різних користувачів. Виконано аналітичне опрацювання даних, що дозволяє отримувати інформацію щодо ефективності використання пристроїв та навантаженості системи. Усі ці аспекти інтегровані у веб-інтерфейс, який забезпечує зручний та інтуїтивно зрозумілий доступ користувачів до функціональності системи.

У третьому розділі описано розробку та впровадження веб-інтерфейсу, який забезпечує користувачам зручний та ефективний доступ до даних у реальному часі. Реалізовано інтерактивні компоненти для відображення параметрів якості повітря. Описано структуру та функціонування бази даних, яка включає всі необхідні таблиці та взаємозв'язки для зберігання та обробки

інформації з давачів. Реалізовано механізми збереження історичних даних та їх доступність для аналізу. Детально описано процеси автентифікації та реєстрації користувачів, забезпечено захист даних та контроль доступу до різних функцій системи.

У розділі «Охорона праці та безпека в надзвичайних ситуаціях» проаналізовано нормативно-правові основи охорони праці в галузі інформаційних технологій. Описано планування заходів цивільного захисту на об'єктах культури у випадку надзвичайних ситуацій

Таким чином, проведені дослідження та розробка веб-інтерфейсу для системи спостереження внутрішнього середовища «розумних приміщень» дозволили створити ефективний інструмент для управління параметрами якості повітря в реальному часі. Інтеграція сучасних технологій забезпечила зручний та надійний доступ користувачів до функціональності системи, що сприяє покращенню умов проживання у «розумних приміщеннях».

**ПЕРЕЛІК ДЖЕРЕЛ**

1. Persaud, Tommy, et al. "Smart city innovations to improve quality of life in urban settings." 2020 IEEE Global Humanitarian Technology Conference (GHTC). IEEE, 2020.
2. Včelák, J., et al. "Smart building monitoring from structure to indoor environment." 2017 Smart City Symposium Prague (SCSP). IEEE, 2017.
3. Hsu, Hui-Huang, Chuan-Yu Chang, and Ching-Hsien Hsu, eds. Big data analytics for sensor-network collected intelligence. Morgan Kaufmann, 2017.
4. Tkachenko, Alla M., and Rodion V. Sevastyanov. "Development of electronic services of the smart city." (2023).
5. Okonta, Donatus Ebere, and Vladimir Vukovic. "Smart Cities Software Applications for Sustainability and Resilience." Heliyon (2024).
6. Sethi, Pallavi, and Smruti R. Sarangi. "Internet of things: architectures, protocols, and applications." Journal of electrical and computer engineering 2017.1 (2017): 9324035.
7. Marques, Goncalo, et al. "Indoor air quality monitoring systems for enhanced living environments: A review toward sustainable smart cities." Sustainability 12.10 (2020): 4024.
8. Narayana, T. Lakshmi, et al. "Advances in real time smart monitoring of environmental parameters using IoT and sensors." Heliyon 10.7 (2024).
9. Gracias, Jose Sanchez, et al. "Smart Cities—A Structured Literature Review." Smart Cities 6.4 (2023): 1719-1743.
10. Addas, Abdullah. "The concept of smart cities: a sustainability aspect for future urban development based on different cities." Frontiers in Environmental Science (2023).
11. Mohanty, Ritu, and Bipin Pradeep Kumar. "Urbanization and smart cities." Solving urban infrastructure problems using smart city technologies. Elsevier, 2021. 143-158.

12. Rai, Hari Mohan, et al. "Use of Internet of Things in the context of execution of smart city applications: a review." *Discover Internet of Things* 3.1 (2023): 8.
13. Shin Hwang, What is a Smart City? Definition and 6 Key Features / [Электронный ресурс] – 2024 – URL: <https://letmewp.com/what-is-a-smart-city-definition-and-6-key-features> (дата звернення: 9.06.2024).
14. Hopkins, Torin, et al. "User interfaces in smart cities." *Handbook of smart cities* (2020): 1-33.
15. Wang, Chao, et al. "Economic and environmental impacts of energy efficiency over smart cities and regulatory measures using a smart technological solution. *Sustain Energy Technol Assess* 47: 101422." (2021).
16. Hernaningsih, T., et al. "Application of the concept of smart city and smart water management for the new capital city." *IOP Conference Series: Earth and Environmental Science*. Vol. 1201. No. 1. IOP Publishing, 2023.
17. Mishra, Priyanka, and Ghanshyam Singh. "Internet of medical things healthcare for sustainable smart cities: current status and future prospects." *Applied Sciences* 13.15 (2023): 8869.
18. Kanellopoulos, Dimitris, et al. "Networking architectures and protocols for IoT applications in smart cities: Recent developments and perspectives." *Electronics* 12.11 (2023): 2490.
19. Zeng, Fan, Chuan Pang, and Huajun Tang. "Sensors on Internet of Things Systems for the Sustainable Development of Smart Cities: A Systematic Literature Review." *Sensors* 24.7 (2024): 2074.
20. Navarro-Ortiz, Jorge, et al. "Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things." *IEEE Communications Magazine* 56.2 (2018): 60-67.
21. Liu, Qian, et al. "Cloud, edge, and mobile computing for smart cities." *Urban Informatics* (2021): 757-795.
22. Hura, V., and L. Monastyrskii. "IOT-BASED SOLUTION FOR DETECTION OF AIR QUALITY USING ESP32."
23. David, Bertrand, et al. "Mobile user interfaces and their utilization in a Smart City." *Proceedings on the international conference on Internet Computing (ICOMP)*.

The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2011.

24. Siddique, Abu Buker, et al. "RETRACTED ARTICLE: An intelligent and secure air quality monitoring system using neural network algorithm and blockchain." IETE Journal of Research 69.11 (2023): CVII-CXX.

25. Vignesh, S & Gandhi, A & Santhosh, Adithyan & Shanmugam, Mr. (2024). SMART AIR QUALITY MONITORING SYSTEM.

26. Reinikainen, L. M., and J. J. K. Jaakkola. "Significance of humidity and temperature on skin and upper airway symptoms." Indoor air 13.4 (2003).

27. Tran, Vinh Van, Duckshin Park, and Young-Chul Lee. "Indoor air pollution, related human diseases, and recent trends in the control and improvement of indoor air quality." International journal of environmental research and public health 17.8 (2020): 2927.

28. TOP 10 Tech Stack That Reign software Development in 2024. URL: <https://www.fingent.com/blog/top-7-tech-stacks-that-reign-software-development> (дата звернення: 10.06.2024).

29. Sung, Yaeseul, et al. "Development of a smart air quality monitoring system and its operation." Asian Journal of Atmospheric Environment 13.1 (2019): 30-38.

30. Hoque, Mohammad Asadul, and Chad Davidson. "Design and implementation of an IoT-based smart home security system." International Journal of Networked and Distributed Computing 7.2 (2019): 85-92.

31. Aserkar, Anushree A., et al. "Real-time Air Quality Monitoring in Smart Cities using IoT-enabled Advanced Optical Sensors." International Journal of Advanced Computer Science & Applications 15.4 (2024).

32. Stanko, A., Palka, O., Matiichuk, L., Martsenko, N., & Matsiuk, O. (2021, September). Smart City: A Review of Model Architecture and Technology. In 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 2, pp. 273-277). IEEE.

33. Duda, O., Mykytyshyn, A., Mytnyk, M., & Stanko, A. Information technology sets formation and" TNTU Smart Campus" services network support.

Proceedings of the 3rd International Workshop on Information Technologies: Theoretical and Applied Problems 2023, Ternopil, Ukraine, Opole, Poland, November 22-24, 2023.

34. Duda O., Kunanets N., Matsiuk O., Pasichnyk V., Cloud-based IT Infrastructure for “Smart City” Projects, in Dependable IoT for Human and Industry: Modeling, Architecting, Implementation. River Publishers, 2018. P. 389-410.

35. Стручок, В. С., Стручок, О. С., & Мудра, Д. В. (2017). Навчальний посібник до написання розділу дипломного проекту та дипломної роботи "Безпека в надзвичайних ситуаціях "для студентів всіх спец. денної, заочної (дистанційної) та екстернатної форм навчання.

36. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / укл.: Стручок В. С. Тернопіль: ФОП Паляниця В. А., 2022. 156 с.

37. Бедрій Я.І. Безпека життєдіяльності : навч. посіб. Київ: Кондор, 2009. 284 с.

38. ДБН В.2.5-67:2013. Опалення, вентиляція та кондиціонування. Інститут «УкрНДІспецбуд». 2013. URL: <https://dbn.co.ua/load/normativy/dbn/1-1-0-1018> (дата звернення 10.06.2024).

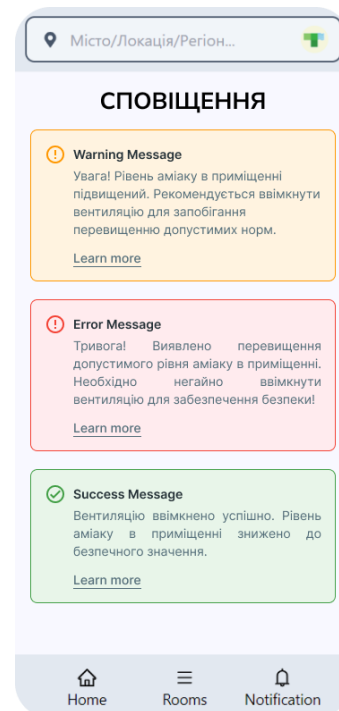
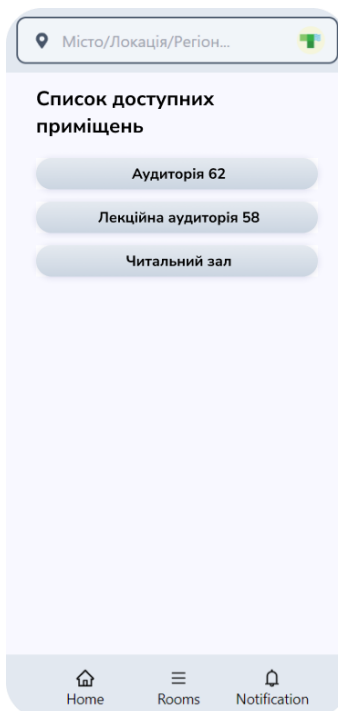
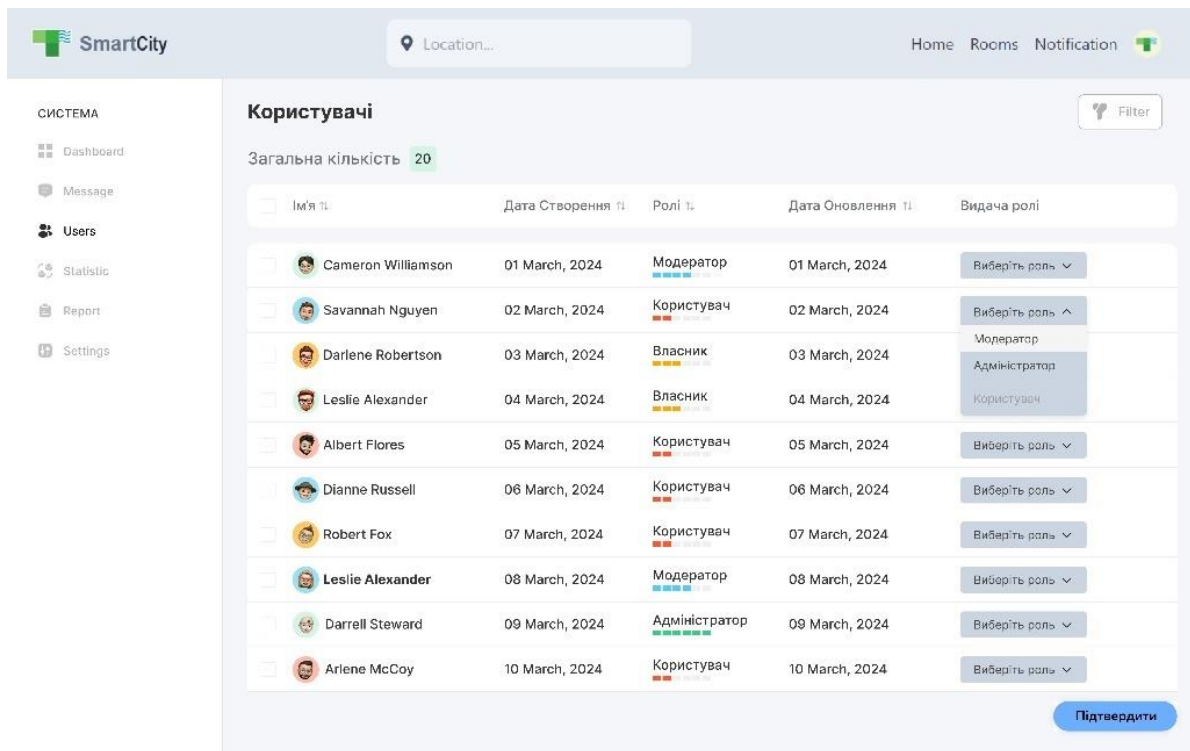
39. Желібо Є. П. Заверуха Н.М., Зацарний В.В. Безпека життєдіяльності. Навчальний посібник. К.: Каравела, 2004. 328 с.

40. Наказ КМУ про затвердження Правил пожежної безпеки в Україні №1417 від 30.12.2014.

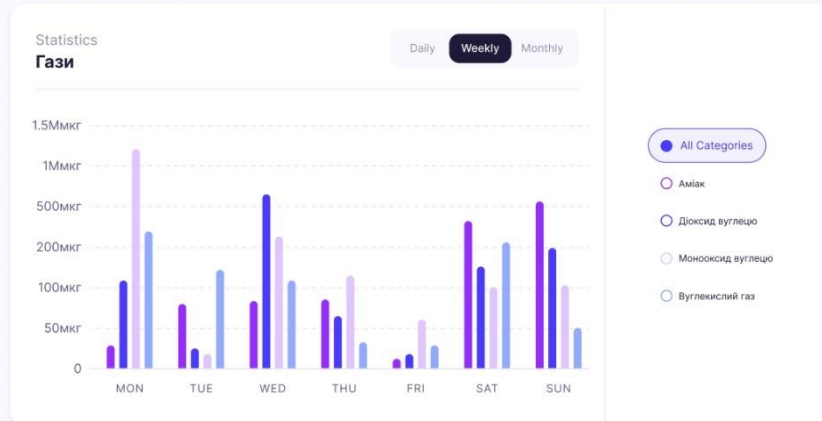
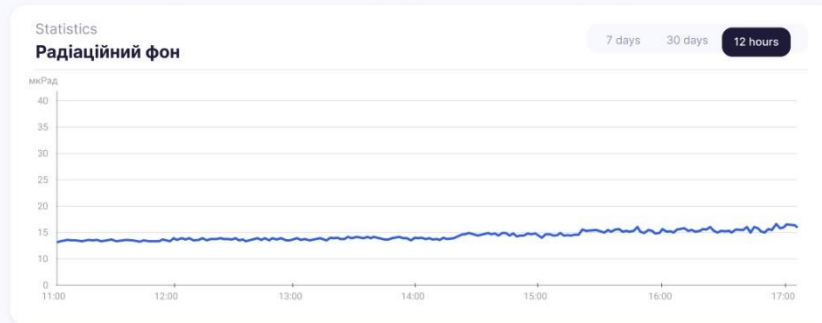
# Додатки



## Перелік рисунків сторінок веб-інтерфейсу



Приміщення: Аудиторія 62



Місто/Локація/Регіон... Sign in


## SignUp

Username




Email


Password

**SIGN UP**


 **CONTINUE WITH GOOGLE**

Have an account? [Sign in](#)

 Home     Rooms     Notification

Місто/Локація/Регіон... 

## Profile






alexkoti

sanya15558764@gmail.com

password

**UPDATE**

 Home     Rooms     Notification

**React код створення компонентів веб-сторінок**

```
//Створення React компонента для відображення списку користувачів
// src/components/UserPanel.js
import React, { useEffect } from 'react';
import { useSelector, useDispatch } from 'react-redux';
import { fetchUsers, updateUserRole } from
'../redux/actions/userActions';

const UserPanel = () => {
  const dispatch = useDispatch();
  const users = useSelector(state => state.users.users);

  useEffect(() => {
    dispatch(fetchUsers());
  }, [dispatch]);

  const handleRoleChange = (userId, event) => {
    const role = event.target.value;
    dispatch(updateUserRole(userId, role));
  };

  return (
    <div className="user-panel">
      <h2>Користувачі</h2>
      <table>
        <thead>
          <tr>
            <th>Ім'я</th>
            <th>Дата Створення</th>
            <th>Роль</th>
            <th>Дата Оновлення</th>
            <th>Видати роль</th>
          </tr>
        </thead>
        <tbody>
          {users.map(user => (
            <tr key={user.id}>
              <td>{user.name}</td>
              <td>{user.createdAt}</td>
              <td>{user.role}</td>
              <td>{user.updatedAt}</td>
              <td>
                <select
                  value={user.role}
                  onChange={e =>
handleRoleChange(user.id, e)}
                >
                  <option
value="Користувач">Користувач</option>
```

```

                                <option
value="Модератор">Модератор</option>
                                <option
value="Власник">Власник</option>
                                <option
value="Адміністратор">Адміністратор</option>
                                </select>
                                </td>
                                </tr>
                                </tbody>
                                </table>
                                </div>
);
};

export default UserPanel;

//Створення React компонента для відображення діаграм
import React, { useEffect } from 'react';
import { useDispatch, useSelector } from 'react-redux';
import { fetchData } from './actions';
import AirQualityChart from './components/AirQualityChart';
import TemperatureChart from './components/TemperatureChart';
import HumidityChart from './components/HumidityChart';
import RadiationChart from './components/RadiationChart';
import ParticulateMatterChart from
'./components/ParticulateMatterChart';
import GasChart from './components/GasChart';

const Dashboard = () => {
  const dispatch = useDispatch();
  const data = useSelector(state => state.data);

  useEffect(() => {
    dispatch(fetchData());
  }, [dispatch]);

  return (
    <div className="dashboard">
      <AirQualityChart data={data.airQuality} />
      <TemperatureChart data={data.temperature} />
      <HumidityChart data={data.humidity} />
      <RadiationChart data={data.radiation} />
      <ParticulateMatterChart data={data.particulateMatter}
/>
      <GasChart data={data.gas} />
    </div>
  );
};

export default Dashboard;

```

```

//Компоненти для отримання діаграм
// AirQualityChart.js
import React from 'react';
import { Line } from 'react-chartjs-2';

const AirQualityChart = ({ data }) => {
  const chartData = {
    labels: data.map(item => item.date),
    datasets: [
      {
        label: 'Air Quality Index',
        data: data.map(item => item.value),
        fill: false,
        backgroundColor: 'rgba(75,192,192,0.4)',
        borderColor: 'rgba(75,192,192,1)'
      }
    ]
  };

  return <Line data={chartData} />;
};

export default AirQualityChart;

//обробник для зміни інтервалу.
import React, { useEffect, useState } from 'react';
import TimeFilter from './components/TimeFilter';

const Dashboard = () => {
  const dispatch = useDispatch();
  const data = useSelector(state => state.data);
  const [interval, setInterval] = useState('daily');

  useEffect(() => {
    dispatch(fetchData(interval));
  }, [dispatch, interval]);

  return (
    <div className="dashboard">
      <TimeFilter onChange={setInterval} />
      <AirQualityChart data={data.airQuality} />
      /* Інші компоненти для діаграм */
    </div>
  );
};

export default Dashboard;

```

## SQL запити створення всіх таблиць бази даних в системі

```
-- User Table
CREATE TABLE User (
    userID INT PRIMARY KEY,
    username VARCHAR(255),
    password VARCHAR(255),
    email VARCHAR(255),
    roleID INT,
    FOREIGN KEY (roleID) REFERENCES Role(roleID)
);
-- Role Table
CREATE TABLE Role (
    roleID INT PRIMARY KEY,
    roleName VARCHAR(255)
);
-- Location Table
CREATE TABLE Location (
    locationID INT PRIMARY KEY,
    country VARCHAR(255),
    city VARCHAR(255),
    address VARCHAR(255)
);
-- Platform_Types Table
CREATE TABLE Platform_Types (
    platformID INT PRIMARY KEY,
    typeName VARCHAR(255),
    modelName VARCHAR(255),
    properties JSON
);
-- IoTDevice Table
CREATE TABLE IoTDevice (
    deviceID INT PRIMARY KEY,
    deviceType VARCHAR(255),
    userID INT,
    sensorID INT,
    actuatorID INT,
    locationID INT,
    platformID INT,
    device_status VARCHAR(255),
    FOREIGN KEY (userID) REFERENCES User(userID),
    FOREIGN KEY (sensorID) REFERENCES Sensor(sensorID),
    FOREIGN KEY (actuatorID) REFERENCES Actuator(actuatorID),
    FOREIGN KEY (locationID) REFERENCES Location(locationID),
    FOREIGN KEY (platformID) REFERENCES Platform_Types(platformID)
);
-- Actuator Table
CREATE TABLE Actuator (
    actuatorID INT PRIMARY KEY,
    actuatorType VARCHAR(255),
    actuator_status VARCHAR(255)
```

```

);
-- Sensor Table
CREATE TABLE Sensor (
    sensorID INT PRIMARY KEY,
    sensorType VARCHAR(255),
    sensor_status VARCHAR(255)
);
-- Humidity Sensor Table
CREATE TABLE HumiditySensor (
    humidityID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    humidityLevel FLOAT,
    FOREIGN KEY (sensorID) REFERENCES Sensor(sensorID)
);
-- Temperature Sensor Table
CREATE TABLE TemperatureSensor (
    temperatureID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    temperature FLOAT,
    FOREIGN KEY (sensorID) REFERENCES Sensor(sensorID)
);
-- Pollution Sensor Table
CREATE TABLE PollutionSensor (
    pollutionID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    pollutionLevel FLOAT,
    pollutionType VARCHAR(255),
    FOREIGN KEY (sensorID) REFERENCES Sensor(sensorID)
);
-- Solid Particles Sensor Table
CREATE TABLE SolidParticlesSensor (
    solidParticlesID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    particleCount INT,
    particleSize FLOAT,
    FOREIGN KEY (sensorID) REFERENCES Sensor(sensorID)
);
-- Radiation Sensor Table
CREATE TABLE RadiationSensor (
    radiationID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    radiationLevel FLOAT,
    FOREIGN KEY (sensorID) REFERENCES Sensor(sensorID)
);
-- Gas Sensor Table
CREATE TABLE GasSensor (
    gasID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,

```



```

        gasType VARCHAR(255),
        gasConcentration FLOAT,
        FOREIGN KEY (sensorID) REFERENCES Sensor(sensorID)
    );
-- Humidity Sensor History Table
CREATE TABLE HumiditySensorHistory (
    historyID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    humidityLevel FLOAT,
    FOREIGN KEY (sensorID) REFERENCES HumiditySensor(sensorID)
);
-- Temperature Sensor History Table
CREATE TABLE TemperatureSensorHistory (
    historyID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    temperature FLOAT,
    FOREIGN KEY (sensorID) REFERENCES TemperatureSensor(sensorID)
);
-- Pollution Sensor History Table
CREATE TABLE PollutionSensorHistory (
    historyID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    pollutionLevel FLOAT,
    pollutionType VARCHAR(255),
    FOREIGN KEY (sensorID) REFERENCES PollutionSensor(sensorID)
);
-- Solid Particles Sensor History Table
CREATE TABLE SolidParticlesSensorHistory (
    historyID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    particleCount INT,
    particleSize FLOAT,
    FOREIGN KEY (sensorID) REFERENCES
SolidParticlesSensor(sensorID)
);
-- Radiation Sensor History Table
CREATE TABLE RadiationSensorHistory (
    historyID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    radiationLevel FLOAT,
    FOREIGN KEY (sensorID) REFERENCES RadiationSensor(sensorID));
-- Gas Sensor History Table
CREATE TABLE GasSensorHistory (
    historyID INT PRIMARY KEY AUTO_INCREMENT,
    sensorID INT,
    timestamp DATETIME,
    gasType VARCHAR(255),
    gasConcentration FLOAT,
    FOREIGN KEY (sensorID) REFERENCES GasSensor(sensorID));

```