

Міністерство освіти і науки України
Кам'янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп'ютерних наук

Кваліфікаційна робота бакалавра
з теми: **«Розробка архітектури хмарного сервісу для збору та
обробки даних від IoT-пристроїв»**

Виконав: здобувач вищої освіти
групи KN1-B21
спеціальності 122 Комп'ютерні
науки

Костинюк Владислав Вікторович

Керівник:

Слободянюк Олександр Васильович
доцент кафедри комп'ютерних наук

Рецензент: _____

Кам'янець-Подільський – 2025 р.

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці хмарного вебсервісу для збору, зберігання, обробки та візуалізації даних, отриманих від IoT-пристроїв. У межах проєкту реалізовано повнофункціональну систему, яка охоплює всі етапи: від надсилання телеметричних даних сенсорами до їх відображення через вебінтерфейс. Розробка базується на використанні мікрофреймворку Flask, хмарного хостингу Render та бази даних MongoDB Atlas.

Застосунок підтримує приймання та обробку даних у реальному часі, формування звітів у форматі PDF, а також надсилання електронних нагадувань про критичні події. Особливу увагу приділено розгортанню системи в хмарному середовищі, що забезпечує доступність сервісу з будь-якого пристрою, підключеного до Інтернету.

Результати роботи демонструють доцільність використання відкритих вебтехнологій для створення масштабованих IoT-рішень. Розроблена система може бути адаптована для задач моніторингу довкілля, агросфери або «розумного дому», а також слугувати навчальним прикладом в освітніх курсах із вебпрограмування, IoT та хмарних технологій.

Ключові слова: IoT, вебсервіс, Flask, MongoDB, хмарні технології, PDF-звіт, сенсорні дані, автоматизація.

ABSTRACT

The qualification thesis is dedicated to the development of a cloud-based web service for collecting, storing, processing, and visualizing data received from IoT devices. The implemented system provides a full data cycle: from sensor-generated telemetry to its display in a user-friendly web interface. The application is built using the Flask microframework, hosted on Render, and integrated with MongoDB Atlas as the cloud database.

The solution supports real-time data processing, PDF report generation, and automated email notifications for critical events. Special attention is given to deploying the system in a cloud environment, ensuring high availability and accessibility from any Internet-connected device.

The results confirm the feasibility of using open-source web technologies to build scalable IoT solutions. The developed system can be adapted for environmental monitoring, smart agriculture, or smart home projects, and may also serve as a teaching tool in courses on web development, IoT, and cloud computing.

Keywords: IoT, web service, Flask, MongoDB, cloud technologies, PDF report, sensor data, automation.

ЗМІСТ

АНОТАЦІЯ	2
ABSTRACT	3
ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1. Сучасні підходи до побудови IoT-систем.....	9
1.2. Основні компоненти IoT-архітектури.....	11
1.3 Огляд протоколів обміну даними в IoT	13
1.4 Хмарні платформи для обробки IoT-даних	16
1.5 Приклади реалізації простих IoT-проєктів	18
1.6 Висновки до розділу	20
РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ.....	22
2.1 Формулювання технічного завдання	22
2.2. Обґрунтування вибору інструментів	23
2.3. Архітектура системи та взаємодія компонентів.....	25
2.4. Діаграма потоків даних.....	27
2.5. Висновки до розділу	29
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ	31
3.1. Реалізація емулятора IoT-пристрою	31
3.2. Розробка серверної частини	32
3.3. Веб-інтерфейс	32
3.4. Розгортання сервісу в хмарі	34

3.5. Тестування та аналіз роботи системи	35
3.6. Висновки до розділу	37
Висновки	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	41

ВСТУП

Сучасна цифрова епоха характеризується стрімким розвитком технологій Інтернету речей (Internet of Things, IoT), які дозволяють автоматизовано збирати, передавати та аналізувати дані з фізичного середовища. Рішення на базі IoT усе ширше впроваджуються у сільське господарство, екологічний моніторинг, міську інфраструктуру, промисловість. У цьому контексті важливим завданням є створення таких архітектур систем, які дозволяють не лише збирати дані від сенсорів, а й ефективно їх обробляти, зберігати та візуалізувати у зручному форматі для користувача.

Особливу роль у цьому відіграють хмарні сервіси, які забезпечують доступність, масштабованість і гнучкість у розгортанні IoT-систем. Використання хмарних платформ дозволяє обробляти дані в режимі реального часу, забезпечувати централізований доступ і мінімізувати локальні ресурси для обробки інформації.

У межах кваліфікаційної роботи розроблено веборієнтовану систему, що дозволяє емулювати роботу IoT-пристрою, збирати телеметричні дані (температура, вологість), передавати їх на сервер через API, зберігати у хмарній базі даних та виводити у вебінтерфейсі. Реалізовано функціонал формування PDF-звітів, нагадувань про події, а також розгортання сервісу на платформі Render. Ознайомитися з результатом роботи можна за посиланнями:

- вебінтерфейс: <https://iot-project-zy46.onrender.com>
- вихідний

код:

<https://github.com/VladislavKostyniuk/iot-project>

Метою роботи є створення хмарного вебсервісу, що дозволяє приймати та обробляти дані від IoT-пристроїв, з використанням відкритого технологічного стеку (Python, Flask, MongoDB, Render).

Для досягнення цієї мети були визначені такі завдання:

- проаналізувати підходи до побудови архітектури IoT-систем;
- обґрунтувати вибір інструментів (Flask, MongoDB, хмарний хостинг);
- реалізувати серверну частину з API для приймання телеметрії;
- створити вебінтерфейс для візуалізації даних;
- реалізувати функціональність генерації PDF-звітів та сповіщень;
- розгорнути готову систему у хмарному середовищі та протестувати її роботу.

Об'єктом дослідження є архітектура систем з IoT-пристроями та хмарною обробкою даних. Предметом дослідження — вебтехнології та інструменти побудови хмарних сервісів для обслуговування IoT-інфраструктури.

Методологія реалізації проєкту включала системний аналіз, моделювання потоків даних, застосування принципів REST-архітектури, об'єктно-орієнтованого підходу до програмування, а також тестування продуктивності й коректності роботи сервісу.

Структура кваліфікаційної роботи охоплює три розділи. Перший розділ присвячено теоретичним основам побудови IoT-архітектур. У другому розділі здійснено проєктування архітектури системи, визначення її компонентів та вибір технологій. У третьому розділі

представлено реалізацію, розгортання сервісу, результати тестування та аналіз роботи системи.

ВИСНОВКИ

У межах кваліфікаційної роботи було реалізовано хмарний сервіс для збору та обробки даних від IoT-пристроїв, який демонструє можливість побудови доступної, масштабованої та функціональної системи з використанням сучасних вебтехнологій та інфраструктур хмарних платформ. Проведений аналіз предметної області засвідчив актуальність застосування IoT-рішень у різних галузях, зокрема в моніторингових системах, «розумних» середовищах, сільському господарстві, енергетиці та екології. Розглянуто архітектурні підходи до побудови IoT-систем, що дало змогу обґрунтовано обрати хмарну модель як найоптимальнішу для даного проєкту.

Проєктування системи охоплювало формування технічного завдання, обґрунтування вибору інструментів, розробку логіки взаємодії між компонентами, опис потоків даних. Реалізацію здійснено за допомогою Python, Flask, MongoDB, а хостинг виконано на платформі Render. Вебінтерфейс забезпечив можливість відображення результатів у зручному вигляді, а серверна частина забезпечила надійний прийом і зберігання даних.

У процесі тестування було підтверджено стабільну роботу системи в умовах різного навантаження, перевірено здатність обробляти неперервний потік вхідних даних, оцінено стійкість до збоїв і втрат зв'язку. Система коректно обробляє HTTP-запити, виконує валідацію даних і забезпечує наочне виведення інформації для користувача.

Розроблений прототип підтвердив ефективність використання відкритого стеку технологій для побудови повноцінної IoT-інфраструктури. Рішення може бути розширене або модифіковане

залежно від прикладної сфери — для аграрного моніторингу, систем безпеки, енергозбереження або інтелектуального управління просторами. Підвищення складності системи в майбутньому може включати роботу з реальними сенсорами, впровадження механізмів авторизації, генерацію статистичних звітів, використання сокетів для реалізації оновлення даних у реальному часі, інтеграцію з мобільними застосунками.

Таким чином, результати виконаної роботи демонструють доцільність використання хмарних технологій у поєднанні з IoT-рішеннями та відкритими засобами розробки для створення адаптивних цифрових сервісів із перспективою подальшого масштабування та інтеграції в різні сфери людської діяльності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Analysis of Cloud Platforms for IoT Services – Elsevier Procedia Computer Science, 2021.
2. Ashton, K. That 'Internet of Things' Thing // RFID Journal. – 2009. – URL: <https://www.rfidjournal.com/that-internet-of-things-thing>
3. AWS IoT Core Documentation – URL: <https://docs.aws.amazon.com/iot>
4. Azure IoT Hub Documentation – URL: learn.microsoft.com/en-us/azure/iot-hub
5. Bandyopadhyay, D., Sen, J. Internet of Things: Applications and Challenges in Technology and Standardization // Wireless Personal Communications. – 2011.
6. Best Practices for Cloud Deployment – Amazon Web Services Blog.
7. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. Cloud computing and emerging IT platforms // Future Generation Computer Systems. – 2009.
8. CoAP Protocol Specification. – IETF RFC 7252. – URL: tools.ietf.org/html/rfc7252
9. Fielding, R. T. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation. – University of California, Irvine, 2000.
10. Flask Documentation – URL: <https://flask.palletsprojects.com>
11. Google Cloud IoT Core Documentation – URL: cloud.google.com/iot-core
12. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions // Future Generation Computer Systems. – 2013.
13. HTTP/1.1 Protocol Specification. – IETF RFC 2616. – URL: <https://www.w3.org/Protocols/>
14. Intelligent Monitoring and Predictive Maintenance using IoT – IEEE Access, 2020.

15. Introduction to MongoDB – MongoDB University. – URL: <https://university.mongodb.com>
16. IoT Data Management Frameworks – Springer, 2022.
17. IoT Security Best Practices – CISCO White Paper, 2020.
18. IoT-based Smart Agriculture: An Overview – International Journal of Computer Applications, 2019.
19. JSON (JavaScript Object Notation) Specification – URL: www.json.org
20. Minerva, R., Biru, A., & Rotondi, D. Towards a Definition of the Internet of Things (IoT). – IEEE Internet Initiative, 2015.
21. MongoDB Documentation – URL: <https://www.mongodb.com/docs/>
22. MQTT Version 5.0. OASIS Standard. – URL: <https://mqtt.org/mqtt-specification/>
23. Python Requests Library Documentation – URL: <https://docs.python-requests.org>
24. Railway Cloud Platform – URL: <https://railway.app>
25. Render Hosting Platform – URL: <https://render.com>
26. REST API Design Rulebook / M. Masse. – O'Reilly Media, 2011.
27. Web API Development with Flask – Real Python Guide.
28. White Paper: Fog Computing and its Role in the Internet of Things. – OpenFog Consortium.
29. Zeng, D., Guo, S., & Cheng, Z. The Web of Things: A Survey // Journal of Communications. – 2011. – №6(6).
- Zhang, Y., & Ansari, N. On Harnessing the Power of Optical Technologies in Green and Smart Cloud Computing // IEEE Communications Surveys & Tutorials. – 2013.