

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

ОПАЛІНСЬКИЙ Владислав Дмитрович
«Вебсервіс моніторингу пристроїв Інтернету речей /
Web service for IoT devices monitoring»

спеціальність: 123 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерна інженерія
Кваліфікаційна робота

Виконав студент групи КІ-42
В.Д. Опалінський

Науковий керівник:
к.т.н. Н.Я. Савка

Кваліфікаційну роботу допущено
до захисту:

" ____ " _____ 20____ р.

Завідувач кафедри
_____ О.Л. Дубчак

Тернопіль – 2025

АНОТАЦІЯ

Опалінський В.Д. Вебсервіс моніторингу пристроїв інтернету речей. – Рукопис.

Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр» за спеціальністю 123 «Комп'ютерна інженерія», освітньо-професійна програма. Західноукраїнський національний університет, Тернопіль, 2025.

Робота написана обсягом 76 сторінок і містить 17 рисунків, 3 додатки та 36 джерел за переліком посилань. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розробка вебсервісу для моніторингу пристроїв інтернету речей в режимі реального часу. Методи дослідження включають методи аналізу та синтезу, порівняльного аналізу, проєктування баз даних, комп'ютерного моделювання, тестування.

У розробленому проєкті розглянуто задачу керування пристроями інтернету речей в режимі реального часу. Проаналізовано особливості архітектури інтернету речей, програмні засоби для їх реалізації та сучасні програмні системи на основі IoT-технологій. Сформульовано функціональні вимоги до вебзастосунку. Розроблено прототип системи моніторингу пристроїв IoT. Побудовано алгоритм проєктування архітектури системи розумного будинку на основі технологій інтернету речей. Здійснено програмну реалізацію розробленої архітектури. Результатами експериментальних досліджень підтверджено перспективність застосування обраних технологій до розробки систем моніторингу.

Практичне значення полягає у розробці веборієнтованої системи моніторингу пристроїв інтернету речей, що дає змогу реагувати на ситуації в режимі реального часу.

Ключові слова: ІНТЕРНЕТ РЕЧЕЙ, ВЕБОРІЄНТОВАНА СИСТЕМА, MYSQL, РОЗУМНИЙ БУДИНОК.

ANNOTATION

Opalinsky V.D. Web service for IoT devices monitoring. – Manuscript.

Qualification work for the Bachelor degree in specialty 123 “Computer Engineering”, educational and professional program. Western Ukrainian National University, Ternopil, 2025.

The work is written in 76 pages and contains 17 figures, 3 appendices and 36 references. The volume of graphic material is 2 sheets of A3 format.

The purpose of the qualification work is to develop a web service for real-time monitoring Internet of Things (IoT) devices.

The research methods include methods of analysis and synthesis, comparative analysis, database design, computer modeling, and testing.

The developed project addresses the task of real-time control over IoT devices. The features of IoT architecture, software tools for implementation, and modern software systems based on IoT technologies are analyzed. Functional requirements for the web application are formulated. A prototype of the IoT device monitoring system is developed. An algorithm for designing the architecture of smart home system based on IoT technologies is built. The software implementation of the developed architecture is carried out. The results of experimental research confirm the effectiveness and potential of the selected technologies for developing monitoring systems.

The practical value lies in the development of web-oriented IoT device monitoring system that enables real-time response to various events and states.

Keywords: INTERNET OF THINGS, WEB-ORIENTED SYSTEM, MYSQL, SMART HOME.

ЗМІСТ

Вступ.....	9
1 Аналіз інтернету речей та вебзастосунків.....	12
1.1 Аналіз інтернету речей.....	12
1.2 Архітектура систем IoT.....	15
1.3 Принципи розробки веб-інтерфейсів	20
1.4 Аналіз засобів керування та моніторингу пристроїв IoT.....	24
1.5 Постановка задачі кваліфікаційної роботи.....	26
2 Вебсервіс для керування та моніторингу IoT.....	28
2.1 Аналіз вимог до вебсервісу.....	28
2.1.1 Функціональні вимоги.....	28
2.1.2 Вимоги до безпеки.....	29
2.1.3 Вимоги до масштабованості.....	31
2.2 Вибір інструментів розробки вебзастосунку.....	32
2.3 Проектування архітектури за стосунку.....	35
2.4 Розробка прототипу вебсервісу.....	38
3 Реалізація вебсервісу для моніторингу ІОТ.....	42
3.1 Реалізація вебзастосунку.....	42
3.2 Інтеграція вебсистеми з пристроями IoT.....	47
3.3 Тестування функціональності та безпеки вебсервісу.....	50
Висновки.....	54
Список використаних джерел.....	55
Додаток А Світлокопія публікації.....	59
Додаток Б Лістинг програмного коду.....	63
Додаток В Техніко-економічне обґрунтування проєкту.....	69

					КР.КІ.10660598.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив	Опалінський В.Д.				ВЕБСЕРВІС МОНІТОРИНГУ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ	Літ.	Арк.	Акрушів
Перевір.	Савка Н.Я.						8	76
Консульт.	Савка Н.Я.					ЗУНУ.ФКІТ. КІ-42		
Н. Контр.	Дубчак Л.О.							
Затвердив	Дубчак Л.О.							

ВСТУП

Моніторинг пристроїв Інтернету речей (IoT) є надзвичайно актуальним в умовах цифрової трансформації. З розвитком технологій та масовим поширенням розумних пристроїв зростає потреба у постійному контролі їхньої роботи, безпеки та ефективності.

Основною причиною підвищеної уваги до моніторингу є стрімке зростання кількості IoT-пристроїв у різних сферах: від промисловості та транспорту до медицини й розумних будинків. Ці пристрої забезпечують автоматизацію процесів, збір даних та їх аналіз, що суттєво покращує продуктивність та комфорт користувачів. Водночас розширення мережі IoT створює нові виклики, пов'язані з кібербезпекою, надійністю роботи та управлінням величезними обсягами даних.

Моніторинг в таких умовах виконує критично важливу роль, уможливлючи своєчасне виявлення несправностей, запобігання можливим загрозам та оптимізацію використання ресурсів. Безперервний контроль параметрів роботи пристроїв допомагає запобігти збоям у системах, що особливо важливо в критичних сферах, таких як медицина чи транспорт. Наприклад, у лікарнях IoT-пристрої відстежують життєві показники пацієнтів, тому при несправності система моніторингу повинна миттєво повідомити персонал про необхідність втручання.

Ще одним важливим аспектом є захист даних, оскільки IoT-пристрої часто працюють із конфіденційною інформацією. Моніторинг дозволяє виявляти аномальну активність, що може свідчити про спроби зламу або несанкціонованого доступу. Таким чином, він виступає ключовим інструментом забезпечення кібербезпеки.

Окрім безпеки, моніторинг сприяє економії ресурсів, оскільки дозволяє оптимізувати використання електроенергії, води та інших важливих параметрів. У сфері розумного міста контроль IoT-систем уможливлює ефективніше керувати

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

транспортними потоками, системами освітлення чи опалення, знижуючи витрати та навантаження на навколишнє середовище.

Таким чином, моніторинг пристроїв Інтернету речей є необхідним елементом для забезпечення стабільної та безпечної роботи цифрових екосистем. Його розвиток сприятиме підвищенню надійності IoT-систем, їхньої безпеки та ефективності, що, у свою чергу, забезпечить подальший прогрес у сфері технологій.

В сучасному світі, де веб-технології набувають все більшого розповсюдження, виникає необхідність розроблення ефективних веб-інтерфейсів для керування та моніторингу IoT-пристроїв. Зважаючи на це, дослідження варто присвятити створенню такого інтерфейсу, який забезпечує потребу в інтуїтивно зрозумілому, надійному та ефективному засобі для взаємодії з широким спектром пристроїв IoT. Адже стрімкий розвиток технологій IoT призводить до збільшення кількості пристроїв, які щодня використовуються – це від смарт-будинків до промислових систем. Однак, незважаючи на такий прогрес, питання зручного та уніфікованого засобу керування пристроями залишається відкритим, що й зумовило мету кваліфікаційної роботи, а саме – розробка вебінтерфейсу для керування та моніторингу пристроїв Інтернету речей.

Зазначена мета вимагає виконання таких задач:

- проаналізувати особливості Інтернету речей;
- дослідження архітектури IoT-систем;
- дослідження сучасних технологій та програмних інструментів розробки веб-інтерфейсів для керування та моніторингу IoT-пристроїв;
- сформулювати вимоги до веб-інтерфейсу системи моніторингу;
- обґрунтувати вибір технологій та інструментів для розробки вебсервісу;
- розробити архітектуру вебсистеми;
- здійснити програмну реалізацію вебсервісу;
- провести експериментальні дослідження розробленої системи;
- обґрунтувати техніко-економічні показники доцільності розробки вебсервісу.

					КР.KI.10660598.00.00.000.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Об’єкт дослідження. Процеси моніторингу пристроїв Інтернету речей.

Предмет дослідження. Алгоритм розробки вебінтерфейсу для керування та моніторингу пристроїв Інтернету речей.

Методи дослідження. Для досягнення цілей проекту застосовані методи системного аналізу для визначення оптимальних способів інтеграції різноманітних пристроїв IoT у єдину систем, методи тезу програмування для мікроконтролерів, методи синтезу при проектуванні веб-інтерфейсу, методи комп’ютерного моделювання при дослідженні працездатності розробленої системи, методи тестування.

Практичне значення результатів дослідження. Розроблений вебінтерфейс управління IoT пристроями може бути використано для керування домашньою IoT системою, підвищуючи рівень комфорту та безпеки життєдіяльності в приватних будинках.

Виконання кваліфікаційної роботи здійснено, враховуючи вимоги, зазначені у [30-32], згідно мети та поставлених задач.

За основними результатами дослідження підготовлено та опубліковано тези доповіді на XII Всеукраїнській науково-практичній конференції молодих учених «Інформаційні технології – 2025». [34]. Копію публікації наведено у додатку А.

					КР.KI.10660598.00.00.000.ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ІНТЕРНЕТУ РЕЧЕЙ ТА ВЕБЗАСТОСУНКІВ

1.1. Аналіз інтернету речей

Концепція Інтернету речей передбачає об'єднання фізичних об'єктів у єдину цифрову мережу завдяки використанню вбудованих сенсорів, програмного забезпечення та інших сучасних технологій. Це дозволяє пристроям обмінюватися інформацією один з одним, а також взаємодіяти із зовнішніми системами через (див. рисунок 1.1).

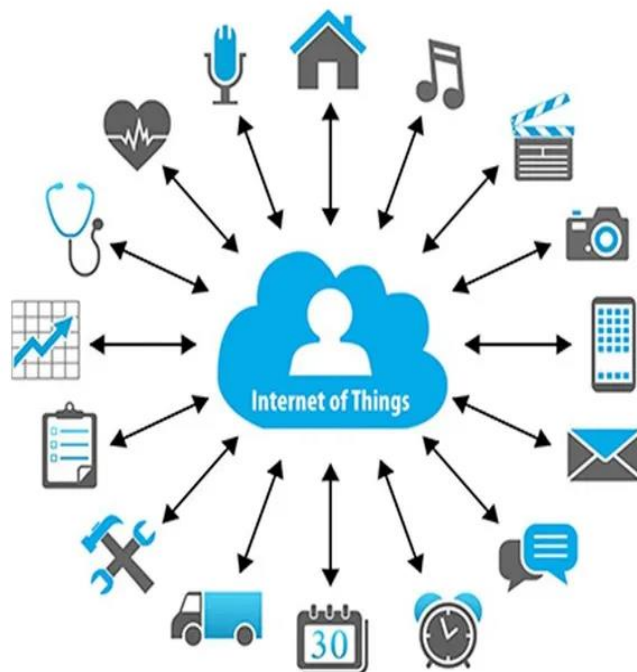


Рисунок 1.1 – Інтернет речей як концепція взаємодії фізичних об'єктів

На початку 1980-х років з'явилися перші пристрої, здатні взаємодіяти через мережу, зокрема, вендинговий апарат для продажу Coca-Cola в Університеті Карнегі-Меллон у 1982 році, який міг передавати інформацію про свій вміст і загальний стан [12].

Протягом 1990-х років відбувалось активне обговорення та дослідження можливостей мережевого зв'язку між пристроями. Видатні думки з цього приводу висловлювались у роботах Марка Вейзера, науковця з Xerox PARC, який

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

розробляв концепцію повсюдного комп'ютинга. Також, у 1999 році Білл Джой виступив з ідеєю "Шести вебів", яка включала прогнози щодо розвитку мережевих технологій.

Термін "Інтернет речей" ввів Кевін Ештон у 1999 році під час презентації для Procter & Gamble, обговорюючи вплив технології RFID на управління ланцюгами постачання. Водночас існує припущення, що Пітер Т. Льюїс використовував цей термін ще у 1985 році [1, 3].

Протягом 2000-х та 2010-х років IoT перетворився з теоретичної концепції у втілені реальні проєкти, як-от розумні міста (див. рис. 1.2), розумне виробництво та розумний транспорт. Цей період позначився суттєвим прогресом в інформаційних технологіях, зокрема у поширенні бездротових з'єднань і розвитку енергоефективних мереж.



Рисунок 1.2 – Розумне місто (Smart City)

Інтернет речей сьогодні є одним з найважливіших та найшвидше зростаючих напрямків у технологічній індустрії. Завдяки прогресу в області мікроелектроніки, телекомунікацій та інформаційних технологій, IoT втілює концепцію розумного зв'язку між різними пристроями та системами на глобальному рівні.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Останнім часом IoT привертає увагу завдяки зростанню доступності високошвидкісних інтернет-з'єднань і падінню вартості датчиків. Технології машинного навчання і штучного інтелекту інтегруються для аналізу великих обсягів даних з різних джерел, що дозволяє підвищити ефективність і автоматизацію процесів у багатьох галузях. IoT технології стали основою для багатьох інноваційних застосувань, включаючи [2, 6, 11]:

- розумні будинки і офіси від систем управління освітленням і температурою до автоматизованих систем безпеки;
- промисловий IoT – оптимізація виробничих процесів, підвищення ефективності роботи обладнання та покращення умов праці;
- смарт-сіті (розумні міста) – від систем управління трафіком до моніторингу якості повітря та води;
- охорона здоров'я – дистанційний моніторинг здоров'я пацієнтів, управління хронічними захворюваннями та оптимізація логістики в медичних закладах.

Подальший розвиток IoT обіцяє суттєві зміни у способі життя та роботі людей, зокрема:

- інтеграція з 5G – наступне покоління мобільного інтернету суттєво підвищить швидкість і знизить затримку зв'язку між пристроями;
- краєві обчислення (Edge computing) – обробка даних на пристроях без необхідності передачі їх на віддалені сервери, що підвищує швидкість реакції систем IoT;
- безпека. З появою більшої кількості IoT пристроїв, питання безпеки стає дедалі актуальнішим, вимагаючи розробки нових методів захисту даних і мереж.

Незважаючи на численні переваги, IoT також стикається з рядом викликів, включаючи проблеми з приватністю даних, управлінням великими обсягами даних та необхідністю стандартизації технологій і протоколів.

Сучасний стан і перспективи розвитку Інтернету речей відображають зростання інтересу та інвестицій у цю сферу. В майбутньому можна очікувати ще

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

більш ширшого впровадження IoT технологій, що забезпечить нові можливості для бізнесу та покращення якості життя.

1.2 Архітектура систем IoT

Архітектура систем Інтернету речей є багат шаровою та включає кілька ключових компонентів, кожен з яких відіграє важливу роль у забезпеченні зв'язку та взаємодії між пристроями та серверами (див. рисунок 1.3) Ця архітектура забезпечує не тільки збір та аналіз даних, але й управління пристроями в реальному часі.

Компоненти Архітектури IoT [14, 16, 20]:

1. сенсори / датчики – це фундаментальні елементи, які збирають різні типи даних зі свого середовища, зокрема, температура, вологість, тиск, тощо.
2. пристрої збору даних, які агрегують та передають дані від сенсорів до обробника даних. Ці пристрої можуть включати проміжні вузли, які допомагають у передачі даних;
3. пристрої з'єднання (шлюзи), які функціонують як мости між IoT пристроями та мережею. Вони здійснюють перетворення протоколів, управління даними та безпеку;
4. хмарні сервери – аналізують вхідні дані, зберігають їх і можуть відправляти команди назад до IoT пристроїв;
5. програмне забезпечення для аналізу даних – спеціалізоване ПЗ, яке обробляє великі обсяги даних, отриманих з IoT пристроїв, для подальшого аналізу і візуалізації.
6. Користувацькі інтерфейси дозволяють користувачам взаємодіяти з IoT системою, встановлювати налаштування, переглядати дані та отримувати сповіщення.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

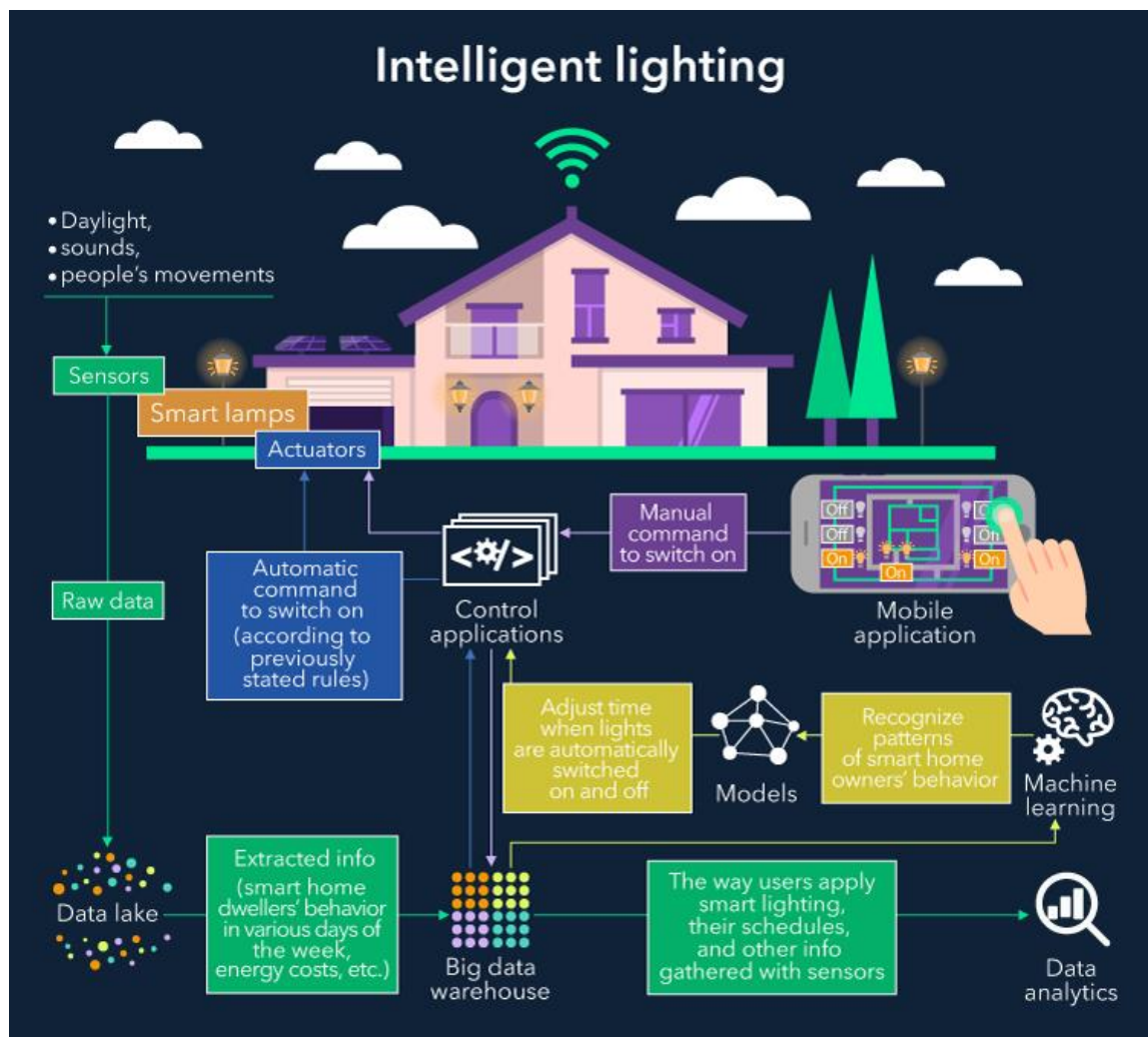


Рисунок 1.3 – Архітектура IoT

Процес взаємодії у системі IoT. Взаємодія в IoT починається з сенсорів, які збирають фізичні параметри з оточення. Дані від сенсорів передаються до приладів збору даних, які відіграють роль первинної обробки та агрегації. Шлюзи перетворюють ці дані у потрібний формат і передають їх через мережу до хмарних серверів або локальних обробних центрів. На хмарних серверах дані обробляються і аналізуються, після чого можуть генеруватися команди, які відправляються назад до пристроїв для виконання дій, наприклад, регулювання температури в будівлі. Користувацькі інтерфейси дозволяють користувачам отримувати звіти та сповіщення, а також контролювати систему в реальному часі.

Безпека в архітектурі IoT є критично важливою, оскільки системи повинні вирішувати питання ідентифікації, аутентифікації, конфіденційності, інтеграції даних і доступу. Розробка захищених мережевих протоколів і застосування

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

шифрування є необхідними заходами для захисту даних, переданих між різними компонентами IoT.

Архітектура IoT включає комплексну взаємодію між різними технічними компонентами та програмним забезпеченням, які спільно забезпечують розумне та ефективне управління ресурсами і процесами в різних сферах людської діяльності. Рівневу архітектуру IoT показано на рисунку 1.4.

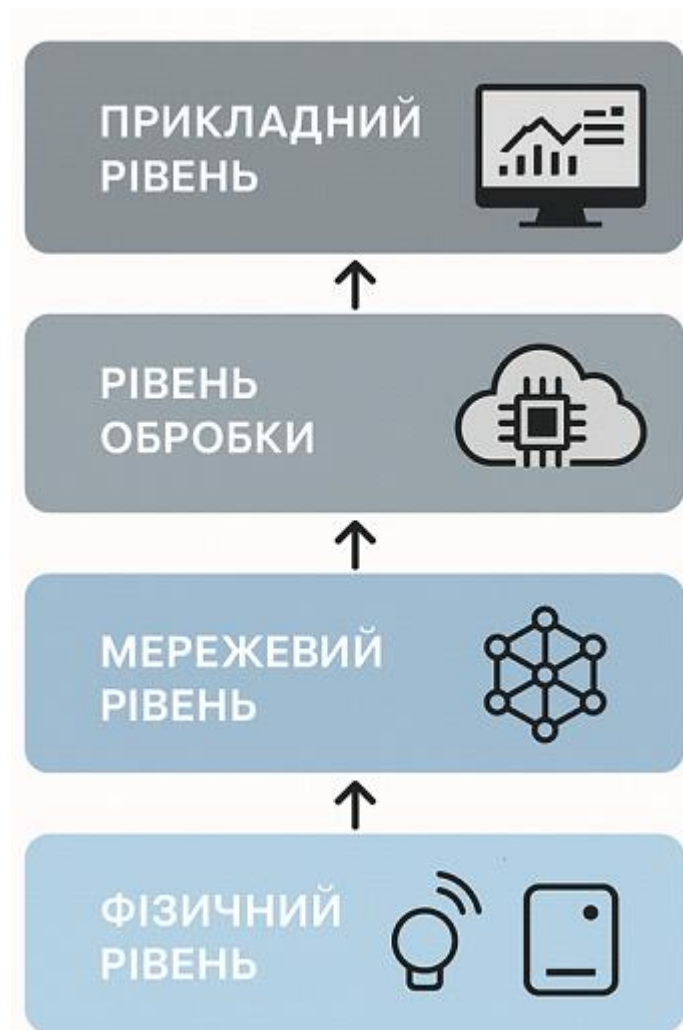


Рисунок 1.4 – Модель архітектури IoT

Деталізацію кожного рівня архітектури із зазначенням процесів, їх переходів із одного рівня до іншого, а також засобів реалізації показано на рисунку 1.5.

Детальний приклад IoT системи для розумного будинку, який використовує Raspberry Pi як центральний контролер розглянемо на рисунку 1.6. Система

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

включає різноманітні датчики, взаємодіє з хмарними сервісами та має інтуїтивно зрозумілий користувацький інтерфейс.

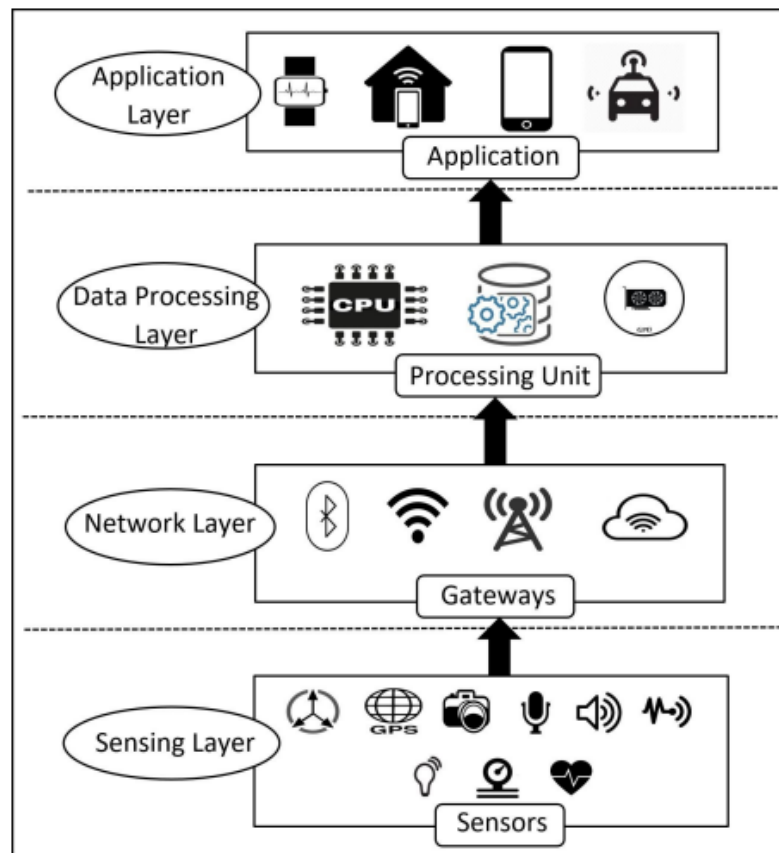


Рисунок 1.5 – Функціональна модель мережі IoT

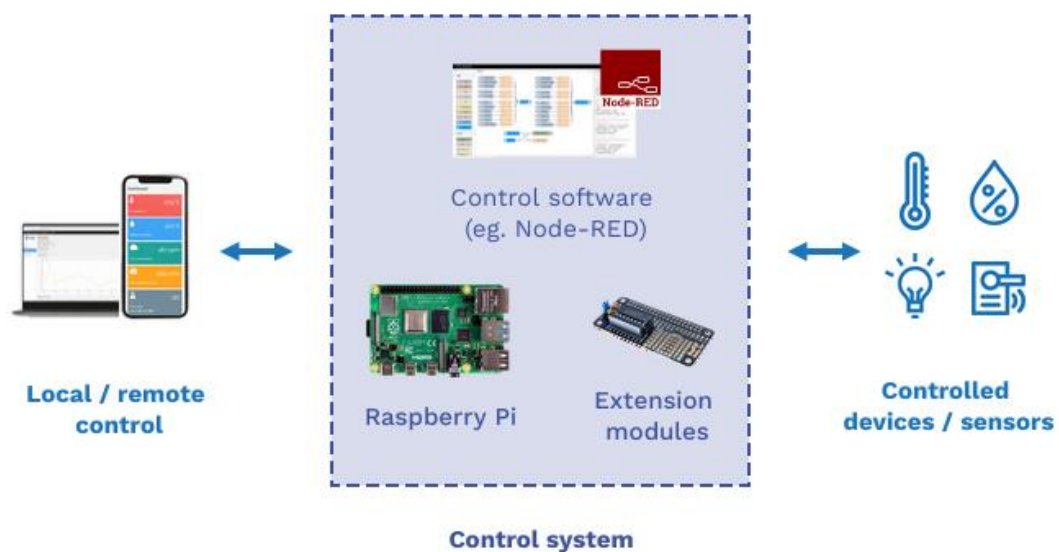


Рисунок 1.6 – Розумний дім на Raspberry Pi

Компоненти системи:

1. контролер Raspberry Pi 4 Model B. Цей міні-комп'ютер виступає як мозок системи, здатний обробляти дані, відправляти та отримувати команди від хмари та управляти іншими пристроями;

2. датчики:

– DHT22 для вимірювання температури та вологості повітря. Цей датчик дозволяє отримувати точні показники стану середовища в реальному часі;

– PIR Motion Sensor (HC-SR501) для виявлення руху в приміщенні, використовується для систем безпеки та автоматизації освітлення;

3. шлюз та мережеві компоненти – Raspberry Pi виконує функції шлюзу, перетворюючи дані з датчиків в цифровий формат та передаючи їх через Ethernet або Wi-Fi;

4. хмарний сервер:

– Amazon AWS IoT Core для обробки та аналізу даних. Платформа надає потужні інструменти для обробки потоків даних, зберігання даних та їх аналізу;

– MongoDB Atlas для зберігання даних у хмарі. Ця база даних ідеально підходить для зберігання великих обсягів даних з високою швидкістю запису та читання;

5. програмне забезпечення:

– Node-RED для візуалізації та управління потоком даних на Raspberry Pi. Цей інструмент дозволяє легко створювати автоматизацію та зв'язувати різні IoT компоненти між собою;

– Grafana для візуалізації даних на користувацькому інтерфейсі. Таке програмне забезпечення дозволяє користувачам переглядати статистику та метрики в зручній і наочній формі;

– користувацький інтерфейс Web dashboard, розроблений за допомогою HTML, CSS та JavaScript, що працює на сервері Node.js. Користувачі можуть взаємодіяти з системою, переглядати датчики та управляти налаштуваннями через веб-інтерфейс.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

Процес взаємодії полягає у наступному. DHT22 збирає дані про температуру і вологість, які Raspberry Pi читає та передає через MQTT на AWS IoT Core. Якщо дані показують підвищення температури вище встановленого порогу, Node-RED відправляє команду на Raspberry Pi активувати кондиціонер через реле. При виявленні руху HC-SR501, система може включити світло або надіслати сповіщення про безпеку. Користувачі можуть моніторити всі параметри через Grafana dashboard і керувати налаштуваннями через веб-інтерфейс.

Такий приклад демонструє, як за допомогою конкретних датчиків, серверів та програмного забезпечення можна створити комплексну інтегровану систему IoT, яка відповідає сучасним вимогам до автоматизації домашніх процесів та управління ресурсами.

1.3 Принципи розробки веб-інтерфейсів

Розглядаючи вимоги до веб-інтерфейсів, насамперед слід відзначити, що основною метою будь-якого веб-інтерфейсу є забезпечення простоти, інтуїтивної зрозумілості і комфортності для користувача. Важливо розробляти інтерфейси, які не тільки привабливі візуально, але й функціональні, ефективні та доступні для всіх користувачів, незалежно від їх технічних навичок або фізичних обмежень.

Користувацький досвід (UX) відіграє особливу роль, кожен елемент дизайну має бути продуманим таким чином, щоб забезпечити зручність користування (див. рисунок 1.7). Це стосується всього – від чіткості навігації та логічної структури сторінок до ефективності інтерактивних елементів. Наприклад, кнопки повинні бути виділені так, щоб їх легко знайти і натиснути, а інформація так структурована, щоб її можна було легко читати і розуміти [23].

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

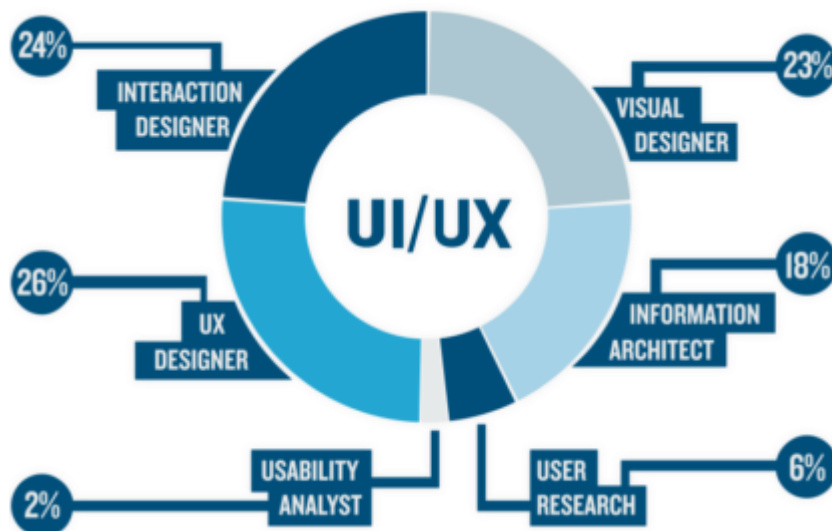


Рисунок 1.7 – UI/UX дизайн

Швидкість завантаження сторінок є ще одним ключовим аспектом. Відомо, що користувачі часто залишають веб-сторінки, які завантажуються більше ніж кілька секунд. Оптимізація зображень, мінімізація коду і використання сучасних технологій кешування уможливлюють скорочення часу завантаження і покращення загального досвіду використання вебсервісу.

Адаптивний дизайн став невід'ємною частиною сучасної веб-розробки. Це означає, що веб-сайт автоматично адаптується до розміру екрану пристрою, з якого він переглядається, будь то настільний комп'ютер, планшет або смартфон [23]. Використання гнучких макетів, зображень і CSS медіа-запитів дозволяє розробляти інтерфейси, які забезпечують зручне переглядання на будь-якому пристрої.

Доступність є фундаментальною вимогою, що забезпечує використання веб-інтерфейсів людьми з різними фізичними обмеженнями, такими як порушення зору, слуху, рухових функцій чи когнітивних здібностей. Це включає розробку сайтів, які можуть ефективно працювати з екранними читачами, мають достатній контраст кольорів, великий розмір шрифту та навігацію, яка може бути використана з клавіатури.

Забезпечення цих елементів може суттєво покращити загальну доступність інтерфейсу, роблячи веб-сайт придатним для широкого кола користувачів, та сприяти вищому ступеню задоволення від користування сайтом.

Стандарти розробки веб-інтерфейсів (див. рисунок 1.8) відіграють вирішальну роль у забезпеченні сумісності, доступності та надійності веб-сайтів [35]. Вони визначають, як веб-сторінки мають бути структуровані, як повинні взаємодіяти з користувачами, і як розробники повинні підходити до процесу їх створення, забезпечуючи дотримання найкращих практик та сучасних технологічних стандартів.

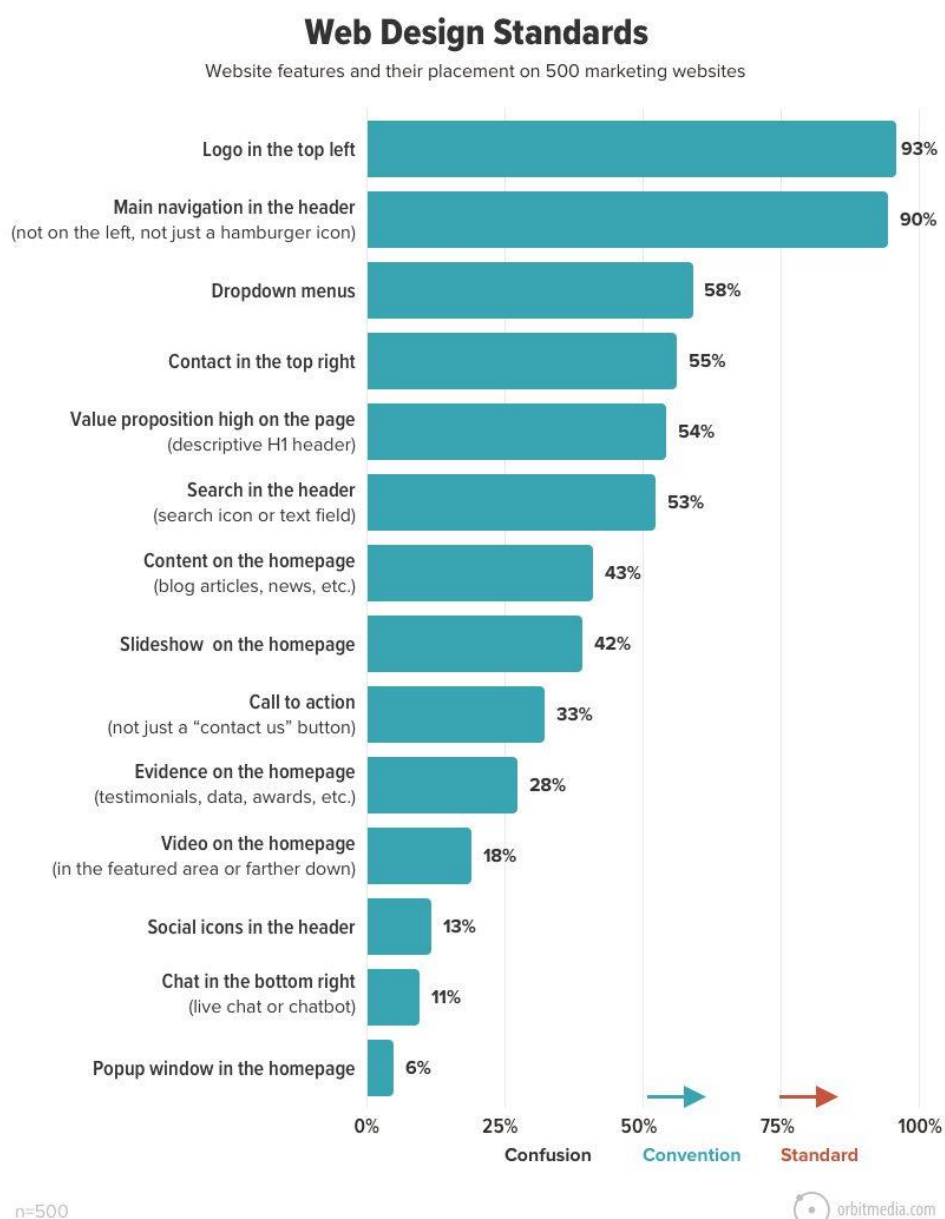


Рисунок 1.8 – Стандарти розробки веб-інтерфейсів

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Стандарти адаптивного веб-дизайну, також відомі як Responsive Web Design (RWD), гарантують, що веб-сайти ефективно працюють на різних пристроях, включаючи настільні комп'ютери, планшети та мобільні телефони. Це досягається за допомогою гнучких макетів, зображень та CSS-медіа-запитів, які автоматично адаптують вміст до розміру екрану користувача. Такий підхід забезпечує кращий досвід перегляду, що є важливим не тільки для користувача, але й для оптимізації пошукових систем.

Загалом, стандарти розробки веб-інтерфейсів важливі для створення високоякісних, доступних і сумісних веб-додатків, які можуть забезпечити доступність користувача на будь-якому пристрої та будь-якій платформі. Дотримання цих стандартів допомагає розробникам уникнути типових помилок та забезпечення функціональності та доступності для широкої аудиторії.

Технології для розробки веб-інтерфейсів безперервно розвиваються, пропонуючи розробникам нові можливості для створення інноваційних, ефективних і вражаючих веб-додатків. Основою для будь-якого веб-сайту є HTML, CSS і JavaScript, які є стандартними технологіями для створення структури, стилізації та інтерактивності веб-сторінок [33].

HTML (HyperText Markup Language) є основною мовою розмітки, що використовується для створення скелета веб-сторінки. Вона дозволяє розміщувати текст, зображення, відео та інші елементи на сторінці. HTML5 включає покращення для аудіо та відео, а також для більш складних аплікацій, таких як ігри та взаємодія з мультимедійними даними.

CSS (Cascading Style Sheets) використовується для контролю зовнішнього вигляду веб-сторінки, дозволяючи розробникам впливати на макет, кольори, шрифти і майже всі аспекти візуального представлення. CSS3 включає можливості анімації, трансформації та плавних переходів, які можуть суттєво покращити користувацький досвід.

JavaScript є мовою програмування, що застосовують для додавання інтерактивності на веб-сторінки. Вона дозволяє розробникам розробляти веб-сайти, які можуть реагувати на дії користувача в реальному часі без необхідності

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

перезавантажувати сторінку. Сучасні JavaScript фреймворки та бібліотеки – React, Angular і Vue.js, забезпечують розробникам потужні інструменти для побудови швидких додатків.

React є бібліотекою для побудови користувацьких інтерфейсів, яка зосереджується на декларативному створенні інтерфейсу та управлінні станом компонентів. Angular надає широкий набір інструментів для розробки веб-додатків, включаючи потужний фреймворк для використання моделі MVC (Model-View-Controller). Vue.js, характеризуючись легкістю і гнучкістю, дозволяє розробникам швидко розпочати розробку та масштабувати за потреби.

Препроцесори CSS, зокрема, SASS і LESS, спрощують процес створення CSS, додаючи можливості змінних, функцій та міксинів, що робить стилі більш читабельними та легшими для управління. Ці інструменти допомагають розробляти структурований та ефективний код, що є важливим для великих проектів.

В сукупності ці технології формують основу для сучасної веб-розробки, надаючи розробникам інструменти необхідні для створення інтерактивних, візуально привабливих і технічно складних веб-сайтів. Інновації в цих технологіях продовжують впливати на якість розробки веб-інтерфейсів, забезпечуючи кращий досвід для користувачів.

1.4 Аналіз засобів керування та моніторингу пристроїв IoT

Системи керування та моніторингу IoT відіграють ключову роль у функціонуванні та ефективності різних IoT-пристроїв та додатків. Ці системи дозволяють користувачам відстежувати стан пристроїв, управляти ними в реальному часі та збирати цінні дані для подальшого аналізу. Сьогодні на ринку існує чимало засобів, що варіюють від відкритих платформ до власних рішень

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

великих технологічних компаній [29, 36]. Нижче представлений огляд найбільш популярних.

AWS IoT Core – це хмарна платформа від Amazon, яка дозволяє з легкістю підключати IoT-пристрої до хмарних сервісів (див. рисунок 1.9). Вона забезпечує безпечне та надійне з'єднання між пристроями та хмарними додатками, дозволяючи обмін даними в режимі реального часу. AWS IoT Core підтримує понад мільйон пристроїв і може обробляти трильйони повідомлень, дозволяючи розробникам легко масштабувати свої додатки.

Azure IoT Hub від Microsoft – це ще одне масштабоване рішення для керування IoT-пристроями. Це хмарна служба, яка допомагає збирати, зберігати та аналізувати дані з множини пристроїв. Azure IoT Hub підтримує широкий спектр протоколів з'єднання та надає інструменти для інтеграції з іншими сервісами Azure, такими як Azure Functions, Machine Learning та Azure Stream Analytics.

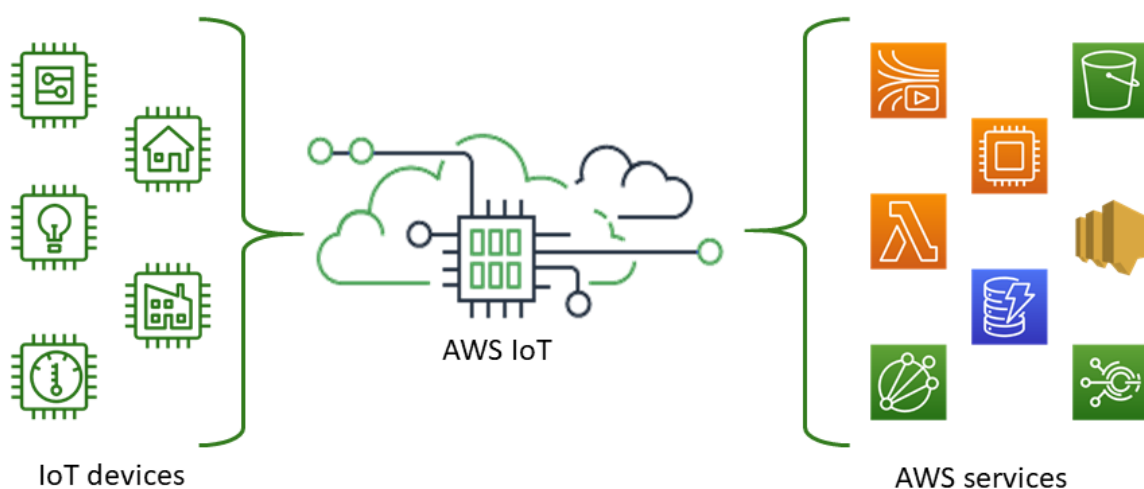


Рисунок 1.9 – Ядро AWS IOT

Google Cloud IoT Core – повнофункціональна платформа для інтеграції IoT-пристроїв з хмарними сервісами Google Cloud. Платформа дозволяє легко управляти пристроями через стандартні протоколи MQTT та HTTP. Google Cloud IoT Core інтегрується з іншими сервісами Google для аналізу даних, зокрема,

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

BigQuery, Cloud Dataflow та Cloud Machine Learning Engine, забезпечуючи потужні аналітичні можливості.

IBM Watson IoT Platform – хмарна платформа, яка надає засоби для підключення, управління та аналізу даних з IoT-пристроїв. Вона включає потужні аналітичні інструменти та інтеграцію з IBM Watson для реалізації можливостей штучного інтелекту та машинного навчання. IBM Watson IoT Platform також забезпечує високий рівень безпеки та приватності даних.

ThingSpeak – відкрита IoT-аналітична платформа, що уможливлює агрегацію, візуалізацію та аналіз потоків даних у реальному часі. Вона досить популярна серед хабістів та дослідників завдяки своїй гнучкості та простоті інтеграції з багатьма популярними IoT-платформами, зокрема, Arduino та Raspberry Pi.

OpenHAB – відкрите програмне забезпечення для домашньої автоматизації, яке дозволяє інтегрувати та керувати різними IoT-пристроями вдома. Воно підтримує велику кількість пристроїв і технологій та пропонує гнучкість для створення індивідуальних засобів домашньої автоматизації.

Наведені платформи та програмні засоби забезпечують широкий спектр функціональності для моніторингу та управління IoT-пристроями, від базових хмарних підключень до складних аналітичних інструментів з використанням штучного інтелекту. Вибір правильної платформи залежить від конкретних потреб проекту, вимог до масштабованості та безпеки.

1.5 Постановка задачі кваліфікаційної роботи

У теперішньому світі технології Інтернету речей активно впроваджуються у різні сфери діяльності, забезпечуючи автоматизацію, віддалений контроль та управління пристроями. Однак із зростанням кількості IoT-пристроїв виникає потреба у надійному моніторингу їхнього стану, продуктивності та безпеки.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Важливим аспектом є розробка ефективного вебсервісу, який дозволяє централізовано відстежувати роботу пристроїв у режимі реального часу, аналізувати отримані дані та оперативно реагувати на можливі відхилення у їх функціонуванні. Існуючі вебзасоби для керування пристроями інтернету речей часто мають важкий для розуміння інтерфейс, що унеможлиблює якісне користування тим пересічному користувачу.

У результаті аналізу обгрунтовано вибір програмних засобів, які забезпечать масштабованість та гнучкість вебсервісу, а також розглянуто технології збору, обробки та візуалізації даних. Особливу увагу у вебсервісі такого типу слід приділити питанням безпеки, оскільки IoT-середовище часто працює через відкриті мережі й є вразливим до атак і несанкціонованого доступу.

Таким чином, метою дослідження виступає розробка ефективного механізму збору даних із пристроїв, оптимізація алгоритмів обробки інформації та створення інтуїтивно зрозумілого веб-інтерфейсу для користувачів, що вимагає виконання таких задач:

- сформулювати вимоги до веб-інтерфейсу системи моніторингу;
- обгрунтувати вибір технологій та інструментів для розробки вебсервісу;
- розробити архітектуру веб-системи;
- здійснити програмну реалізацію вебсервісу;
- провести експериментальні дослідження розробленої системи;
- обгрунтувати техніко-економічні показники доцільності розробки вебсервісу.

У наступних розділах кваліфікаційної роботи зупинимося на розв’язуванні вищесформульованих задач із детальним викладенням усіх проведених досліджень та описом здійснених розробок.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ВЕБСЕРВІС ДЛЯ КЕРУВАННЯ ТА МОНИТОРИНГУ ІОТ

2.1 Аналіз вимог до вебсервісу

2.1.1 Функціональні вимоги

Розглянемо детально формальні та неформальні вимоги до розробки вебсервісу. Насамперед опишемо вимоги до керування пристроями.

Інтерфейс керування – веб-інтерфейс повинен забезпечувати кнопки або перемикачі для керування станом пристроїв (наприклад, включення / виключення світла, регулювання температури, тощо).

Налаштування пристроїв – користувачі можуть налаштовувати параметри пристроїв, такі як температурні пороги для системи опалення чи кондиціонування.

Автоматичні сценарії – веб-інтерфейс дозволяє розробляти та активувати сценарії, що автоматично керують пристроями залежно від визначених умов (наприклад, включення опалення при падінні температури нижче заданого рівня).

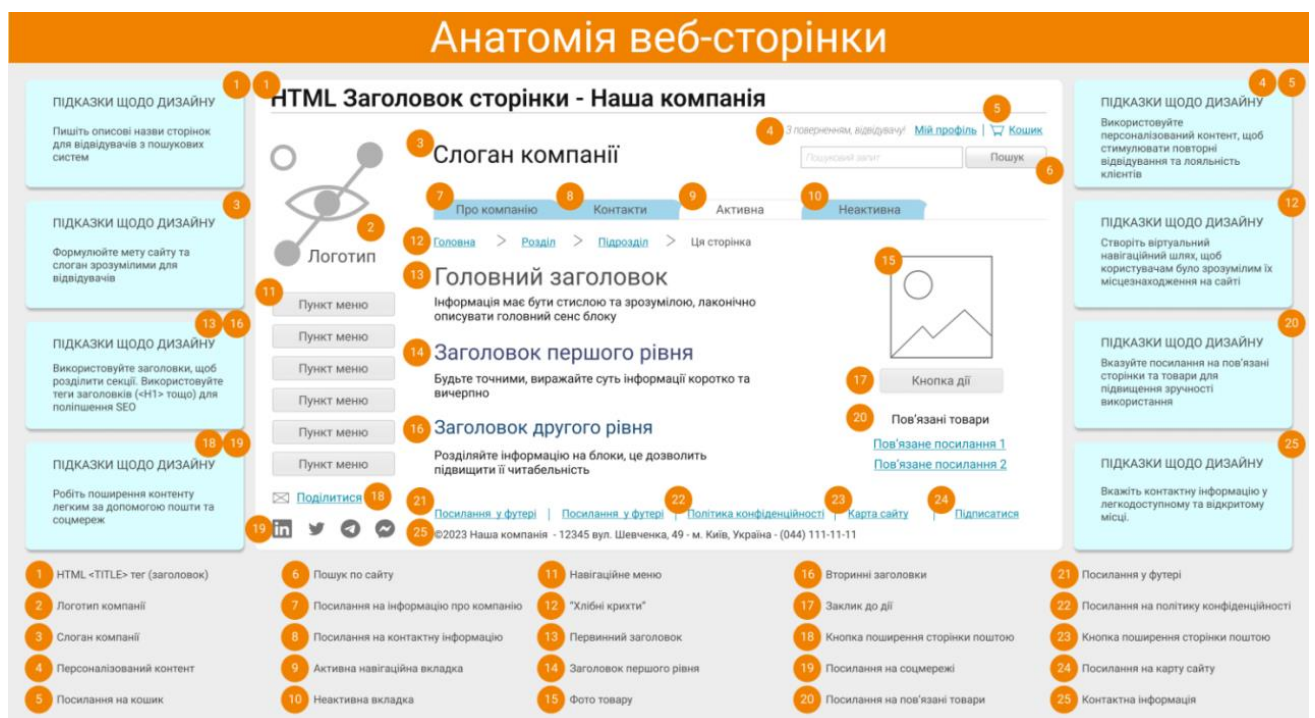


Рисунок 2.1 – Структура вимог до веб-інтерфейсу

Вимоги до системи моніторингу стану:

- відображення даних у реальному часі – веб-інтерфейс надає інформацію про поточний стан температури та вологості в приміщенні, отриману з датчиків;
- історія показників – користувачі можуть переглядати історію змін показників за обраний період. Дані відображаються у вигляді графіків та таблиць;
- сповіщення – веб-інтерфейс надсилає сповіщення користувачам через електронну пошту або SMS при виявленні критичних змін у показниках або станах системи.

Система аналізу даних повинна забезпечувати такі функції:

- збір та зберігання даних – всі дані з датчиків збираються та зберігаються в базі даних MySQL для подальшого аналізу;
- генерація звітів – користувачі можуть генерувати звіти про стан системи та зміни показників для аналізу ефективності роботи обладнання;
- візуалізація даних – використання графічних компонентів для наглядного представлення аналітичних даних, таких як графіки зміни температури та вологості.

Такі вимоги забезпечать розробку зручного, функціонального та відповідального веб-інтерфейсу для ефективного керування та моніторингу пристроїв IoT.

2.1.2 Вимоги до безпеки

Безпека є ключовою складовою будь-якої системи, що керує та моніторить IoT пристрої, оскільки вона забезпечує захист даних та систем від несанкціонованого доступу або шкідливих атак. Нижче наведено детальний опис вимог до безпеки для спроектованого веб-інтерфейсу.

Автентифікація та авторизація:

					КР.KI.10660598.00.00.000.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

– багаторівнева автентифікація – впровадження двофакторної автентифікації (2FA) для забезпечення додаткового рівня безпеки при вході в систему;

– обмежений доступ – використання ролей та дозволів для контролю доступу користувачів до різних частин інтерфейсу в залежності від їхніх повноважень;

– аудит та логування – запис всіх операцій користувачів та системних подій для можливості відстежування та аналізу інцидентів безпеки.

Шифрування даних:

– шифрування трафіку – використання протоколу SSL/TLS для захисту даних, що передаються між клієнтськими пристроями та сервером;

– захист даних на сервері – шифрування чутливих даних, збережених у базі даних для захисту їх від можливих витоків у разі взлому системи;

– безпечне зберігання паролів – використання сучасних методів хешування паролів, таких як bcrypt або Argon2, для забезпечення безпеки паролів користувачів.

Захист від зловмисних атак:

– захист від SQL-ін'єкцій – застосування підготовлених запитів (prepared statements) та ORM систем для взаємодії з базою даних для мінімізації ризику SQL-ін'єкцій;

– захист від XSS атак – запровадження строгих правил валідації та санітації вводу даних на стороні сервера і клієнта для запобігання виконання шкідливих скриптів;

– захист від CSRF атак – використання токенів для захисту від міжсайтових запитів на подання (CSRF) шляхом перевірки справжності запитів, що відправляються користувачами.

Зазначені вимоги до безпеки уможливають розробку міцної основи для захисту системи управління та моніторингу IoT-пристроїв, забезпечуючи конфіденційність, цілісність та доступність даних та ресурсів.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

2.1.3 Вимоги до масштабованості

Вимоги до масштабованості веб-інтерфейсу для керування та моніторингу пристроїв Інтернету речей повинні бути ретельно продумані, щоб система могла адаптуватися до зростаючої кількості пристроїв, користувачів та даних без втрати продуктивності та ефективності.

На першому етапі важливо розглянути горизонтальне масштабування системи. Це означає здатність системи розширюватися за допомогою додавання нових серверів або інстанцій до вже існуючої інфраструктури, що дозволяє обробляти більшу кількість запитів та даних. Такий підхід передбачає використання безстанових веб-серверів, де сесії користувачів управляються через централізоване сховище сесій, що дозволяє додавати нові сервери без переривання роботи системи.

Поряд із горизонтальним масштабуванням, необхідно враховувати модульність системи. Система має бути розроблена таким чином, що до неї легко додавати нові компоненти або оновлювати існуючі без змін в основному коді. Це означає, що кожен компонент системи повинен мати чітко визначений інтерфейс взаємодії з іншими компонентами, що спрощує інтеграцію та заміну частин системи.

Важливим аспектом масштабованості є також відмовостійкість. Система повинна бути здатна продовжувати роботу навіть у випадку часткової втрати компонентів або відмов обладнання. Це можна досягти за допомогою редундантності критичних компонентів та використання стратегій реплікації даних. Також слід забезпечити постійний моніторинг системи, щоб оперативно виявляти та усувати несправності.

Забезпечення масштабованості системи дозволить витримувати зростання навантаження, як від збільшення кількості підключених пристроїв, так і від розширення функціональності системи. Такий підхід допоможе зберегти високу продуктивність та забезпечити надійність в умовах змінних обсягів використання.

					КР.KI.10660598.00.00.000.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2 Вибір інструментів розробки вебзастосунку

Розробка веб-інтерфейсу системи моніторингу ґрунтується на технологіях HTML, CSS, Bootstrap та JavaScript, а також взаємодії з контролером і платою керування, що використовують середовище Arduino IDE та мову програмування Arduino C (див. рисунок 2.2) [27, 28, 33].



Рисунок 2.2 – Frontend-технології розробки системи моніторингу IoT-пристроїв

В першу чергу, необхідно забезпечити взаємодію між веб-інтерфейсом та контролером через веб-сервер, вбудований в контролер або використовуючи протоколи зв'язку, такі як HTTP або WebSocket. Для цього потрібно створити API на контролері, яке може обробляти запити від веб-інтерфейсу та відправляти необхідні команди або дані.

Пізніше можна використовувати JavaScript для реалізації веб-інтерфейсу, який відображатиме дані, зчитані з контролера, та дозволить користувачу взаємодіяти з пристроями IoT через інтуїтивний інтерфейс. Використання Bootstrap допоможе забезпечити респонсивний дизайн та швидку розробку інтерфейсу.

Крім того, зручність розробки забезпечується за рахунок використання середовища Arduino IDE, що здійснює програмування контролера, яке надасть доступ до бібліотек та інструментів розробки програмного забезпечення для IoT пристроїв. Мова програмування Arduino C є основною мовою розробки програм для контролерів Arduino та інших сумісних плат.

Загалом, вибір таких технологій та інструментів дозволить ефективно розробляти веб-інтерфейс керування та моніторингу пристроїв IoT, забезпечуючи зручну взаємодію користувача та ефективну роботу пристроїв у системі.

Для бекенду веб-інтерфейсу, що включає обробку запитів від клієнтів, взаємодію з базою даних та виконання бізнес-логіки, можна використати PHP та MySQL [33], що продемонстровано на рисунку 2.3.



Рисунок 2.3 – Backend технології

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

PHP – це серверна мова програмування, яка широко використовується для розробки динамічних веб-сайтів та веб-додатків. Вона включає набір функцій для роботи з HTTP-запитами, обробки форм, роботи з базами даних. PHP є досить популярним для проєктів невеликої та середньої складності.

MySQL – реляційна система управління базами даних, яку часто використовують разом з PHP для зберігання та керування даними. Система надає можливості зберігання, організації та маніпулювання даними у вигляді таблиць зі зв'язкам, підтримує різні типи даних, операції з базами даних та механізми забезпечення цілісності та безпеки даних.

У проєктованому веб-інтерфейсі на базі PHP та MySQL можна реалізувати логіку обробки запитів від клієнтів, взаємодію з базою даних для отримання або збереження інформації та виконання необхідних операцій з даними. Наприклад, можна розробити PHP-скрипти, які обробляють запити на отримання даних про стан пристроїв IoT з бази даних та відправляють їх клієнту через веб-сервер, або приймають дані від клієнта та зберігають їх у базі даних для подальшого аналізу чи використання.

Розміщення веб-додатку з інтерфейсом та приватними даними вимагає ретельного вибору хостингового сервісу або хмарної технології, щоб забезпечити безпеку та конфіденційність інформації [17, 21, 35] (див. рисунок 2.4).



Рисунок 2.4 – Хмарні технології

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

Опишемо найбільш популярні варіанти. Amazon Web Services (AWS) пропонує різноманітні сервіси, які можна сконфігурувати для розміщення веб-додатків з високим рівнем безпеки. Наприклад для розгортання серверів, Amazon RDS для баз даних, Amazon S3 для зберігання статичних файлів та резервних копій даних.

Google Cloud Platform (GCP) надає широкий спектр сервісів для розміщення веб-додатків та забезпечення їх безпеки. Зокрема, можна застосувати Google Compute Engine для серверів, Cloud SQL для баз даних та Google Cloud Storage для зберігання даних.

Microsoft Azure пропонує послуги хмарного хостингу, які дозволяють розміщувати веб-додатки та бази даних в безпечному середовищі. Azure App Service, Azure SQL Database та Azure Storage – це деякі з сервісів, які можуть бути використані із вказаною метою.

Heroku – це платформа як сервіс (PaaS), яка спрощує розгортання та управління веб-додатками. Вона забезпечує безпеку даних за допомогою інструментів автентифікації, авторизації та захисту даних.

Незалежно від обраного сервісу, важливо правильно налаштувати права доступу, фаєрволи та інші механізми безпеки для забезпечення обмеженого доступу до даних лише для адміністратора або користувачів системи. Також можна розглянути використання віртуальних приватних мереж (VPN) та інших технологій для додаткового захисту даних.

2.3. Проектування архітектури застосунку

Основні компоненти системи IoT забезпечують архітектуру системи моніторингу пристроїв. Периферійні пристрої (Edge Devices) – це датчики, які здійснюють збір інформації, зокрема, світлові, звукові, електромагнітні,

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

спеціалізовані (газ, спектр). Виконавчі механізми забезпечують механічну роботу, зокрема, управління освітленням, температурою, механічними замками тощо.

Рівень периферійного обчислення (Edge Computing Layer) здійснює перетворення даних, зокрема, мінімальна функціональність для перетворення аналогових сигналів від датчиків у цифровий формат (див. КР.KI.10660598.00.00.000 С1).

Взаємодію з веб-інтерфейсом забезпечує:

- Backend (центр обробки даних) – зберігання і обробка даних від периферійних пристроїв, управління логікою IoT рішень та обробка запитів від веб-інтерфейсу, API для комунікації між backend та frontend;

- Frontend (клієнтська частина) – розробка користувацьких інтерфейсів для моніторингу та управління IoT пристроями, інтерактивність з користувачем через веб-браузери або мобільні додатки.

Проектування системи моніторингу IoT-пристроїв включає забезпечення таких вимог:

- масштабованість – система повинна підтримувати можливість збільшення кількості пристроїв та користувачів без втрати продуктивності;

- безпека – захист даних і управління доступом через сучасні методи аутентифікації та шифрування;

- низьке енергоспоживання – оптимізація роботи периферійних пристроїв для зниження витрат на енергію.

Така архітектура лягає в основу детального проектування веб-інтерфейсу, забезпечуючи зручний та ефективний доступ до функцій управління та моніторингу IoT рішень.

При проектуванні користувацького інтерфейсу визначають кольорову палітру візуальних елементів, підбирають шрифти, візуально представляють функціонал системи та безпосередньо реалізують функції інтерфейсу. Таким чином, проектування користувацького інтерфейсу в межах веб-інтерфейсу керування та моніторингу IoT-засобів, вимагає реалізації певних кроків.

					КР.KI.10660598.00.00.000.ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Визначення основних візуальних елементів – вибір кольорів, які сприятимуть зручності сприйняття інформації та не будуть втомлювати очі користувачів при тривалій роботі.

Типографіка – підбір шрифтів, які добре читаються на екранах різних пристроїв і при різному освітленні.

Іконографія та символіка – розробка інтуїтивно зрозумілих ікон, які візуально представляють функціонал системи.

Розробка макетів та елементів управління:

- структура інтерфейсу – визначення місця розташування ключових компонентів інтерфейсу, таких як навігаційні меню, панелі інструментів, основне робоче вікно;

- адаптивний дизайн – забезпечення зручного перегляду та взаємодії з веб-інтерфейсом на різних пристроях: десктопах, планшетах, смартфонах;

- взаємодія з елементами управління – проектування кнопок, перемикачів, списків вибору для забезпечення зручної взаємодії користувача з системою.

Тестування інтерфейсу на зручність використання (usability):

- прототипування – розробка та тестування інтерактивних прототипів для оцінки зручності використання інтерфейсу до його фінальної розробки;

- залучення кінцевих користувачів – проведення тестувань з реальними користувачами для збору даних зворотного зв'язку і внесення необхідних корективів;

- аналіз зворотного зв'язку – оцінка відгуків користувачів та корекція дизайну на основі отриманих даних.

Реалізація таких кроків уможливить розробку ефективного та інтуїтивно зрозумілого інтерфейсу користувача, який сприятиме зручному моніторингу та керуванню IoT-пристроями.

Для моделювання взаємодій користувача з системою основну увагу варто приділити визначенню та аналізу основних користувацьких сценаріїв. Це включає розробку детальних сценаріїв взаємодії користувачів із системою, які дозволяють виявити ключові потреби та очікування користувачів. Особливу увагу слід

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

приділити забезпеченню інтуїтивно зрозумілої навігації та доступу до функцій системи, щоб користувачі могли ефективно управляти IoT-пристроями та отримувати необхідну інформацію. Впровадження методів тестування інтерфейсу з реальними користувачами також є критично важливим для аналізу зручності користування та задоволення їхніх потреб. Такі дії дозволяють виявити можливі проблеми в інтерфейсі та внести корективи до фінальної версії продукту, оптимізувавши взаємодію користувачів із системою з метою досягнення кращих результатів.

2.4 Розробка прототипу вебсервісу

Розробка інтерфейсу користувача для веб-інтерфейсу IoT, створеного на основі HTML, CSS, Bootstrap та JavaScript [26-29], включає чимало важливих аспектів, які забезпечують зручність та функціональність. Зокрема, інтерфейс поділено на модулі, кожен з яких візуально представлений іконками та графіками, що сприяє інтуїтивному розумінню та зручності управління та моніторингу даних. Ці модулі включають інструментальні панелі для швидкого доступу до основних функцій системи, а також графічні відображення даних, які автоматично оновлюються в реальному часі. Приклад розробленого модуля програмного коду що забезпечує веб-інтерфейс наведено нижче:

```
<script>
```

```
function updateTemperature() {
```

```
    const temperature = (20 + Math.random() * 5).toFixed(2); //
```

Випадкова температура від 20 до 25

```
    document.getElementById('average-temperature').textContent =
```

```
    temperature + ' °C';
```

```
}
```

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

setInterval(updateTemperature, 5000); // Оновлювати кожні 5 секунд
updateTemperature(); // Початкове оновлення
</script>
<div class="col-md-6">
  <div class="card">
    <div class="card-body">
      <h5 class="card-title">Керування Воріт</h5>
      <p id="gate-icon" style="font-size: 2rem;">▼</p> <!-- Стартова
іконка (закриті ворота) -->
      <button id="gate-toggle" class="btn btn-primary">Відкрити
Ворота</button>
    </div>
  </div>
</div>

```

З метою підвищення зручності користувачів, інтерфейс адаптований для перегляду на мобільних пристроях. Використання Bootstrap уможливорює забезпечення адаптивності та правильності відображення на різних пристроях без потреби в розробці окремих версій для кожного типу пристроїв. Розробка реагує на зміни розміру екрану та орієнтації, оптимізуючи інтерфейс з метою покращення взаємодії та зручності користувача.

Структуру прототипу інтерфейсу користувача представлено на рисунку 2.5. Інтерфейс включає модулі з можливістю налаштування їх користувачем, дозволяючи кожному адаптувати робоче середовище під свої потреби, що передбачає персоналізацію дашбордів, вибір віджетів, які найбільше використовуються, та налаштування параметрів відображення даних. Це дозволяє кожному користувачу створити найбільш комфортне та ефективне робоче місце.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Панель моніторингу та керування IoT приватного будинку



Рисунок 2.5 – Структура прототипу інтерфейсу користувача

Веб-інтерфейс також передбачає впровадження додаткових функцій безпеки, зокрема шифрування даних та безпечна аутентифікація користувачів, що забезпечує захист чутливої інформації та системних налаштувань. Враховуючи важливість безперервного доступу та управління системою IoT, інтерфейс користувача враховує не тільки зручність та адаптивність, але й надійність та безпеку взаємодій з системою.

					КР.KI.10660598.00.00.000.ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Розробка серверної частини веб-інтерфейсу для систем управління та моніторингу IoT є критично важливою задачею, що забезпечує надійність, безпеку та ефективність обробки даних. Основною метою серверної частини є збір, обробка та зберігання даних, отриманих від IoT-пристроїв, а також забезпечення функціональності, необхідної для взаємодії з користувацьким інтерфейсом.

Серверну архітектуру розроблено з врахуванням потреб швидкого відгуку та високої доступності. Застосування масштабованих технологій та фреймворків, зокрема, Node.js або Python Flask, дозволяє обробляти запити в реальному часі, що є важливим для IoT систем [25]. Зазначені технології підтримують асинхронне виконання, що оптимально для взаємодії з великою кількістю одночасних користувацьких сесій та IoT-пристроїв.

Зберігання даних організовано на основі бази даних, які забезпечують надійність та гнучкість у роботі з великими об'ємами даних та їхніми структурами. Це дозволяє ефективно виконувати операції CRUD (створення, читання, оновлення, видалення), важливі для управління даними в IoT.

Безпека серверної частини є ключовою, оскільки IoT системи часто включають управління критично важливими ресурсами та обробку чутливих даних. Впровадження заходів безпеки, таких як шифрування даних, безпечні API ключі, та методи аутентифікації, є обов'язковими для захисту системи від несанкціонованого доступу та можливих зловмисних атак.

Також серверна частина включає логіку для автоматичного виявлення та реагування на події від IoT пристроїв, що уможливорює оперативну обробку ситуацій із перевищенням порогових значень датчиків або відмови обладнання. Реалізація логіки обробки та розподілу повідомлень між модулями системи забезпечує високий рівень інтегрованості та модульності всієї IoT платформи.

В сукупності, серверна частина веб-інтерфейсу формує фундамент для надійного та безпечного управління IoT системами, підтримуючи постійну взаємодію між інтерфейсом користувача та периферійними IoT-пристроями.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ВЕБСЕРВІСУ ДЛЯ МОНІТОРИНГУ ІОТ

3.1. Реалізація вебзастосунку

Розробку веб-інтерфейсу здійснено на основі розв'язування задачі керування розумним будинком, яка на сьогодні є досить поширеною та актуальною. Лістинг програмного коду наведено у додатку Б. Розберемо структуру та використані технології коду HTML, CSS, і JavaScript, який імітує керування розумним будинком в IoT:

Структура HTML:

- основна структура – документ включає базову структуру HTML з `<head>` і `<body>` секціями;
- метадані – в `<head>` використовуються метатеги для встановлення набору символів і відображення веб-сторінки на різних пристроях;
- залежності – підключені зовнішні бібліотеки Bootstrap для CSS і JS, а також бібліотека Chart.js для відображення графіків;
- стилі – вбудовані стилі для карточок і тексту, що дає можливість налаштувати відображення елементів на сторінці;
- структура контенту – в `<body>` міститься `div` з класом `container`, в якому розміщені різні `div`-елементи з класом `col-md-6` для карточок, що показують різні аспекти моніторингу і керування будинком.

Елементи CSS:

- Bootstrap використовується для стилізації та відповідного відображення веб-інтерфейсу на різних пристроях;
- власні стилі – стилі в `<style>` секції задають маржини для карточок і кольорові відтінки для тексту, що забезпечує покращення візуальної чіткості і організацію інформації.

JavaScript елементи:

- динаміка інтерфейсу – скрипти відповідають за динамічне оновлення даних (температури, вологості, вживання енергії);

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

- інтерактивність – обробники подій на кнопках і перемикачах дозволяють користувачам взаємодіяти з веб-інтерфейсом (відкриття воріт, управління світлом);
- візуалізація даних – використання Chart.js для розробки графіків, що показують зміну вологості в часі.

Програмний код ефективно інтегрує HTML, CSS, і JavaScript для розробки інтерактивного веб-інтерфейсу, який може відображати дані моніторингу в реальному часі при керуванні розумним будинком на основі IoT. Приклад програмного коду, який реалізує модуль розробки аудіооб'єкта наведено нижче:

```
<script>
    // Створення аудіооб'єкта
    var audio = new Audio('zvuk.mp3');
    // Функція для активації звукового сповіщення
    document.getElementById('sound-alert').addEventListener('click', function()
{
    audio.play();
});
    // Функція для зупинки звуку
    document.getElementById('stop-sound').addEventListener('click', function()
{
    audio.pause();
    audio.currentTime = 0; // Скидування часу відтворення на початок
});
</script>
```

Візуалізація даних є важливим інструментом для розуміння складних концепцій та прийняття обґрунтованих рішень. У сфері енергетики візуалізація даних про споживання електроенергії уможливорює відстежувати та контролювати

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

енергоспоживання, виявляти потенційні можливості економії та приймати більш екологічні рішення.

Розробка картки HTML здійснюється на основі таких кроків:

1. створення контейнера картки на основі елемента `<div>` з класом `col-md-6` для розміщення картки на шести стовпчиках сітки Bootstrap;
2. створення елемента картки:
 - a) елемент `<div>` з класом `card` та атрибутом `text-center` для розробки візуальної картки та центрування її вмісту;
 - b) заголовок картки – елемент `<div>` з класом `card-header` для розміщення заголовка "Розхід Електроенергії";
 - c) тіло картки – елемент `<div>` з класом `card-body` для розміщення вмісту картки;
 - d) заголовок картки – елемент `<h5>` з класом `card-title` для розміщення заголовка "Поточний Розхід";
 - e) вмісту картки – два елементи `<p>`.

Елемент з класом `card-text` та ідентифікатором `power-usage` для відображення поточного споживання електроенергії, а елемент з класом `devices` та ідентифікатором `active-devices` для відображення списку активних приладів. У результаті реалізації таких кроків отримуємо картку HTML, яка візуалізує дані про поточне споживання електроенергії та активні прилади (див. рисунок 3.1) Картка має чіткий та лаконічний дизайн, який полегшує користувачам розуміння даних.

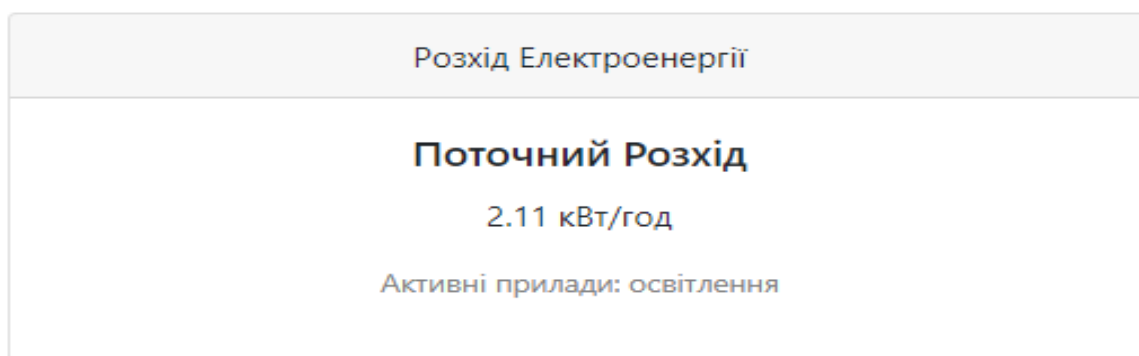


Рисунок 3.1 – Екранна форма елемента інтерфейсу

					КР.KI.10660598.00.00.000.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Опишемо дизайн веб-інтерфейсу для керування розумним будинком. Загальний дизайн веб-інтерфейсу створений таким чином, щоб бути інтуїтивно зрозумілим та візуально привабливим, з використанням сучасних веб-стандартів і фреймворків. Основні елементи дизайну описано нижче.

Використання кольорів. Дизайн використовує стандартну палітру Bootstrap, яка забезпечує чистий і професійний вигляд. Колірні акценти, такі як сірий для тексту підзаголовків (`class="devices"`), забезпечують просту орієнтацію в інформації, забезпечуючи візуальне розділення різних типів даних.

Кольори використовуються для індикації різних станів (зелений, жовтий, червоний для індикації рівня CO₂), що робить візуальне сприйняття статусу швидким і ефективним.

Типографіка. Веб-інтерфейс використовує стандартні шрифти, задані Bootstrap, що сприяє читабельності та естетичній консистенції. Розмір шрифтів і жирність тексту використовуються для визначення ієрархії інформації, з більшим і жирним текстом для заголовків.

Візуальні компоненти і лейаут. Інформація організована у вигляді карточок (`class="card"`), що дозволяє чітко відокремлювати різні секції управління та моніторингу. Це також спрощує користувачам зорове сканування сторінки для пошуку потрібної інформації. Використання кнопок і перемикачів (`class="btn"` та `class="custom-control-input"`) забезпечує легке та зрозуміле управління функціями розумного будинку.

Адаптивність. Завдяки використанню Bootstrap, веб-інтерфейс адаптується до різних розмірів екранів, забезпечуючи зручне використання як на настільних комп'ютерах, так і на мобільних пристроях.

Такий дизайн веб-інтерфейсу ефективно поєднує естетичну привабливість з функціональністю, створюючи приємний і зручний досвід для користувачів. Візуальна ієрархія, інтуїтивно зрозумілі контрольні елементи і адаптивність сприяють доступності та зручності широкій користувацькій аудиторії, що є суттєвим для систем управління розумним будинком.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтерактивність веб-інтерфейсу забезпечує користувачам можливість активно взаємодіяти з системою керування розумним будинком, здійснюючи моніторинг та управління в реальному часі. На рисунку 3.2 продемонстровано приклад системи керування розумним будинком на основі IoT. Розглянемо ключові інтерактивні елементи, використані у програмному коді.

Панель моніторингу та керування IoT приватного будинку

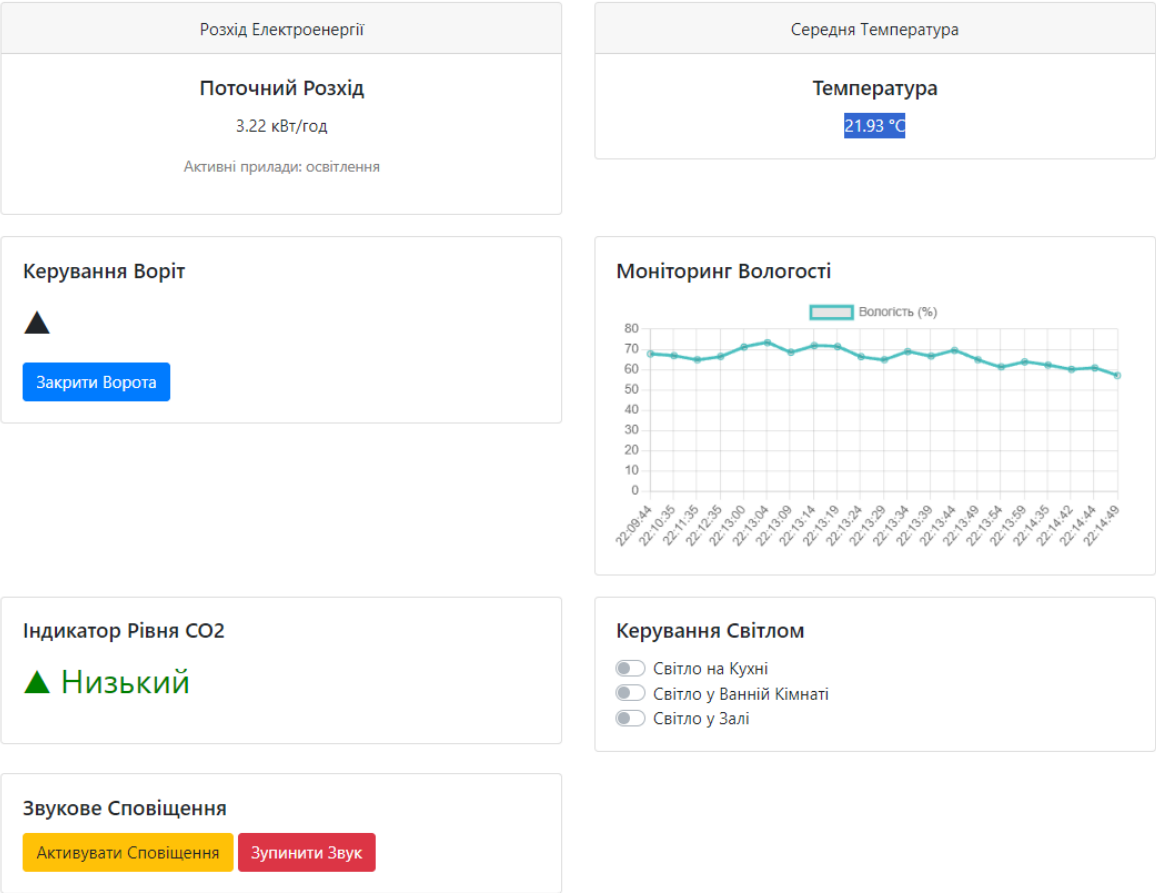


Рисунок 3.2 – Елементи вебінтерфейсу для керування розумним будинком

Кнопки керування воротами. Кнопка з текстом "Відкрити Ворота" дозволяє користувачам відкривати та закривати ворота. Стан воріт відображається зміною тексту кнопки на "Закрити Ворота" та іконкою воріт (▲ для відкритих та ▼ для закритих), що додає зрозумілості та сприяє зворотному зв'язку.

Перемикачі для керування світлом у різних кімнатах (кухня, ванна кімната,

вітальня) дозволяють користувачам легко управляти освітленням через інтерфейс. Вони реагують на зміни стану (ввімкнено/вимкнено), що відображається у системі управління.

Інтерактивний графік вологості, розроблений на основі Chart.js, динамічно оновлюється кожні 5 секунд. Користувачі можуть спостерігати зміну вологості у реальному часі, що є важливим для моніторингу стану параметрів будинку.

Індикатор CO₂ використовує кольорове кодування (зелений, жовтий, червоний) для показу рівнів CO₂, забезпечуючи користувачам швидке розуміння потенційної небезпеки без необхідності читати точні цифри.

Скрипти, які регулярно оновлюють дані про використання електроенергії, температуру, та інші показники, забезпечують динамічне оновлення інформації на інтерфейсі. Це включає як автоматичне оновлення тексту на веб-сторінці, так і зміну даних на графіках.

Кнопки для активації та зупинки звукових сповіщень дозволяють користувачам використовувати аудіосигнали для привернення уваги у випадку небезпеки або для інших повідомлень. Це додає ще один рівень інтерактивності та взаємодії.

У наступному підрозділі розглянемо детально приклад реалізації системи керування розумним будинком із пристроями IoT.

3.2. Інтеграція вебсистеми з пристроями IoT

Для реалізації проекту з інтеграції IoT-систем у розумному будинку використано мікроконтролер ESP-WROOM-32 ESP32, який підтримує Wi-Fi та Bluetooth. Така плата є ідеальною для контролю інтелектуальних домашніх пристроїв, завдяки своїй багатофункціональності і можливості збирати дані з різноманітних джерел та керувати ними через бездротові інтерфейси (див. рисунок 3.3).

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.3 – Плата розробника ESP-WROOM-32 ESP-32

Опишемо детально реалізацію процедури збору та відображення даних про показники розумного будинку.

Для реалізації процедури збору даних використано такі засоби:

- датчики температури, вологості та CO₂, які призначені для постійного моніторингу показників у приміщенні та надсилають дані на ESP32. Датчики вимірюють відповідні показники, які передаються через Wi-Fi до веб-сервера, інтегрованого в систему управління. Використання протоколу MQTT або HTTP(S) дозволяє забезпечити надійну та швидку передачу даних до веб-інтерфейсу.
- ватметри для моніторингу споживання електроенергії. Електроприлади в будинку оснащені ватметрами, які збирають дані про енергоспоживання кожного пристрою. Ця інформація надходить на ESP32, а звітти передається на сервер, де вона агрегується та відображається через веб-інтерфейс у реальному часі.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

– система контролю доступу через ворота з Wi-Fi. Стан воріт, які контролюються через Wi-Fi, також моніториться. Веб-інтерфейс отримує сигнали про відкриття чи закриття воріт і може керувати ними дистанційно.

Реалізацію процедури відображення даних здійснено на основі:

– інтерфейс користувача. Веб-інтерфейс, розроблений з використанням HTML, CSS, і JavaScript, динамічно відображає всі дані в реальному часі. Графіки та індикатори, створені на основі Chart.js та інших JavaScript бібліотек, показують поточні показники температури, вологості, рівнів CO₂, а також споживання енергії у вигляді легкодоступних діаграм;

– оновлення в реальному часі. Використання WebSockets або подібних технологій для встановлення двостороннього з'єднання між веб-інтерфейсом і сервером дозволяє забезпечити оновлення даних у реальному часі без необхідності перезавантаження сторінки;

– сповіщення. У разі виявлення аномалій у показниках датчиків або систем керування (наприклад, несподіване відкриття воріт чи високий рівень CO₂), система генерує сповіщення, які можуть бути представлені у вигляді звукових сигналів або спливаючих вікон у веб-інтерфейсі.

Такі механізми збору та відображення даних дозволяють користувачам ефективно моніторити та керувати IoT-пристроями, забезпечуючи високий рівень зручності та безпеки (див. KP.KI.10660598.00.00.000 C2).

Керування освітленням та воротами через веб-інтерфейс є критичним аспектом інтеграції веб-інтерфейсу з IoT-системами в контексті розумного будинку. Цей процес включає розробку і реалізацію функціоналу, який дозволяє користувачам дистанційно управляти освітленням та воротами через зручний веб-базований інтерфейс.

Система управління освітленням забезпечує можливість включення та виключення світла в різних кімнатах будинку. Це реалізовано через графічний інтерфейс у вигляді перемикачів (sliders) та кнопок, які з'єднані з IoT-контролерами, такими як ESP32. Користувач може керувати освітленням, змінюючи його інтенсивність або статус (ввімкнено/вимкнено) з будь-якого місця,

					KP.KI.10660598.00.00.000.ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

де є доступ до інтернету. Стан кожного освітлювального пристрою відображається в реальному часі, забезпечуючи актуальну інформацію про освітлення в будинку.

Керування воротами через веб-інтерфейс дозволяє користувачам відкривати та закривати ворота без необхідності фізичного контакту. Ворота з Wi-Fi контролерами можуть отримувати команди від сервера, що використовує веб-інтерфейс для відправлення запитів через захищене з'єднання. Ця система також може надсилати сповіщення користувачу про стан воріт, наприклад, чи вони зараз відкриті чи закриті, а також інформувати про будь-які несанкціоновані спроби доступу.

Управління цими системами через веб-інтерфейс вимагає належної безпеки, щоб запобігти несанкціонованому доступу та керуванню. Використання захищених протоколів зв'язку та аутентифікації необхідні для забезпечення захисту від несанкціонованого доступу.

3.3 Тестування функціональності та безпеки вебсервісу

Тестування відповідності вимогам реального часу важливе для перевірки здатності веб-інтерфейсу ефективно відображати дані та реагувати на користувацькі команди без затримок. Це включає перевірку швидкості відгуку інтерфейсу, час завантаження даних, а також спроможність системи оновлювати інформацію динамічно і стабільно.

Методологія тестування [13]. Тести швидкості реакції (Latency Tests) використано для перевірки часу відгуку системи на запити користувачів від моменту надсилання запиту до моменту отримання відповіді. Це включає тестування затримки мережі та швидкості обробки даних на сервері.

Тести продуктивності (Performance Tests) застосовано для оцінки здатності веб-інтерфейсу підтримувати стабільну роботу під час високого навантаження. Це

					КР.KI.10660598.00.00.000.ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

включає симуляцію великої кількості одночасних користувачів та запитів до сервера.

На основі тестів на реалізм оновлень (Real-time Update Tests) проведено верифікацію спроможності системи оновлювати інтерфейс в реальному часі без помилок та збоїв. Використання WebSockets чи інших технологій для забезпечення неперервної трансмісії даних між сервером та клієнтом.

Тести часу завантаження (Load Time Tests) слугують для оцінки загального часу, необхідного для завантаження веб-інтерфейсу та відображення початкових даних, що включає аналіз часу завантаження ресурсів, таких як скрипти, стилі та зображення.

Інструменти для тестування продуктивності, такі як JMeter, LoadRunner допомагають моделювати різні сценарії користування та вимірювати відповіді системи. Browser DevTools призначений для оцінки часу завантаження ресурсів та виконання веб-інтерфейсу, включаючи час до першого байта (TTFB). WebSocket тестування призначені для тестування імплементації WebSocket, які забезпечують перевірку стабільності та швидкості передачі даних.

Ефективне тестування відповідності вимогам реального часу забезпечує перевірку здатності сервісу ефективно взаємодіяти з користувачами, зменшуючи при цьому можливість помилок або збоїв під час експлуатації.

Тестування взаємодії веб-інтерфейсу з IoT компонентами є ключовим аспектом валідації, що забезпечує здатність системи коректно управляти різними пристроями в розумному будинку та отримувати від них дані. Цей процес включає ряд заходів, спрямованих на перевірку надійності, безпеки і швидкості взаємодії між веб-інтерфейсом, IoT-датчиками та контролерами.

Основою тестування є верифікація протоколів зв'язку, які використовуються для передачі команд і даних між сервером та IoT-пристроями. Це, зазвичай, включає HTTP, MQTT або WebSockets. Тести зосереджуються на здатності системи здійснювати бідирекційний обмін даними, що дозволяє не тільки відправляти команди на пристрої, але й приймати від них статусні повідомлення.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Важливим елементом є також тестування автоматизованих сценаріїв взаємодії, зокрема, автоматичного включення освітлення в певний час або реагування системи на зміну параметрів середовища, зафіксованих датчиками температури або вологості. Сценарії тестування включають імітацію змін умов середовища для перевірки реакції системи, а також тести на відмовостійкість, де система повинна коректно обробляти помилкові або нестандартні сигнали від пристроїв.

Тести безпеки є невід'ємною частиною процесу, оскільки інтерфейс досить часто зв'язується з різними компонентами IoT через відкриті мережі. Таке тестування включає верифікацію захисту переданих даних через шифрування, аутентифікацію пристроїв та засоби захисту від несанкціонованого доступу.

Комплексний підхід до тестування забезпечує не тільки функціональну працездатність системи, а й її здатність безпечно і ефективно інтегруватися з розгалуженою мережею IoT компонентів. Виконання цього тестування здійснює перевірку на здатність веб-інтерфейсу надійно управляти всіма аспектами розумного будинку: від освітлення до безпекових систем, забезпечуючи користувачам комфорт та надійний контроль.

Після завершення серії тестів на веб-інтерфейс для керування та моніторингу розумного будинку в IoT, аналіз результатів показав, що інтерфейс відповідає багатьом з вимог. Веб-інтерфейс демонструє продуктивність і стабільність під час оперування в реальному часі та ефективно взаємодіє з IoT компонентами. Проте, виявлено кілька недоліків, які варто виправити, зокрема:

- затримки при оновленні даних. При високому навантаженні на сервер, час оновлення даних може збільшуватися, що не відповідає критеріям реального часу.
- інтерфейс користувача. Деякі користувачі зазначили про перевантаженість інтерфейсу, особливо на мобільних пристроях, що ускладнює навігацію та контроль.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

– взаємодія з деякими IoT пристроями. Траплялися випадки, коли команди відправлені через веб-інтерфейс, не виконувались пристроями через проблеми з сумісністю або помилки в протоколах зв'язку.

Для покращення швидкості оновлення даних можна оптимізувати серверні запити та впровадити кешування даних, що знизить навантаження і покращить відгук.

Варто спростити користувацький інтерфейс, зробити його більш інтуїтивним і адаптивним до різних розмірів екранів. Із збереженням легко доступності основних функцій.

Також слід реалізувати додаткові протоколи зв'язку та стандарти, що покращать сумісність з різними типами пристроїв. При цьому доцільно провести додаткові тести з різними конфігураціями IoT компонентів.

У наступних дослідженнях варто розглянути можливість інтеграції голосових команд і машинного навчання для аналізу поведінки користувачів і автоматичного керування пристроями на основі звичок користувачів. Впровадження двофакторної аутентифікації та сучасних методів шифрування підвищить рівень захисту даних та управлінських команд. Розрахунок показників, які підтверджують економічну ефективність розробки вебсервісу наведено у додатку В.

					КР.KI.10660598.00.00.000.ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У кваліфакаційній роботі розв'язано актуальну практичну задачу розробки вебсервісу для моніторингу пристроїв Інтернету речей й при цьому отримано висновки.

1. Проаналізовано особливості Інтернету речей, їх архітектури, а також заводи моніторингу пристроїв IoT й при цьому зазначено, що існуючі сервіси часто є складними для розуміння.

2. Проаналізовано вимоги до вебсервісів для керування пристроями IoT й на цій основі обґрунтовано вибір програмних інструментів для проектування веб-системи моніторингу.

3. Розроблено прототип веб-системи, що відповідає поставленим вимогам до сучасних систем такого типу, із детальним описом її архітектури та функціональних можливостей.

4. Проведено програмну реалізацію вебсервісу на основі сучасних технологій, що забезпечує адаптивність та інтуїтивність при взаємодії з користувачами у режимі реального часу.

5. Здійснено тестування та експериментальні дослідження вебсервісу. Результатами досліджень підтверджено її працездатність при керуванні та моніторингу системи розумного будинку на основі IoT-пристроїв.

6. Проаналізовано техніко-економічні показники розробки вебсервісу й сформульовано висновки щодо доцільності розробки та перспектив щодо впровадження.

					КР.KI.10660598.00.00.000.ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Comput. Netw, 2010. Vol. 54, No. 15, pp. 2787–2805.
2. Bandyopadhyay D., Sen J. Internet of things: Applications and challenges in technology and standardization // Wireless Pers. Commun. 2011. Vol. 58, No. 1. P. 49-69.
3. He W., Xu L. Integration of distributed enterprise applications: A survey // IEEE Trans. Ind. Informat, 2014. Vol. 10, No. 1. P. 35-42.
4. ITU NGN-GSI Rapporteur Group. Requirements for Support of USN Applications and Services in NGN Environment. Switzerland, Geneva: International Telecommunication Union (ITU), 2010: веб-сайт. URL: https://www.itu.int/dms_pub/itu-t/oth/23/01/t23010000040002pdf.pdf (дата звернення 25.03.2025 р.).
5. Jia X., Feng O., Fan T., Lei Q. RFID technology and its applications in internet of things (IoT) // Proc. 2nd IEEE Int. Conf. Consum. Electron., Commun. Netw. (CECNet). China, Yichang, 2012. P. 1282-1285.
6. Kodali, R. K., Jain, V., Bose, S., and Boppana, L. «IoT Based Smart Security and Home Automation System» // Матеріали міжнародної конференції з обчислювальної техніки, зв'язку та автоматизації (ICCCA), IEEE, 2016. С. 1286-1289.
7. Li Q., Wang Z., Li W., Li J., Wang C., Du R. Applications integration in a hybrid cloud computing environment: Modelling and platform // Enterp. Inf. Syst. 2013. Vol. 7, No. 3. P. 1-35.
8. Li S., Xu L., Wang X. Compressed sensing signal and data acquisition in wireless sensor networks and internet of things // IEEE Trans. Ind. Informat. 2013. Vol. 9, No. 4. P. 2177-2186.

9. Li S., Xu L., Wang X., Wang J. Integration of hybrid wireless networks in cloud services oriented enterprise information systems // Enterp. Inf. Syst. 2012. Vol. 6, No. 2. P. 165-187.
10. Li Y., Hou M., Liu H., Liu Y. Towards a theoretical framework of strategic decision, supporting capability and information sharing under the context of Internet of Things // Inf. Technol. Manage. 2012. Vol. 13, No. 4. P 205-216.
11. Miorandi D., Sicari S., De Pellegrini F., Chlamtac I. Internet of things: Vision, applications and research challenges // Ad Hoc Netw. 2012. Vol. 10, No. 7. P. 1497-1516.
12. Ngai E. W. T., Moon K. K., Riggins F. J., Yi C. Y. RFID research: An academic literature review (1995–2005) and future research directions // Int. J. Prod. Econ. 2008. Vol. 112, No. 2. P. 510-520.
13. Ren L., Zhang L., Tao F., Zhang X., Luo Y., Zhang Y. A methodology towards virtualization-based high performance simulation platform supporting multidisciplinary design of complex products // Enterp. Inf. Syst. 2012. Vol. 6, No. 3. P. 267-290.
14. Sinha, P., and Wang, Y. «A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures.» IEEE Access, 2018. Vol. 6. P. 82721-82743.
15. Sun C. Application of RFID technology for logistics on internet of things // AASRI Procedia. 2012. Vol. 1. P. 106-111.
16. Tan L., Wang N. Future internet: The internet of things // Proc. 3rd Int. Conf. Adv. Comput. Theory Eng. (ICACTE). China, Chengdu, 2010. P.376-380.
17. Tao F., Laili Y., Xu L., Zhang L. FC-PACO-RM: A parallel method for service composition optimal-selection in cloud manufacturing system // IEEE Trans. Ind. Informat. 2013. Vol. 9, No. 4. P. 2023-2033.
18. Uckelmann D., Harrison M., Michahelles F. An architectural approach towards the future internet of things // Uckelmann D., Harrison M., Michahelles F. Architecting the Internet of Things. USA, NY: Springer, 2011. P. 1-24.

19. V. Gritsenko, L. Timashova “The Internet of Things in the Structure of the Clever Enterprise”, Control Systems and Computers, 2017. No. 1. P. 19–25.
20. Van Kranenburg R. The Internet of Things: A Critique of Ambient Technology and the All-Seeing Network of RFID. The Netherlands, Amsterdam: Institute of Network Cultures, 2007. 62 p.
21. Van Kranenburg R., Anzelmo E., Bassi A., Caprio D., Dodson S., Ratto M. The internet of things // Proc. 1st Berlin Symp. Internet Soc. Germany, Berlin, 2011. 84 p.
22. Wang L., Xu L., Bi Z., Xu Y. Data filtering for RFID and WSN integration // IEEE Trans. Ind. Informat. 2014. Vol. 10, No. 1. P. 408-418.
23. Ахмедова А., Сидоров Д., & Кузнєцова К. Розробка адаптивного веб-інтерфейсу для систем Інтернету речей // Матеріали міжнародної конференції з Інтернету речей та інтелектуальних систем, 2021. С. 123-128.
24. Вагнер, К. та ін. «Методи машинного навчання для прогнозованого обслуговування» // Матеріали міжнародної конференції IEEE з великих даних, 2014. С. 362-367.
25. Всесвітній консорціум W3C: веб-сайт URL: <https://www.w3.org/> (дата звернення 10.03.2015 р.).
26. Документація Bootstrap. «Вступ до Bootstrap»: веб-сайт. URL: <https://getbootstrap.com> (дата звернення 15.03.2015 р.).
27. Документація з HTML5: веб-сайт URL: <https://developer.mozilla.org/en-US/docs/Glossary/HTML5> (дата звернення 08.03.2015 р.).
28. Документація з JavaScript: веб-сайт URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення 18.03.2015 р.).
29. Коельо, В. Н. та ін. «Інтернет речей: Комплексний огляд технологій, архітектури та викликів». Технічний огляд IETЕ, 2018. Т. 35, № 2. С. 205-220.
30. Методичні вказівки до виконання практичних робіт з дисципліни «Техніко-економічне обґрунтування розробки комп’ютерних систем»/ Н.Я. Савка, І.Р. Паздрій / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 40 с.

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

31. Методичні вказівки до оформлення курсових, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп’ютерна інженерія» / І.В. Гураль, Л.О.Дубчак / під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 34 с.

32. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Бакалавр” спеціальності 123 «Комп’ютерна інженерія» галузі знань 12 Інформаційні технології / О.М. Березький, Л.О.Дубчак, Г.М. Мельник, Ю.М. Батько, О.Й. Піцун / Під ред. Л.О.Дубчак. Тернопіль: ЗУНУ, 2024. 54 с.

33. Морін, Д.Х., Харт, Е. і Раві, С. «Веб-технології: HTML, CSS, JavaScript, PHP, вебсервіси та Python». Prentice Hall, 3-тє видання, 2019. 576 с.

34. Опалінський В.Д., Лепа Р. Трансформація бізнес-процесів на основі технологій інтернету речей // Матеріали XII Всеукраїнської науково-практичної конференції молодих учених «Інформаційні технології – 2025». Київ: КСУ ім. Б. Грінченка, 15 травня 2025 р. С. 173-174.

35. Смірнов І., Петренко О., & Іванов С. Веб-інтерфейс для керування та моніторингу пристроїв IoT на основі хмарних обчислень // Журнал Інтернету речей, 2020. Вип. 12(4). С. 1234-1245.

36. Хілі, Д. «Платформи Інтернету речей для підключення підключених пристроїв». IoT for All, 22 березня 2021 року: веб-сайт. URL: <https://iotforall.com> (дата звернення 20.03.2025 р.).

					КР.КІ.10660598.00.00.000.ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		