

DOI: <https://doi.org/10.36910/6775-2524-0560-2025-59-31>

УДК: 004.75

Міскевич Оксана Іванівна, ст. викладач

<https://orcid.org/0000-0002-5009-2391>

Шульгач Іван Сергійович, студент

Луцький національний технічний університет, м. Луцьк, Україна

МЕТОДИ ТА ЗАСОБИ РОЗРОБКИ ВЕБ-ІНТЕРФЕЙСУ КЕРУВАННЯ АПАРАТНИМИ МОДУЛЯМИ ARDUINO В СКЛАДІ СИСТЕМИ ІНТЕРНЕТУ РЕЧЕЙ

Міскевич О. І., Шульгач І. С. **Методи та засоби розробки веб-інтерфейсу керування апаратними модулями Arduino в складі системи інтернету речей.** Активне поширення технологій Інтернету речей і широке застосування у численних галузях, від домашньої автоматизації до промислових рішень, створюють потребу у розробці зрозумілих та функціональних засобів взаємодії з такими системами. Ключовим елементом, що забезпечує зручний моніторинг, збір даних та дистанційне керування пристроями, є веб-інтерфейс. У даній статті здійснено аналіз сучасних методів та інструментальних засобів, призначених для розробки веб-інтерфейсів, орієнтованих на керування апаратними модулями Arduino у рамках систем Інтернету речей, детально розглянуто етапи проектування, включаючи вибір оптимальних ефективних протоколів зв'язку та актуальних стеків технологій для фронтенд і бекенд розробки. Особливу увагу приділено специфіці інтеграції веб-інтерфейсів із мікроконтролерами Arduino, аналізу популярних бібліотек, програмних платформ та підходів до забезпечення безпеки передачі даних. У статті також висвітлено основні труднощі та запропоновано рекомендації щодо вибору інструментарію для створення масштабованих веб-рішень для проектів на базі Arduino.

Ключові слова: веб-інтерфейс, Інтернет речей, Arduino, IoT, керування апаратними модулями, методи розробки, засоби розробки, протоколи зв'язку, архітектура програмного забезпечення, мікроконтролери.

Miskevych O., Shulhach I., **Methods and tools for developing a web interface to control hardware modules of Arduino within an internet of things system.** The active proliferation of Internet of Things (IoT) technologies and the widespread use of Arduino based hardware modules across numerous domains—from home automation to industrial solutions—have created an urgent need for developing intuitive and functional means of interacting with such systems. A web interface is the key component that provides convenient monitoring, data collection, and remote control of IoT devices. In this article, we present a comprehensive analysis of modern methods and toolsets designed for developing web interfaces aimed at controlling Arduino hardware modules within IoT systems. We examine in detail the design stages, including the selection of optimal architectural patterns, efficient communication protocols (such as HTTP, MQTT, and WebSockets), and up-to-date technology stacks for frontend development (e.g., React, Vue, Angular) and backend development (Node.js, Python/Flask/Django). Special attention is paid to the specifics of integrating web interfaces with Arduino microcontrollers, the analysis of popular libraries and software platforms (for example, Arduino IoT Cloud and Blynk), and approaches to ensuring secure data transmission. The article also highlights the main challenges and offers recommendations for choosing tools to create scalable and reliable web solutions for Arduino-compatible IoT projects.

Keywords: web interface, Internet of Things, Arduino, IoT, hardware module control, development methods, development tools, communication protocols, software architecture, microcontrollers.

Постановка проблеми.

Сучасний розвиток Інтернету речей сприяє масовому використанню апаратних модулів на базі Arduino в різних сферах, від домашньої автоматизації до промислового моніторингу. Водночас існуючі веб-інтерфейси, для керування такими системами, зазвичай, обмежуються відображенням базових показників або простим дистанційним увімкненням та вимкненням пристроїв, ігноруючи потреби масштабованості, адаптивності та безпеки. Через обмежені ресурси мікроконтролерів Arduino та різноманіття апаратних модулів виникає необхідність у виборі оптимальних архітектурних рішень і комунікаційних протоколів, що забезпечують мінімальні затримки, надійність передачі даних та можливість легкої інтеграції нових пристроїв. Окремою проблемою є забезпечення зручного користувацького досвіду на фронтенді, сучасні фреймворки потребують адаптивних рішень для відображення даних та керування модулями, а бекенд-інструменти мають поєднувати гнучкість із високим рівнем безпеки. Таким чином, наукова проблема полягає у розробці методів і засобів, що дозволяють створювати масштабовані, безпечні та інтуїтивно зрозумілі веб-інтерфейси для ефективного керування апаратними модулями Arduino в складі IoT систем. Це рішення відкриває можливості для універсальних платформ керування в різноманітних прикладних галузях.

Аналіз останніх досліджень та публікацій. Активне поширення технологій Інтернету речей (IoT) та широке застосування апаратних модулів на платформі Arduino зумовлюють зростаючий інтерес до розробки зручних та функціональних веб-інтерфейсів керування такими

системами. Останні дослідження в цій галузі спрямовані на вирішення проблем масштабованості, безпеки та інтуїтивності взаємодії з IoT-пристроями на базі Arduino.

Так, у роботі [1] Соколової Н.О. та Белова А.С. розглядається розробка програмного забезпечення IoT-системи на апаратній платформі Arduino для моніторингу параметрів мікроклімату, таких як температура, вологість, рівень CO₂ та освітленість. Дослідження демонструє практичне застосування платформи Arduino, сервісу ThingsSpeak для агрегації, візуалізації та аналізу даних у хмарі, а також використання Telegram-бота як інтерфейсної частини для взаємодії з користувачем. Це підкреслює важливість мережевої взаємодії та доступних інтерфейсів для ефективного збору даних та сповіщення в IoT-системах.

У свою чергу, магістерська дисертація [2] Півня Н.О. присвячена створенню системи охоронної сигналізації офісного приміщення на базі Arduino UNO. Система використовує різноманітні датчики та забезпечує моніторинг і захист у режимі реального часу з можливістю відстеження інформації через веб-інтерфейс. Важливим аспектом є передача даних на сервер для зберігання та аналізу, а також інформування власника про порушення через мобільний додаток або повідомлення, зокрема з використанням GSM-модуля. Це дослідження акцентує увагу на питаннях безпеки, інтеграції датчиків та важливості веб-інтерфейсу для контролю та управління IoT-системами.

Дослідження [3] Цирульника С.М. та Моторної Л.В. фокусується на використанні модуля ESP8266 та хмарного сервісу IFTTT для проектного навчання студентів створенню пристроїв Інтернету речей, на прикладі системи "розумний будинок". Робота розкриває особливості підключення та взаємодії модуля ESP8266, використання хмарного IoT-сервісу aRest для доступу до модуля та передачі даних, а також інтеграцію з такими сервісами, як Weather Underground, через IFTTT для автоматизованого керування. Це підкреслює можливості застосування універсальних інструментальних засобів та хмарних платформ для створення гнучких та функціональних веб-орієнтованих систем керування IoT-пристроями, а також важливість формування відповідних компетенцій у майбутніх фахівців, що включають знання методів передачі даних та програмування мережевих додатків.

Проведений аналіз наведених публікацій свідчить про значну активність наукових досліджень у сфері розробки веб-інтерфейсів та програмних рішень, призначених для керування Arduino-сумісними системами Інтернету речей. Однак, багато існуючих робіт часто фокусуються на вирішенні окремих аспектів розробки, таких як моніторинг конкретних параметрів або реалізація простого дистанційного керування через специфічні платформи чи месенджери. Залишаються недостатньо дослідженими комплексні методи та універсальні інструментальні засоби, що дозволяли б створювати масштабовані, захищені та інтуїтивно зрозумілі веб-інтерфейси для ефективного керування широким спектром апаратних модулів Arduino в складі складних IoT-систем.

Мета дослідження. Метою даного дослідження є систематизація та аналіз сучасних методів та засобів розробки веб-інтерфейсів для керування апаратними модулями Arduino в системах Інтернету речей, а також розробка рекомендацій щодо вибору оптимальних технологічних стеків та архітектурних рішень. Для досягнення поставленої мети необхідно проаналізувати існуючі підходи до проектування та реалізації веб-інтерфейсів для IoT-систем на базі Arduino, дослідити особливості застосування різних комунікаційних протоколів для взаємодії веб-інтерфейсу з мікроконтролерами, оцінити переваги та недоліки популярних фронтенд та бекенд-технологій у контексті розробки для Arduino IoT, визначити ключові аспекти забезпечення безпеки та масштабованості веб-інтерфейсів.

Виклад основного матеріалу дослідження. У рамках даного дослідження було проведено аналіз і систематизацію методів і засобів, спрямованих на розробку ефективного веб-інтерфейсу для керування апаратними модулями Arduino в системах Інтернету речей. Робота зосереджена на знаходженні оптимальних архітектурних рішень, протоколів зв'язку та технологічних стеків, що враховують як специфіку сучасних веб-технологій, так і обмеження платформи Arduino.

Вибір архітектури впливає на її гнучкість, масштабованість, продуктивність. Для систем на базі Arduino, в ході дослідження було проаналізовано три фундаментальні архітектурні моделі, їх узагальнену взаємодію компонентів, які представлено на рисунку 1.

Найпростіший варіант, де веб-сервер реалізується на мікроконтролері з підтримкою мережевих функцій називається архітектура прямої взаємодії. Такий підхід є доцільним для невеликих локальних мереж та завдань з мінімальними вимогами до обчислювальних ресурсів. Перевагами є низька початкова вартість та простота розгортання. Однак, обмежені ресурси

мікроконтролера та підвищені ризики безпеки стають суттєвими недоліками при масштабуванні системи або підключенні до глобальної мережі.

Архітектура взаємодії через хмарну платформу враховує використання проміжних хмарних сервісів, таких як Arduino IoT Cloud, Blynk, AWS IoT, Google Cloud IoT Core. Пристрій на базі arduino надсилає дані на хмарну платформу, а веб-інтерфейс користувача взаємодіє з цією платформою через наданий API. Такий підхід дозволяє значно розвантажити мікроконтролер, забезпечити надійне зберігання даних, реалізувати складну бізнес-логіку на стороні хмари та спростити доступ до пристроїв. Це гарантує високу масштабованість та надійність такого рішення. Разом з тим є потенційний недолік у вигляді залежності від стороннього сервісу та можливих пов'язаних витрат.

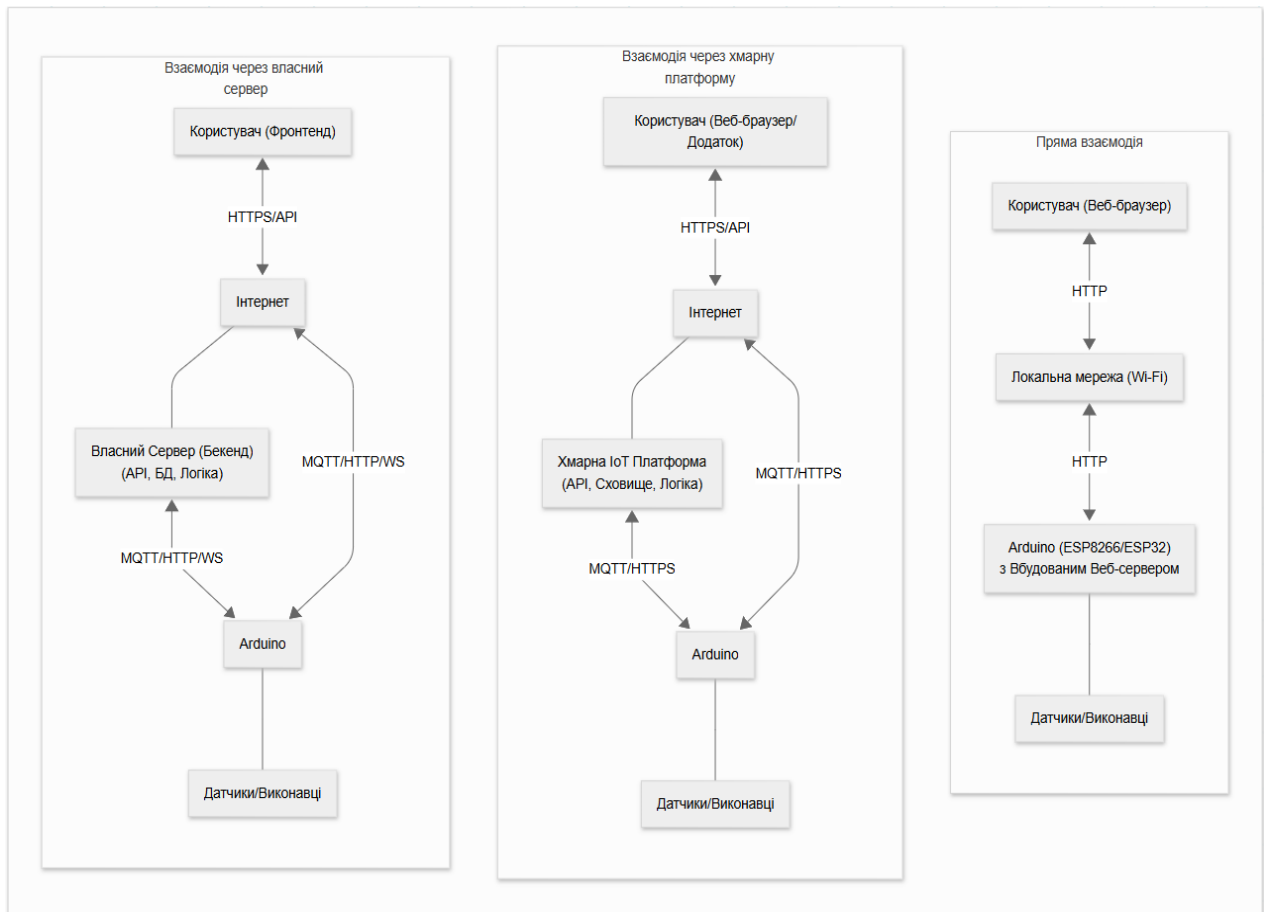


Рис. 1– Архітектурні моделі

Архітектура взаємодії через власний сервер. Цей підхід, що передбачає розробку власного серверного додатку, який виступає посередником між Arduino та клієнтським веб-інтерфейсом. Дослідження показало, що дана архітектура забезпечує максимальну гнучкість у виборі технологій, кастомізації функціоналу та контролі над даними і безпекою. Однак, це вимагає значно більших зусиль на розробку, розгортання та подальшу підтримку серверної інфраструктури.

На основі аналізу цих архітектур було зроблено висновок, що вибір конкретної моделі має ґрунтуватися на специфічних вимогах проєкту, що видно на таблиці 1.

Важливим є ефективність обміну даними між модулями Arduino та веб-інтерфейсом. Основні протоколи зв'язку, що застосовуються в IoT-системах HTTP, як традиційний протокол для веб-комунікацій, був розглянутий для сценаріїв, де Arduino може виступати як HTTP-клієнт, для відправки даних на сервер, або як HTTP-сервер для прямої взаємодії. Використання HTTP визнано обов'язковим для забезпечення захищеної передачі даних. Однак, для IoT-застосунків виявлено недолік у вигляді відносно великого розміру заголовків, що може бути критичним для пристроїв з обмеженою пропускну здатністю.

Таблиця 1– Аналіз взаємодій

Критерій	Пряма взаємодія	Взаємодія через хмарну платформу	Взаємодія через власний сервер
Основний принцип реалізації	Веб-сервер на мікроконтролері	Використання проміжних хмарних сервісів	Розробка власного серверного додатку.
Переваги	Низька вартість, простота розгортання.	Дозволяє розвантажити мікроконтролер, забезпечити зберігання даних, реалізувати складну логіку; висока масштабованість та надійність.	Забезпечує максимальну гнучкість.
Недоліки	Обмежені ресурси мікроконтролера та питання безпеки.	Може бути залежність від стороннього сервісу.	Вимагає більших зусиль на розробку та підтримку.
Доцільність застосування	Для локальних мереж та простих завдань	Для систем, що потребують розвантаження мікроконтролера, зберігання даних, реалізації складної логіки, високої масштабованості та надійності.	Для складніших систем, де потрібна максимальна гнучкість

MQTT, легкий протокол обміну повідомленнями за моделлю «видавець-підписник» Встановлено, що MQTT є високоефективним в умовах нестабільного зв'язку та для пристроїв з обмеженими ресурсами, завдяки малим заголовкам та низькому енергоспоживанню. Роль MQTT-брокера як посередника забезпечує гнучкість у маршрутизації повідомлень.

Протокол WebSockets є оптимальним для завдань, що вимагають повнодуплексного зв'язку та оновлення даних на веб-інтерфейсі в реальному часі без необхідності постійних HTTP-запитів. Це забезпечує низьку затримку та ефективне використання ресурсів.

CoAP, спеціально розроблений для пристроїв з обмеженими ресурсами та мереж з низькою пропускну здатністю. Його схожість з HTTP та оптимізація для низького енергоспоживання роблять його перспективним для певних IoT-застосунків.

Обґрунтування вибору конкретного протоколу залежить від специфічних вимог проєкту до швидкості передачі даних, надійності, енергоефективності, складності реалізації та існуючої інфраструктури. Схему, що ілюструє типові потоки даних при використанні різних протоколів, наведено на рисунку 2.

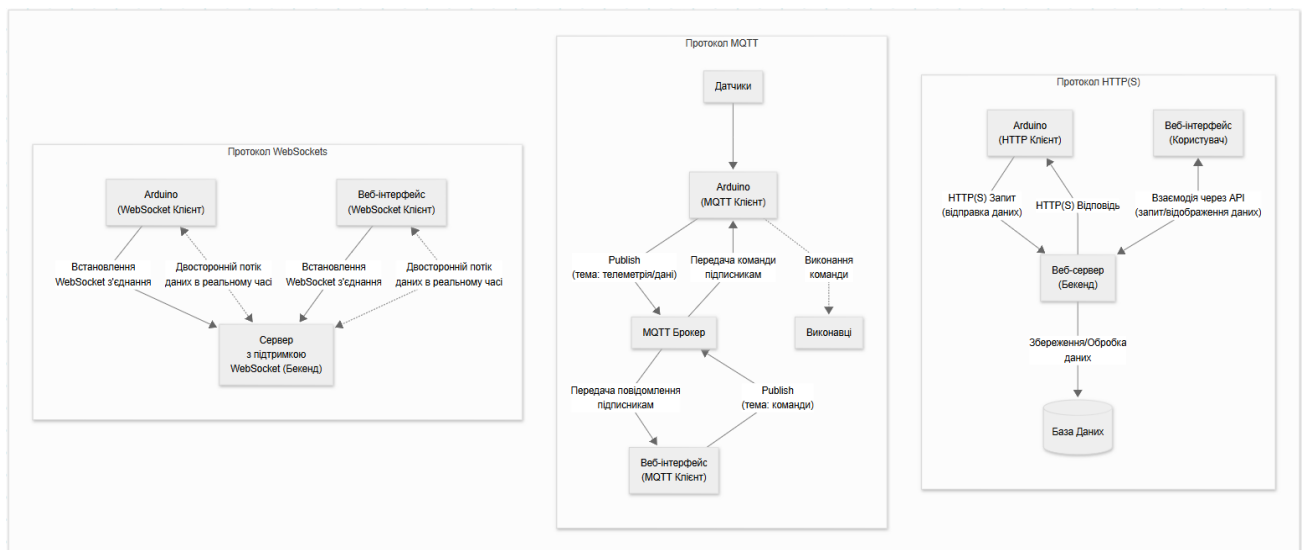


Рис. 2 – Потоки даних при використанні різних протоколів

Фронтенд частина веб-інтерфейсу відповідає за візуальне представлення даних, що надходять від модулів Arduino, та забезпечення інтерактивної взаємодії з користувачем. Базові веб-технології а саме HTML, CSS, JavaScript є основою будь-якого веб-інтерфейсу і можуть бути достатніми для простих завдань візуалізації та управління.

Сучасні JavaScript-фреймворки React, Angular, Vue.js чудово підходять для створення складних, динамічних та інтерактивних односторінкових додатків. Вони надають структурований підхід до розробки, компоненту модель, управління станом та інші інструменти, що прискорюють розробку та покращують якість коду.

Використання готових UI-компонентів таких, як Bootstrap, Materialize, Tailwind CSS та значно прискорює процес розробки візуальної частини інтерфейсу та забезпечує його адаптивність для різних пристроїв.

У випадках, коли обрано архітектуру з власним сервером, бекенд виконує ключові функції обробку запитів від фронтенду, взаємодію з базою даних для зберігання та отримання даних телеметрії, а також комунікацію з апаратними модулями Arduino. Node.js має високу ефективність для IoT-застосунків завдяки його асинхронній неблокуючій моделі вводу-виводу, що добре підходить для обробки великої кількості одночасних підключень від пристроїв та користувачів.

Python є популярним вибором завдяки своїй простоті, швидкості розробки та великій кількості бібліотек, зокрема для аналізу даних та машинного навчання, що може бути корисним для обробки даних з IoT-пристроїв.

Java, потужна та надійна платформа, що підходить для розробки великих, високопродуктивних та масштабованих корпоративних систем, в тому числі й IoT-платформ.

RНР популярна технологія для веб-розробки, що також може бути використана для створення бекенду IoT-систем.

Вибір конкретної бекенд-технології, як показало дослідження, залежить від таких факторів, як складність проекту, вимоги до продуктивності, навантаження, досвід команди розробників та наявність необхідних бібліотек.

Безпека та масштабованість є критично важливими аспектами при розробці веб-інтерфейсів для IoT-систем. Шифрування каналів зв'язку є необхідністю використання HTTPS, TLS, SSL для захисту даних, що передаються між веб-інтерфейсом, сервером та пристроями. Для автентифікації та авторизації є різні механізми, наприклад, токени JWT, OAuth 2.0, для надійної перевірки ідентичності користувачів та пристроїв, а також для розмежування прав доступу до функцій системи та даних.

Проведене дослідження методів та засобів розробки веб-інтерфейсів для керування апаратними модулями Arduino дозволило систематизувати ключові підходи та технології, що застосовуються на різних етапах життєвого циклу таких систем. На основі аналізу було виявлено, що вибір конкретних технологій, архітектурних рішень та протоколів зв'язку не є універсальним і значною мірою детермінується специфікою конкретного IoT-проекту. До таких визначальних факторів належать кількість пристроїв та користувачів, вимоги до рівня безпеки та конфіденційності даних, доступний бюджет, необхідна функціональність веб-інтерфейсу, а також наявний досвід та компетенції команди розробників.

Для реалізації простих локальних завдань, таких як керування кількома датчиками або виконавчими пристроями в межах домашньої мережі, цілком достатнім може бути застосування архітектури прямої взаємодії з реалізацією веб-сервера безпосередньо на мікроконтролерах типу ESP8266 або ESP32. Це мінімізує витрати та складність розгортання.

Для більш складних, розподілених або комерційних систем, де важливі надійність, масштабованість та розширений функціонал, дослідження обґрунтовує доцільність використання архітектур з проміжним сервером. У таких випадках протоколи MQTT та WebSockets продемонстрували свою ефективність для забезпечення надійного та швидкого обміну даними в реальному часі, що є критичним для багатьох IoT-застосунків.

Застосування сучасних JavaScript-фреймворків для розробки фронтенду, згідно з результатами дозволяє створювати інтерактивні, адаптивні та функціонально насичені користувацькі інтерфейси. Вибір потужних бекенд-платформ забезпечує необхідну продуктивність та гнучкість серверної частини.

Висновки та перспективи подальшого дослідження. У ході дослідження проведено аналіз сучасних методів та інструментів для розробки веб-інтерфейсів керування модулями Arduino в IoT-системах. Розглянуто архітектурні підходи, протоколи зв'язку, технології розробки та аспекти

безпеки. Вибір стеку технологій залежить від складності завдання, кількості пристроїв, вимог до продуктивності, безпеки та бюджету. Для простих систем доцільні вбудовані веб-сервери; для складних – архітектури з проміжним сервером. Протоколи MQTT та WebSockets ефективні для комунікації в реальному часі. Сучасні JavaScript-фреймворки та серверні платформи надають потужні інструменти розробки.

Список бібліографічного опису

1. Соколова Н.О., Белов А.С Розробка програмного забезпечення IoT-системи на апаратній платформі Arduino URL : <https://surl.li/edglnc> (дата звернення: 13.05.2025).
2. Півень Н.О. Система охоронної сигналізації офісного приміщення. URL : <https://surl.li/rhxyeg> (дата звернення: 13.05.2025).
3. Цирульник С. М. Вплив особистісних та професійних рис на ефективність діяльності менеджера. URL: <https://surl.li/wffpye> (дата звернення: 13.05.2025).

References

1. Sokolova N.O., Bielov A.S. Rozrobka prohramnoho zabezpechennia IoT-systemy na aparatnii platformi arduino [Development of IoT system software on the Arduino hardware platform]. URL: <https://surl.li/edglnc> (accessed: May 13, 2025).
2. Piven N.O. Systema okhoronnoi syhnalizatsii ofisnoho prymishchennia [Office security alarm system]. URL: <https://surl.li/rhxyeg> (accessed: May 13, 2025).
3. Tsyurulnyk S. M. Vplyv osobystisnykh ta profesiinykh rys na efektyvnist diialnosti menedzhera [The influence of personal and professional traits on the effectiveness of a manager's activity]. URL: <https://surl.li/wffpye> (accessed: May 13, 2025).