МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: АВТОМАТИЗОВАНИЙ АНАЛІЗ ТА SEO ОПТИМІЗАЦІЯ ВЕБ САЙТІВ

Здобувача вищої освіти другого (магістерського) рівня,
ІІ курсу, групи КМІ-24-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології в економіці»

Кушнір Андрій Олексійович
(підпис) (прізвище, ім'я, по батькові)

Науковий керівник
Горбовий Артур Юліанович, д-р техн. наук, професор
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-професійної програми
Омельчук Антон Анатолійович, канд. техн. наук
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри протокол від «02» січня 2026 № 10

Завідувач кафедри
Філоненко Михайло Миколайович, канд. фіз.-мат. наук, доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2026

Реєстрація № 10 19.01.2026 _____
(дата) (підпис відповідальної особи)

Результати перевірки до захисту 02.01.2026 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 4 (добре) 09.02.2026 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ О.С. Дюбанов
(підпис)

Члени комісії:

_____ А.Ю. Горбовий
(підпис)

_____ М.М. Філоненко
(підпис)

_____ І.Д. Погорєловська
(підпис)

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст кваліфікаційної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	25		09.02.2026 Дюбанов О.С. Горбовий А.Ю. Філоненко М.М. Погореловська І.Д. Суховецька Л.П.
2. Дотримання вимог щодо оформлення	від 0 до 10	9		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		
4. Рецензент	від 0 до 5	4		
РАЗОМ БАЛІВ	від 0 до 50	42		

Оцінювання кваліфікаційної роботи науковим керівником - 41 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
83	B	4	09.02.2026	Дюбанов О.С. Горбовий А.Ю. Філоненко М.М. Погореловська І.Д. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань 12 «Інформаційні технології»
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Інформаційні управляючі системи і технології в економіці»

Затверджую
Завідувач кафедри _____ Філоненко М.М.
«11» грудня 2024

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Кушніру Андрію Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Автоматизований аналіз та SEO оптимізація веб сайтів.
Затверджена наказом ректора від «10» грудня 2024 № 1852
2. Строк здачі закінченої роботи «02» січня 2026
3. Мета роботи Спроєктувати та реалізувати вебсервіс з використанням сучасних технологій веб-розробки та мікросервісної архітектури для автоматизованого аналізу та оптимізації seo вебсайтів з метою покращення їх видимості у пошукових системах.
Об'єкт дослідження Процеси автоматизації моніторингу веб-сайтів та управління інформаційними системами.
Предмет дослідження Методи та засоби розробки веб-додатків з використанням Django, Celery, Redis та мікросервісної архітектури для реалізації систем автоматизації.
Орієнтовна база дослідження Відкриті веб-ресурси мережі Інтернет, документація пошукових систем (Google Search Central), технічна документація фреймворків Django та Celery.
4. Вихідні дані до роботи Технічне завдання на розробку сервісу; вимоги до SEO-аналізу (мета-теги, швидкість завантаження, структура заголовків); набір інструментів для парсингу (BeautifulSoup/Selenium); API пошукових систем.
5. Перелік графічного матеріалу Схема архітектури системи, інтерфейси користувач, графіки результатів тестування продуктивності, алгоритм роботи вебсервісу.
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом) ВСТУП, РОЗДІЛ 1: Історія та сучасний стан автоматизації управління веб-проектами, Методологічні засади та інструментарій дослідження систем моніторингу, РОЗДІЛ 2: Аналіз вимог та технічне обгрунтування розробки, Проєктування та обгрунтування вибору інструментарію розробки, Розробка алгоритму роботи веб-додатку, Верифікація ефективності розробленого вебсервісу та аналіз результатів оптимізації, РОЗДІЛ 3: План впровадження системи автоматизації в діяльність підприємства, Прогноз результатів та покращення показників видимості сайту, Обгрунтування економічного ефекту та практична значиміст, Напрями подальшого вдосконалення та масштабування системи,висновки,список використаних джерел,додатки, ВИСНОВКИ, СПИСОК.

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 15.12.2024	До 15.12.2024
2	**І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	28.05.2025	28.05.2025
3	**ІІ розділ** (аналіз експериментальної частини про методики досліджень)	08.09.2025	08.09.2025
4	**ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	17.11.2025	17.11.2025
5	**Висновки**	08.12.2025	08.12.2025
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	22.12.2025	22.12.2025
7	Отримання відгуку керівника	20.01.2026	20.01.2026
8	Отримання зовнішньої рецензії	23.01.2026	23.01.2026

8. Дата видачі завдання «11» грудня 2024

9. Керівник кваліфікаційної роботи _____ Горбовий А.Ю.
(підпис) (прізвище, ініціали)

10. Керівник освітньо-професійної програми _____ Омельчук А.А.
(підпис) (прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти _____ Кушнір А.О.
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Автоматизований аналіз та SEO оптимізація веб сайтів.

(назва кваліфікаційної роботи)

Мета магістерської роботи полягає у розробці комплексної веб-системи автоматизації DoDoBot для моніторингу веб-ресурсів, аналізу SEO-показників та оптимізації видимості веб-сайтів у пошукових системах. Завданнями роботи є: аналіз існуючих систем автоматизації моніторингу та управління; дослідження сучасних підходів до побудови веб-сервісів та мікросервісних архітектур; обґрунтування вибору технологічного стеку; проєктування архітектури системи; розробка моделей даних користувачів і проєктів; реалізація механізмів асинхронної обробки завдань; інтеграція із зовнішніми сервісами сповіщень та API; створення адаптивного веб-інтерфейсу; тестування та оцінка ефективності розробленого рішення.

Об’єктом дослідження є процеси автоматизації моніторингу веб-сайтів та управління інформаційними системами.

Предметом дослідження є архітектурні рішення та програмні засоби розробки веб-систем автоматизації з використанням Django, Celery, Redis та мікросервісного підходу.

Базою дослідження є веб-система автоматизації DoDoBot, що реалізується як модульний веб-сервіс із підтримкою асинхронного виконання завдань та інтеграції із зовнішніми платформами.

За результатами дослідження спроєктовано та реалізовано веб-систему, що забезпечує автоматизований моніторинг доступності веб-ресурсів, аналіз SEO-параметрів, генерацію контенту з використанням GPT-моделей та надсилання сповіщень через різні канали зв’язку. Реалізовано мікросервісну архітектуру з асинхронною обробкою задач і адаптивний інтерфейс для взаємодії з користувачами.

Одержані результати можуть бути використані для автоматизації процесів у сфері моніторингу веб-сайтів, SEO-аналізу, управління контентом та комунікації з клієнтами в малих і середніх підприємствах.

Кваліфікаційна робота містить _54_ сторінок, _3_ таблиць, _28_ рисунків, список використаних джерел з _41_ найменувань, _5_ додатків.

Ключові слова: автоматизація, веб-моніторинг, SEO-аналіз, веб-сервіс, мікросервісна архітектура, Django, Celery, Redis, GPT.

Рік виконання магістерської роботи 2025–2026.

Рік захисту роботи 2026.

ABSTRACT

Automated analysis and SEO optimization of websites.

(title of the qualification work)

The purpose of the master's thesis is to develop a comprehensive web-based automation system, DoDoBot, for monitoring web resources, analyzing SEO metrics, and optimizing website visibility in search engines. The objectives of the work include: analysis of existing automation systems for monitoring and management; study of modern approaches to building web services and microservice architectures; justification of the selected technology stack; design of the system architecture; development of user and project data models; implementation of asynchronous task processing mechanisms; integration with external notification services and APIs; creation of an adaptive web interface; testing and evaluation of the effectiveness of the developed solution.

The object of the research is the processes of automating website monitoring and managing information systems.

The subject of the research is architectural solutions and software tools for developing web-based automation systems using Django, Celery, Redis, and a microservice approach.

The research base is the DoDoBot web automation system, implemented as a modular web service with support for asynchronous task execution and integration with external platforms.

As a result of the research, a web system has been designed and implemented that provides automated monitoring of web resource availability, analysis of SEO parameters, content generation using GPT models, and delivery of notifications through various communication channels. A microservice architecture with asynchronous task processing and an adaptive user interface for interaction with users has been implemented.

The obtained results can be used to automate processes in the fields of website monitoring, SEO analysis, content management, and client communication in small and medium-sized enterprises.

The qualification work contains _54_ pages, _3_ tables, _28_ figures, a list of references with _41_ sources, and _5_ appendices.

Keywords: automation, web monitoring, SEO analysis, web service, microservice architecture, Django, Celery, Redis, GPT.

Year of completion of the master’s thesis: 2025–2026.

Year of thesis defense: 2026.

ЗМІСТ

ВСТУП **3**
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ **5**
1.1. Історія та сучасний стан автоматизації управління веб-проектами 5
1.2. Методологічні засади та інструментарій дослідження систем моніторингу 11
РОЗДІЛ 2. ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ВЕБ-ДОДАТКУ DODoBot . **16**
2.1. Аналіз вимог та технічне обґрунтування розробки 16
2.2. Проєктування та обґрунтування вибору інструментарію розробки 16
2.3. Розробка алгоритму роботи веб-додатку 20
2.4. Верифікація ефективності розробленого вебсервісу та аналіз результатів оптимізації 42
РОЗДІЛ 3. НАПРЯМИ ВПРОВАДЖЕННЯ ТА ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ СИСТЕМИ DODOBOT **43**
3.1. План впровадження системи автоматизації в діяльність підприємства 43
3.2. Прогноз результатів та покращення показників видимості сайту 44
3.3. Обґрунтування економічного ефекту та практична значиміст 45
3.4. Напрями подальшого вдосконалення та масштабування системи 45
ВИСНОВКИ **48**
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **50**
ДОДАТКИ **54**

ВСТУП

Сучасний розвиток інформаційних технологій зумовлює зростаючу потребу в автоматизації бізнес-процесів, зокрема у сфері моніторингу веб-ресурсів, управління контентом та аналітики. Більшість компаній стикаються з проблемами ефективного контролю за працездатністю веб-сайтів, своєчасного аналізу SEO-параметрів та оперативного інформування відповідальних осіб про критичні зміни. Вирішення цих завдань вручну є затратним, вимагає значних часових ресурсів та підвищує ризик людських помилок.

Актуальність даної магістерської роботи полягає у розробці комплексної веб-системи автоматизації DoDoBot, що дозволяє бізнес-користувачам створювати власні «роботи» для виконання рутинних завдань: моніторингу доступності веб-сайтів, аналізу SEO-показників, генерації контенту за допомогою GPT-моделей, перевірки SEO-тегів та відправки сповіщень через різні канали зв’язку. Подібні рішення сприяють підвищенню продуктивності бізнесу, зниженню операційних витрат та забезпечують своєчасність прийняття управлінських рішень.

Мета магістерської роботи: спроєктувати та реалізувати вебсервіс з використанням сучасних технологій веб-розробки та мікросервісної архітектури для автоматизованого аналізу та оптимізації seo вебсайтів з метою покращення їх видимості у пошукових системах.

Об’єкт дослідження: процеси автоматизації моніторингу веб-сайтів та управління інформаційними системами.

Предмет дослідження: методи та засоби розробки веб-додатків з використанням Django, Celery, Redis та мікросервісної архітектури для реалізації систем автоматизації.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналітичний огляд існуючих систем автоматизації моніторингу та управління;
- визначити архітектуру системи та методи реалізації її складових;

- розробити моделі даних для користувачів, проектів, блоків та повідомлень;
- реалізувати інтеграцію з зовнішніми сервісами (Telegram, Gmail, SMS, WayForPay, OpenAI API);
- впровадити механізм асинхронного виконання завдань за допомогою Celery та Redis;
- забезпечити багаторівневу систему аутентифікації та захисту даних;
- створити адаптивний інтерфейс для взаємодії з користувачами;
- провести тестування та аналіз продуктивності розробленої системи.

Гіпотеза дослідження полягає у припущенні, що використання модульної архітектури на базі Django та мікросервісів дозволить створити масштабовану систему автоматизації, здатну ефективно виконувати задачі моніторингу та управління проектами з мінімальними витратами ресурсів.

Наукова новизна роботи полягає у комплексній реалізації системи, яка поєднує модульність Django-додатків із незалежними Flask-мікросервісами для відправки повідомлень та підтримує динамічне керування налаштуваннями без перезапуску сервера. Такий підхід забезпечує гнучкість, масштабованість та зручність впровадження системи у реальних бізнес-процесах.

Практичне значення отриманих результатів полягає у створенні програмного продукту, який може бути використаний малими та середніми підприємствами для автоматизації рутинних процесів у сфері моніторингу веб-сайтів, управління контентом та комунікації з клієнтами.

Впровадження такого рішення сприятиме підвищенню ефективності роботи компаній та зниженню витрат.

Методи дослідження: при виконанні роботи використано методи аналізу та синтезу, порівняння існуючих рішень, проєктування програмних систем, моделювання архітектури, а також експериментальні методи тестування та верифікації роботи веб-додатка.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Історія та сучасний стан автоматизації управління веб-проектами

На початкових етапах розвитку веб-технологій управління веб-проектами, підтримка сайтів та контроль їх стану виконувалися переважно вручну.

Адміністратори та розробники самостійно перевіряли доступність ресурсів, аналізували журнали серверів, відстежували помилки в роботі сайтів та зміни в контенті. SEO-оптимізація зводилася до періодичного аналізу позицій у пошукових системах та ручного перегляду статистики відвідувань. Такий підхід був ефективним лише для невеликих сайтів і не масштабувався при зростанні кількості сторінок, доменів або клієнтських проектів.

Зі збільшенням складності веб-проектів з’явилася низка проблем: неможливість оперативно реагувати на падіння сайтів, затримки у виявленні SEO-помилок (биті посилання, помилки індексації, дубльований контент), високі витрати часу на рутинні перевірки та людський фактор.

Особливо критично це стало для агентств та фріланс-команд, які одночасно обслуговують десятки або сотні сайтів.

Розвиток мов програмування високого рівня, зокрема Python, а також поява сучасних веб-фреймворків Django, Flask та FastAPI суттєво змінили підхід до автоматизації. Вони дозволили створювати масштабовані системи управління веб-проектами, які поєднують моніторинг технічного стану сайтів, аналіз SEO-показників, автоматичний парсинг даних і централізоване сповіщення користувачів. Завдяки цьому більшість рутинних операцій була переведена в автоматичний режим. Важливим напрямком стала автоматизація SEO-процесів.

SEO-оптимізація в сучасних умовах базується на комплексі технічних, контентних і поведінкових факторів, які використовуються алгоритмами Google для оцінки якості веб-сторінок і формування пошукової видачі.

Сучасні алгоритми ранжування орієнтовані не лише на наявність ключових слів, а й на зручність ресурсу для користувача, швидкість його роботи, структурованість контенту та рівень довіри до сайту.

До ключових технічних SEO-критеріїв належать коректна структура URL-адрес, використання протоколу HTTPS, швидкість завантаження сторінок, коректні HTTP-статуси відповідей, наявність файлів sitemap.xml і robots.txt, а також правильне використання canonical-посилань.

Наприклад, URL виду /services/web-development є більш зрозумілим для пошукових роботів і користувачів, ніж параметричні адреси типу /?id=15, які ускладнюють індексацію та можуть призводити до дублювання контенту. Важливу роль відіграють також показники Core Web Vitals (LCP, CLS, INP), мобільна адаптивність і відсутність технічних помилок, таких як биті посилання або некоректні редиректи [40].

Автоматизовані SEO-системи дозволяють регулярно перевіряти ці параметри та своєчасно виявляти відхилення, ще до того, як вони негативно вплинуть на позиції сайту в пошуковій видачі.

Основні SEO-критерії та приклади коректної й некоректної реалізації (табл. 1.1):

Таблиця 1.1.

Основні SEO-критерії та приклади коректної й некоректної реалізації

Критерій	**Добре (оптимізовано)**	**Погано (помилка)**
1	**2**	**3**
URL-адреси	/blog/seo-audit	/?id=123&cat=5
HTTP-статус	200 OK, 301 Moved Permanently	404 Not Found, 500 Error
Редиректи	301 при зміні адреси	Ланцюги 302
HTTPS	HTTPS без mixed content	HTTP або помилки SSL

Продовження табл. 1.1.

1	2	3
Mobile-friendly	Адаптивна верстка	Десктоп-версія на мобільних
Швидкість завантаження	LCP < 2,5 с	LCP > 4 с
Title	Унікальний, 50–60 символів	Однаковий на всіх сторінках
Description	Описує зміст сторінки	Відсутній
H1	Один логічний H1	Кілька H1 або відсутній
Контент	Унікальний, корисний	Дубльований
Canonical	Вказаний коректно	Відсутній
Sitemap.xml	Актуальний, без помилок	Містить 404 сторінки
Robots.txt	Дозволяє індексацію	Блокує важливі URL
Внутрішні посилання	Логічна перелінковка	Ізольовані сторінки
Schema.org	Розмітка Article, Product	Відсутня
Core Web Vitals	У нормі	Порушені показники

Джерело: [40]

Сучасні системи здатні регулярно перевіряти індексацію сторінок, статуси HTTP-відповідей, швидкість завантаження, коректність мета-тегів, наявність canonical-посилань та структуру sitemap. Автоматичний аналіз дозволяє виявляти проблеми ще до того, як вони суттєво вплинуть на позиції сайту в пошукових системах, що значно підвищує ефективність SEO-стратегії [35].

Окрему роль відіграють Telegram-боти та веб-додатки як інтерфейс взаємодії з системами автоматизації. Вони забезпечують швидкий доступ до актуальної інформації в режимі реального часу, дозволяють отримувати

критичні сповіщення миттєво та керувати проектами без необхідності входу до складних адміністративних панелей [29]. Це особливо зручно для розробників, SEO-фахівців і менеджерів, які працюють віддалено або з мобільних пристроїв.

Сучасні системи автоматизації управління веб-проектами характеризуються такими ключовими особливостями:

- Модульність — сучасні системи автоматизації будуються на основі модульної або мікросервісної архітектури, де кожен функціональний блок є логічно та технічно відокремленим компонентом. Це означає, що модулі моніторингу доступності (uptime), SEO-аналізу, парсингу контенту, фінансового контролю, аналітики трафіку або логування помилок можуть розроблятися, тестуватися та впроваджуватися незалежно один від одного. Такий підхід знижує ризик відмови всієї системи при помилці в одному з компонентів, спрощує масштабування та дозволяє адаптувати систему під різні бізнес-потреби без зміни ядра. Модульність також полегшує повторне використання компонентів у інших проєктах;
- Асинхронна обробка завдань — використання асинхронних механізмів виконання дозволяє ефективно працювати з великою кількістю однотипних або довготривалих операцій. Завдяки asyncio забезпечується неблокуюча обробка HTTP-запитів, а Celery у поєднанні з брокерами повідомлень (Redis, RabbitMQ) дозволяє виносити ресурсоємні задачі у фонові черги. Це особливо важливо для автоматизованого SEO-сканування, перевірки тисяч сторінок на помилки індексації, аналізу швидкості завантаження та збору статистики з різних джерел. Асинхронний підхід підвищує продуктивність системи, зменшує навантаження на сервер і забезпечує стабільну роботу навіть при пікових навантаженнях [1];
- Інтеграція з зовнішніми сервісами — ключовою характеристикою сучасних систем автоматизації є активне використання API сторонніх платформ. Підключення до Google Analytics 4 та Google Search Console дозволяє отримувати актуальні дані про трафік, джерела переходів, поведінку користувачів і проблеми індексації. Інтеграція з SEO-сервісами Ahrefs та

Semrush дає змогу автоматично аналізувати зворотні посилання, ключові слова та конкурентне середовище. Додатково, взаємодія з CDN та хмарними платформами дозволяє контролювати стабільність інфраструктури, SSL-сертифікати та час відповіді серверів, формуючи єдину централізовану систему моніторингу;

- Гнучкість і автоматичні сценарії реагування — важливою перевагою є можливість створення користувацьких правил та сценаріїв автоматизації. Система може самостійно аналізувати зміни показників і виконувати визначені дії без втручання людини. Наприклад, при різкому падінні органічного трафіку або позицій у пошуковій видачі автоматично формується звіт, надсилається термінове сповіщення та запускається діагностичний аналіз. Подібні механізми дозволяють скоротити час реагування з годин або днів до кількох хвилин, що безпосередньо впливає на збереження трафіку, конверсій і фінансової стабільності веб-проєкту.

Таким чином, сучасна автоматизація управління веб-проектами є критично важливим інструментом для ефективної підтримки сайтів, оптимізації SEO-процесів та забезпечення стабільної роботи цифрових продуктів в умовах зростаючої конкуренції.

Ринок програмних рішень пропонує велику кількість інструментів для автоматизації управління веб-проєктами, технічного моніторингу та SEO-контролю. Зі зростанням кількості веб-ресурсів і складності цифрової інфраструктури виникла потреба у системах, здатних забезпечувати постійний контроль стану сайтів, швидке реагування на збої та централізований аналіз даних. Умовно такі рішення можна поділити на дві основні категорії: комплексні веб-платформи (SaaS) та спеціалізовані Telegram-боти для оперативного моніторингу.

SaaS-платформи орієнтовані на комплексне управління веб-ресурсами та зазвичай надають розширений функціонал: глибокий технічний і SEO-аудит, візуалізацію даних у вигляді дашбордів, історію змін і збереження логів за

тривалий період. Такі системи часто використовуються великими компаніями, digital-агентствами та SEO-відділами. Зокрема, до таких сервісів належать:

- UptimeRobot є одним із найбільш поширених сервісів моніторингу доступності сайтів. Платформа підтримує перевірку HTTP(S)-запитів, контроль портів, Heartbeat-моніторинг та перевірку SSL-сертифікатів. Важливою функцією є створення публічних сторінок статусу сервісів, що підвищує довіру користувачів. Водночас обмеження інтервалу перевірок у безкоштовній версії (5 хвилин) знижує ефективність сервісу для критично важливих систем [26];
- SEMrush є багатофункціональною платформою, орієнтованою насамперед на SEO та цифровий маркетинг. Вона дозволяє автоматично аналізувати технічний стан сайту, показники Core Web Vitals, структуру внутрішніх і зовнішніх посилань, видимість у пошукових системах та конкурентне середовище. Перевагою є детальні аналітичні звіти, однак висока вартість підписки та складність інтерфейсу роблять платформу малодоступною для невеликих проєктів і індивідуальних розробників [25];
- Zapier та Make (Integromat) належать до класу low-code/no-code платформ автоматизації бізнес-процесів. Вони не виконують моніторинг безпосередньо, але дозволяють об’єднувати різні сервіси в єдині автоматизовані сценарії. Наприклад, при виявленні проблеми з доступністю сайту або зміні контенту може автоматично створюватися задача в системі управління проєктами або надсилатися сповіщення відповідальній особі. Основними недоліками таких рішень є залежність від сторонніх сервісів, затримки виконання сценаріїв та зростання вартості при масштабуванні [41].

Telegram-боти займають окрему нішу інструментів швидкого реагування. Їх головною перевагою є мінімальний поріг входу, миттєві push-сповіщення та зручність використання з мобільних пристроїв. Такі боти зазвичай виконують допоміжну роль і не замінюють повноцінні веб-платформи, але значно підвищують оперативність контролю. Серед них можна виділити:

- @WebMonitorBot використовується для відстеження доступності веб-ресурсів та змін у коді сторінок. Він є зручним інструментом для фрілансерів

і невеликих команд, оскільки дозволяє оперативно дізнаватися про збої без входу до окремих систем управління;

- @SEOBot орієнтований на швидке отримання базових SEO-показників, таких як авторитет домену, кількість зворотних посилань або індексація сторінок. Завдяки інтеграції з API аналітичних сервісів бот надає стислий огляд стану сайту, проте не забезпечує глибокого технічного аналізу;
- @ContentNotifierBot призначений для моніторингу змін контенту, наприклад оновлення цін, появи нових матеріалів або змін на сайтах конкурентів. Водночас використання універсальних алгоритмів парсингу обмежує точність роботи бота на динамічних або SPA-сайтах, що знижує його практичну цінність у складних проєктах.

1.2. Методологічні засади та інструментарій дослідження систем моніторингу

Методологічною основою дослідження є системний підхід, що дозволяє розглядати веб-проєкт як складну структуру взаємопов'язаних технічних та контентних елементів. У процесі роботи для досягнення поставленої мети використано такі наукові методи:

- метод порівняльного аналізу — для зіставлення функціональних можливостей існуючих платформ;
- метод системного аналізу — для визначення структури та архітектурних рівнів майбутньої системи автоматизації;
- статистичні та аналітичні методи — для оцінки технічних показників Core Web Vitals, швидкості завантаження та SEO-параметрів;

Інформаційною базою дослідження послужили дані технічної документації API сучасних сервісів, результати спостережень за роботою веб-ресурсів протягом 2023–2025 років, а також аналітичні звіти Google Search Console.

Дизайн сучасних систем автоматизації управління веб-проєктами відіграє ключову роль у забезпеченні зручності використання, швидкості доступу до інформації та ефективності роботи користувачів. Основною вимогою до

інтерфейсу є інтуїтивна зрозумілість, що дозволяє мінімізувати час на навчання та знизити кількість помилок під час взаємодії з системою [2].

Важливою характеристикою є чітка і логічно структурована навігація, яка забезпечує швидкий доступ до основних функцій системи: перегляду стану сайтів, аналітики, звітів та налаштувань. Навігаційні елементи повинні бути уніфікованими та передбачуваними, що позитивно впливає на користувацький досвід [17].

Сучасні рішення також орієнтовані на адаптивний дизайн, який забезпечує коректне відображення інтерфейсу на різних типах пристроїв — від настільних комп’ютерів до смартфонів і планшетів. Для цього активно застосовуються HTML та CSS, що відповідають за структуру сторінок і їхнє візуальне оформлення, а також дозволяють гнучко змінювати макет залежно від розміру екрана [34]. Додатково використовується фреймворк Bootstrap, який надає готову сіткову систему та адаптивні компоненти, що значно пришвидшує розробку інтерфейсів [30]. Для створення динамічних і інтерактивних елементів широко застосовується ReactJS, який дозволяє оновлювати окремі частини інтерфейсу без перезавантаження сторінки та забезпечує зручну взаємодію з користувачем [38]. У сукупності ці технології роблять адаптивний дизайн особливо актуальним для спеціалістів, які працюють віддалено та потребують постійного доступу до даних у будь-який час.

Для підвищення наочності інформації широко використовується графічне відображення даних у вигляді діаграм, графіків і таблиць. Візуалізація показників дозволяє швидко оцінювати динаміку змін, виявляти аномалії та приймати управлінські рішення без детального аналізу сирих даних [14].

З функціональної точки зору сучасні системи автоматизації забезпечують моніторинг доступності веб-сайтів із можливістю налаштування інтервалів перевірок, що дозволяє адаптувати систему під вимоги конкретного проєкту.

Важливою складовою є генерація аналітичних звітів, які надають узагальнену інформацію про стабільність роботи, ефективність SEO та інші ключові показники.

Окрему увагу приділено системам автоматичного сповіщення, які інформують користувачів про критичні події через Telegram, електронну пошту або SMS. Такий підхід значно скорочує час реагування на проблеми та знижує ризики простою веб-ресурсів.

Додатково, інтеграція з платіжними платформами створює можливості для монетизації сервісу шляхом впровадження підписок або платних тарифних планів.

Сучасні системи автоматизації управління веб-проєктами будуються на основі багаторівневої архітектури, що забезпечує гнучкість, масштабованість і високу надійність. Такий підхід передбачає логічний поділ системи на окремі рівні, кожен з яких виконує визначені функції.

Модульна архітектура дозволяє розширювати функціональність системи без необхідності зміни базового коду, що спрощує підтримку та масштабування.

Для обґрунтування вибору архітектурного рішення в межах дослідження було проведено порівняльний аналіз монолітного та мікросервісного (модульного) підходів (табл. 1.2).

Таблиця 1.2.

Порівняльна характеристика архітектурних підходів до побудови систем автоматизації

Критерій порівняння	**Монолітна архітектура**	**Мікросервісна (модульна) архітектура**
1	**2**	**3**
Швидкість розгортання	Високана початкових етапах	Потребує складного налаштування
Масштабованість	Обмежена (тільки всієї системи)	Висока (окремих модулів/сервісів)
Стійкість до помилок	Помилка в одному модулі зупиняє всю систему	Помилка в одному модулі не впливає на інші

Продовження табл. 1.2

1	2	3
Складність підтримки	Зростає експоненціально з кодом	Потребує навичок оркестрації
Стек технологій	Єдиний для всього проєкту	Можливість використання різних мов/БД
Тестування	Потребує повного регресійного тестування системи	Дозволяє проводити ізольоване тестування окремих функцій
Складність оновлення	Потребує перезбірки та перезапуску всього додатка	Можливість оновлювати окремі частини без зупинки сервісу
Витрати ресурсів	Нижчі на старті, але зростають при розширенні	Вищі через необхідність підтримки зв'язків між модулями

Джерело: [18]

Клієнтський рівень (Frontend) відповідає за безпосередню взаємодію з користувачем та формування зручного користувацького досвіду. Він реалізується за допомогою сучасних JavaScript-фреймворків, таких як React, Vue.js або Angular, які дозволяють створювати модульні, масштабовані та високопродуктивні інтерфейси [3]. Завдяки використанню компонентного підходу спрощується підтримка коду та його подальший розвиток. Адаптивний дизайн забезпечує коректне відображення системи на різних пристроях, а інтерактивна візуалізація аналітичних даних реалізується за допомогою бібліотек Chart.js або D3.js, що дозволяє користувачам швидко аналізувати показники та приймати обґрунтовані рішення [36].

Серверний рівень (Backend) реалізується з використанням мови програмування Python та фреймворків Django, Flask або FastAPI і виконує ключову роль у функціонуванні всієї системи. Він відповідає за обробку

клієнтських запитів, реалізацію бізнес-логіки, перевірку прав доступу, взаємодію з базами даних та інтеграцію із зовнішніми сервісами [7].

Архітектура бекенду зазвичай передбачає поділ на окремі логічні модулі, зокрема модулі моніторингу, аналітики, управління користувачами та взаємодії з Telegram API. Для підвищення швидкодії та обробки великої кількості запитів застосовується асинхронне виконання фонових завдань із використанням Celery або RQ [8].

Рівень баз даних забезпечує надійне зберігання як структурованої, так і неструктурованої інформації. Реляційні системи управління базами даних, такі як PostgreSQL або MySQL, використовуються для зберігання даних про користувачів, проєкти, налаштування та історію моніторингу. У свою чергу, NoSQL-рішення, наприклад MongoDB, є доцільними для збереження логів, подій та великих масивів неструктурованих даних. Використання ORM-інструментів спрощує роботу з базами даних, зменшує кількість помилок і підвищує рівень безпеки доступу до інформації [16].

Інтеграційний рівень забезпечує взаємодію системи з зовнішніми сервісами та платформами. Зокрема, використовується Telegram API для автоматизованого надсилання повідомлень і сповіщень користувачам, аналітичні сервіси — для збору та обробки статистичних даних, а також платіжні системи (Stripe, LiqPay, Fondy) для реалізації механізмів підписок, білінгу та оплати послуг. Такий підхід дозволяє розширювати функціональність системи без суттєвих змін її основної архітектури.

Рівень автоматизації та моніторингу включає використання планувальників завдань, таких як Cron або Celery Beat, які забезпечують регулярне виконання перевірок, оновлення даних і запуск фонових процесів.

Додатково здійснюється збір логів, метрик продуктивності та помилок, що дозволяє оцінювати стабільність роботи системи, своєчасно виявляти проблеми та виконувати подальшу оптимізацію і масштабування.

РОЗДІЛ 2. ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ВЕБ-ДОДАТКУ DODoBot

2.1. Аналіз вимог та технічне обґрунтування розробки

Першим етапом у розробці веб-застосунку DoDoBot стало вивчення потреб майбутніх користувачів. Для цього використовувалися прості, але ефективні способи: перегляд публічних відгуків у мережі, аналіз коментарів у соціальних медіа та тематичних форумах, а також власні спостереження за роботою подібних сервісів. Такий підхід дав змогу сформувати початкове уявлення про очікування та проблеми, з якими стикаються користувачі.

Зібрані дані були структуровані та поділені на дві основні групи:

- функціональні вимоги (які можливості повинен мати DoDoBot);
- нефункціональні вимоги (зручність використання, стабільність роботи, швидкодія, простота інтерфейсу).

Особлива увага приділялась аналізу конкурентного середовища. Для цього було обрано кілька найбільш популярних аналогів. Порівнювалися їхні ключові функції, дизайн інтерфейсу, рівень зручності, швидкість роботи, а також реакція користувачів у вигляді відгуків і рейтингу. Окремо відзначалися сильні сторони конкурентів, які варто взяти за приклад, та слабкі місця, яких необхідно уникати у DoDoBot.

Такий підхід дозволив визначити оптимальний набір функцій і зрозуміти, у якому напрямку рухатися, щоб застосунок був не просто копією існуючих рішень, а мав власну цінність і переваги. У результаті був сформований чіткий перелік вимог, що став основою для подальшого проєктування і розробки DoDoBot.

2.2. Проєктування та обґрунтування вибору інструментарію розробки

Формування користувацького інтерфейсу є одним із центральних аспектів розробки вебзастосунку DoDoBot, адже саме UI/UX-дизайн визначає якість взаємодії користувача з системою та рівень задоволеності продуктом. Для цього

етапу було обрано платформу Figma, яка відповідає сучасним стандартам інструментів для проєктування інтерфейсів.

Вибір Figma обґрунтований її функціональними можливостями. Серед ключових переваг можна відзначити широкий спектр інструментів для створення високоточних wireframes і mockups, побудову інтерактивних прототипів та організацію інформаційної архітектури застосунку. Це дозволяє ще на ранніх етапах перевірити логіку навігації та сценарії користувацької взаємодії (user flows) [13].

Важливим чинником є також підтримка collaborative design. Figma забезпечує синхронну роботу членів команди в режимі реального часу, що дає змогу дизайнерам, розробникам і аналітикам одночасно вносити правки та узгоджувати рішення без додаткових витрат часу. Такий підхід суттєво підвищує ефективність розробки й мінімізує ризик комунікаційних помилок [32].

Окремої уваги заслуговує можливість проведення попереднього юзабіліті-тестування. Завдяки інтерактивним прототипам можна оцінити зручність навігації, швидкість доступу до основних функцій та зрозумілість інтерфейсних елементів ще до початку програмної реалізації [39]. Це дозволяє вчасно виявити проблемні зони та оптимізувати дизайн на ранніх стадіях, що знижує вартість виправлень у майбутньому.

Таким чином, використання Figma як базового інструменту UI/UX-дизайну забезпечило можливість інтегрувати сучасні підходи до проєктування, реалізувати принципи юзабіліті та отримати якісний прототип DoDoBot, який став основою для подальшої розробки.

Після завершення етапу аналізу вимог до DoDoBot постало питання щодо вибору оптимальної мови програмування для реалізації серверної частини. Було вирішено використати Python, що пояснюється його простотою, гнучкістю та широким спектром застосувань у розробці веб-застосунків [21].

Однією з ключових переваг Python є менеджер пакетів pip, який значно спрощує інтеграцію додаткових бібліотек і модулів у проєкт. Завдяки pip розробник може швидко додавати необхідну функціональність, встановлюючи

пакети з великого онлайн-репозиторію Python, що прискорює процес створення і тестування системи [19].

Основні переваги використання pip у розробці:

- простота інтеграції та оновлення: пакети можна встановлювати або оновлювати кількома командами, без складних конфігурацій, що економить час на підготовку середовища;
- широкий спектр бібліотек: pip надає доступ до численних модулів для різних завдань – від веб-фреймворків і роботи з базами даних до бібліотек для обробки даних і машинного навчання;
- контроль версій та залежностей: можна точно вказати версію пакета для проєкту, що дозволяє уникнути конфліктів між бібліотеками та забезпечує стабільність роботи системи;
- інтеграція з IDE: більшість сучасних середовищ розробки, таких як PyCharm, Visual Studio Code або Sublime Text, підтримують pip «з коробки», що робить керування пакетами швидким і зручним.

Завдяки цим можливостям, Python з pip стає надійним вибором для веб-розробки, забезпечуючи гнучкість, швидкість інтеграції нових модулів та стабільність функціонування проєкту [20].

Для реалізації серверної частини DoDoBot обрано Django як основний фреймворк проєкту. Це рішення ґрунтується на кількох критично важливих чинниках. Django дозволяє швидко розробляти веб-застосунки завдяки вбудованим компонентам та чітким конвенціям, що прискорює реалізацію бізнес-логіки і дозволяє ефективно інтегрувати функціональні модулі без додаткових витрат часу [31].

Фреймворк також відомий високим рівнем безпеки: він забезпечує захист від SQL-ін’єкцій, Cross-Site Scripting, CSRF-атак та інших поширених загроз. Наявність готових рішень для аутентифікації користувачів, управління базами даних та адміністративного інтерфейсу значно спрощує розробку, дозволяючи зосередитися на унікальних функціях застосунку [10].

Проєкт реалізований із використанням мікросервісної архітектури, де частина окремих сервісів створюється на Flask, а при потребі нові модулі можуть бути реалізовані на FastAPI або інших технологіях. Такий підхід забезпечує гнучкість у виборі інструментів під конкретні завдання, одночасно зберігаючи інтеграцію у єдину систему.

Ключовою перевагою цієї архітектури є можливість розміщення мікросервісів на різних серверах або навіть в різних середовищах виконання. Це дозволяє масштабувати систему горизонтально, оптимізувати використання ресурсів та підвищувати відмовостійкість: у випадку збоїв одного сервісу інші продовжують працювати без перебоїв [4].

Крім того, така модель полегшує незалежну розробку, тестування та оновлення окремих модулів, що значно прискорює релізи і зменшує ризики інтеграційних помилок.

Використання мікросервісів на різних серверах також дозволяє обирати оптимальні технології для конкретних задач. Наприклад, сервіс, що обробляє велику кількість запитів у реальному часі, може бути реалізований на високопродуктивному FastAPI, тоді як інший, менш критичний сервіс – на Flask або Django [12]. Такий підхід підвищує ефективність, знижує навантаження на основний сервер і забезпечує гнучкість архітектури.

Отже, поєднання Django для основної логіки з мікросервісами на Flask, можливістю інтеграції FastAPI та розміщенням сервісів на різних серверах дозволяє досягти оптимального балансу між швидкістю розробки, безпекою, масштабованістю та гнучкістю системи. Така архітектура робить DoDoBot сучасним, надійним і здатним легко адаптуватися до майбутніх потреб.

Для забезпечення надійного зберігання та обробки даних у DoDoBot основною базою даних обрано PostgreSQL, яка використовується у головному сервісі на Django. Такий вибір пояснюється необхідністю високої стабільності, масштабованості та широких функціональних можливостей. PostgreSQL довела свою надійність у великих проєктах та при значних навантаженнях. Механізми резервного копіювання та реплікації дозволяють забезпечити високу доступність

даних і запобігати їх втраті, що особливо важливо для ключових компонентів системи [37].

Основні переваги PostgreSQL включають підтримку складних типів даних, роботу з JSON, геопросторові функції та потужні інструменти для аналітики. Це робить її гнучкою та універсальною СКБД, здатною ефективно обробляти різноманітні завдання та великі обсяги даних [15].

Мікросервісна архітектура DoDoBot передбачає, що окремі сервіси можуть використовувати власні бази даних за потребою. Важливо, що для цих сервісів не застосовується SQLite: хоча ця СКБД добре підходить для невеликих проєктів чи прототипів, вона не забезпечує достатньої продуктивності та масштабованості для завдань, що потребують одночасної роботи великої кількості запитів або обробки значних обсягів даних. Натомість сервіси можуть використовувати PostgreSQL, MySQL або інші СКБД, обираючи оптимальне рішення залежно від специфіки конкретного модуля.

Таким чином, поєднання головного сервісу на Django з PostgreSQL і гнучких баз даних для мікросервісів забезпечує надійність, масштабованість і адаптивність системи. Такий підхід дозволяє ефективно розподіляти навантаження, оптимізувати ресурси та зберігати високу продуктивність незалежно від розвитку та масштабування проєкту.

2.3. Розробка алгоритму роботи веб-додатку

Веб-додаток DoDoBot побудований за модульною архітектурою з використанням патерну «Модель-Вид-Контролер» (MVC), що забезпечує чітке розділення відповідальностей між логікою бізнес-процесів, представленням даних та контролем взаємодії користувача з системою.

Основна частина проєкту – Django-додаток mysite, який містить кілька піддодатків (apps), а також набір мікросервісів для виконання специфічних завдань. Мікросервіси можуть бути реалізовані на Flask, FastAPI або інших технологіях, що дозволяє адаптувати систему під різні сценарії та оптимально масштабувати її [33].

Функціонал адміністратора Передбачає використання окремого інструментарію для повного контролю над системою, а саме:

- управління блогами та контентом: додавання, редагування та публікація статей, керування категоріями та SEO-полями;
- мікросервіси: можливість додавати нові сервіси, керувати існуючими, налаштовувати їх параметри та інтеграцію з головним додатком;
- управління користувачами та проектами: перегляд активних користувачів, контроль їхніх проектів, управління правами доступу та історією транзакцій.

Це дозволяє адміністратору повністю контролювати функціонування системи та забезпечувати безперебійну роботу всіх сервісів.

Під час відвідування веб-додатку користувач бачить:

- головну сторінку, що відображає загальну інформацію про сервіс, доступні інструменти та поточні проекти;
- сторінку блогів/новин, де можна переглядати публікації, категорії та новини платформи;
- можливість реєстрації та авторизації, що забезпечує доступ до персонального кабінету (дашборду) (див. додаток рис. А. 2.1.)

Після авторизації користувач отримує доступ до дашборду, який включає:

- перелік проектів з можливістю додавання нового проекту, управління блоками та налаштуваннями;
- мої тикети – інтегрована система обміну повідомленнями з технічною підтримкою;
- баланс у різних валютах, включно з доларом США та криптовалютами (наприклад, біткоїн), для зручності фінансових операцій;
- панель керування, де доступні наступні розділи

(див. додаток рис. А. 2.2.).:

  - особисті дані – відображення профільної інформації користувача;

- редагувати профіль – можливість змінити Ім’я, Прізвище, Email, Номер телефону, а також завантажити фото для аватарки;
- SEO інструменти – окрема сторінка з набором стандартних інструментів для оптимізації.

Сторінка «Особисті дані» призначена для відображення та управління профільною інформацією користувача. Вона забезпечує доступ до основних параметрів облікового запису та надає можливість їх редагування (рис. 2.3). Основний зміст сторінки складається з двох блоків:

- особисті дані – відображення профільної інформації користувача у вигляді структурованих полів: ім’я, прізвище, логін, email, номер телефону, баланс;
- редагувати профіль – функціональний блок, який дозволяє користувачеві вносити зміни до своєї інформації. Доступні можливості: змінити Ім’я, Прізвище, Email, Номер телефону; завантажити фото для встановлення аватарки.

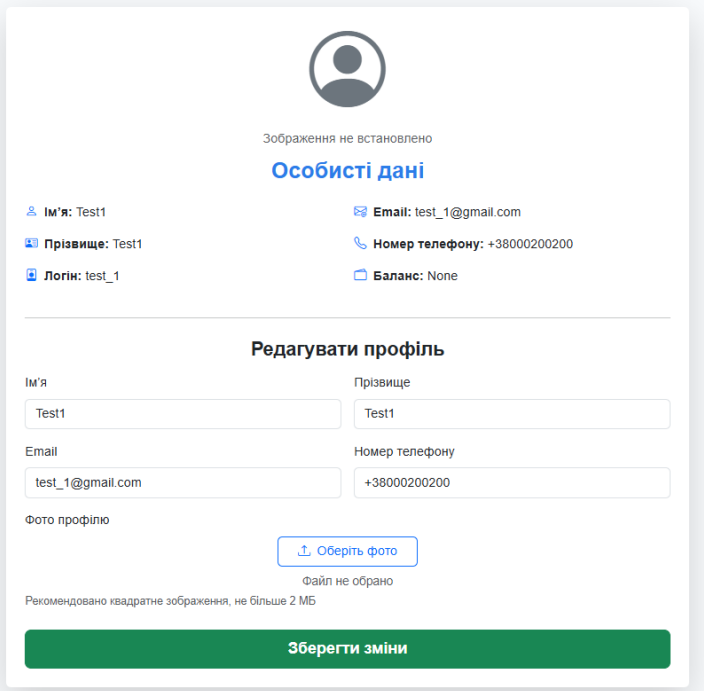


Рис. 2.3 Особисті дані

Сторінка «Поповнення балансу» надає користувачам можливість здійснювати фінансові операції з поповнення власного рахунку у системі.

Інтерфейс побудований таким чином, щоб процес був максимально простим, гнучким та зручним для кінцевого користувача.

Основні елементи сторінки:

- поле введення суми – «Сума поповнення (USD)». Користувач вводить бажану суму у доларах США, при цьому система автоматично перевіряє відповідність значення мінімальним та максимальним обмеженням, встановленим адміністратором (див. додаток рис. А. 2.4.);
- вибір способу оплати – випадаючий список «Виберіть спосіб оплати», у якому доступні варіанти: WayForPay – інтеграція з популярним платіжним сервісом для роботи з банківськими картками. CryptoBot – сервіс для оплати за допомогою криптовалют. Завдяки гнучкій архітектурі системи, адміністратор може у майбутньому додавати нові платіжні методи без необхідності змінювати програмний код, а також легко змінювати мінімальні та максимальні значення для поповнення;
- кнопка підтвердження операції – «Поповнити». Після натискання користувач автоматично перенаправляється на сторінку вибраного платіжного сервісу для завершення транзакції.

Функціонал сторінки передбачає:

- інтерактивну перевірку коректності введених даних;
- динамічне керування параметрами поповнення (мінімальна/максимальна сума);
- можливість масштабування платіжних інструментів без додаткового доопрацювання програмного забезпечення.

Сторінка «Історія поповнень» призначена для відображення всіх фінансових операцій користувача, пов’язаних із поповненням рахунку. Основним елементом інтерфейсу є інтерактивна таблиця, яка дозволяє швидко отримати інформацію про минулі транзакції та керувати ними. Основні компоненти таблиці (див. додаток рис. А. 2.5.):

- дата – дата та час створення операції;
- сума (USD) – розмір поповнення у доларах США;

- метод – обраний спосіб оплати (наприклад, WayForPay або CryptoBot);
- статус – поточний стан операції: Очікується – платіж створено, але ще не підтверджено. Сплачено – транзакцію успішно завершено. Прострочено – користувач не здійснив оплату у відведений час. Відмінено – операцію скасовано користувачем або системою;
- опис – додаткова інформація щодо операції. Наприклад: Поповнення через WayForPay, OrderRef: DH6220838651;
- кнопка «Деталі» – дозволяє переглянути розширену інформацію про конкретну транзакцію;

Після натискання кнопки «Деталі» відкривається спливаюче вікно, де користувач бачить (див. додаток рис. А. 2.6.):

- повну інформацію про транзакцію (ідентифікатор, час створення, вибраний метод, статус, технічний опис);
- якщо статус = «Очікується», доступна додаткова кнопка «Оплатити», яка перенаправляє користувача на сторінку оплати відповідного сервісу.

Особливості реалізації:

- вся інформація відображається у зручному табличному форматі з можливістю сортування та фільтрації записів;
- система підтримує масштабованість: при збільшенні кількості операцій передбачено пагінацію (посторінковий перегляд);
- використання модального вікна дозволяє не залишати основну сторінку та взаємодіяти з даними у більш інтерактивному форматі.

Важливим елементом системи є модуль «Останні тікети», який забезпечує користувачам можливість ефективної взаємодії з технічною підтримкою.

Основною метою даного функціоналу є спрощення процесу комунікації між користувачем та адміністраторами системи, надання швидкої допомоги у вирішенні проблем, відповідей на запитання та супроводу під час роботи з програмним продуктом (див. додаток рис. А. 2.7.).

Інтерфейс модуля реалізований у вигляді окремого блоку з двома основними кнопками:

- «усі тікети» – відкриває список усіх звернень користувача, включаючи як активні, так і закриті;
- «відкрити тікет» – ініціює створення нового звернення до служби підтримки.

При натисканні на кнопку «Відкрити тікет» користувачеві відкривається модальне вікно з формою для заповнення. Форма містить такі поля:

- тема – короткий заголовок звернення, який дозволяє швидко зрозуміти його зміст;
- повідомлення – детальний опис проблеми, запитання або пропозиції;
- прикріпити файл (необов’язково) – можливість додати зображення, документ чи інший матеріал, що допоможе точніше ідентифікувати проблему (див. додаток рис. А. 2.8.).

Після заповнення та відправки форми звернення автоматично зберігається у базі даних, а адміністратор отримує відповідне повідомлення у панелі керування. Це дає змогу оперативно реагувати на вхідні тікети та забезпечує швидкий цикл «питання–відповідь».

Користувач має змогу у будь-який момент переглянути історію своїх звернень: як відкритих (що перебувають на розгляді), так і закритих (вирішених). Для зручності реалізовано сортування та фільтрацію тікетів, що спрощує навігацію навіть при великій кількості звернень (див. додаток рис. А. 2.9.).

Особливістю модуля є можливість ведення повноцінного чату між користувачем та адміністратором. У рамках кожного відкритого тікета формується діалог, де сторони можуть обмінюватися повідомленнями у режимі реального часу. Це дозволяє уточнювати деталі проблеми, надсилати додаткові файли або отримувати проміжні консультації, не створюючи нових звернень (див. додаток рис. А. 2.10.).

Для адміністратора передбачено інструменти управління тікетами: зміна їх статусу (наприклад, «новий», «у роботі», «закритий»), залишення коментарів, відповідей, а також ведення аналітики за кількістю отриманих і опрацьованих запитів.

Завдяки впровадженню модуля «Останні тікети» користувачі отримують:

- зручний інструмент для комунікації з технічною підтримкою;
- прозору систему відстеження статусу своїх звернень;
- можливість отримувати швидку допомогу у форматі діалогу;
- впевненість у тому, що їхня проблема не залишиться без уваги.

Функціональний блок «Ваші проєкти» призначений для управління проєктами користувача та забезпечує зручний інтерфейс для роботи з ними. У даному розділі зібрані всі наявні проєкти користувача з відображенням ключових характеристик кожного з них, а також інструменти для редагування, керування та організації робочих процесів.

У процесі перегляду списку проєктів користувачеві відображається інформація для кожного елемента, а саме:

- назва проєкту – унікальне ім’я, яке користувач задає при створенні;
- статус – показує поточний стан («Призупинено» / «Працює»);
- дата створення – зручна для контролю часу ініціалізації;
- кількість блоків – відображає, скільки робочих завдань (процесів) включає проєкт.

Ця інформація дозволяє швидко зорієнтуватися у переліку навіть за наявності великої кількості проєктів (рис. 2.11).

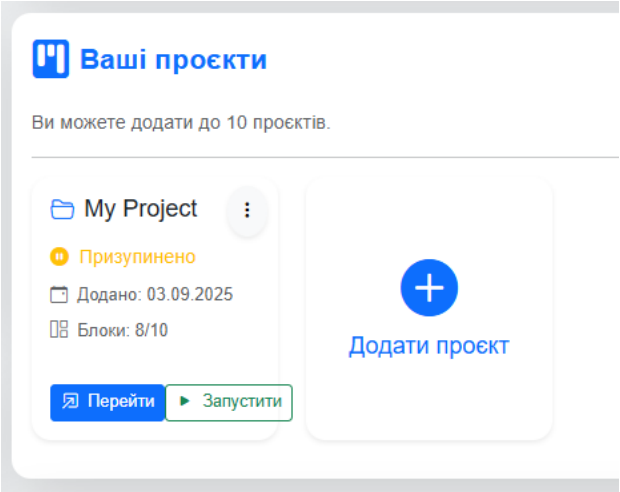


Рис. 2.11 Список проєктів

Для зручності взаємодії з проєктами в інтерфейсі передбачені інструменти керування. У нижній частині кожної картки проєкту розташовані кнопки для взаємодії:

- «перейти» – відкриває робочу область конкретного проєкту, де користувач має можливість взаємодіяти з усіма блоками й процесами;
- «запустити / зупинити» – дозволяє одночасно запускати або призупиняти всі робочі процеси всередині проєкту. Це забезпечує швидкий контроль без необхідності вручну змінювати статус кожного окремого завдання;
- кнопка у вигляді трьох вертикальних точок – відкриває модальне вікно «налаштування проєкту», де доступні додаткові опції (рис. 2.12).

У модальному вікні «Налаштування проєкту» користувач має можливість змінювати базові параметри: назва проєкту, опис проєкту.

Також передбачені додаткові дії:

- архівувати – переміщення проєкту до архіву (аналог корзини). Проєкт зберігається, але не відображається у загальному списку активних;
- дублювати – створення копії з усіма існуючими процесами та налаштуваннями;
- експортувати – збереження проєкту у форматі .json, що дозволяє отримати локальну копію зі всіма параметрами;
- видалити – повне видалення проєкту з системи (рис. 2.13).

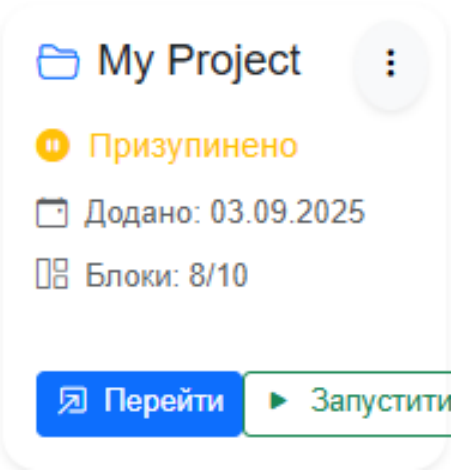


Рис. 2.12 Керування проєктом

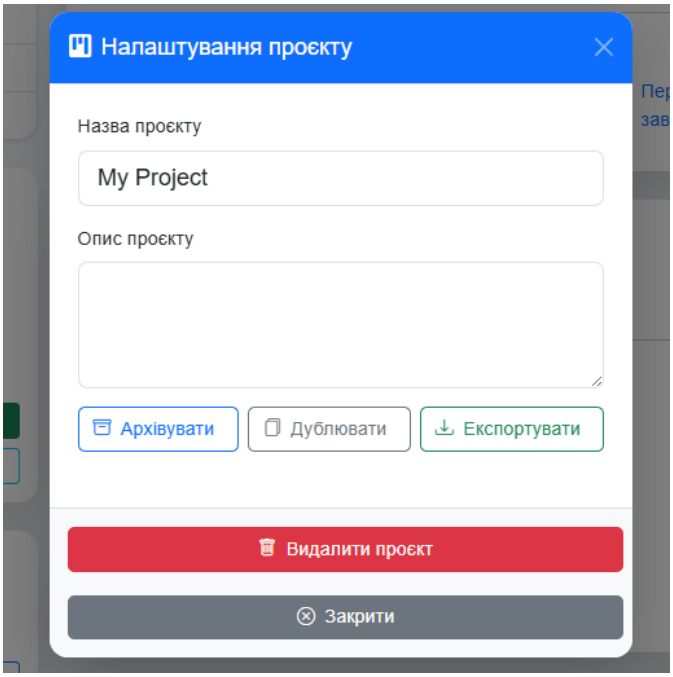


Рис. 2.13 Налаштування проєкту

За замовчуванням користувач може створити до 10 проєктів, однак кількість може змінюватися залежно від тарифного плану. Ліміти на кількість проєктів, а також інші параметри тарифів налаштовуються адміністратором у панелі керування (рис. 2.14).

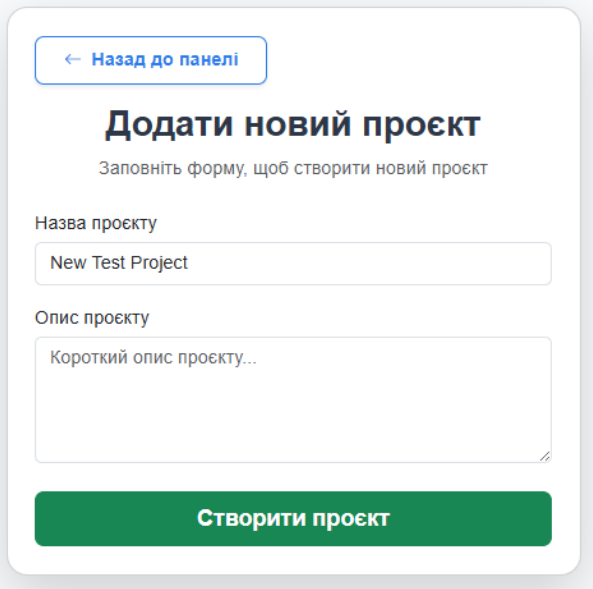


Рис. 2.14 Додавання нового проєкту

Таким чином, блок «Ваші проєкти» виконує роль централізованої панелі управління, де користувач може швидко орієнтуватися між своїми проєктами, редагувати їх, керувати робочими процесами, а також здійснювати експорт чи архівацію для зручності організації.

Робоча область проєкту в DoDoBot призначена для організації та управління процесами автоматизації всередині конкретного проєкту. Вона забезпечує користувачу зручний та інтуїтивно зрозумілий інтерфейс для перегляду доступних work-блоків, їх налаштування, запуску та контролю у реальному часі (див. додаток рис. А. 2.15.).

Робоча область поділена на дві основні секції:

Під час роботи з проєктом користувачеві доступна ліва секція інтерфейсу, у якій зосереджені доступні work-блоки. У цій секції відображаються всі доступні блоки (процеси автоматизації), які користувач може додати до проєкту (рис. 2.16). Кожен work-блок відповідає за певну функціональність і має власні параметри налаштувань.

На даний момент доступні наступні work-блоки:

- перевірка сайту – автоматична перевірка сайту на відповідність стандартам SEO та відстеження позицій у пошукових системах. Перевіряє:

- позиції сайту за ключовими словами;
- якість зовнішніх та внутрішніх посилань;
- оптимізацію контенту (мета-теги, ключові слова, заголовки, структуровані дані);
- швидкість завантаження та адаптивність під мобільні пристрої;
- статистику та рейтинг сайту, що дозволяє відстежувати динаміку змін.

- порівняння сайтів - Аналізує два сайти одночасно для порівняння SEO-показників і просування. Перевіряє:
  - позиції сайтів у пошукових системах;
  - якість посилань і контенту;
  - загальний рейтинг просування;
  - дає змогу швидко визначати, який сайт має кращу стратегію SEO.

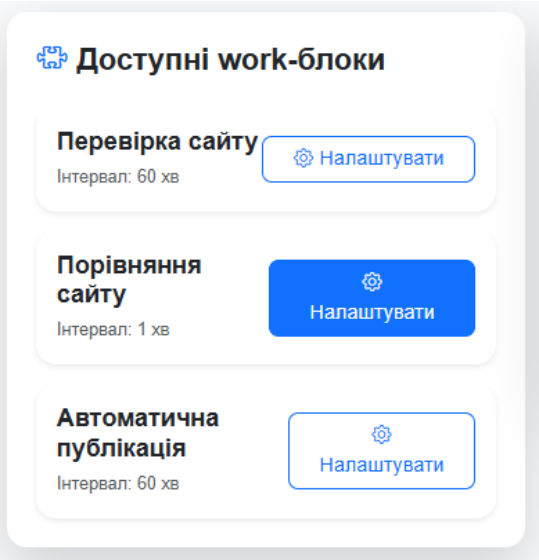


Рис. 2.16 Доступні work-блоки

Щоб додати work-блок до проекту:

1. натискаємо кнопку «Налаштувати» поруч із потрібним блоком;

2. відкривається модальне вікно з полями налаштувань (поля різняться в залежності від блока);

3. після заповнення всіх параметрів натискаємо «Додати до проекту», і блок переноситься до правої секції для запуску та управління (рис. 2.17);

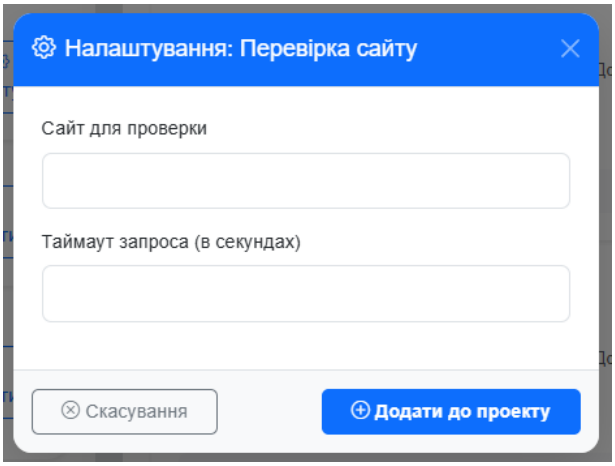


Рис. 2.17 Налаштування блока

Під час роботи з робочою областю проєкту користувачеві доступна права секція інтерфейсу, у якій відображаються work-блоки, вже додані до проєкту. Це ті блоки, що мають налаштовані параметри та готові до роботи (рис. 2.18).
До основних можливостей цієї секції належать:

- моніторинг стану блоків – відображення статусу (активний, призупинено, завершено);
- управління процесами – запуск, зупинка або тимчасове призупинення роботи блоків;
- редагування налаштувань – можливість змінювати параметри work-блока через модальне вікно;
- історія виконання – перегляд логів та статистики роботи блоків для оцінки ефективності;
- динамічне розширення – додавання нових блоків у міру оновлення системи або розширення функціоналу.

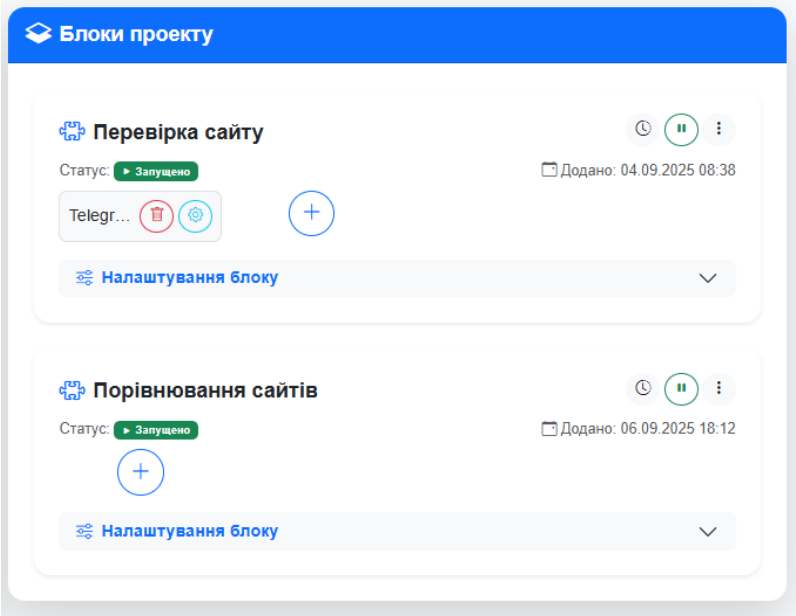


Рис. 2.18 Список доданих блоків

Кожен work-блок може генерувати звіт або результат своєї роботи, і щоб користувачу не доводилося постійно перевіряти історію на сайті, є можливість налаштувати сповіщення про результати роботи (див. додаток рис. А. 2.19.).

Для додавання сповіщення натискаємо кнопку «+» на певному work-блоці. Після цього відкриється модальне вікно модальне «Доступні повідомлення», де можна вибрати куди відправляти звіт (Gmail, Telegram Bot) та спосіб доставки (платні або безкоштовні опції).

Приклад:

- відправка сповіщень Gmail – платно: 0.80 грн за повідомлення;
- telegram bot – безкоштовно або за окремою тарифною опцією;
- після вибору способу натискаємо «Додати», і відкривається модальне вікно «Налаштування повідомлення» з полями, специфічними для кожного способу. Для Telegram Bot це номер телефону, пріоритет повідомлення (середній, високий, низький), токен бота, ID чату;
- після натискання «Зберегти» налаштування зберігаються і всі звіти від цього work-блока будуть надсилатися відповідно до заданих параметрів (див. додаток рис. А. 2.10.).

Додатково для кожного способу повідомлень є:

- кнопка налаштування блоку – для перегляду або редагування налаштувань (рис. 2.21);

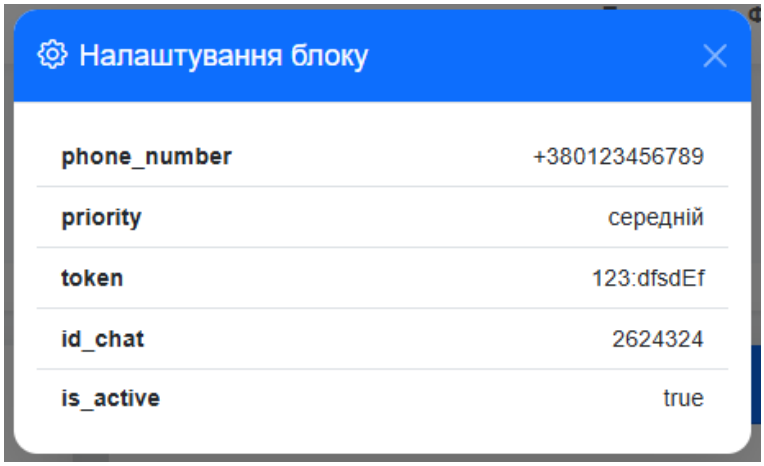


Рис. 2.21 Перегляд налаштування блоку

- кнопка видалення – для видалення способу сповіщення.

Таким чином, права секція робочої області дозволяє не лише запускати та контролювати процеси, а й організувати автоматичне надсилання звітів та результатів роботи work-блоків у зручний для користувача спосіб (рис. 2.22).

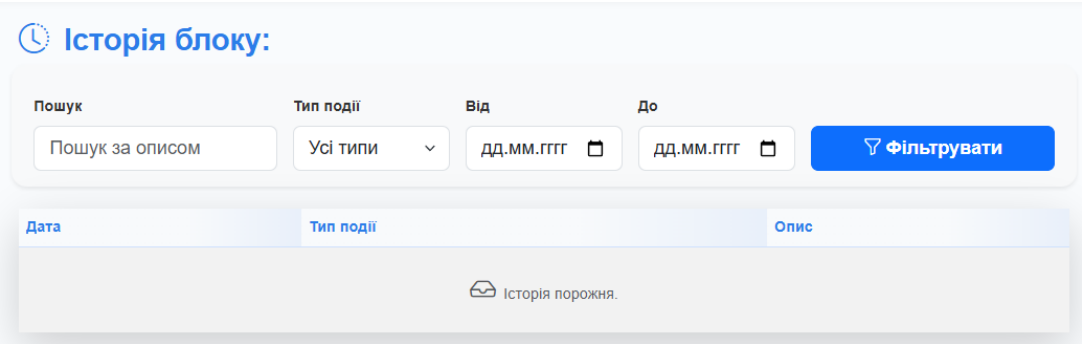


Рис. 2.22 Історя блоку

Для керування системою та її складовими використовується адміністративна панель DoDoBot. Вона є центральним інструментом управління, доступна за адресою /admin та відкривається лише після авторизації адміністратора (рис. 2.23).

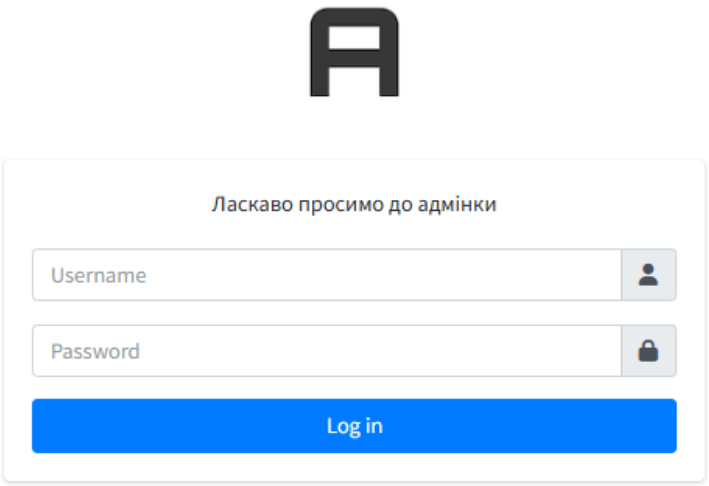


Рис. 2.23 Авторизація до адмін панелі

Після успішної авторизації користувач переходить до панелі адміністратора (дашборду), яка забезпечує швидкий доступ до ключових модулів і налаштувань системи (див. додаток рис. А. 2.24.).

Основні можливості адміністративної панелі поділені на окремі модулі. Зокрема, для роботи з контентом використовується модуль Blog App, який дозволяє:

- редагувати та видаляти публікації;
- прикріплювати зображення, задавати категорії та теги;
- використовувати як канал комунікації з користувачами — для публікації оновлень, анонсів нових work-блоків чи інструкцій.

Для управління ключовими параметрами системи передбачений модуль Constance, який забезпечує швидку зміну налаштувань без необхідності втручання у вихідний код. Зокрема, він надає такі можливості:

- налаштування тарифних лімітів (кількість доступних проєктів), параметрів оплати, системи сповіщень, API-ключів тощо;
- забезпечує централізоване зберігання конфігурацій, що є важливим для стабільної роботи системи.

Модуль Robot App є основним функціональним компонентом системи та відповідає за управління work-блоками.

У цьому розділі користувачам доступний набір work-блоків (Available blocks), а розробник має можливість додавати нові блоки, редагувати наявні та видаляти застарілі.

Додавання нового work-блоку — це не просто створення нового запису в базі даних. Це повноцінна інтеграція окремого мікросервісу, який може бути реалізований навіть у вигляді іншої програми (наприклад, Flask-сервіс, що працює на окремому сервері та виконує сканування SEO-показників сайту) (див. додаток рис. А. 2.25.).

Після натискання на кнопку Add available block відкривається сторінка, де розробник може вказати (див. додаток рис. А. 2.26.):

- name – назва work-блоку;
- block type – тип блоку (аналітика, сканування, моніторинг тощо);
- description – короткий опис його функціоналу;
- default interval – стандартний інтервал запуску;
- api endpoint – URL мікросервісу, з яким буде взаємодіяти система;
- default settings – стандартні налаштування у форматі JSON (які підтягуються користувачу за замовчуванням);
- settings schema – JSON-схема, яка визначає, які поля з налаштуваннями буде бачити користувач у своєму інтерфейсі. Вона також використовується мікросервісом для правильної обробки даних.

Завдяки такій архітектурі система є максимально гнучкою: будь-який новий мікросервіс можна підключити до DoDoBot без зміни основного коду, лише через адмін-панель (див. додаток рис. А. 2.27.).

Для управління користувачами та їх фінансовими операціями в адміністративній панелі передбачений модуль User App. Він об’єднує кілька основних підсекцій, що забезпечують зручний контроль даних користувачів.

У підсекції Custom users відображаються всі користувачі, зареєстровані в системі. Для кожного користувача доступна детальна інформація, зокрема:

- username – унікальний логін;
- email – електронна адреса;

- full name – повне ім’я;
- phone number – номер телефону;
- balance – поточний баланс користувача;
- is active – статус активності акаунта;
- date joined – дата створення облікового запису.

Завдяки цьому адміністратор може швидко перевірити стан акаунтів, відстежити активність та за потреби виконати адміністративні дії (заблокувати чи активувати користувача).

Підсекція Payment histories призначена для зберігання та перегляду історії всіх поповнень і фінансових транзакцій користувачів. У цій секції для кожного платежу відображається така інформація:

- user – хто виконав платіж;
- amount – сума;
- method – спосіб оплати;
- status – статус транзакції;
- order reference – унікальний ідентифікатор замовлення;
- description – опис платежу;
- json – додаткова технічна інформація у форматі JSON.

Приклад JSON-запису:

{
"invoiceUrl": "https://secure.wayforpay.com/invoice/i53f7e0adf1f0",
"reason": "Ok",
"reasonCode": 1100,
"qrCode": "https://wayforpay.com/qr/img/i53f7e0adf1f0?type=i&size=200",
"orderReference": "DH6220838651"
}

Який характеризує відповідь платіжного сервісу після успішного формування рахунку на оплату.

Підсекція Tickets призначена для роботи з користувацькими зверненнями до служби технічної підтримки. Система реалізована у форматі чату, що дозволяє

адміністратору або оператору підтримки вести діалог із клієнтом у режимі реального часу.

У списку звернень відображаються ключові параметри:

- title – тема звернення;
- user – користувач, який відкрив тікет;
- оператор – адміністратор або співробітник підтримки, що закріплений за цим зверненням;
- status – поточний стан тікету (відкритий, у роботі, закритий);
- created at – дата та час створення звернення.

Завдяки цьому адміністратори мають можливість оперативно реагувати на проблеми користувачів, вести переписку, залишати відповіді та закривати тікети після вирішення питання. Такий підхід значно підвищує рівень комунікації та забезпечує прозорість роботи служби підтримки.

Розділ Recent Actions призначений для відображення останніх дій адміністратора та інших членів команди. У ньому зосереджена інформація про всі ключові події в системі, зокрема:

- хронологічний журнал подій: створення або видалення користувачів, редагування налаштувань, публікації у блозі, додавання work-блоків;
- функцію що дозволяє відслідковувати всі ключові зміни в системі в режимі реального часу.

Цей розділ є важливим елементом контролю та безпеки, оскільки допомагає зрозуміти, хто і коли виконав певну дію.

Розділ Groups призначений для створення груп адміністраторів із різними рівнями доступу та використовується для розмежування обов’язків між членами команди. У межах цього розділу можуть бути налаштовані такі групи:

- група «Техпідтримка» – адміністратори, які мають доступ до системи тікетів, можуть відповідати на звернення користувачів та вести комунікацію через чат;
- група «Контент-менеджери» – мають доступ до Blog_App і відповідають за публікації новин, статей і матеріалів для користувачів;

- група «Системні адміністратори» – повний доступ до Constance, Robot_App і User_App, відповідають за технічне функціонування сервісу;
- група «Аналітики» – мають доступ до статистики використання work-блоків, відслідковування ефективності та підготовки звітів.

Завдяки такій системі груп адміністратор головного рівня може делегувати завдання окремим працівникам без ризику надати їм надмірні права.

Таким чином, адміністративна панель DoDoBot є багатофункціональним інструментом, що об’єднує керування контентом, користувачами, налаштуваннями та процесами автоматизації. Вона забезпечує зручність адміністрування, контроль над безпекою та гнучкість у розвитку системи.

Мікросервіс SiteCompetitorAnalyzer – це окремий сервіс на Flask, який відповідає за автоматичне порівняння власного сайту користувача із сайтом конкурента. Його основна задача – аналізувати два веб-ресурси за ключовими параметрами (швидкість завантаження, SEO-показники, технічна оптимізація, структурованість контенту, мета-теги тощо) та формувати звіт із рекомендаціями, що саме потрібно покращити.

Особливість мікросервісу полягає в тому, що він працює незалежно від основної системи на Django, і його можна масштабувати чи замінити без зупинки центрального ядра (див. додаток рис. Б. 2.1.).

Як працює мікросервіс:

1. користувач у центральному сервісі створює work-блок типу SiteCompetitorAnalyzer;
2. за допомогою Settings schema користувач вказує: адресу свого сайту, адресу сайту конкурента, інтервал перевірки (наприклад, раз на добу), спосіб доставки результатів (Telegram, email тощо), параметри доступу до Telegram-бота (Bot Token, Chat ID);
3. за заданим інтервалом центральна система через Celery + Redis надсилає запит до мікросервісу;

4. flask-додаток виконує серію перевірок: робить запити до обох сайтів, вимірює час відгуку та швидкість, перевіряє базові SEO-параметри (теги title, description, h1, ключові слова), проводить базовий аналіз структури сторінок;

5. результати передаються у внутрішній модуль AI, який формує статистику «що краще у конкурента» та «де перевага у користувача», генерує текстові рекомендації (наприклад, «додати більше ключових слів у description», «оптимізувати зображення», «зменшити час відповіді сервера») та готовий результат надсилається у Telegram-чат користувача у вигляді повідомлення-звіту з рекомендаціями.

Переваги такого підходу:

- гнучкість: можна підключати необмежену кількість конкурентів для різних проектів;
- масштабованість: мікросервіс можна розгорнути на іншому сервері, навіть на іншій мові програмування;
- розширюваність: у майбутньому можна додати інтеграцію з Google PageSpeed Insights чи Ahrefs API;
- автоматизація: результат не просто зберігається, а одразу надходить користувачу у зручний канал (Telegram);
- ai-аналітика: замість сухої статистики користувач отримує зрозумілі поради.

Мікросервіс SEOAnalyzer є окремим сервісом, реалізованим на базі Flask, який відповідає за комплексний SEO-аналіз веб-сайтів. Його основним завданням є проведення детальної перевірки веб-ресурсів за 28 ключовими SEO-параметрами (технічна оптимізація, структурованість контенту, мета-теги, швидкість завантаження, мобільна адаптивність, доступність тощо) та формувати детальний звіт із конкретними рекомендаціями для покращення позицій у пошукових системах.

Особливість мікросервісу полягає в тому, що він працює незалежно від основної системи на Django, і його можна масштабувати чи замінити без зупинки центрального ядра. Крім того, він підтримує автоматичну відправку звітів у

Telegram, що робить його зручним для регулярного моніторингу SEO-показників.

Мікросервіс працює таким чином: користувач у центральному сервісі створює work-блок типу SEOAnalyzer та за допомогою Settings schema вказує URL сайту для аналізу, параметри доступу до Telegram-бота (Bot Token і Chat ID) для отримання звітів, а також інтервал автоматичних перевірок у разі потреби регулярного моніторингу.

За заданим інтервалом або за запитом центральна система через Celery + Redis надсилає запит до мікросервісу. Flask-додаток виконує комплексний аналіз сайту:

1. завантажує HTML-сторінку та вимірює швидкість відповіді сервера;
2. перевіряє основні SEO-елементи (title, meta description, H1-H6 теги);
3. аналізує технічні аспекти (robots.txt, sitemap.xml, canonical URL, Schema разметку);
4. оцінює мобільну дружелюбність (viewport, responsive зображення, media queries);
5. перевіряє доступність (ALT атрибути, labels форм);
6. аналізує структуру URL та їх дружелюбність;
7. оцінює якість та обсяг контенту;
8. перевіряє безпеку (HTTPS) та швидкість завантаження.

Результати обробляються внутрішньою системою оцінювання:

1. кожен параметр отримує відповідний бал (максимум 100 балів загалом);
2. генеруються конкретні рекомендації для покращення показників;
3. формується детальний звіт українською мовою з візуальними індикаторами;
4. готовий результат надсилається у Telegram-чат користувача у вигляді структурованого повідомлення-звіту з детальним розбором за категоріями та практичними рекомендаціями.

Також результат повертається у JSON-форматі для програмної обробки (див. додаток. В 2.1).

Переваги такого підход:

- комплексність: мікросервіс аналізує 28 різних SEO-параметрів, охоплюючи всі ключові аспекти оптимізації сайту;
- гнучкість: можна налаштувати різні типи аналізу (базовий, повний, технічний) залежно від потреб користувача;
- автоматизація: результат не просто зберігається, а одразу надходить користувачу у зручний канал (Telegram) з детальними рекомендаціями;
- масштабованість: мікросервіс можна розгорнути на окремому сервері та масштабувати незалежно від основної системи;
- локалізація: всі звіти та рекомендації надаються українською мовою для зручності користувачів;
- інтелектуальна оцінка: замість сухої статистики користувач отримує зрозумілі поради та конкретні кроки для покращення SEO;
- візуальна звітність: звіти містять емодзі-індикатори, кольорове кодування та структуровану подачу інформації;
- технічна досконалість: перевірка охоплює як базові елементи (title, description), так і складні технічні аспекти (Schema розмітка, Canonical URL, доступність);
- моніторинг продуктивності: вимірювання швидкості завантаження, розміру файлів та загальної продуктивності сайту;
- мобільна адаптивність: спеціальні перевірки для забезпечення коректного відображення на мобільних пристроях.

Такий детальний звіт дозволяє швидко оцінити стан SEO сайту та отримати конкретні рекомендації для покращення (див. додаток. В 2.2).

2.4. Верифікація ефективності розробленого вебсервісу та аналіз результатів оптимізації

Для підтвердження ефективності розробленого вебсервісу було проведено апробацію його функціоналу на базі вебсайту RoboCactus.

Процес дослідження мав аналітико-дослідницький характер і базувався на порівнянні фактичних показників діяльності сайту до та після використання розробленого інструментарію.

На початковому етапі (серпень 2025 р.) розроблена система зафіксувала низку критичних SEO-проблем, які суттєво погіршували індексацію та видимість сайту в пошукових системах. Зокрема, було виявлено відсутність файлів sitemap.xml та robots.txt, некоректну структуру заголовків H1, дублювання мета-тегів та відсутність атрибутів ALT у зображень
(див. додаток рис. Г. 2.1.). (див. додаток. Г 2.1).

Після того, як за порадами сервісу всі технічні помилки були виправлені, сайт став «чистим» для пошукових систем. Щоб підтвердити успіх, було дані Google Analytics. На графіку видно, що як тільки помилки зникли, відвідуваність сайту різко пішла вгору: якщо в серпні користувачів майже не було, то у вересні їхня кількість зросла до 194 осіб, а в окремі дні на сайт заходило понад 70 людей (див. додаток рис. Г. 2.2.).

Ці цифри та графіки доводять: вебсервіс дійсно допомагає автоматично знаходити проблеми, виправлення яких реально піднімає сайт у рейтингу та приводить нових відвідувачів.

РОЗДІЛ 3. НАПРЯМИ ВПРОВАДЖЕННЯ ТА ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ СИСТЕМИ DODOBOT

3.1. План впровадження системи автоматизації в діяльність підприємства

Для досягнення максимальної ефективності використання системи DoDoBot доцільно реалізувати поетапний план впровадження, який мінімізує ризики та забезпечить стабільний перехід від ручного контролю до автоматизованого управління веб-ресурсами.

Етап 1. Підготовчий (1–2 тижні). На даному етапі проводиться аудит поточних бізнес-процесів підприємства, визначаються ключові точки контролю (доступність сайту, SEO-показники, швидкість завантаження сторінок, коректність контенту). Орієнтовні витрати на підготовчий етап визначаються за формулою (3.1):

$$C = T \cdot R, \tag{3.1}$$

де C - витрати на підготовчий етап, грн;

T - тривалість аналітичних робіт, (10–15 год);

R - середня погодинна ставка спеціаліста, (300–350 грн/год).

За даних умов витрати становлять 3 000–5 000 грн.

Етап 2. Конфігурація процесів. Передбачається створення індивідуальних «роботів» під конкретні задачі бізнесу. Наприклад, робот «SEO-Монітор» для сайту RoboCactus здійснює перевірку мета-тегів, структури заголовків (H1–H6), наявності атрибутів alt у зображеннях та коректності внутрішньої перелінковки. Орієнтовна вартість розробки одного такого модуля в ринкових умовах складає 8 000–12 000 грн, проте у випадку власної системи ці витрати є разовими.

Етап 3. Налаштування тригерів і сценаріїв. На цьому етапі встановлюється періодичність перевірок: щогодинний моніторинг доступності сайту, щоденний аналіз критичних SEO-параметрів та щотижневий повний

SEO-аудит. Такий підхід дозволяє зменшити ймовірність тривалих збоїв майже до нуля та оперативно реагувати на проблеми.

Етап 4. Інтеграція каналів сповіщень. DoDoBot інтегрується з Telegram-ботами та Email-розсилками для миттєвого інформування відповідальних осіб. У випадку виявлення критичної помилки повідомлення надходить протягом 1–2 хвилин, що дозволяє скоротити середній час реакції з 2–3 годин до 10–15 хвилин. Таким чином, разовий аудит перетворюється на безперервний процес контролю якості веб-ресурсу.

3.2. Прогноз результатів та покращення показників видимості сайту

На основі практичних даних, отриманих у другому розділі, можна спрогнозувати конкретні кількісні та якісні результати від постійного використання системи DoDoBot.

Технічна стабільність. Завдяки модулю моніторингу доступності очікується скорочення часу простою (downtime) сайту на 90–95%. Якщо до впровадження система простоювала в середньому 10 годин на місяць, то після автоматизації цей показник зменшиться до 20–30 хвилин.

Ріст видимості в пошукових системах. Постійний контроль SEO-помилок (відсутні meta description, дубльовані title, биті посилання) забезпечує стабільний ріст органічного трафіку. За прогнозом, органічний трафік може зрости з 300 відвідувачів/місяць до 800–1 000 відвідувачів/місяць протягом 3–4 місяців. Очікується вихід сайту в ТОП-10 за основними цільовими запитами.

Якість контенту. Використання інтегрованих GPT-моделей для генерації мета-тегів та описів сторінок дозволяє уникнути дублювання контенту та підвищити релевантність. Середній CTR у пошуковій видачі може зрости з 2,5% до 4–5%, що напряму впливає на кількість переходів.

3.3. Обґрунтування економічного ефекту та практична значиміст

Впровадження системи DoDoBot має чітко виражений економічний ефект, який полягає у зниженні операційних витрат та підвищенні ефективності роботи персоналу.

Заміна платних аналогів. Використання власної системи дозволяє відмовитися від підписок на сторонні сервіси (Ahrefs, Semrush, Screaming Frog). Середня вартість таких сервісів становить близько 1 000–1 200 $ на рік, що еквівалентно 38 000–46 000 грн.

Оптимізація фонду оплати праці. Автоматизація рутинних завдань вивільняє до 20–25 робочих годин SEO-спеціаліста щомісяця. Економія коштів за рахунок автоматизації рутинних операцій визначається за формулою (3.2).

$$E = T \cdot R, \tag{3.2}$$

де E - економія витрат на оплату праці, грн/міс;

T - зекономлений робочий час, 20–25 год;

R - погодинна ставка спеціаліста, 300 грн/год.

За даних умов місячна економія становить 6 000–7 500 грн, що в річному вимірі перевищує 70 000 грн.

Швидкість прийняття управлінських рішень. Завдяки автоматизованим звітам керівництво отримує актуальні дані в режимі реального часу, що зменшує ризик фінансових втрат та дозволяє швидше реагувати на зміну ринкових умов.

3.4. Напрями подальшого вдосконалення та масштабування системи

Система DoDoBot має значний потенціал для подальшого розвитку та масштабування. З цією метою визначено такі напрями подальшого вдосконалення системи:

1. розширення ШІ-модуля генерації контенту. Одним із напрямів розвитку є впровадження повноцінного модуля штучного інтелекту для генерації контенту. Це включає: автоматичне створення SEO-оптимізованих статей; генерацію описів товарів та послуг; створення унікальних meta title та meta description для великих сайтів. Використання ШІ дозволить зменшити витрати на копірайтинг у 5–8 разів. Наприклад, якщо середня вартість написання однієї статті становить 800–1 200 грн, то автоматизована генерація контенту знижує цю суму до 100–150 грн за одиницю контенту;

2. інтеграція з Google Search Console та Google Analytics. Інтеграція з офіційними API Google Search Console та Google Analytics відкриє доступ до реальних пошукових запитів, показників CTR, позицій сторінок та індексації. Це дозволить: отримувати точні дані без використання сторонніх сервісів; автоматично виявляти сторінки з падінням позицій; формувати рекомендації щодо оптимізації на основі реальних даних користувачів. Очікується, що така інтеграція підвищить ефективність SEO-рішень на 20–30%;

3. модуль фінансової аналітики та прогнозування. Подальший розвиток DoDoBot може включати модуль економічної аналітики, який оцінює вплив SEO-показників на дохід бізнесу. Система зможе: прогнозувати зростання прибутку від органічного трафіку, оцінювати рентабельність SEO-інвестицій, формувати фінансові звіти для керівництва.

Інтеграція з CMS та маркетинговими платформами Інтеграція з популярними CMS (WordPress, Laravel, Shopify) та CRM-системами дозволить автоматично застосовувати рекомендації DoDoBot без участі розробників. Це зменшить час впровадження змін з декількох днів до кількох хвилин.

Навіть з урахуванням платної підписки DoDoBot загальні витрати на SEO-моніторинг скорочуються приблизно на 64%. При цьому підприємство отримує безперервний автоматизований контроль, швидку реакцію на помилки та стабільніше зростання SEO-показників (табл. 3.1).

Таблиця 3.1

Економічна ефективність від впровадження DoDoBot

Список витрат	**До впровадження DoDoBot**	**Після впровадження DoDoBot**	**Економія, грн/рік**
Підписка Ahrefs	18 000 грн/рік	0 грн	18 000
Підписка Semrush	20 000 грн/рік	0 грн	20 000
Робочий час SEO-спеціаліста (20 год/міс × 300 грн)	72 000 грн/рік	24 000 грн/рік	48 000
Підписка DoDoBot (1 200 грн/міс)	0 грн	14 400 грн/рік	–14 400
Серверні витрати (хостинг, API)	0 грн	6 000 грн/рік	–6 000
Загальні витрати	128 000 грн/рік	46 400 грн/рік	81 600 грн

Джерело: [6]

ВИСНОВКИ

У ході виконання магістерської роботи було спроєктовано та реалізовано комплексну веб-систему автоматизації DoDoBot, орієнтовану на вирішення актуальних завдань автоматизації бізнес-процесів у сфері моніторингу веб-ресурсів, управління контентом та аналітики.

У результаті дослідження та практичної реалізації досягнуто поставленої мети — створено веб-додаток для автоматизації процесів моніторингу та управління проєктами з використанням сучасних технологій веб-розробки та мікросервісної архітектури.

Під час роботи було вирішено такі основні завдання:

- проведено аналіз існуючих систем автоматизації, визначено архітектуру розроблюваної системи;
- спроєктовано та реалізовано моделі даних користувачів, проєктів, блоків і повідомлень;
- реалізовано інтеграцію з зовнішніми сервісами, зокрема Telegram, Gmail, SMS, WayForPay та OpenAI API;
- впроваджено механізм асинхронного виконання завдань із використанням Celery та Redis, а також забезпечено багаторівневу систему аутентифікації та захисту даних;
- створено адаптивний інтерфейс для взаємодії з користувачами та проведено тестування і аналіз продуктивності системи.

Результати роботи підтверджують висунуту гіпотезу про те, що використання модульної архітектури на базі Django у поєднанні з мікросервісами дозволяє створити масштабовану та ефективну систему автоматизації з мінімальними витратами ресурсів.

Наукова новизна роботи полягає у комплексному поєднанні Django-додатків із незалежними мікросервісами та підтримці динамічного керування налаштуваннями без перезапуску сервера, що забезпечує гнучкість і масштабованість системи.

Практичне значення отриманих результатів полягає у можливості використання розробленої системи малими та середніми підприємствами для автоматизації рутинних процесів моніторингу веб-сайтів, управління контентом та комунікації. Впровадження DoDoBot сприяє підвищенню ефективності роботи компаній і зниженню операційних витрат.

Результати магістерської роботи були апробовані на міжнародній науково-практичній конференції, (див. додаток. Д).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Асинхронне програмування: переваги, недоліки та сфери застосування : Foxminded : вебсайт. URL: https://foxminded.ua/asynkhronne-prohramuvannia/ (дата звернення: 04.01.2026).

2. Дизайн інтерфейсів користувача : навч. посіб. : StudFile. URL: https://studfile.net/preview/9445425/page:52/ (дата звернення: 04.01.2026).

3. Розробка з боку Front-end: що це таке і чим відрізняється від Back-end : Dan-IT : вебсайт. URL: https://dan-it.com.ua/uk/blog/rozrobka-z-boku-front-end-shho-ce-take-i-chim-vidriznjaietsja-vid-back-end/ (дата звернення: 04.01.2026).

4. Мікросервісна архітектура? : PayPro Global : вебсайт. URL: https://payproglobal.com/uk/відповіді/що-таке-мікросервісна-архітектура/ (дата звернення: 05.01.2026).

5. Кушнір Андрій. Збірник матеріалів науково-практичної конференції / Львівський національний університет імені Івана Франка, Факультет фінансів та бізнес-менеджменту. – Львів : ЛНУ ім. І. Франка, 2025. – URL: https://financial.lnu.edu.ua/wp-content/uploads/2025/06/Zbirnyk-konferentsii-15-travnia-2025.pdf (дата звернення: 05.01.2026).

6. Фінансова звітність : https://www.crunchbase.com/ (дата звернення: 05.01.2026).

7. Backend Development Guide : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/blogs/backend-development/ (дата звернення: 05.01.2026).

8. Celery: Distributed Task Queue : documentation. URL: https://docs.celeryq.dev/en/stable/ (дата звернення: 06.01.2026).

9. Create a Telegram Bot using Python : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/python/create-a-telegram-bot-using-python/ (дата звернення: 06.01.2026).

10. Cross-Site Request Forgery (CSRF) : MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/Security/Attacks/CSRF (дата звернення: 07.01.2026).

11. Explain working of HTTPS : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/html/explain-working-of-https/ (дата звернення: 07.01.2026).

12. FastAPI: The Modern Python Framework : Kinsta : website. URL: https://kinsta.com/blog/fastapi/ (дата звернення: 07.01.2026).

13. Figma vs Photoshop: A Comprehensive Comparison : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/blogs/figma-vs-photoshop-comparison/ (дата звернення: 07.01.2026).

14. Graphical Representation of Data : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/maths/graphical-representation-of-data/ (дата звернення: 07.01.2026).

15. Introducing JSON : official website. URL: https://www.json.org/json-en.html (дата звернення: 07.01.2026).

16. MySQL Tutorial : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/mysql/mysql-tutorial/ (дата звернення: 07.01.2026).

17. Navigation Element in Web Design : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/websites-apps/navigation-element-in-web-design/ (дата звернення: 07.01.2026).

18. Performance Metrics for SEO : Moz : website. URL: https://moz.com/learn/seo/performance-metrics (дата звернення: 07.01.2026).

19. PHP vs Python: Which language to choose in 2025 : NT Blog. URL: https://nt.ua/blog/php-vs-python-which-programming-language-to-choose-to-start-your-career-in-2025 (дата звернення: 07.01.2026).

20. pip: The Python Package Installer : Real Python : website. URL: https://realpython.com/what-is-pip/ (дата звернення: 07.01.2026).

21. Python Executive Summary : official website. URL: https://www.python.org/doc/essays/blurb/ (дата звернення: 07.01.2026).

22. Python PIP Guide : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/python/python-pip/ (дата звернення: 07.01.2026).

23. Robots.txt Optimization : Moz : website. URL: https://moz.com/learn/seo/robotstxt (дата звернення: 07.01.2026).

24. Search Engine Optimization (SEO) Basics : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/techtips/search-engine-optimization-seo-basics/ (дата звернення: 07.01.2026).

25. Site Audit Tool : Semrush : website. URL: https://www.semrush.com/siteaudit/ (дата звернення: 07.01.2026).

26. UptimeRobot: Free Website Monitoring Service : official website. URL: https://uptimerobot.com/ (дата звернення: 07.01.2026).

27. VS Code vs PyCharm: Choosing the Best IDE : Geekflare : website. URL: https://geekflare.com/es/vs-code-vs-pycharm/ (дата звернення: 07.01.2026).

28. Visual Studio vs Visual Studio Code : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/blogs/visual-studio-vs-visual-studio-code/ (дата звернення: 07.01.2026).

29. What is a Telegram Bot? : OPC Router : website. URL: https://www.opc-router.com/what-is-a-telegram-bot/ (дата звернення: 07.01.2026).

30. What is Bootstrap? : Coursera : website. URL: https://www.coursera.org/articles/what-is-bootstrap (дата звернення: 07.01.2026).

31. What is Django? : STX Next : website. URL: https://www.stxnext.com/blog/what-is-django (дата звернення: 07.01.2026).

32. What is Figma? : Noble Desktop : website. URL: https://www.nobledesktop.com/learn/figma/what-is-figma (дата звернення: 07.01.2026).

33. What is Flask? : Noble Desktop : website. URL: https://www.nobledesktop.com/blog/what-is-flask (дата звернення: 07.01.2026).

34. What is HTML? : Kinsta : website. URL: https://kinsta.com/blog/what-is-html/ (дата звернення: 11.01.2026).

35. What is HTTP? : GeeksforGeeks : website. URL: https://www.geeksforgeeks.org/html/what-is-http/ (дата звернення: 08.01.2026).

36. What is JavaScript? : MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/What_is_JavaScript (дата звернення: 08.01.2026).

37. What is PostgreSQL? : Kinsta : website. URL: https://kinsta.com/blog/what-is-postgresql/ (дата звернення: 08.01.2026).

38. What is React.js? : Kinsta : website. URL: https://kinsta.com/blog/what-is-react-js/ (дата звернення: 08.01.2026).

39. What is Selenium Python? : Coursera : website. URL: https://www.coursera.org/articles/what-is-selenium-python (дата звернення: 08.01.2026).

40. What is SEO? : Semrush : website. URL: https://www.semrush.com/blog/what-is-seo/ (дата звернення: 08.01.2026).

41. What is Zapier? : CloudSpot Help Center. URL: https://help.cloudspot.io/en/articles/4439091-what-is-zapier (дата звернення: 08.01.2026).

ДОДАТКИ

Додаток А

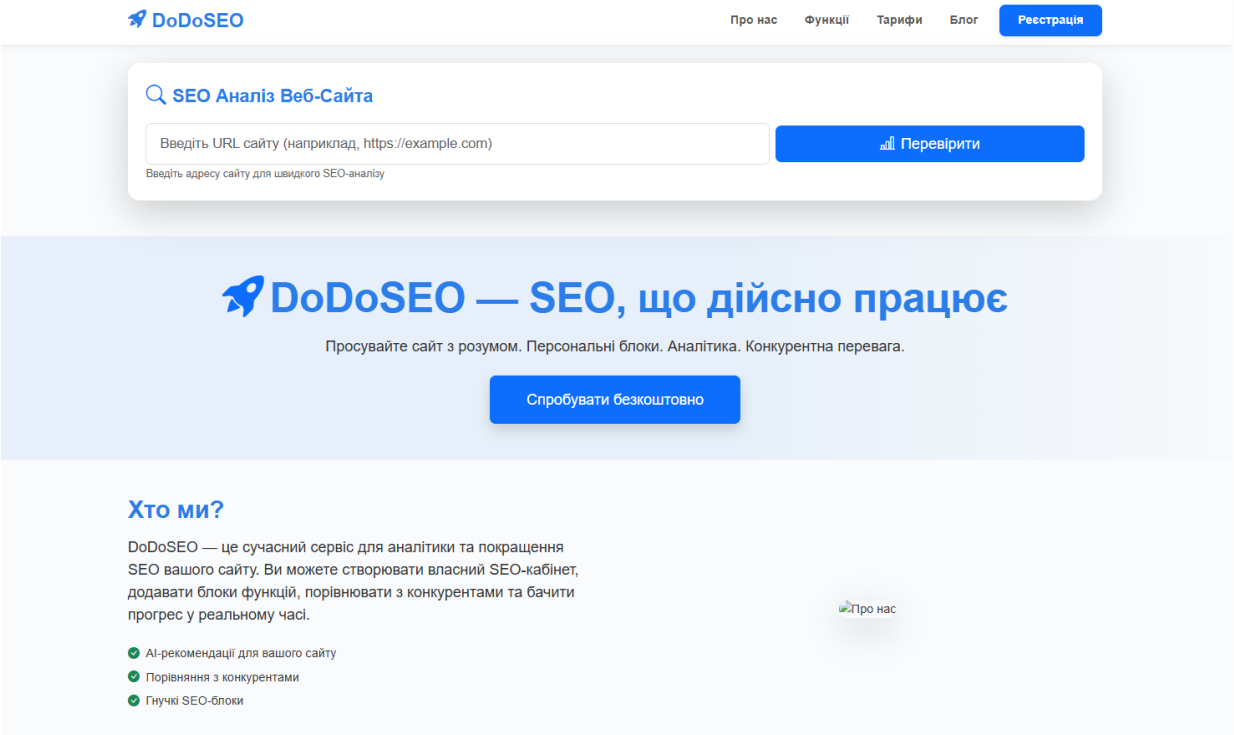


Рис. А.2.1. Головна сторінка сайту

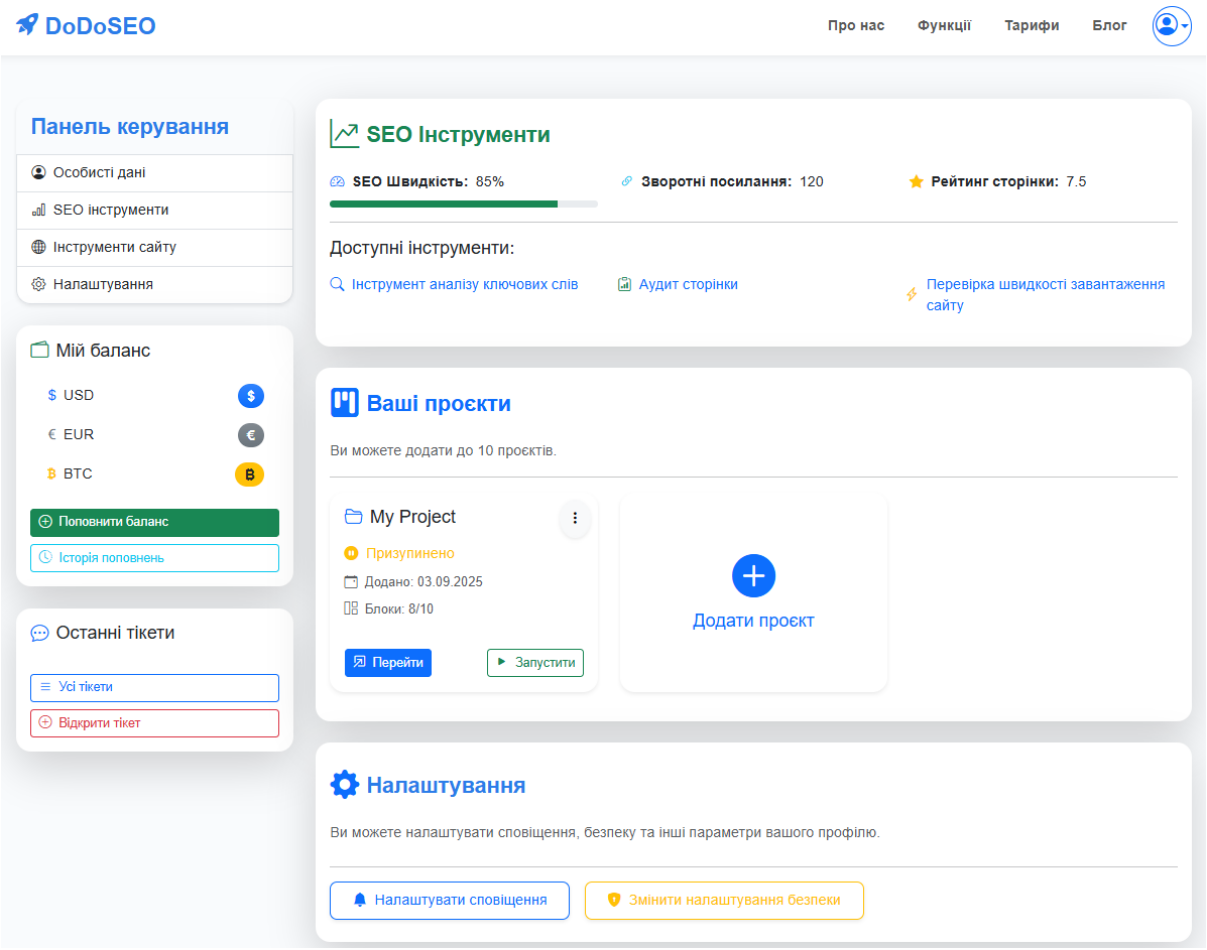


Рис. А.2.2. Панель керування

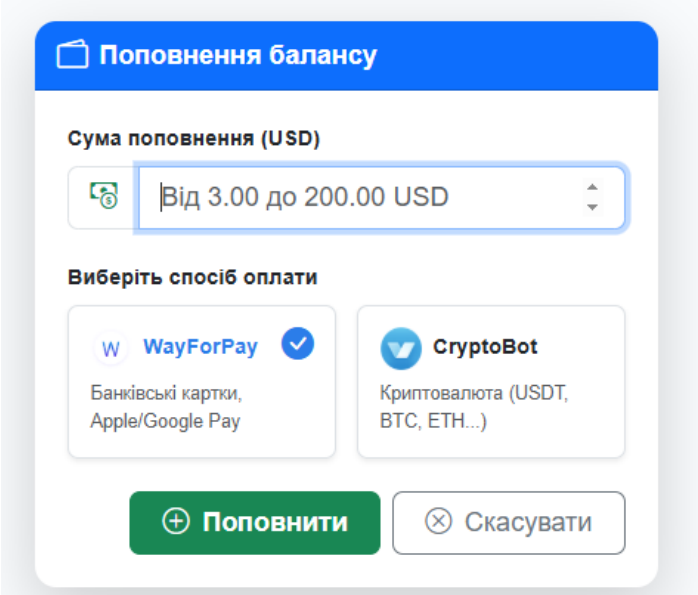


Рис. А.2.4. Поповнення балансу

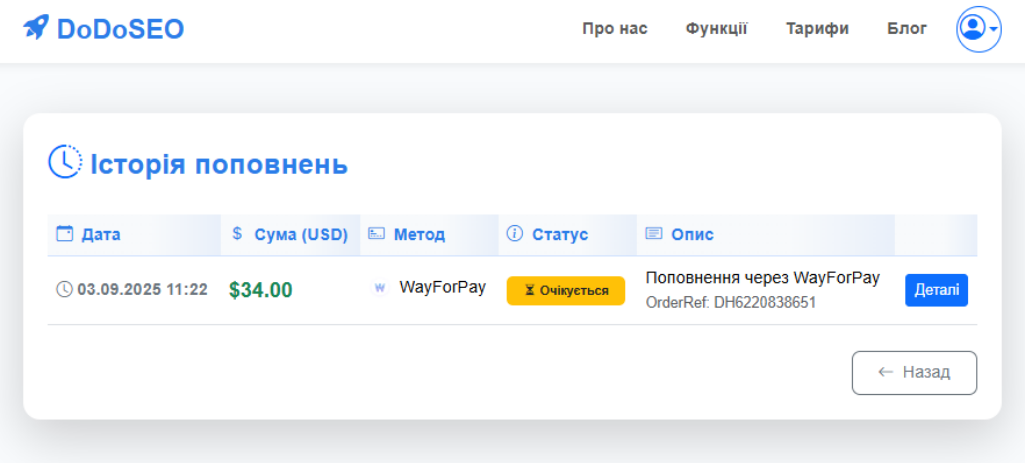


Рис. А.2.5. Історія поповнень

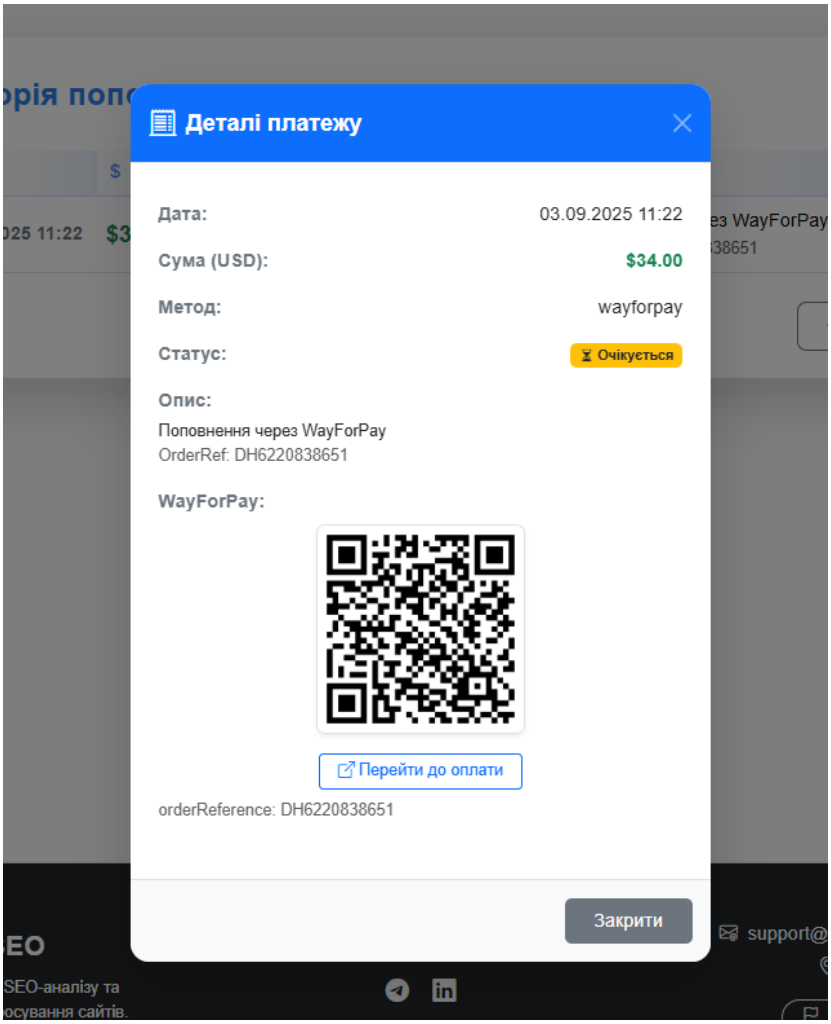


Рис. А.2.6. Деталі платежу

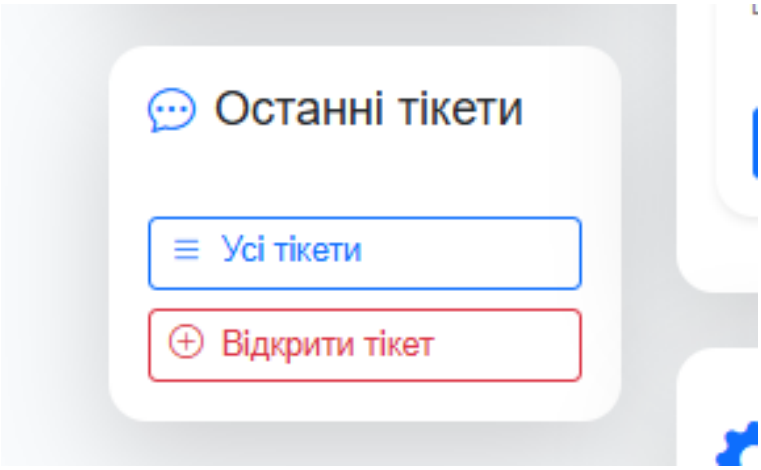


Рис. А.2.7. Останні тікети

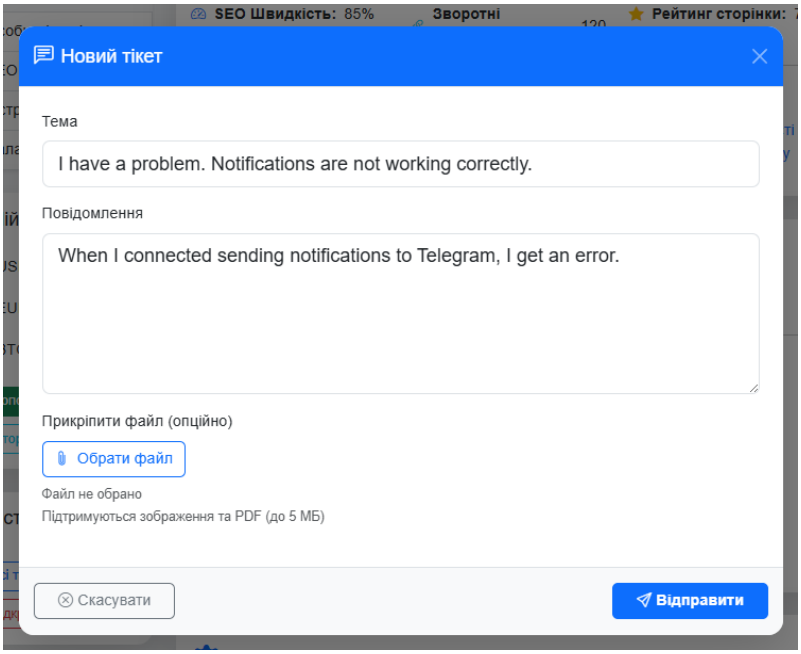


Рис. А.2.8. Створення нового тікету

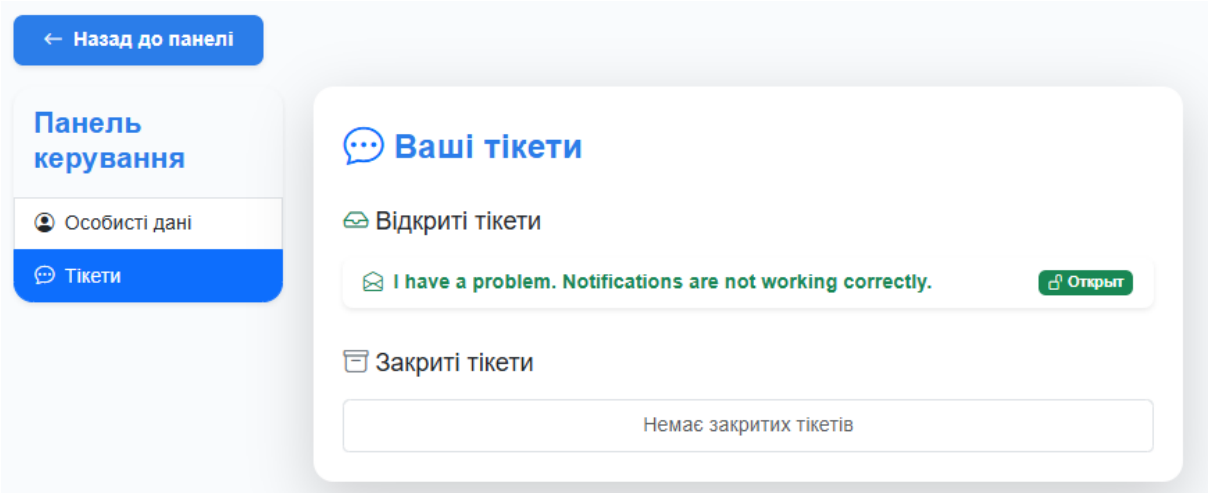


Рис. А.2.9. Перегляд тікетів

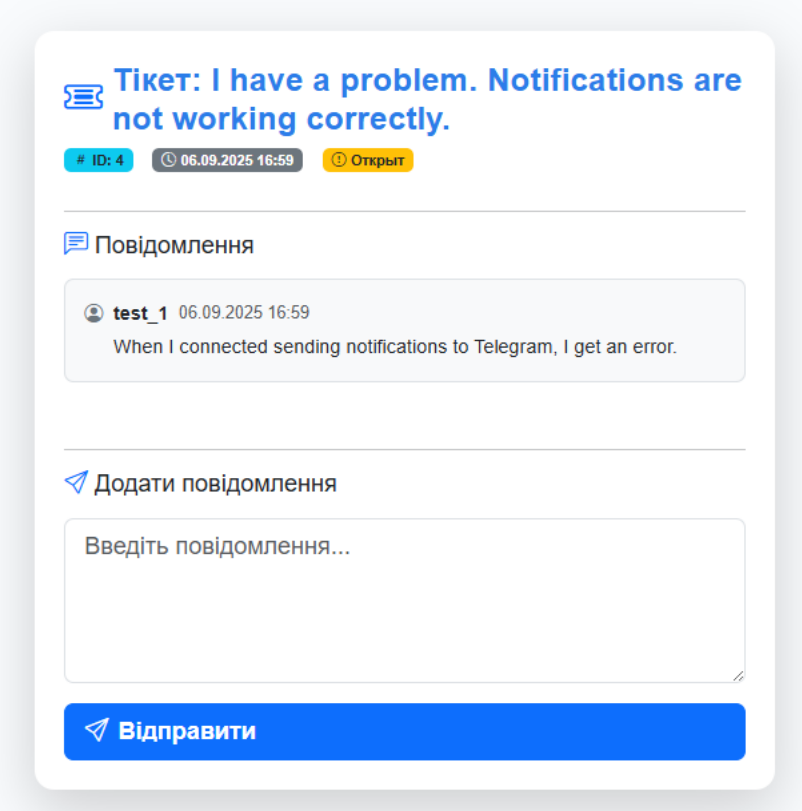


Рис. А.2.10. Відправка повідомлень

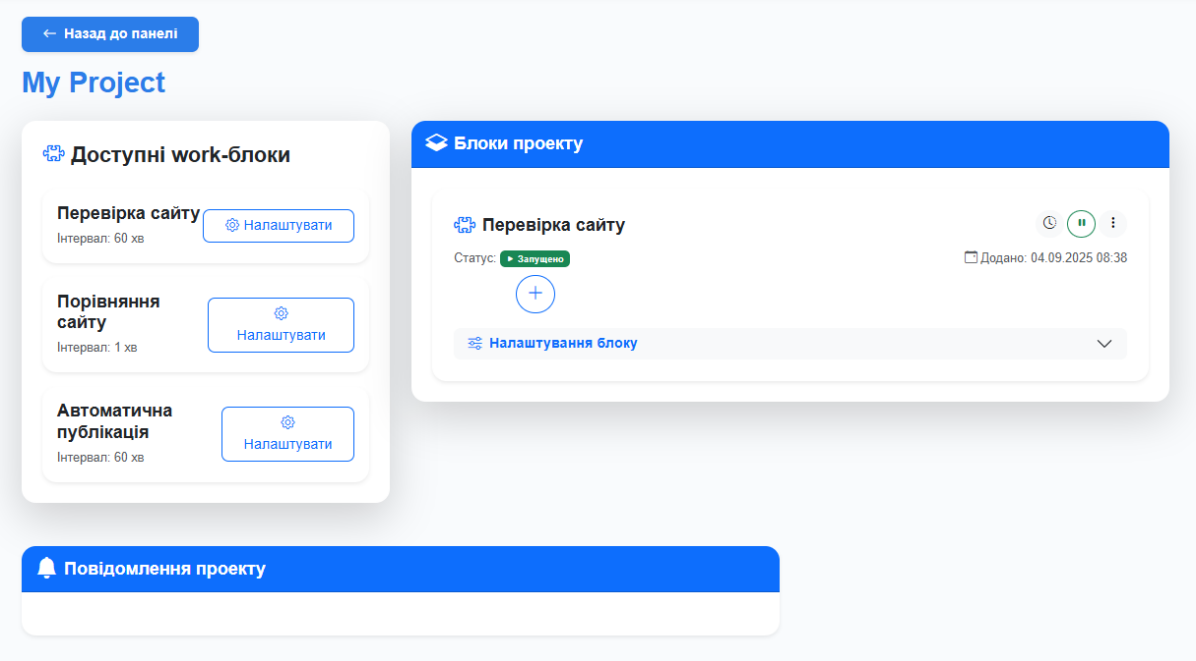


Рис. А.2.15. Робоча область проекту

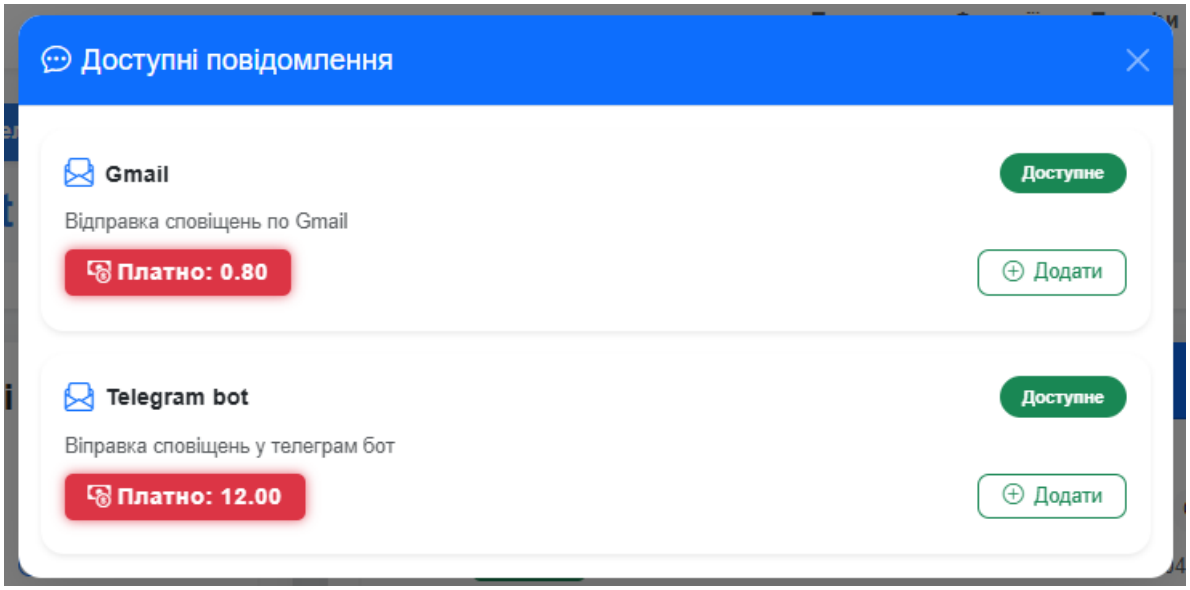


Рис. А.2.19. Доступні повідомлення

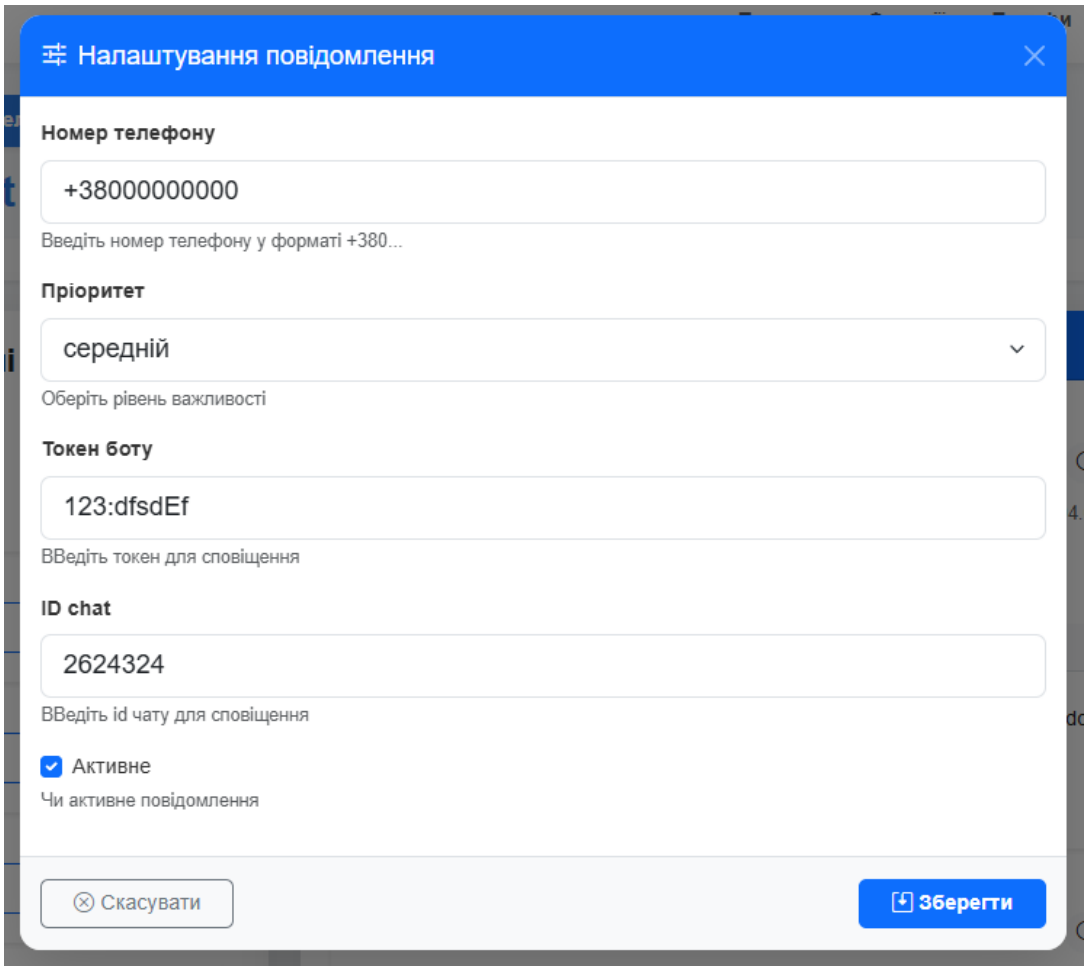


Рис. А.2.20. Налаштування повідомлення

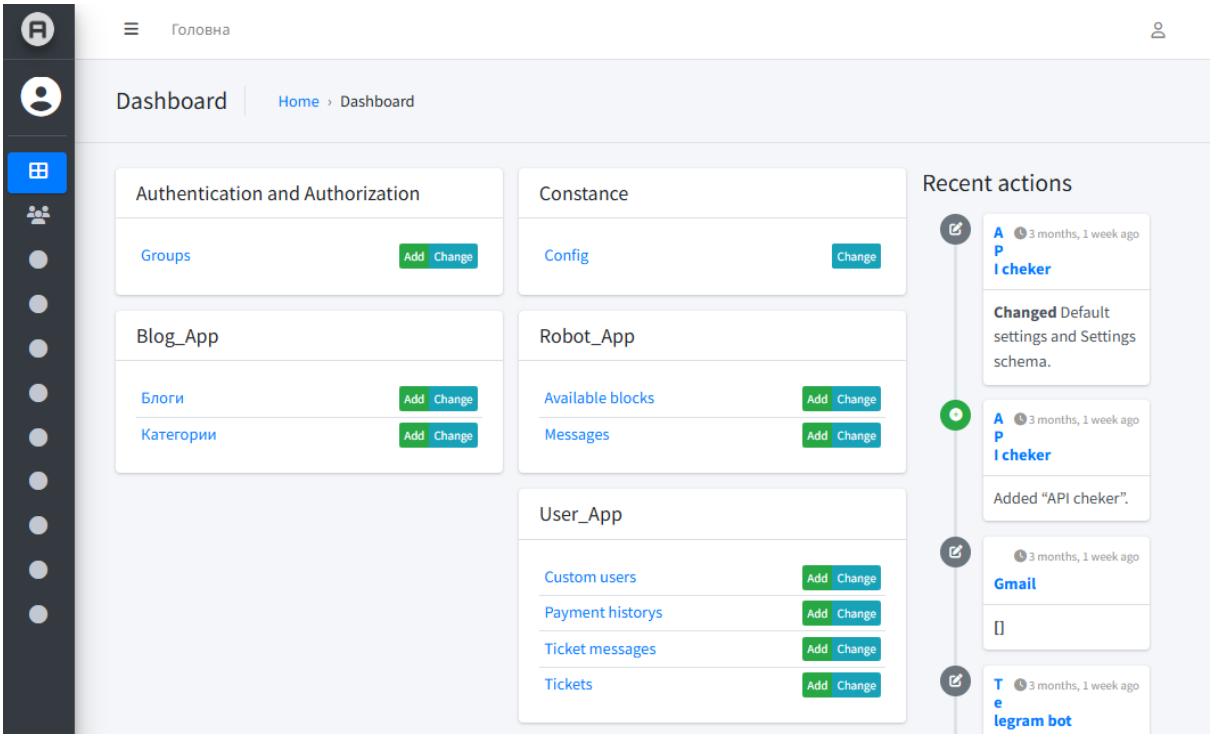


Рис. А.2.24. Панель керування адміністратора

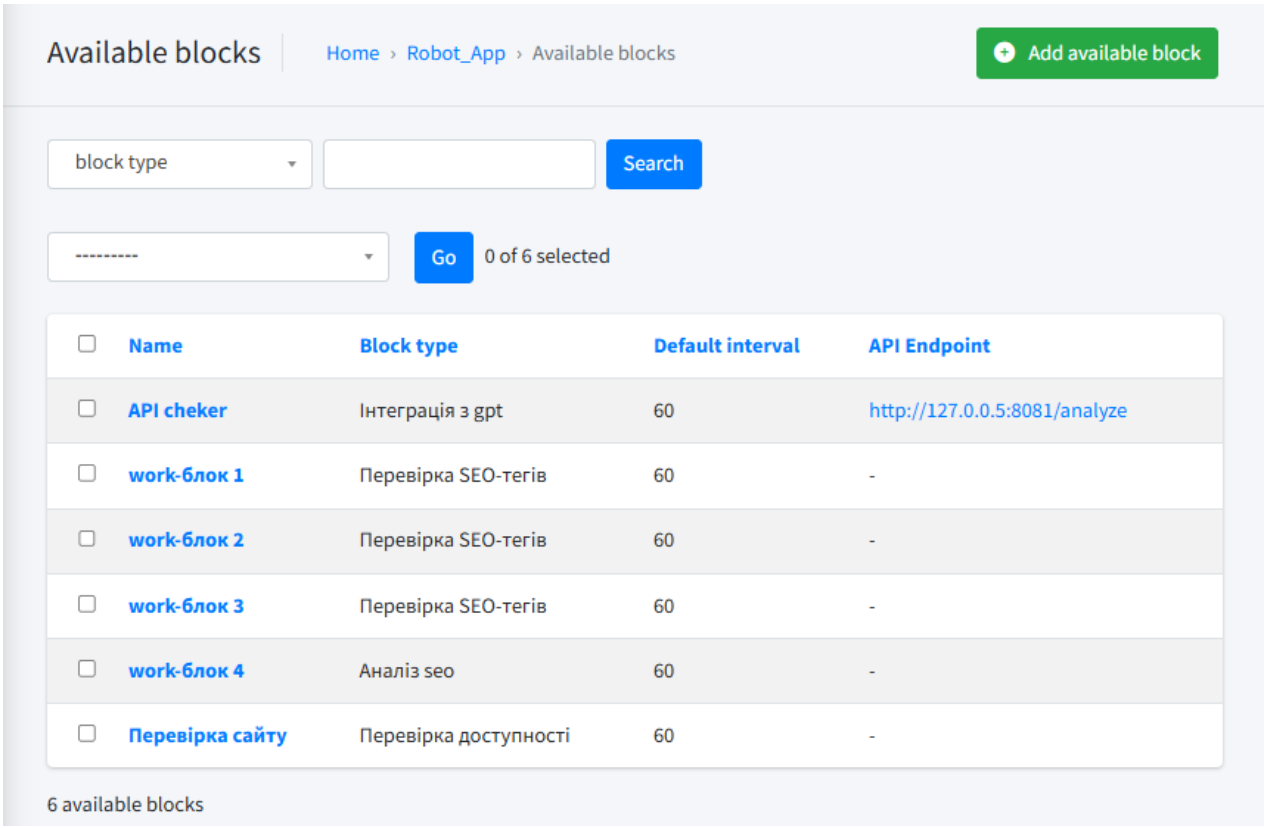


Рис. А.2.25. Авторизація до адмін панелі

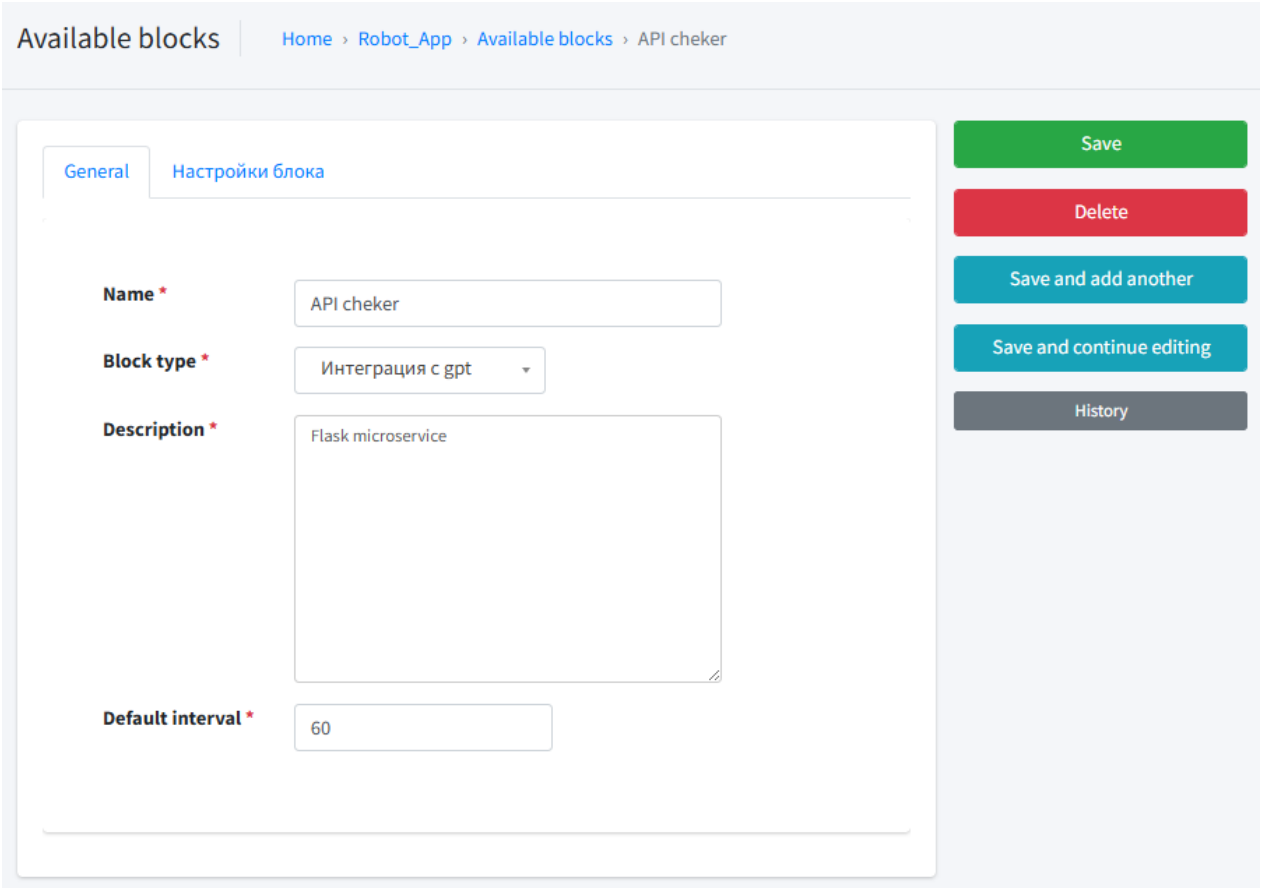


Рис. А.2.26. Налаштування блока

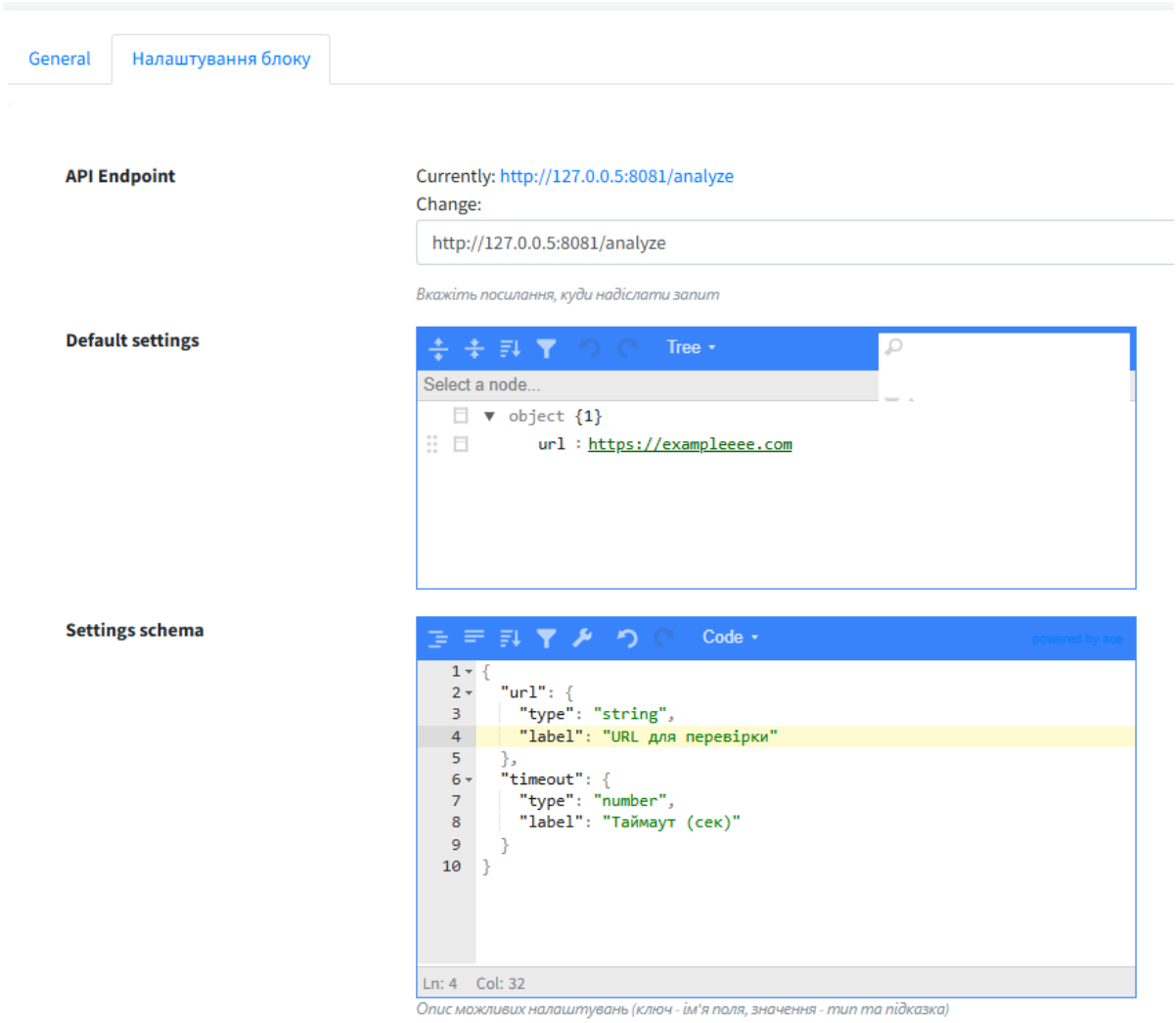


Рис. А.2.27. Налаштування JSON схеми для блока

Додаток Б

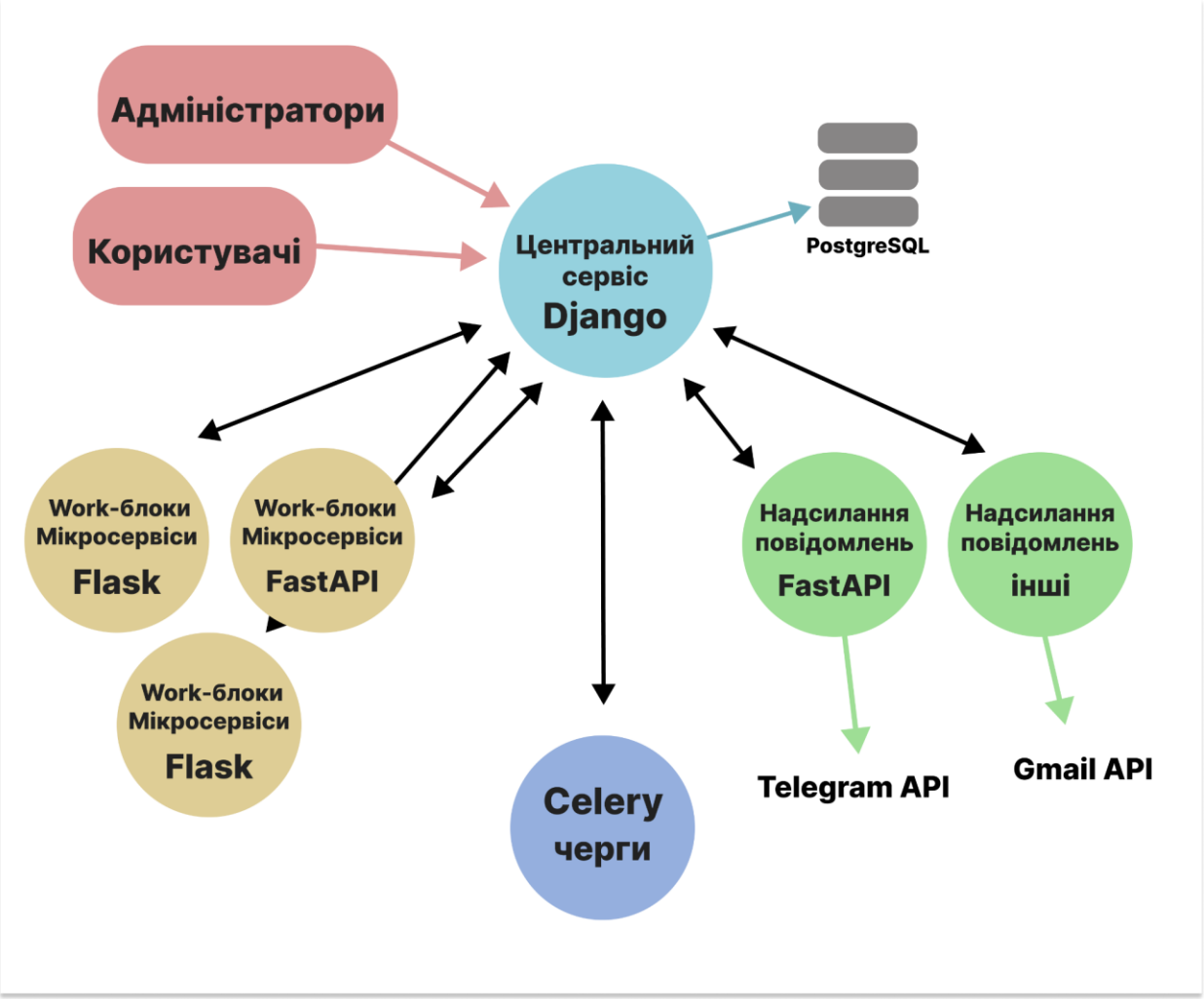


Рис. Б.2.1. Схема мікро сервісної архітектури

Додаток В.1

Settings schema для SEO-аналізу


```
{
  "website_url": {
    "type": "string",
    "title": "Website URL",
    "description": "URL веб-сайту для SEO-аналізу"
  },
  "analysis_type": {
    "type": "string",
    "enum": ["basic", "full", "technical"],
    "default": "full",
    "title": "Analysis Type",
    "description": "Тип аналізу (basic – базові SEO елементи, full – повний аналіз, technical – фокус на технічних аспектах)"
  },
  "check_interval": {
    "type": "integer",
    "title": "Check Interval (minutes)",
    "default": 1440,
    "description": "Інтервал автоматичних перевірок у хвилинах (за замовчуванням 24 години)"
  },
  "telegram_settings": {
    "type": "object",
    "title": "Telegram Settings",
    "properties": {
      "bot_token": {
        "type": "string",
```

```
  "title": "Bot Token",
      "description": "API-токен Telegram-бота для відправки звітів"
    },
    "chat_id": {
      "type": "string",
      "title": "Chat ID",
      "description": "ID чату або користувача для отримання SEO-звітів"
    },
    "send_immediately": {
      "type": "boolean",
      "default": true,
      "title": "Send Immediately",
      "description": "Відправляти звіт одразу після аналізу"
    }
  }
},
"notification_thresholds": {
  "type": "object",
  "title": "Notification Settings",
  "properties": {
    "min_score_alert": {
      "type": "integer",
      "default": 70,
      "title": "Minimum Score Alert",
      "description": "Мінімальний SEO-бал для сповіщення про проблеми"
    },
    "critical_issues_only": {
      "type": "boolean",
```

«Продовження додатка В.1»


```
    "default": false,
        "title": "Critical Issues Only",
        "description": "Сповіщати тільки про критичні SEO-проблеми"
      }
    }
  }
}
```

Додаток В.2

Приклад роботи мікросервісу: запуск тестового SEO-аналізу


```
{
  "report_sent_to_telegram": null,
  "seo_analysis": {
    "title_tag": {
      "content": "Robocactus - Робототехніка та програмування",
      "score": 15,
      "status": true
    },
    "meta_description": {
      "content": "Robocactus — школа робототехніки та програмування в Ірпені. Навчання дітей Python, розробці сайтів та робототехніці. Інтерактивні курси для майбутніх IT-спеціалістів.",
      "score": 15,
      "status": true
    },
    "h1_tags": {
      "count": 1,
      "score": 15,
      "status": "ІДЕАЛЬНО (1 H1 тег)"
    },
    "images_with_alt": {
      "percentage": 75,
      "score": 4,
      "total": 8,
      "with_alt": 6
    },
    "meta_tags": {
      "charset": true,
```

```
  "open_graph": {

   "description": true,
    "image": true,
    "title": true
  },
  "twitter_card": {
    "card": false,
    "title": false
  },
  "viewport": true
},
"meta_tags_score": 9,
"heading_structure": {
  "headings": [
    {
      "count": 1,
      "level": 1,
      "tags": ["Вивчення програмування та робототехніки стало дост"]
    },
    {
      "count": 8,
      "level": 2,
      "tags": ["Основні напрямки", "Розробка сайтів", "Робототехніка"]
    },
    {
      "count": 8,
      "level": 3,
      "tags": ["🔥 Відкрито набір у групу!", "Python Start", "Web Beginner"]
```

```
    }

],
  "issues": [],
  "proper_structure": true
},
"heading_structure_score": 8,
"page_speed": {
  "css_files": {
    "good": false,
    "total": 5
  },
  "external_scripts": {
    "good": true,
    "total": 3
  },
  "html_size": {
    "bytes": 40864,
    "good": true,
    "kb": 39.91
  },
  "images_count": {
    "good": true,
    "total": 8
  }
},
"page_speed_score": 6,
"content_quality": {
  "links": {
```

«Продовження додатка В.2»


```
"external": 6,

      "internal": 12,
      "total": 18
    },
    "readability": {
      "avg_words_per_sentence": 15.1,
      "good": true
    },
    "word_count": {
      "good": true,
      "total": 452
    }
  },
  "content_quality_score": 8,
  "technical_seo": {
    "canonical": {
      "exists": false,
      "url": null
    },
    "hreflang": {
      "count": 0,
      "exists": false
    },
    "meta_robots": {
      "content": "index, follow",
      "exists": true
    },
    "schema_markup": {
```

"count": 1,
"exists": true
}
},
"technical_seo_score": 5,
"security_performance": {
"https": true,
"response_time": {
"good": true,
"seconds": 0.33
}
},
"security_performance_score": 5,
"mobile_friendly": {
"media_queries": false,
"responsive_images": {
"total": 8
},
"viewport_meta": true
},
"mobile_friendly_score": 2,
"accessibility": {
"color_styles": 25,
"form_labels": {
"inputs": 11,
"labels": 7
},
"images_alt": {
"percentage": 75,

```
        "total": 8,
        "with_alt": 6
      }
    },
    "accessibility_score": 1,
    "robots_txt": true,
    "robots_txt_score": 8,
    "sitemap_xml": true,
    "sitemap_xml_score": 8,
    "friendly_urls": {
      "score": 6,
      "status": true,
      "unfriendly_links": []
    },
    "total_score": 88
  },
  "total_score": 88,
  "url": "https://robocactus.com.ua"
}
```

Додаток В.3

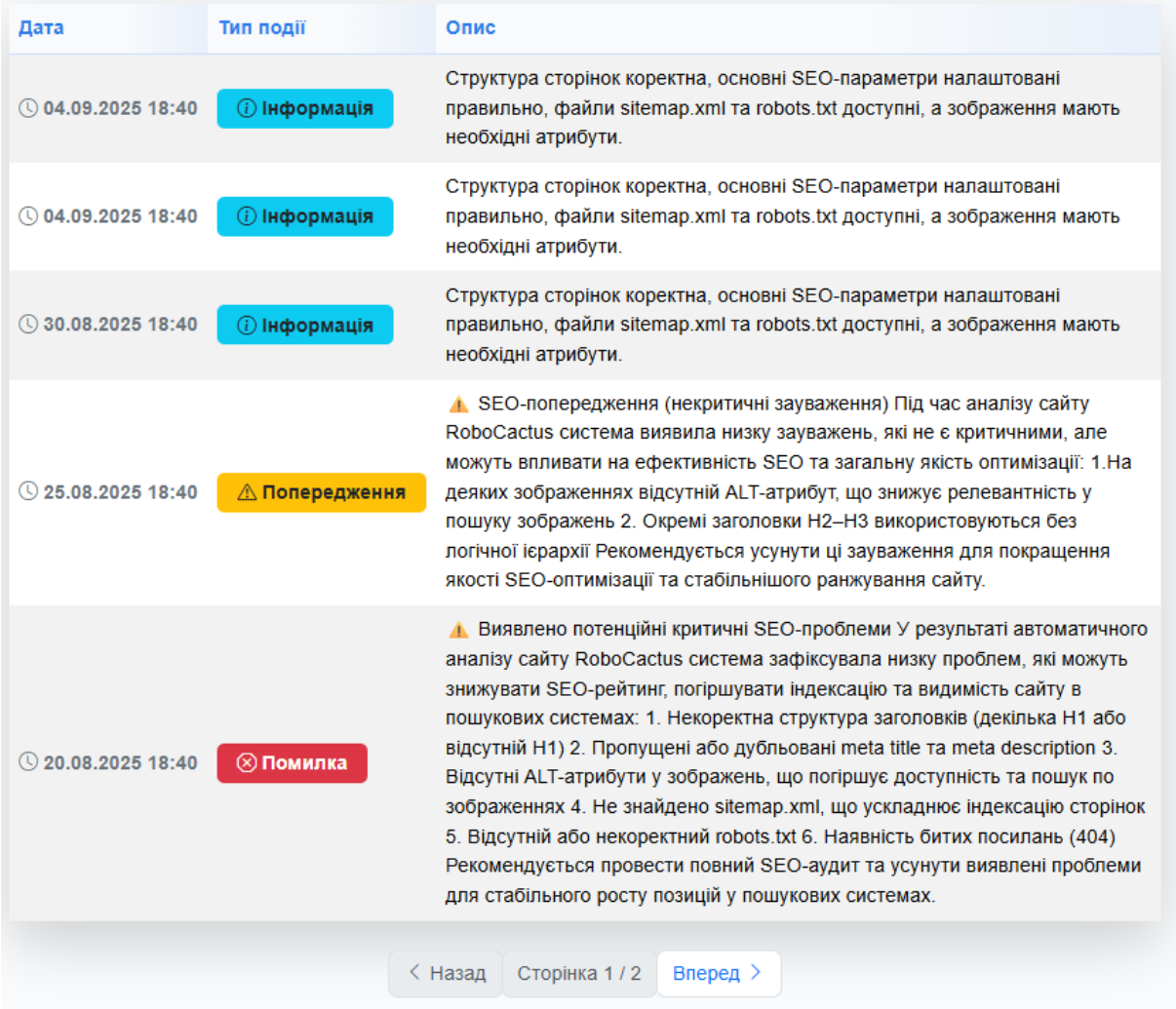
Settings schema для порівняльного аналізу сайтів

```
{
 "user_site_url": {
  "type": "string",
  "title": "Your Website URL",
  "description": "Адреса сайту, який потрібно перевіряти"
 },
 "competitor_site_url": {
  "type": "string",
  "title": "Competitor Website URL",
  "description": "Адреса сайту конкурента для порівняння"
 },
 "check_interval": {
  "type": "integer",
  "title": "Check Interval (minutes)",
  "default": 1440,
  "description": "Як часто виконувати перевірку (у хвилинах)"
 },
 "telegram_settings": {
  "type": "object",
  "title": "Telegram Settings",
  "properties": {
   "bot_token": {
    "type": "string",
    "title": "Bot Token",
    "description": "API-токен для доступу до Telegram-бота"
   },
   "chat_id": {
    "type": "string",
```

"title": "Chat ID",

"description": "Чат або користувач, куди надсилати результати"
}
}
},
"analysis_depth": {
"type": "string",
"enum": ["basic", "extended"],
"default": "basic",
"title": "Analysis Depth",
"description": "Рівень аналізу (basic – базовий, extended – з додатковими SEO-показниками)"
}
}

Додаток Г



Дата	Тип події	Опис
04.09.2025 18:40	Інформація	Структура сторінок коректна, основні SEO-параметри налаштовані правильно, файли sitemap.xml та robots.txt доступні, а зображення мають необхідні атрибути.
04.09.2025 18:40	Інформація	Структура сторінок коректна, основні SEO-параметри налаштовані правильно, файли sitemap.xml та robots.txt доступні, а зображення мають необхідні атрибути.
30.08.2025 18:40	Інформація	Структура сторінок коректна, основні SEO-параметри налаштовані правильно, файли sitemap.xml та robots.txt доступні, а зображення мають необхідні атрибути.
25.08.2025 18:40	Попередження	⚠ SEO-попередження (некритичні зауваження) Під час аналізу сайту RoboCactus система виявила низку зауважень, які не є критичними, але можуть впливати на ефективність SEO та загальну якість оптимізації: 1.На деяких зображеннях відсутній ALT-атрибут, що знижує релевантність у пошуку зображень 2. Окремі заголовки H2–H3 використовуються без логічної ієрархії Рекомендується усунути ці зауваження для покращення якості SEO-оптимізації та стабільнішого ранжування сайту.
20.08.2025 18:40	Помилка	⚠ Виявлено потенційні критичні SEO-проблеми У результаті автоматичного аналізу сайту RoboCactus система зафіксувала низку проблем, які можуть знижувати SEO-рейтинг, погіршувати індексацію та видимість сайту в пошукових системах: 1. Некоректна структура заголовків (декілька H1 або відсутній H1) 2. Пропущені або дубльовані meta title та meta description 3. Відсутні ALT-атрибути у зображень, що погіршує доступність та пошук по зображеннях 4. Не знайдено sitemap.xml, що ускладнює індексацію сторінок 5. Відсутній або некоректний robots.txt 6. Наявність битих посилань (404) Рекомендується провести повний SEO-аудит та усунути виявлені проблеми для стабільного росту позицій у пошукових системах.

< Назад | Сторінка 1 / 2 | Вперед >

Рис. Г. 2.1 Результати автоматизованого аналізу технічного стану сайту RoboCactus до та в процесі оптимізації.

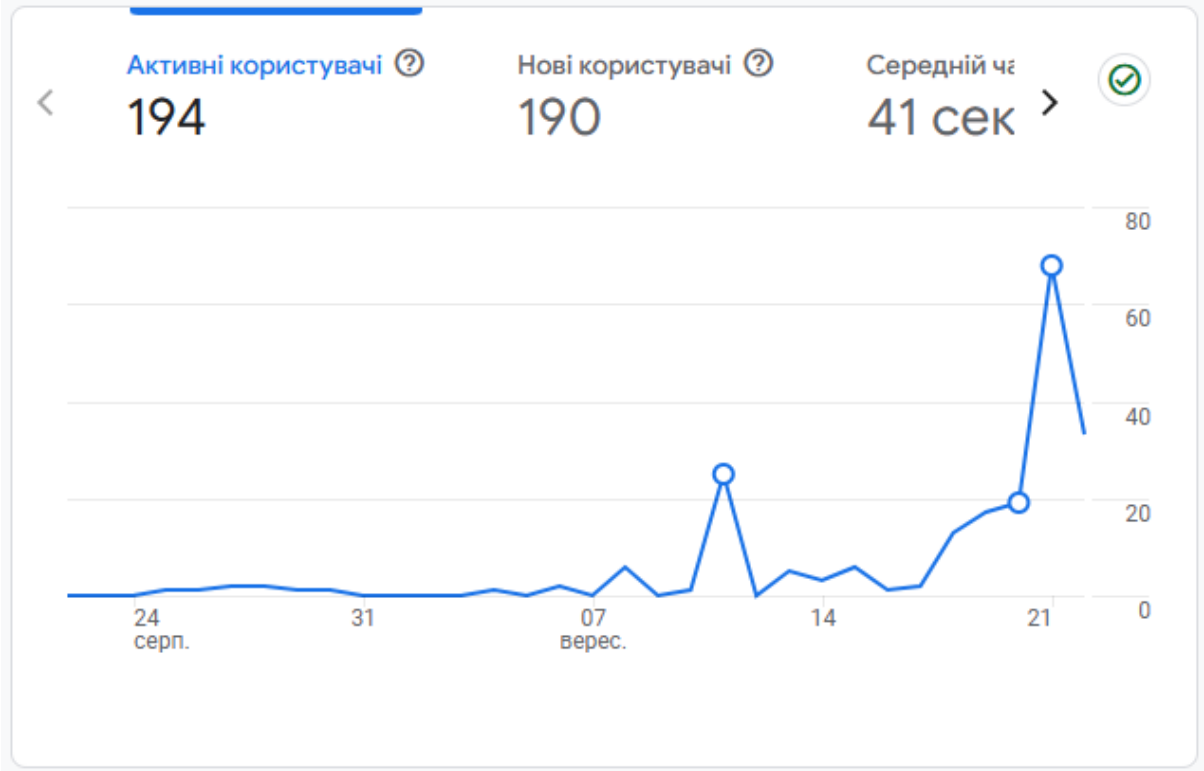


Рис. Г. 2.2. Ріст відвідуваності сайту після виправлення помилок.

Додаток Д

Сертифікат учасника конференції



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ

Сертифікат

Андрій КУШНІР

взяв(ла) участь у роботі

І Міжнародної науково-практичної конференції здобувачів вищої освіти та молодих вчених
«ЕКОНОМІКА І БІЗНЕС: НОВІ РЕАЛІЇ, ТЕХНОЛОГІЧНІ ТРЕНДИ, ЦИФРОВІ ІННОВАЦІЇ»
ECONOMY AND BUSINESS: NEW REALITIES, TECHNOLOGICAL TRENDS, DIGITAL INNOVATIONS

Андрій ГУКАЛЮК
Перший проректор

15 травня 2025 р.

Олександра СМІРНОВА
Проректор з наукової роботи