

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА СЕРВІСУ ДЛЯ ПОШУКУ ОПТИМАЛЬНИХ ТОЧОК
ОБМІНУ ВАЛЮТ З ВИКОРИСТАННЯМ LARAVEL, REACT, MYSQL,
CLICKHOUSE, REDIS TA ELASTICSEARCH**

**DEVELOPMENT OF A SERVICE FOR FINDING OPTIMAL CURRENCY
EXCHANGE POINTS USING LARAVEL, REACT, MYSQL, CLICKHOUSE,
REDIS AND ELASTICSEARCH**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗ-43
Олійник Андрій Сергійович

Керівник:
Ящук Андрій Анатолійович

Кваліфікаційну роботу
допущено до захисту
«__» _____ 20__ р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

«__» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Олійнику Андрію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка сервісу для пошуку оптимальних точок обміну валют з використанням Laravel, React, MySQL, Clickhouse, Redis та Elasticsearch»

Керівник роботи: к.т.н., доцент Ящук А. А.

затверджені наказом закладу вищої освіти від «31» грудня 2025 р. № 541/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «29» травня 2026 р.

3. Вихідні дані до роботи: предметна область моніторингу валютних курсів та пошуку оптимальних точок обміну валют, технології розробки вебзастосунків (Laravel, React), системи керування базами даних та аналітичні сховища даних (MySQL, ClickHouse), засоби кешування та організації черг повідомлень (Redis), пошукові системи (Elasticsearch), вимоги до веборієнтованих інформаційних систем та забезпечення їх функціонування.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): аналіз сучасного стану сервісів моніторингу валютних курсів та пошуку точок обміну валют, постановка завдання, формування вимог до розроблюваної інформаційної системи, обґрунтування вибору технологій та засобів розробки, практична реалізація сервісу пошуку оптимальних точок обміну валют, розробка структури бази даних, оптимізація інформаційної системи для пошукових систем, розробка документації програмного продукту, тестування та налагодження програмного забезпечення.

5. Перелік графічного матеріалу: 19 рисунків, 12 лістингів.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Ящук А. А.		
Специфікація вимог до розробленої системи	Ящук А. А.		
Розробка об'єкта проектування	Ящук А. А.		
Нормоконтроль	Суринович О. М.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту	_____ %		
Академічна доброчесність	Ящук А. А.		

7. Дата видачі завдання «3» лютого 2026 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 14.02.2026 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 03.03.2026 р.	
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 14.03.2026 р.	
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 14.04.2026 р.	
5	Практична реалізація об'єкта проектування та розробка бази даних	до 20.04.2026 р.	
6	Тестування та налагодження об'єкта проектування	до 04.05.2026 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 29.05.2026 р.	

Здобувач вищої освіти _____

Андрій ОЛІЙНИК

Керівник кваліфікаційної роботи _____

Андрій ЯЩУК

АНОТАЦІЯ

Олійник А. С. Розробка сервісу для пошуку оптимальних точок обміну валют з використанням Laravel, React, MySQL, ClickHouse, Redis та Elasticsearch. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків та списку використаних джерел.

У першому розділі проведено дослідження предметної області, пов'язаної з моніторингом валютних курсів та пошуком оптимальних точок обміну валют, проаналізовано існуючі програмні рішення та визначено їх переваги і недоліки, а також обґрунтовано доцільність створення власного сервісу. У другому розділі сформовано вимоги до програмного продукту, здійснено проектування інформаційної системи із використанням UML-нотації та обґрунтовано вибір технологій, методів і засобів розробки. У третьому розділі описано процес практичної реалізації сервісу, розроблено структуру бази даних, розглянуто питання пошукової оптимізації, підготовки документації, а також наведено результати тестування та налагодження програмного забезпечення. У висновках підсумовано результати виконаної роботи, підтверджено досягнення поставленої мети та повне виконання визначених завдань.

Ключові слова: вебсервіс, валютний курс, обмін валют, інформаційна система, Laravel, React, MySQL, ClickHouse, Redis, Elasticsearch..

ABSTRACT

Oliinyk A. S. Development of a Service for Finding Optimal Currency Exchange Points Using Laravel, React, MySQL, ClickHouse, Redis and Elasticsearch. Manuscript. Bachelor's Qualification Thesis of the Educational Program “Software Engineering” in the specialty “Software Engineering”. Lutsk National Technical University. Lutsk, 2026.

The bachelor's qualification thesis consists of an introduction, three chapters, conclusions, and a list of references.

The first chapter investigates the subject area related to exchange rate monitoring and finding optimal currency exchange points, analyzes existing software solutions and identifies their advantages and disadvantages, as well as substantiates the feasibility of developing a dedicated service. The second chapter formulates the requirements for the software product, presents the design of the information system using UML notation, and justifies the choice of technologies, methods, and development tools. The third chapter describes the process of practical implementation of the service, develops the database structure, considers search engine optimization issues and documentation preparation, and presents the results of software testing and debugging. The conclusions summarize the results of the work, confirm the achievement of the stated goal, and demonstrate the complete fulfillment of the defined tasks.

Keywords: web service, exchange rate, currency exchange, information system, Laravel, React, MySQL, ClickHouse, Redis, Elasticsearch.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ СЕРВІСІВ ДЛЯ ПОШУКУ ОПТИМАЛЬНИХ ТОЧОК ОБМІНУ ВАЛЮТ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	13
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	14
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	14
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	19
РОЗДІЛ 3 РОЗРОБКА СЕРВІСУ ДЛЯ ПОШУКУ ОПТИМАЛЬНИХ ТОЧОК ОБМІНУ ВАЛЮТ З ВИКОРИСТАННЯМ LARAVEL, REACT, MYSQL, CLICKHOUSE, REDIS ТА ELASTICSEARCH	27
3.1 Практична реалізація об'єкта проектування	27
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	39
3.3 Розробка бази даних	44
3.4 SEO інформаційно-комп'ютерної системи	48
3.5 Розробка документації для програмного продукту	52
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56

ВСТУП

Актуальність теми. У сучасних умовах розвитку цифрової економіки та фінансових ринків інформація про валютні курси є важливим чинником під час прийняття фінансових рішень як для фізичних осіб, так і для суб'єктів господарювання. Постійні зміни валютного ринку, наявність великої кількості банківських установ і пунктів обміну валют, а також відмінності між курсами купівлі та продажу валют створюють потребу у швидкому доступі до актуальної та достовірної інформації. Використання сучасних вебтехнологій дозволяє автоматизувати процес збору та обробки даних про валютні курси, забезпечуючи користувачам можливість оперативного пошуку найбільш вигідних умов для здійснення валютних операцій. У зв'язку з цим розробка сервісу для пошуку оптимальних точок обміну валют є актуальним завданням, спрямованим на підвищення доступності фінансової інформації та ефективності її використання.

Сучасний стан розвитку інформаційних технологій характеризується активним використанням веборієнтованих інформаційних систем, технологій автоматизованого збору даних, аналітичних платформ та засобів обробки великих обсягів інформації. Значна кількість існуючих сервісів надає можливість перегляду валютних курсів і фінансової статистики, проте часто має обмежені можливості щодо аналізу історичних даних, фільтрації інформації та інтеграції із зовнішніми джерелами. Тенденції розвитку таких систем передбачають використання масштабованих клієнт-серверних архітектур, спеціалізованих сховищ даних, пошукових платформ та механізмів автоматизованого оновлення інформації, що створює передумови для розробки сучасних вебсервісів із високою продуктивністю, актуальністю даних і зручністю використання.

Метою кваліфікаційної роботи бакалавра є розробка сервісу для пошуку оптимальних точок обміну валют з використанням Laravel, React, MySQL, ClickHouse, Redis та Elasticsearch, який забезпечуватиме автоматизований збір даних про валютні курси, їх обробку, зберігання та представлення користувачам.

Об'єктом дослідження є процеси збору, обробки та зберігання інформації про валютні курси в інформаційних системах фінансового призначення.

Предметом дослідження є методи, засоби та технології розробки веборієнтованих інформаційних систем для моніторингу валютних курсів і пошуку оптимальних точок здійснення валютно-обмінних операцій.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз предметної області, існуючих сервісів моніторингу валютних курсів та сучасних технологій розробки вебсистем;
- спроектувати архітектуру інформаційної системи на основі Laravel та React;
- розробити структуру зберігання даних із використанням MySQL і ClickHouse;
- реалізувати механізм автоматизованого збору та оновлення інформації про курси валют засобами Laravel та програмних модулів парсингу;
- розробити користувацький вебінтерфейс із використанням React для пошуку та перегляду інформації про валютні курси;
- реалізувати засоби аналізу історичних даних і візуалізації динаміки зміни курсів валют;
- провести тестування та налагодження програмного забезпечення, а також підготувати документацію щодо його використання та розгортання.

Практичне значення роботи полягає у створенні програмного продукту для моніторингу валютного ринку, пошуку оптимальних умов обміну валют та аналізу динаміки зміни курсів. Розроблений сервіс може використовуватися користувачами, фінансовими установами й аналітичними сервісами, діяльність яких пов'язана з обробкою фінансової інформації.

Дана робота виконана в межах досліджень у сфері веброзробки, проєктування інформаційних систем та обробки даних із застосуванням сучасних підходів до побудови клієнт-серверних застосунків, організації сховищ даних і реалізації механізмів пошуку та аналітичної обробки інформації.

РОЗДІЛ 1

АНАЛІЗ СЕРВІСІВ ДЛЯ ПОШУКУ ОПТИМАЛЬНИХ ТОЧОК ОБМІНУ ВАЛЮТ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Розвиток цифрових технологій та стрімке зростання обсягів доступної інформації суттєво вплинули на фінансову сферу, сприяючи появі нових інформаційних сервісів і підходів до обробки фінансових даних. За даними наукових досліджень використання сучасних програмних засобів у фінансовому секторі дозволяє автоматизувати процеси збору, аналізу та обробки інформації, підвищити швидкість отримання даних та покращити якість надання послуг у зв'язку з чим значного поширення набули технології, які дозволяють поєднати традиційні фінансові механізми із сучасними цифровими платформами.

Нині фінансові системи активно використовують значні обсяги даних для підтримки процесів прийняття рішень. Застосування цифрових технологій дозволяє збирати й аналізувати інформацію в режимі реального часу, забезпечуючи користувачів актуальними даними щодо фінансових показників та змін ринку. Розвиток онлайн-платформ та мережевих технологій посприяв переходу від традиційних фінансових моделей до більш гнучких інформаційних систем, орієнтованих на швидке отримання й опрацювання інформації. Проте зростання кількості даних та швидкості їх оновлення створює потребу у використанні спеціалізованих програмних рішень для автоматизованого збору, зберігання й аналізу інформації [1].

Дослідження показують, що одним із найбільш поширених рішень для реалізації фінансових систем стала API-орієнтована архітектура, яка передбачає розділення клієнтської та серверної частин застосунку. Такий підхід забезпечує більш гнучку структуру системи, дозволяє незалежно розробляти окремі компоненти та спрощує їх подальшу підтримку. Крім того, використання окремого серверного API підвищує можливості масштабування системи та спрощує інтеграцію із зовнішніми сервісами та мобільними застосунками.

Сучасні підходи до веброзробки також передбачають використання серверних фреймворків для реалізації бізнес-логіки та спеціалізованих бібліотек для створення інтерактивного користувацького інтерфейсу, що дозволяє забезпечити швидке оновлення інформації без необхідності повного перезавантаження сторінок, підвищити зручність використання системи та покращити взаємодію користувача із сервісом. Також значна увага приділяється безпечному обміну даними, асинхронній передачі інформації та документуванню API, що забезпечує стабільну взаємодію між окремими компонентами системи [2].

Згідно з науковим дослідженням [3], сучасні системи працюють із великими обсягами даних і потребують швидкої обробки інформації, що зумовлює необхідність використання ефективних засобів зберігання та аналізу даних. Для вирішення таких задач застосовуються системи керування базами даних, які забезпечують високу продуктивність отримання й обробки великої кількості інформації. Одним із таких рішень є ClickHouse, який забезпечує швидке виконання аналітичних запитів з підтримкою розподілу даних між декількома вузлами, що дозволяє зменшити навантаження на обчислювальні ресурси та підвищити продуктивність системи. У роботі зазначається, що застосування подібних технологій дозволяє ефективно працювати з історичними даними, аналізувати зміни показників та ефективно виконувати запити агрегації.

Проведений аналіз літературних джерел показав, що сучасні фінансові системи орієнтовані на використання масштабованих вебтехнологій, засобів автоматизованого збору інформації та високопродуктивної обробки даних. Для сервісів моніторингу валютних курсів особливого значення набувають підтримка аналітичної обробки даних та забезпечення швидкого доступу користувачів до актуальної та достовірної інформації.

Враховуючи особливості сучасних фінансових систем та потребу у швидкому доступі до актуальної інформації про валютні курси, доцільним є аналіз існуючих цифрових сервісів у сфері моніторингу валютного ринку. Дослідження подібних рішень дозволяє оцінити підходи до реалізації таких систем, визначити особливості їх функціонування, а також виявити основні

переваги та недоліки наявних сервісів. Крім того, аналіз існуючих платформ дає можливість визначити функціональні можливості, які доцільно врахувати під час розробки власного програмного продукту.

Серед популярних українських сервісів фінансової інформації можна виділити Minfin та Finance.ua, які надають користувачам доступ до актуальних курсів валют, інформації про банки та обмінні пункти. Обидві платформи також містять економічні новини, аналітичні матеріали, дані щодо кредитування, депозитів і страхових послуг, що дозволяє використовувати їх як комплексні джерела фінансової інформації. Наявність великого обсягу актуальних даних забезпечує користувачам можливість оперативно отримувати відомості про стан фінансового ринку та зміни валютних курсів.

Сервіс Minfin забезпечує доступ до історичних даних і статистичних матеріалів, що створює можливість аналізу змін валютних курсів та проведення порівняння показників за певний період часу. Платформа дозволяє отримувати значний обсяг фінансової інформації в межах однієї системи, що зменшує необхідність використання сторонніх ресурсів для пошуку додаткових даних. Крім цього, користувачі можуть відстежувати динаміку фінансових показників та аналізувати тенденції розвитку валютного ринку. Водночас велика кількість додаткових розділів і функціональних можливостей, може ускладнювати швидкий пошук необхідної інформації та знижувати зручність використання сервісу для виконання вузькоспеціалізованих задач.

Finance.ua також об'єднує значну кількість інформаційних та аналітичних сервісів, надаючи користувачам можливість перегляду курсів валют та історії їх змін. Система забезпечує доступ до інформації щодо стану ринку, що дозволяє використовувати платформу для комплексного аналізу фінансових даних. Наявність широкого набору функцій створює можливість вирішення різних фінансових задач у межах одного сервісу. Проте значна кількість інформаційних матеріалів і додаткових сервісів може ускладнювати пошук потрібних даних, особливо у випадках, коли користувачу необхідно виконати конкретне завдання.

На відміну від багатофункціональних фінансових платформ, сервіс kurs.com.ua орієнтований на моніторинг валютного ринку та надання актуальної інформації про курси валют. Ресурс забезпечує відображення змін валютних показників, пошук даних залежно від місця розташування користувача, а також надає інформацію про конкретні пункти обміну валют. Завдяки вузькій спеціалізації сервіс забезпечує швидкий і зручний доступ до інформації про валютний ринок, що дозволяє користувачам оперативно переглядати та порівнювати курси валют у різних регіонах. Регулярне оновлення даних сприяє швидкому реагуванню на зміни валютного ринку, а зручне представлення інформації спрощує пошук даних і підвищує ефективність використання ресурсу.

Крім того, сервіс підтримує відображення статистичних даних та динаміки змін курсів валют за певний проміжок часу, що дає змогу користувачам аналізувати тенденції валютного ринку. Водночас система має певні обмеження, пов'язані з недостатнім рівнем аналітичної обробки інформації та обмеженими можливостями інтеграції з іншими сервісами і програмними рішеннями.

Аналіз наведених програмних рішень демонструє, що сучасні системи забезпечують реалізацію окремих функцій, пов'язаних із моніторингом валютних курсів, переглядом статистичних показників та аналізом фінансової інформації. Водночас більшість існуючих рішень орієнтована або на широкий спектр фінансових послуг, або на виконання окремих задач, пов'язаних із відображенням курсів валют. При цьому не всі системи забезпечують комплексне поєднання автоматизованого збору інформації з різних джерел, фільтрації за містом, аналізу історичних даних та підтримки інтеграції із системами.

У зв'язку з цим виникає необхідність створення нового програмного рішення, яке поєднуватиме можливості моніторингу курсів, пошуку оптимальних точок купівлі, продажу та обміну валют, аналітичної обробки історичних даних та підтримки інтеграцій для взаємодії із зовнішніми сервісами. Реалізація такого підходу дозволить створити більш вузькоспеціалізоване програмне рішення, орієнтоване на потреби користувачів, які зацікавлені у

швидкому отриманні актуальної інформації та виборі найбільш вигідних умов для здійснення операцій обміну валют.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

На основі проведеного аналізу предметної області, сучасних технологій розробки вебсистем та існуючих рішень було визначено основні вимоги до продукту. Для досягнення поставленої мети, яка полягає у розробці сервісу для пошуку оптимальних точок обміну валют з використанням Laravel, React, MySQL, ClickHouse, Redis та Elasticsearch, необхідно вирішити такі завдання:

- провести аналіз предметної області, існуючих сервісів моніторингу валютних курсів та сучасних технологій розробки вебсистем з метою визначення функціональних вимог до програмного продукту;
- спроектувати архітектуру інформаційної системи на основі клієнт-серверного підходу із використанням Laravel для реалізації серверної частини та React для реалізації користувацького інтерфейсу;
- розробити структуру зберігання даних із використанням MySQL для роботи з основними сутностями системи та ClickHouse для зберігання й аналітичної обробки історичних даних про курси валют;
- реалізувати механізм автоматизованого збору та оновлення інформації про курси валют із використанням серверних засобів Laravel та програмних модулів парсингу даних;
- розробити вебінтерфейс користувача за допомогою React, який забезпечуватиме перегляд актуальних курсів валют, пошук обмінних пунктів, фільтрацію даних за містом та перегляд історії змін курсів;
- реалізувати засоби аналізу історичних даних та візуалізації змін курсів валют за допомогою інтерактивних графіків і статистичних показників;
- провести тестування та налагодження розробленої системи, усунути можливі недоліки та підготувати документацію щодо використання і розгортання програмного продукту.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Проектування сучасних інформаційних систем потребує попереднього аналізу предметної області та потреб користувачів, визначення вимог до програмного забезпечення і моделювання основних елементів системи. Під час формування вимог враховуються особливості роботи сервісів моніторингу валютних курсів, необхідність оперативного отримання актуальної інформації, підтримка фільтрації за містом, можливість перегляду історичних даних, а також вимоги до продуктивності, масштабованості, безпеки даних і стабільності роботи системи. Такий підхід дозволяє сформулювати цілісне уявлення про структуру майбутнього програмного продукту, особливості взаємодії його компонентів, інформаційні потоки та процеси обробки даних.

Розроблюваний сервіс повинен буде відповідати функціональним і нефункціональним вимогам, що визначатимуть умови використання системи та її якісні характеристики. Функціональні вимоги охоплюватимуть засоби отримання, обробки, пошуку, фільтрації та відображення інформації про валютні курси, а також механізми взаємодії із зовнішніми сервісами. Нефункціональні вимоги стосуватимуться швидкодії, зручності користування, стабільності роботи, масштабованості та захисту даних.

Для моделювання розроблюваної системи обрано об'єктно-орієнтований підхід, оскільки він дозволяє представити програмне забезпечення у вигляді взаємопов'язаних компонентів, кожен з яких відповідатиме за виконання окремих функцій. Такий підхід забезпечує гнучкість архітектури, спрощує масштабування системи та сприяє незалежній взаємодії між окремими модулями.

З метою формалізованого представлення функціональних можливостей розроблюваного програмного забезпечення та опису взаємодії користувачів із системою доцільно використати діаграму прецедентів. Вона дозволяє

відобразити основні функції системи, визначити учасників взаємодії та встановити взаємозв'язки між окремими сценаріями, які повинні будуть підтримуватися розроблюваним сервісом.

У межах розроблюваного програмного рішення основними учасниками взаємодії виступатимуть користувач та модуль автоматизованого збору даних. Користувач повинен буде взаємодіяти із сервісом для пошуку актуальних курсів валют, перегляду інформації про обмінні пункти та вибору оптимальних умов обміну. Модуль збору даних повинен буде забезпечувати автоматичне отримання інформації із зовнішніх джерел, оновлення курсів валют та накопичення історичних даних. Діаграма прецедентів з основними функціональними можливостями розроблюваної системи наведена на рисунку 2.1.

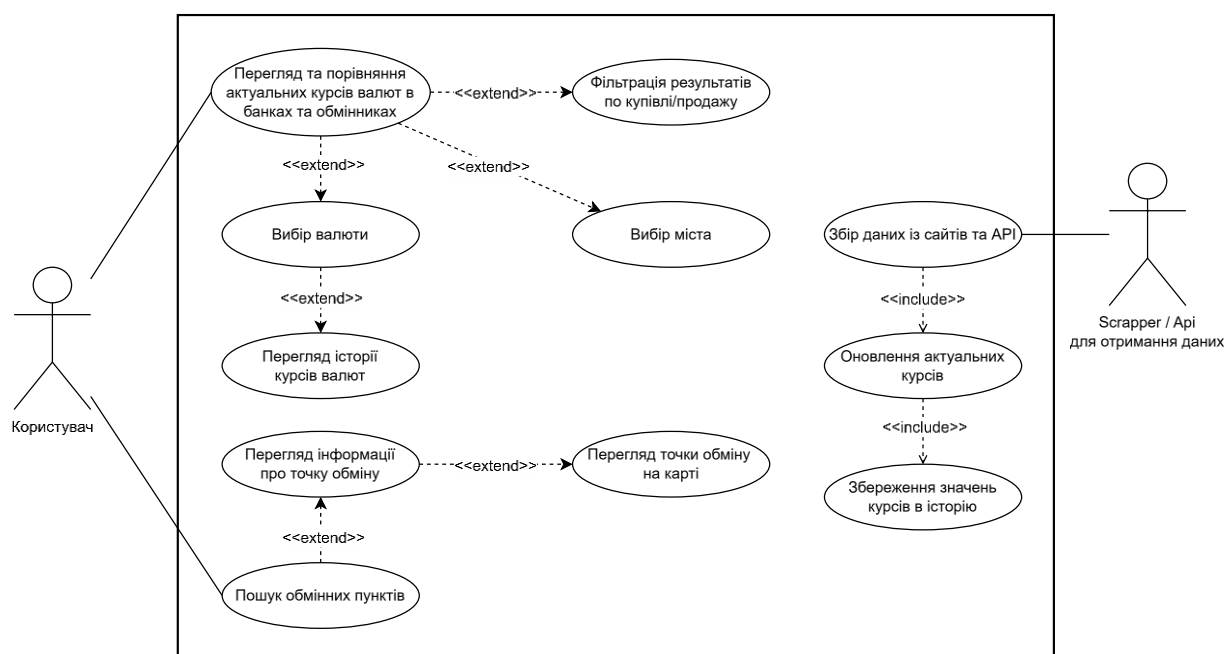


Рисунок 2.1 – Діаграма прецедентів сервісу для пошуку оптимальних точок обміну валют

На діаграмі прецедентів відображено основні сценарії взаємодії користувача та модуля автоматизованого збору даних із сервісом для пошуку оптимальних точок здійснення валютних операцій. Центральним елементом діаграми є система, у межах якої визначено набір функціональних можливостей, реалізація яких передбачається під час розробки сервісу.

Користувач повинен буде мати можливість переглядати та порівнювати актуальні курси валют у банках та обмінниках. Для уточнення результатів пошуку система повинна буде підтримувати вибір валюти, вибір міста та фільтрацію результатів залежно від типу операції купівлі або продажу валют. Також користувач повинен мати доступ до перегляду історії змін курсів валют, що дозволить відстежувати динаміку валютного ринку за певний період часу.

Окремою функціональною можливістю системи стане пошук обмінних пунктів із можливістю перегляду детальної інформації про конкретну точку обміну. Після отримання результатів пошуку користувач повинен мати можливість переглядати розташування обмінних пунктів на карті та порівнювати доступні варіанти залежно від курсу валют і місця розташування.

Модуль автоматизованого збору даних повинен буде виконувати отримання інформації із зовнішніх сайтів та API банків і обмінників. Отримані дані використовуватимуться для оновлення актуальних курсів валют у системі, а також для збереження історичних значень курсів з метою подальшого аналізу та відображення статистичної інформації користувачам.

Для детального аналізу логіки взаємодії користувача із розробленим сервісом варто побудувати діаграму діяльності. Вона дозволяє відобразити послідовність дій, які повинні будуть виконуватися під час роботи із сервісом, а також розмежувати операції користувача та процеси, що виконуватимуться системою. На відміну від діаграми прецедентів, діаграма діяльності дає можливість детально описати порядок виконання окремих операцій та переходів між ними, а її використання сприяє кращому розумінню логіки функціонування сервісу та послідовності обробки користувацьких запитів.

У межах розроблюваного сервісу діаграма діяльності, наведена на рисунку 2.2, відображає сценарій перегляду курсів валют, вибору типу курсу, переходу до розширеної сторінки, уточнення міста, перегляду динаміки зміни курсу та переходу до інформації про конкретний банк або обмінник.

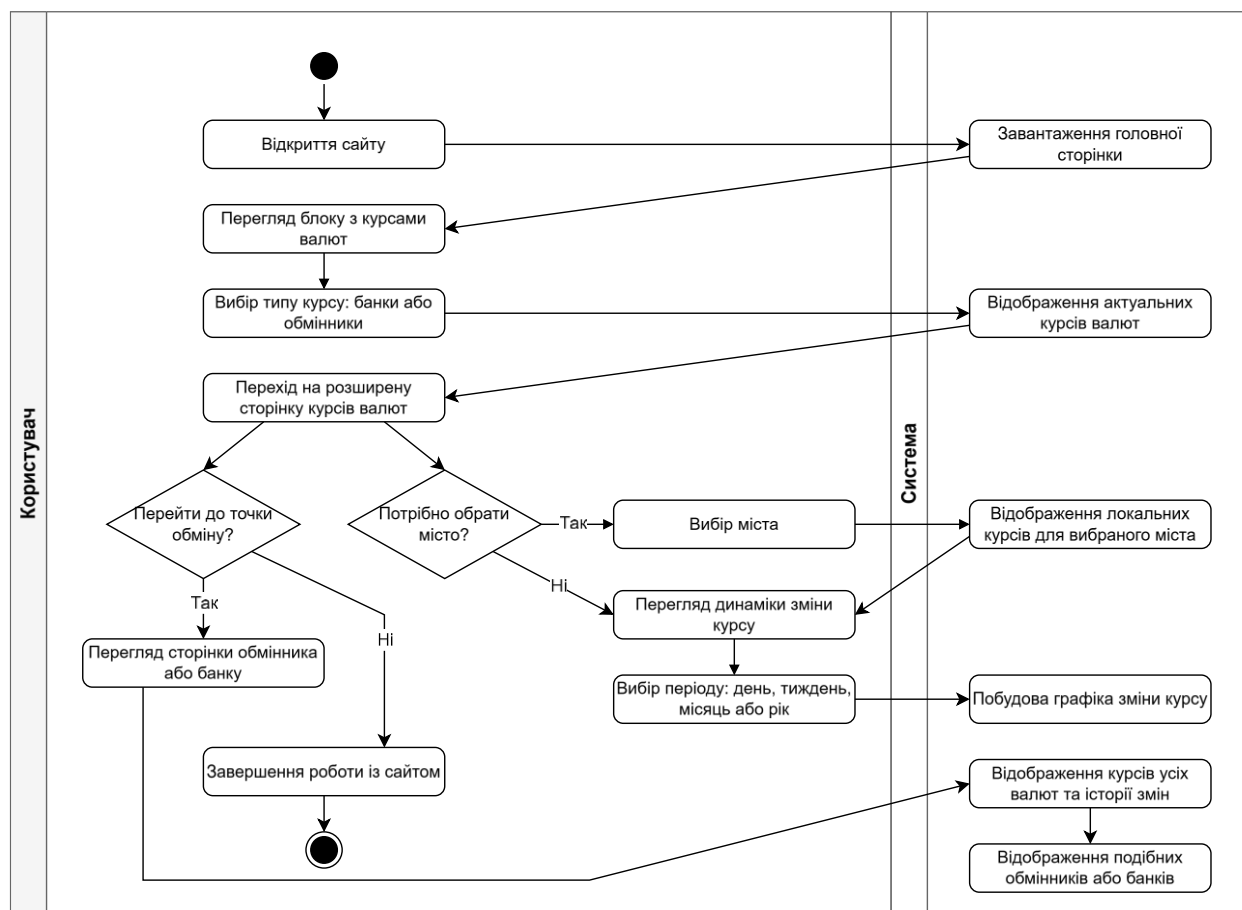


Рисунок 2.2 – Діаграма діяльності сценарію перегляду курсів валют

На діаграмі діяльності зображено основний сценарій взаємодії користувача із сервісом для пошуку оптимальних точок обміну валют. Сценарій починається з відкриття користувачем сайту, після чого система повинна завантажити головну сторінку. Далі користувач переглядатиме блок із курсами валют та обиратиме курси банків або обмінників. Після вибору одного з варіантів система повинна відобразити актуальні курси валют відповідно до обраного типу.

Наступним етапом буде перехід користувача на розширену сторінку курсів валют. За потреби користувач зможе обрати своє місто, після чого система повинна буде показати локальні курси валют для вибраного міста. Такий сценарій дозволить враховувати географічний чинник, оскільки курси валют можуть відрізнятися залежно від регіону та конкретних обмінних пунктів. Після перегляду локальних курсів користувач зможе перейти до перегляду динаміки зміни курсу. Для цього він обиратиме період перегляду: день, тиждень, місяць або рік, а система повинна буде відобразити графік зміни курсу за обраний

проміжок часу, що дозволить користувачеві оцінити динаміку змін курсу обраної валюти.

За потреби користувач зможе перейти до сторінки конкретної точки обміну, банку або обмінника. На цьому етапі система повинна буде відобразити курси всіх доступних валют в конкретній точці обміну, історію зміни курсів, інформацію про відділення, а також подібні банки та обмінники.

Також важливою складовою проектування системи є визначення її функціональних характеристик. Система повинна буде забезпечувати доступ до актуальної інформації про валютні курси та її відображення користувачам. Доступ до можливостей сервісу здійснюватиметься через мережу Інтернет із використанням протоколів сімейства TCP/IP. Для забезпечення надійності роботи передбачатиметься контроль коректності вхідних даних, підтримка цілісності інформації та розмежування доступу до окремих функціональних можливостей.

Інформаційні потоки системи передбачатимуть отримання даних із зовнішніх джерел, їх обробку та зберігання у базах даних з подальшим відображенням користувачеві через вебінтерфейс та програмний інтерфейс. З огляду на те, що основною інформаційною потребою користувача є швидке отримання актуальної інформації про курси валют та можливість порівняння доступних варіантів обміну, структура інформаційних потоків повинна буде забезпечувати своєчасне оновлення даних, мінімальний час їх обробки та швидке відображення результатів пошуку.

Особливості інформаційних потоків та функціональних можливостей системи безпосередньо впливатимуть на структуру користувацького інтерфейсу та організацію взаємодії користувача із сервісом. У зв'язку з цим під час проектування інтерфейсу особлива увага приділятиметься зручності використання системи та швидкому доступу до її основних функціональних можливостей. Основними елементами інтерфейсу будуть: головна сторінка з актуальними курсами валют, розширена сторінка перегляду курсів із можливістю вибору міста та параметрів фільтрації, сторінка окремого банку або обмінного пункту, а також засоби візуалізації історичних даних про зміну валютних курсів.

Результати моделювання дозволили відобразити логіку функціонування розроблюваної системи та визначити її основні функціональні можливості, а побудовані UML-моделі створюють основу для подальшого проєктування архітектури програмного забезпечення, вибору технологій реалізації та розробки окремих функціональних модулів.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Для розробки сервісу для пошуку оптимальних точок обміну валют необхідно обрати програмні засоби, які забезпечуватимуть пошук, збір, зберігання та обробку інформації, а також підтримуватимуть стабільну роботу системи при зростанні обсягів даних і кількості користувачів. Вибір технологій є одним із ключових етапів проєктування програмного забезпечення, оскільки від нього залежать продуктивність, масштабованість, надійність та можливість подальшого розвитку програмного продукту. З огляду на особливості предметної області потрібно проаналізувати сучасні засоби розробки та визначити інструменти, які найкраще підійдуть для реалізації поставлених завдань.

Для реалізації серверної частини можуть використовуватися різні сучасні фреймворки, зокрема Laravel, Symfony та NestJS. Symfony характеризується високою гнучкістю та широкими можливостями налаштування архітектури застосунку, проте його використання зазвичай потребує більшої кількості налаштувань, що підвищує складність розробки. NestJS є популярним рішенням для створення серверних застосунків мовою TypeScript та підтримує модульний підхід до побудови систем, однак потребує використання окремого технологічного стеку. Тому для розроблюваного сервісу обрано фреймворк Laravel, оскільки він надає готові механізми реалізації REST API, спрощує роботу з базами даних, чергами повідомлень, кешуванням та засобами безпеки. Крім того, Laravel дозволяє швидко створювати масштабовані серверні застосунки та має безліч додаткових інструментів, бібліотек та сторонніх інтеграцій.

Під час вибору технології для реалізації клієнтської частини було розглянуто React, Angular та Vue. Angular являє собою повноцінний фреймворк із великою кількістю вбудованих можливостей, однак його використання пов'язане зі складнішою структурою проєкту та більшими витратами часу на розробку. Vue відзначається простотою використання та невисоким порогом входження, проте має меншу поширеність у великих корпоративних проєктах. Тому для розробки інтерфейсу користувача у даній роботі обрано React, який використовує компонентний підхід до побудови інтерфейсів і забезпечує високу швидкодію завдяки ефективному оновленню елементів сторінки. Використання React також дозволяє створити динамічний інтерфейс із можливістю швидкого відображення актуальної інформації без повного перезавантаження сторінок.

Зберігання основної інформації про валюти, банки, обмінні пункти та міста повинне здійснюватися за допомогою реляційних систем керування базами даних, серед яких найбільш поширеними є MySQL та PostgreSQL. Обидві системи забезпечують високу надійність та підтримують складні SQL-запити. Для реалізації даного проєкту обрано саме MySQL, оскільки вона характеризується стабільністю роботи, широким поширенням, простотою адміністрування та ефективною інтеграцією з Laravel. Використання MySQL дозволить забезпечити надійне зберігання структурованих даних та підтримку цілісності інформації під час виконання великої кількості операцій.

Особливістю розроблюваного сервісу є необхідність накопичення та аналізу історичних даних щодо зміни валютних курсів. Для вирішення таких задач можуть використовуватися різні сховища даних, проте найбільш доцільним рішенням є використання бази даних ClickHouse. Вона орієнтована на швидку аналітичну обробку великих обсягів даних та дозволяє виконувати складні запити значно швидше порівняно з реляційними базами даних. Застосування ClickHouse забезпечить ефективне зберігання історії курсів валют та формування статистичних даних для відображення графіків зміни валютних курсів.

Для прискорення роботи системи та організації фонові обробки даних обрано систему кешування Redis. Альтернативою можуть виступати і інші

системи кешування однак Redis поєднує можливості швидкого зберігання даних у пам'яті та підтримку механізмів черг повідомлень. У межах розроблюваної системи Redis планується використовувати для обробки задач автоматичного оновлення курсів валют, виконання фонових процесів та зменшення навантаження на основні компоненти системи.

Важливою вимогою до сервісу є забезпечення швидкого пошуку інформації серед великої кількості записів. Для реалізації пошукових можливостей можуть використовуватися різні спеціалізовані системи, зокрема Elasticsearch та Apache Solr. Для даного проєкту обрано Elasticsearch, який забезпечує високу швидкість пошуку, підтримує фільтрацію даних та дозволяє виконувати складні пошукові запити в режимі реального часу. Його використання дасть можливість швидко знаходити інформацію про банки, обмінники та курси валют відповідно до параметрів, заданих користувачем.

Таким чином, для реалізації сервісу для пошуку оптимальних точок обміну валют буде використано фреймворк Laravel як основу серверної частини [4], React для побудови користувацького інтерфейсу [5], MySQL для зберігання основних даних [6], ClickHouse для аналітичної обробки історичних даних [7], Redis для організації черг та кешування [8], а Elasticsearch для реалізації механізмів пошуку [9]. Обраний технологічний стек дозволить забезпечити продуктивну, масштабовану та зручну інформаційну систему, здатну ефективно вирішувати поставлені задачі.

Для детального представлення структури розроблюваної системи та ролі обраних технологій побудовано UML-діаграму компонентів зображену на рисунку 2.3, яка відображає основні програмні модулі, зовнішні джерела даних і зв'язки між ними. У розроблюваній системі основними компонентами виступатимуть клієнтська та серверна частини, модуль автоматизованого збору даних, бази даних, пошуковий механізм та система черг. Клієнтський застосунок відповідатиме за взаємодію з користувачем, серверна частина забезпечуватиме обробку запитів і реалізацію бізнес-логіки, а модуль збору даних виконуватиме отримання інформації із зовнішніх сайтів та API банків і обмінників.

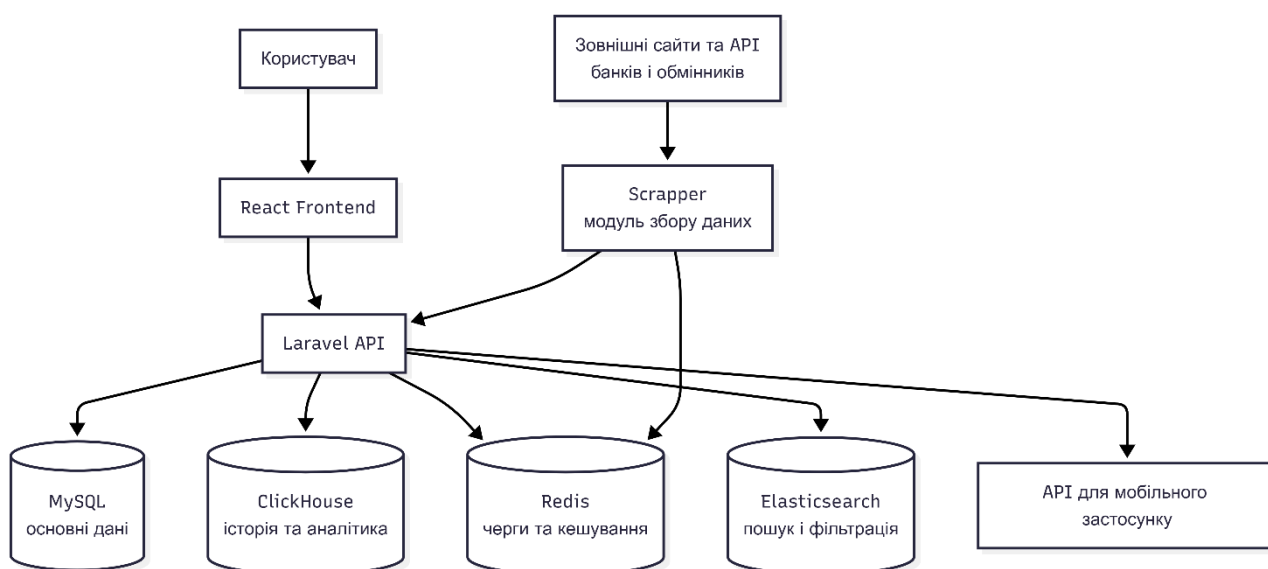


Рисунок 2.3 – Діаграма компонентів розроблюваного сервісу

На діаграмі компонентів зображено загальну архітектуру сервісу для пошуку оптимальних точок обміну валют. Користувач взаємодіятиме із системою через клієнтську частину, реалізовану за допомогою React, яка буде забезпечувати відображення курсів валют, сторінок банків і обмінників, графіків динаміки зміни курсів та результатів пошуку. Серверна частина, реалізована на основі Laravel API, буде центральним компонентом системи, який оброблятиме запити від клієнтської частини, забезпечуватиме взаємодію з базами даних, пошуковою системою та службовими модулями, а також надаватиме програмний інтерфейс для взаємодії з мобільним застосунком.

Для зберігання основних даних використовуватиметься MySQL. У цій базі даних повинна буде зберігатися інформація про міста, валюти, банки, обмінні пункти та актуальні значення курсів. ClickHouse використовуватиметься для збереження історичних даних і виконання аналітичних запитів, пов'язаних із побудовою графіків та аналізом динаміки зміни валютних курсів.

Redis у структурі системи виконуватиме роль інструмента для роботи з чергами та кешуванням. Його використання дозволить організувати фонове виконання задач, пов'язаних з оновленням курсів валют, та зменшити навантаження на основні компоненти системи. Elasticsearch відповідатиме за швидкий пошук і фільтрацію інформації про банки та обмінники відповідно до

параметрів, заданих користувачем. Окремим компонентом архітектури є модуль автоматизованого збору даних, який буде отримувати інформацію із зовнішніх сайтів та API банків і обмінників, після чого передавати її для подальшої обробки та збереження, що дозволить підтримувати актуальність курсів валют і формувати історичні дані для подальшого аналізу.

Для відображення логіки взаємодії компонентів розроблюваного сервісу побудовано блок-схему одного з його ключових процесів, а саме оновлення курсів валют, який забезпечуватиме отримання інформації із зовнішніх джерел, її обробку та збереження в системі (рис. 2.4).

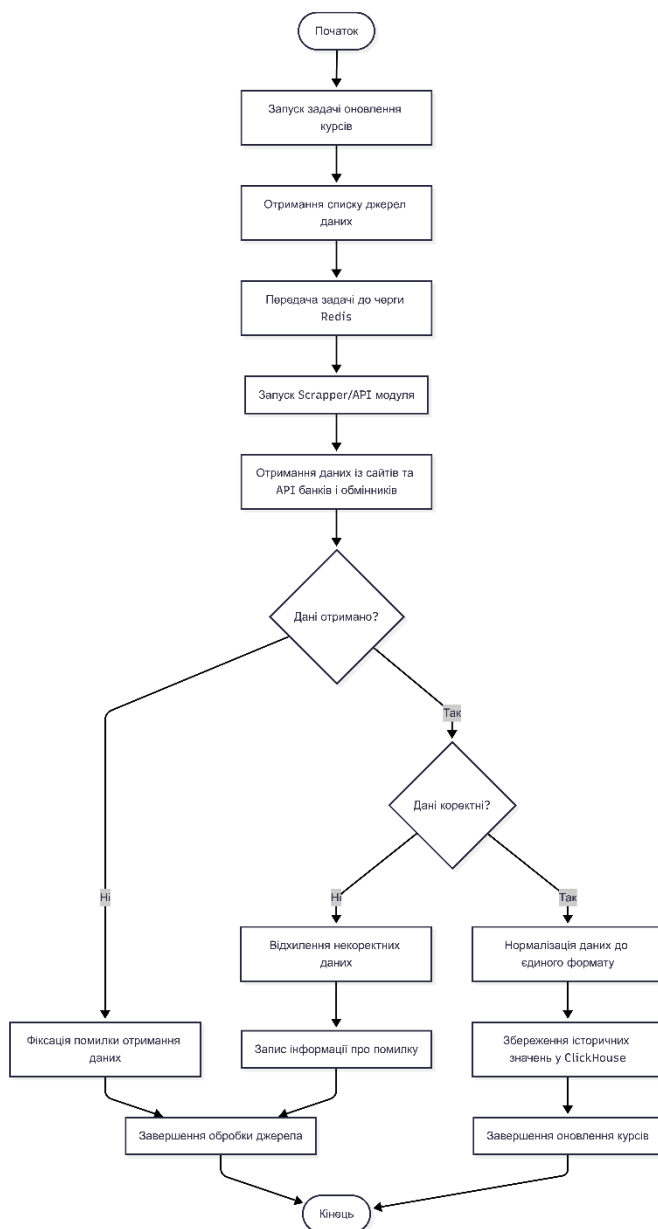


Рисунок 2.4 – Алгоритм оновлення курсів валют

На блок-схемі зображено послідовність дій, які будуть виконуватися під час автоматизованого оновлення курсів валют. Процес буде починатися із запуску задачі оновлення курсів, після чого система буде отримати перелік зовнішніх джерел даних. Такими джерелами можуть бути сайти та API банків і обмінників, з яких надходитиме інформація про актуальні курси валют. Для організації фонові обробки задачі оновлення курсу повинні будуть передаватися до черги в Redis, що дозволить виконувати збір даних без блокування основної роботи системи та зменшити навантаження на серверну частину. Після цього модуль Scrapper/API буде звертатися до зовнішніх джерел і отримувати необхідну інформацію про курси валют.

Після отримання даних система буде перевіряти, чи вдалося отримати відповідь від джерела. Якщо дані не були отримані, помилку буде зафіксовано, а обробку відповідного джерела завершено. У разі успішного отримання інформації система буде виконати перевірку її коректності. Якщо отримані дані містять помилки або не відповідають очікуваному формату, вони будуть відхилятися, а інформація про помилку зберігатиметься для подальшого аналізу та контролю роботи джерел даних.

Після успішного проходження перевірки дані будуть нормалізуватися до єдиного формату. Це необхідно через те, що різні банки та обмінники можуть надавати інформацію в різній структурі та з використанням різних форматів, а нормалізація забезпечуватиме уніфіковане зберігання інформації та її подальшу обробку в межах системи. Завершальним етапом алгоритму є збереження історичних значень курсів валют у ClickHouse, після чого процес оновлення курсів вважається завершеним.

Для деталізації процесу обробки користувацького запиту побудовано діаграму послідовності (рис. 2.5). На відміну від діаграми компонентів, яка показує загальну структуру системи, діаграма послідовності дозволяє описати порядок взаємодії між окремими частинами програмного забезпечення під час виконання конкретного сценарію. У межах розроблюваного сервісу таким

сценарієм є пошук курсів валют користувачем, оскільки він охоплює взаємодію клієнтської частини, серверного API, пошукового механізму та сховищ даних.

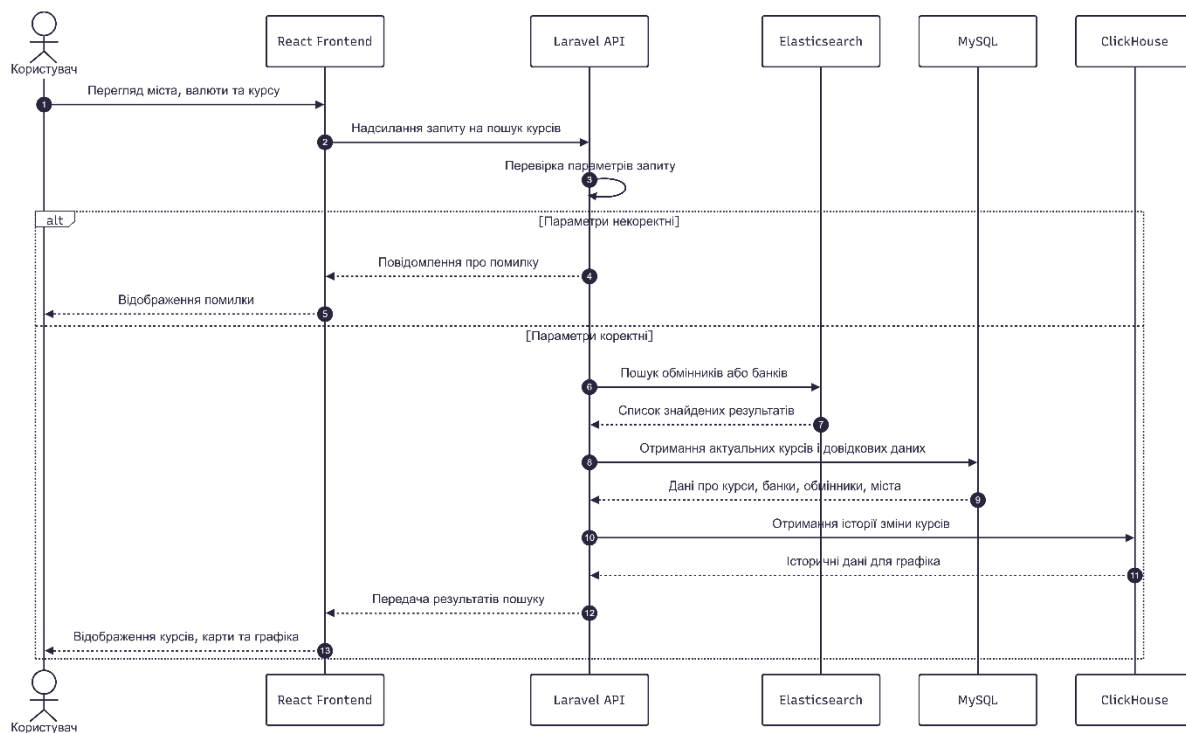


Рисунок 2.5 – Діаграма послідовності процесу пошуку курсів валют

Сценарій на діаграмі починається з того, що користувач обирає місто, валюту та тип курсу. Після цього клієнтська частина, реалізована за допомогою React, буде формувати запит і передавати його до серверної частини системи, де Laravel API отримуватиме запит та виконуватиме перевірку переданих параметрів. Якщо параметри будуть некоректними або неповними, система буде повертати повідомлення про помилку, яке відображатиметься користувачеві у вебінтерфейсі. У разі коректності параметрів серверна частина звертатиметься до Elasticsearch для виконання пошуку банків або обмінних пунктів відповідно до заданих умов. Пошукова система повертатиме список знайдених результатів, після чого Laravel API отримуватиме з MySQL актуальні курси валют та довідкову інформацію про банки, обмінники й міста, що дозволить сформувати повний набір даних, необхідних для відображення користувачеві.

Для побудови графіків динаміки зміни курсу сервер буде звертатися до ClickHouse, де зберігатимуться історичні значення валютних курсів. Отримані історичні дані передаватимуться разом із результатами пошуку до клієнтської частини, після чого користувачеві буде відображено список курсів, карту розташування банку або обмінника, а також графік курсу за обраний період.

Побудована діаграма компонентів дозволила сформувану архітектурну сервісу та визначити взаємозв'язки між його основними складовими. Розроблена блок-схема алгоритму оновлення курсів валют відобразила послідовність дій, необхідних для автоматизованого отримання, перевірки та збереження інформації з зовнішніх джерел, а діаграма послідовності деталізувала процес обробки користувацьких запитів і продемонструвала порядок взаємодії між окремими компонентами системи під час виконання основних функцій.

Обраний технологічний стек та запропонована структура системи забезпечуватиме ефективну взаємодію між компонентами, підтримуватиме масштабування програмного забезпечення та створюватиме основу для реалізації сервісу для пошуку оптимальних точок обміну валют.

РОЗДІЛ 3

РОЗРОБКА СЕРВІСУ ДЛЯ ПОШУКУ ОПТИМАЛЬНИХ ТОЧОК ОБМІНУ ВАЛЮТ З ВИКОРИСТАННЯМ LARAVEL, REACT, MYSQL, CLICKHOUSE, REDIS TA ELASTICSEARCH

3.1 Практична реалізація об'єкта проєктування

Реалізація сервісу для пошуку оптимальних точок обміну валют здійснювалась з використанням сучасного стеку вебтехнологій, який включає Laravel для розробки серверної частини та React для створення користувацького інтерфейсу. Такий підхід дозволяє розділити логіку обробки даних та відображення інформації, забезпечуючи незалежний розвиток компонентів системи та спрощуючи їх подальшу підтримку. Взаємодія між клієнтською та серверною частинами реалізована через програмний інтерфейс, який забезпечує передачу даних між компонентами системи та підтримує роботу вебінтерфейсу.

У процесі розробки особлива увага приділялася реалізації механізмів автоматизованого отримання актуальних курсів валют, пошуку та фільтрації інформації, а також відображенню результатів у зручному для користувача вигляді. Для забезпечення актуальності даних передбачено використання модуля автоматизованого збору інформації із зовнішніх джерел, а для обробки користувацьких запитів реалізовано набір серверних сервісів та програмних інтерфейсів, що дозволяють забезпечити оперативне оновлення інформації та швидке надання результатів користувачам.

Архітектура розроблюваної системи побудована за клієнт-серверним принципом, що забезпечує масштабованість програмного забезпечення, спрощує інтеграцію з додатковими сервісами та підтримує незалежний розвиток окремих функціональних модулів. Важливою складовою серверної частини розроблюваного сервісу є програмний інтерфейс API, через який клієнтський застосунок отримує доступ до актуальних курсів валют, історичних даних та іншої інформації. Для реалізації взаємодії між клієнтською та серверною

частинами у Laravel було створено набір API-маршрутів наведених в лістингу 3.1, які забезпечують доступ до необхідних функціональних можливостей системи.

Лістинг 3.1 – Реалізація API-маршрутів для роботи з валютними курсами

```
Route::prefix('app-api')
->group(function () {
    Route::controller(PrefetchApi\ProvidersController::class) -
>group(function () {
    Route::get('exchangers/latest-rates', 'apiLatestRates')
        ->name('api.exchangers.latest-rates');

    Route::get('exchangers/rates-history', 'apiProviderHistoryRates')
        ->name('api.exchangers.exchanger-rates-history');
    });

    Route::controller(PrefetchApi\CurrenciesController::class) -
>group(function () {
    Route::get('currencies/rates-history', 'apiCurrencyHistoryRates')
        ->name('api.currencies.currency-rates-history');
    });
});
```

кінець лістингу 3.1

Наведений фрагмент коду демонструє реалізацію групування маршрутів із використанням спільного префікса `app-api`, що дозволяє логічно відокремити API системи від інших маршрутів застосунку та спрощує подальшу підтримку програмного забезпечення. Через визначені маршрути користувачі можуть отримувати актуальні курси валют, а також історичні дані щодо змін курсів окремих валют і обмінних пунктів, а використання контролерів дозволяє розділити логіку маршрутизації та бізнес-логіку застосунку відповідно до принципів архітектури MVC.

Після отримання запиту від клієнтської частини його обробка виконується відповідним контролером, який відповідає за перевірку вхідних параметрів, формування запитів до сервісного шару та підготовку даних для подальшого відображення користувачеві. Фрагмент реалізації обробки сторінки курсів валют наведено в лістингу 3.2.

Лістинг 3.2 – Фрагмент обробки сторінки курсів валют у контролері

```

$countryRatesService = app(CountryRatesService::class)
    ->setCountry($locale->country);

if ($city) {
    $countryRatesService->setCity($city);
}

$dateIntervalWithRates = $countryRatesService->getRatesScannedAtInterval(
    ignorePrimaryProvider: false
);

$cities = $countryRatesService->filterCitiesWithRates();
$ratesCharts = $countryRatesService->resolveRatesChartsBlockData();

return $renderer->prefetch(
    fn () => ApiLatestRatesPQ::make(
        $city,
        $date,
        LatestRatesTableTabs::ratesPageMainTab()
    ),
);

```

кінець лістингу 3.2

Представлений фрагмент коду демонструє використання сервісного шару для роботи з даними про валютні курси. Під час виконання запиту система враховує обране користувачем місто, визначає доступний часовий діапазон даних, перевіряє коректність запиту та формує набір інформації, необхідної для відображення на сторінці. Окремо здійснюється підготовка даних для побудови графіків динаміки зміни курсів валют та отримання актуальних значень курсів для подальшого відображення в інтерфейсі користувача. Це дозволило відокремити бізнес-логіку від контролерів і забезпечити більш гнучку структуру програмного забезпечення. Такий підхід спрощує подальше розширення функціональних можливостей системи та надає можливість повторного використання програмного коду. У результаті реалізації серверної частини було створено механізм обробки запитів користувачів, отримання актуальної інформації про валютні курси та підготовки даних для їх подальшого відображення у вебінтерфейсі.

Для реалізації автоматизованого збору курсів валют у розроблюваній системі було створено окремий модуль парсерів, який забезпечує отримання

інформації із зовнішніх сайтів та API банків і обмінних пунктів. Основним призначенням даного модуля є автоматичне оновлення актуальних курсів валют без необхідності ручного введення інформації, що дозволяє забезпечити оперативне оновлення інформації та підтримувати актуальність даних у сервісі.

Робота модуля збору даних базується на використанні окремих класів-парсерів для кожного джерела інформації. Кожен парсер відповідає за отримання даних із конкретного сайту або API, обробку отриманої відповіді та формування структури даних, придатної для подальшого використання в межах системи. Приклад реалізації отримання курсів валют із зовнішнього API наведено у лістингу 3.3.

Лістинг 3.3 – Реалізація парсера для отримання курсів валют із зовнішнього API

```
public function parse(): array
{
    $client = new GuzzleClient([
        'connect_timeout' => 10,
    ]);

    $response = $client->request('GET', static::URL);
    $response = json_decode($response->getBody(), true, flags:
JSON_THROW_ON_ERROR);

    $data = Arr::get($response, static::CONTAINER_KEY);

    $rates = [];
    foreach ($data as $item) {
        $rates[] = $this->mapRate(
            $this->getCity(),
            $this->getCurrencyByCode(
                Arr::get($item, static::CURRENCY_KEY)
            ),
            $this->getCurrencyByCode('UAH'),
            $ask,
            $bid
        );
    }

    return $rates;
}
```

кінець лістингу 3.3

У наведеній функції використовується HTTP-клієнт Guzzle для виконання запиту до зовнішнього API [10]. Після отримання відповіді виконується її

перетворення з формату JSON у масив даних, який надалі обробляється програмою. Для кожного отриманого запису формується окремий об'єкт курсу із зазначенням валюти, міста та значеннями купівлі й продажу, після чого отримані дані передаються для подальшої обробки.

Оскільки різні банки та обмінники можуть надавати інформацію в різних форматах, після етапу отримання даних виконується їх нормалізація, фрагмент реалізації якої наведено в лістингу 3.4. Метою даного процесу є уніфікація всіх курсів валют до єдиної внутрішньої структури, яка буде використовуватися під час збереження та подальшої обробки інформації.

Лістинг 3.4 – Реалізація нормалізації даних валютного курсу

```
protected function mapRate(
    ?City $city,
    ?Currency $baseCurrency,
    ?Currency $targetCurrency,
    float $ask,
    float $bid
): ?array {
    if (! $baseCurrency || ! $targetCurrency || min($ask, $bid) <= 0) {
        return null;
    }

    return [
        'provider_id' => $this->getProvider()->getKey(),
        'city_id' => $city?->getKey(),
        'base_currency_id' => $baseCurrency->getKey(),
        'target_currency_id' => $targetCurrency->getKey(),
        'ask' => $ask,
        'bid' => $bid,
    ];
}
```

кінець лістингу 3.4

Наведений фрагмент демонструє формування єдиної структури даних для всіх курсів валют незалежно від джерела їх отримання. Перед створенням запису виконується перевірка наявності необхідних валют та коректності значень курсу. Після успішної перевірки формується масив, який містить ідентифікатор джерела даних, місто, валютну пару та значення курсів купівлі й продажу.

Щоб забезпечити зручну роботу користувача з валютними курсами у системі було реалізовано механізм фільтрації актуальних даних. Основним

призначенням якого є відбір курсів валют відповідно до параметрів, які задаються користувачем під час роботи з інтерфейсом. До таких параметрів належать тип курсу, обрана валюта, місто, а також спосіб сортування результатів. Завдяки чому користувач отримує не весь набір даних, а лише релевантну інформацію, необхідну для порівняння банків або обмінних пунктів.

Фільтрація актуальних курсів реалізована на рівні сервісного шару, що дозволяє відокремити логіку обробки даних від контролерів та інтерфейсу користувача. Це дозволяє змінювати правила відбору даних без суттєвого впливу на інші частини програмного забезпечення. Приклад реалізації механізму фільтрації актуальних курсів валют наведено в лістингу 3.5.

Лістинг 3.5 – Реалізація фільтрації курсів валют за параметрами запиту

```
$filters = fn (ClickhouseBuilder $query): ClickhouseBuilder => $query
    ->whereIn(
        'base_currency_id',
        $this->country->nonPrimaryCurrencies->modelKeys()
    )
    ->where(
        'target_currency_id',
        $this->country->primaryCurrency->getKey()
    )
    ->where(
        fn (ClickhouseBuilder $subQuery) =>
            $lastRatesTableTab->clickhouseFilters($this, $subQuery)
    );

$paginator = $this->clickhouseLastRatesTableRepository->getLatestRates(
    $date,
    $filters,
    $orderByByName,
    $orderByByValue,
    $page,
    $limit
);
```

кінець лістингу 3.5

У представленому фрагменті коду формується набір умов, які використовуються для отримання актуальних курсів валют. Спочатку система обмежує вибірку валютами, що не є основною валютою країни, після чого задається цільова валюта, відносно якої відображаються курси. Додаткові умови

фільтрації визначаються через метод `clickhouseFilters`, який враховує вибраний тип таблиці курсів, зокрема перегляд курсів банків або обмінників.

Після формування фільтрів вони передаються до класу-репозиторію, який відповідає за отримання актуальних курсів із відповідного сховища даних. Разом із фільтрами також передаються параметри сортування, номер сторінки та кількість записів на сторінці. Це дозволяє реалізувати посторінкове відображення результатів і не завантажувати надмірний обсяг інформації за один запит, що є важливим для даного сервісу, оскільки він працює з великою кількістю банків, обмінних пунктів, валют і міст.

Реалізований механізм фільтрації дозволяє користувачеві швидко отримувати потрібну інформацію про актуальні курси валют залежно від вибраних параметрів. Винесення логіки фільтрації до сервісного шару забезпечує гнучкість програмного рішення та створює можливість для подальшого розширення функціональності, наприклад додавання нових типів фільтрів, способів сортування або окремих режимів перегляду курсів валют.

Після реалізації серверної частини, механізмів автоматизованого збору даних та пошуку інформації наступним етапом розробки стало створення клієнтської частини із використанням `React`. Саме інтерфейс користувача забезпечує взаємодію із сервісом та надає можливість переглядати актуальні курси валют, виконувати пошук і фільтрацію даних, аналізувати динаміку зміни курсів та отримувати інформацію про банки й обмінні пункти.

Структура клієнтської частини побудована у вигляді набору взаємопов'язаних компонентів, де основний компонент сторінки відповідає за отримання даних від серверної частини, передачу їх дочірнім компонентам та відображення основних елементів інтерфейсу, фрагмент реалізації якого наведено в лістингу 3.6. До таких елементів належать таблиця курсів валют, навігаційні елементи та блок відображення графіків зміни валютних курсів.

Лістинг 3.6 – Реалізація головного компонента сторінки курсів валют

```
export default function Index(props: Props) {
  const indexQuery = currencyExchangeApi.useIndexQuery({
```

```

        city: props.city,
        date: props.date
    });

    return (
        <PageQueryProvider query={ indexQuery }>
            {({ data }) => (
                <Container>
                    <Title bold tighter>
                        { data.h1 }
                    </Title>

                    <ProvidersTable
                        city={ props.city }
                        date={ props.date }
                        cities={ data.tableFilters.cities }
                        tabs={ data.tableTabs }
                    />

                    {
                        data.dynamicsExchangeRates &&
                        <DynamicsExchangeRatesGraphCarousel
                            data={ data.dynamicsExchangeRates }
                        />
                    }
                </Container>
            )}
        </PageQueryProvider>
    );
}

```

кінець лістингу 3.6

Даний фрагмент коду демонструє реалізацію головного компонента сторінки перегляду курсів валют, який після отримання даних із серверної частини розподіляє їх до компонентів, які відповідають за відображення таблиці курсів та графіків зміни валютних показників.

Також важливою складовою клієнтської частини є механізм отримання інформації від серверного API (ліст. 3.7). Для цього використовується бібліотека Redux Toolkit Query, яка дозволяє керувати запитами до сервера, автоматично кешувати результати та повторно використовувати отримані дані [11].

Лістинг 3.7 – Реалізація API-запиту для отримання курсів валют

```

const currencyExchangeApi = createApi({
    reducerPath: 'currencyExchange',
    baseQuery: createBaseQuery(),
    endpoints: build => ({

```

```

index: build.query({
  query: ({ date, city }) => ({
    url: pagesCurrencyExchangerUrl(city, date),
    method: 'get'
  })
})
});

```

кінець лістингу 3.7

Представлений фрагмент коду демонструє реалізацію API-модуля, який забезпечує отримання інформації про курси валют із серверної частини системи.

Результатом реалізації клієнтської частини став вебінтерфейс, який забезпечує зручний доступ до функціональних можливостей сервісу. На головній сторінці, зображеній на рисунку 3.1, користувач отримує доступ до актуальної інформації про валютні курси та основних інструментів навігації системою.

Все про фінанси

Ваш корисний довідник щодо курсів іноземних валют в Україні. Незалежно від того, чи є ви резидентом України, іноземним мандрівником, який відвідує країну, інвестором, який прагне залучити українську економіку, чи просто вам цікаво дізнатися про динаміку валюти в регіоні.

КУРС У БАНКАХ

КУРС В ОБМІННИКАХ



Курс валют

Курс у твоєму місті →

Банки Обмінники

	USD	EUR	PLN	GBP	
НБУ	44,268	51,554	12,194	59,439	31,194
Обмінник	Купівля Продаж	Купівля Продаж	Купівля Продаж	Купівля Продаж	Купівля Продаж
Центро Фінанс Львів, Станом на 21:36	▲ 44,000 +0.100	▲ 44,250 +0.050	▼ 51,200 -0.100	▲ 51,800 +0.200	▲ 11,950 +0.050
OBMIN24 Дніпро, Станом на 21:36	▼ 44,000 -0.100	▼ 44,300 -0.050	▼ 51,620 -0.030	▼ 51,750 -0.050	▼ 12,050 -0.010

Рисунок 3.1 – Головна сторінка сервісу

На сторінці перегляду курсів валют (рис. 3.2) реалізовано можливість фільтрації даних, вибору міста, перегляду таблиць курсів та аналізу динаміки їх зміни за допомогою графіків, що дозволяє користувачеві швидко знаходити необхідну інформацію та порівнювати доступні варіанти обміну валют.

Обмін валют в банках та обмінниках України

Всі Банки Обмінники 01.06.2026 Micro

	USD	EUR	PLN	GBP	
НБУ	44,268	51,554	12,194	59,439	31,
Обмінник	Купівля Продаж	Купівля Продаж	Купівля Продаж	Купівля Продаж	Купівля
KURS.SUMY.UA Суми, Станом на 21:46	▲ 44,050 +0.050	▲ 44,550 +0.050	▲ 51,550 +0.100	▲ 52,150 +0.100	▲ 11,950 +0.050
Обмін валют Львів, Станом на 21:46	▼ 44,000 -0.050	▼ 44,150 -0.050	▼ 51,500 -0.050	▼ 51,650 -0.050	▼ 12,100 -0.050
Артада Київ, Станом на 21:46	▼ 43,900 -0.150	▼ 44,330 -0.020	▼ 51,400 -0.100	▼ 51,900 -0.020	▼ 12,000 -0.050
Money24/7 Суми, Станом на 21:46	▲ 44,000 +0.100	▲ 44,500 +0.100	▲ 51,350 +0.150	▲ 51,950 +0.150	▲ 11,800 +0.100
Garant Money	▲ 43,950	▲ 44,180	▼ 51,400	▼ 51,700	▼ 12,030

Рисунок 3.2 – Сторінка перегляду курсів валют

Реалізація клієнтської частини забезпечила створення зручного та інтуїтивно зрозумілого інтерфейсу користувача, який дозволяє ефективно взаємодіяти з сервісом, отримувати актуальну інформацію про курси валют та використовувати функції фільтрації й аналізу даних.

Після реалізації клієнтської частини сервісу наступним етапом стало відображення історичних даних у зручному для користувача вигляді. Оскільки сервіс працює не лише з актуальними курсами валют, а й з історією їх зміни, важливою функціональною можливістю стало формування графіків динаміки курсів. Це дозволяє користувачеві не тільки переглядати поточні значення, а й аналізувати зміну курсу за різні проміжки часу.

Для побудови графіків використано компонент на основі бібліотеки `react-chartjs-2` [12], фрагмент реалізації якого наведено у лістингу 3.8. Він отримує підготовлені дані про курси валют, формує підписи осі часу та набори значень для відображення ліній графіка. При цьому користувач може переглядати дані за різні періоди та перемикає тип курсу, наприклад купівлю або продаж.

Лістинг 3.8 – Реалізація побудови графіка динаміки курсів валют

```
const currentData = useMemo(() => {
  const period = props.data.find(({ id }) => id === activeTab) ??
  props.data[0];

  return {
```

```

    labels: period.data.labels,
    datasets: period.data.datasets.map(({ label, chartExchangeRates, color
  }) => ({
    label,
    data: chartExchangeRates.map(item =>
      typeof item === 'object' && item
        ? item[activeRateType]
        : item
    ),
    borderColor: color,
    backgroundColor: color,
    tension: 0.3
  }))
  };
}, [activeTab, activeRateType, props.data]);

return <Line options={options} data={currentData} height={300} />;

```

кінець лістингу 3.8

У наведеному фрагменті коду формується набір даних, який передається до компонента графіка. Спочатку визначається активний період перегляду, після чого з отриманих даних формуються підписи для осі часу та набори значень курсів валют. Якщо значення курсу містить окремі показники купівлі та продажу, система відображає той показник, що відповідає активному типу курсу. Після підготовки дані передаються до компонента Line для побудови лінійного графіка.

Реалізований механізм дозволяє динамічно оновлювати графік при зміні періоду перегляду або типу курсу. Завдяки цьому користувач може аналізувати зміну валютних курсів у часі та порівнювати різні значення без необхідності переходу на інші сторінки. Приклад відображення графіка динаміки зміни курсу валют наведено на рисунку 3.3.

Динаміка курсів валют

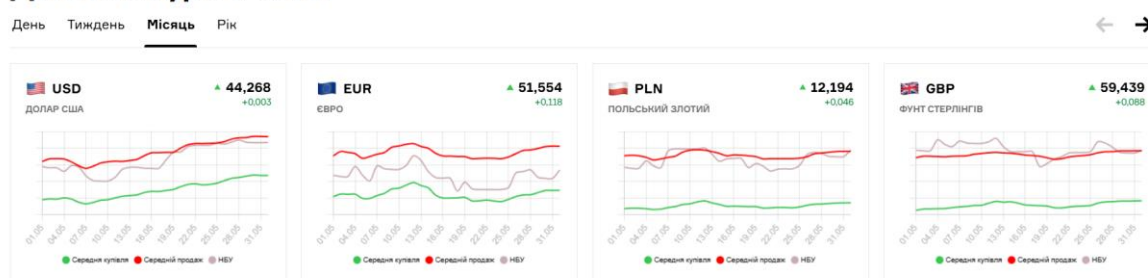


Рисунок 3.3 – Графік динаміки зміни курсу валют

Окрім перегляду загальної динаміки курсів, у системі також передбачено сторінку окремого банку або обмінного пункту (рис. 3.4) на якій користувач може переглянути доступні курси валют, історію їх зміни та додаткову інформацію про вибрану точку обміну. Такий підхід забезпечує детальніший аналіз умов обміну та дозволяє користувачеві приймати рішення на основі ширшого набору даних.

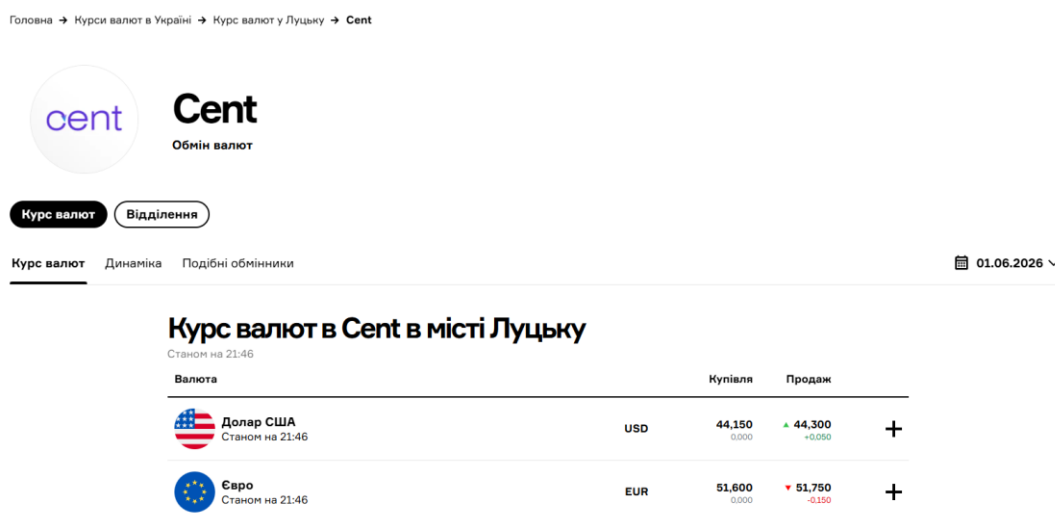


Рисунок 3.4 – Сторінка окремого обмінного пункту

За наявності інформації про відділення банків або обмінних пунктів користувач також може переглядати їх розташування на карті, що є важливим для вибору оптимальної точки обміну, оскільки користувач враховує не лише курс валют, а й географічне розташування відповідного банку або обмінника (рис. 3.5).

Відділення IFS в місті Київ

	IFS просп. Науки, 1, Центральний автовокзал	Зачинено Відчиняється в вівторок о 08:00 пн - нд 08:00 - 20:00
	IFS вул. Глубоцька, 29-31	Зачинено Відчиняється в вівторок о 09:00 пн - нд 09:00 - 19:00
	IFS площа Бессарабська, 2	Зачинено Відчиняється в вівторок о 09:00 пн - нд 09:00 - 20:00
	IFS просп. Берестейський, 136Ж	Зачинено Відчиняється в вівторок о 08:00 пн - нд 08:00 - 20:00

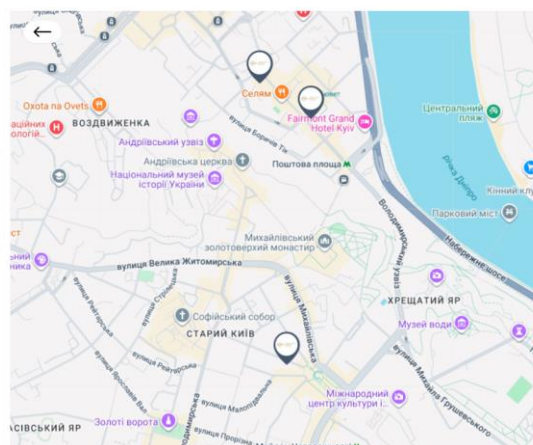


Рисунок 3.5 – Карта відділень обмінних пунктів

Отже, реалізація візуалізації історії зміни курсів валют дозволила розширити функціональні можливості клієнтської частини сервісу та надати користувачеві інструменти для аналізу валютної динаміки. Використання графіків, сторінок окремих обмінних пунктів і карти відділень забезпечує зручне представлення даних та сприяє оптимальному вибору точки для обміну валют.

Таким чином, у результаті практичної реалізації об'єкта проектування було створено основні програмні модулі сервісу для пошуку оптимальних точок обміну валют. Розроблене програмне забезпечення забезпечує автоматизований збір, зберігання та відображення інформації про валютні курси та повністю відповідає функціональним вимогам, визначеним на етапі проектування системи.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Тестування є важливим етапом розробки програмного забезпечення, оскільки воно дозволяє перевірити правильність реалізації функціональних можливостей системи, виявити потенційні помилки та оцінити стабільність роботи програмного продукту. У процесі розробки сервісу для пошуку оптимальних точок обміну валют проводилося поетапне тестування як серверної, так і клієнтської частини системи. Особлива увага приділялася перевірці коректності роботи API, обробці помилкових запитів, тестуванню адаптивності користувацького інтерфейсу та перевірці механізмів валідації даних.

Для перевірки коректного функціонування програмного забезпечення використовувалося ручне функціональне тестування, тестування програмного інтерфейсу за допомогою спеціалізованих інструментів, а також перевірка роботи вебінтерфейсу на різних типах пристроїв, що дозволило своєчасно виявити помилки та усунути недоліки ще на етапі розробки сервісу.

Для перевірки коректності роботи серверної частини та програмного інтерфейсу застосовувався інструмент Postman [13]. За його допомогою виконувалося надсилання HTTP-запитів до API та аналіз отриманих

відповідей [14]. Одним із перевірених сценаріїв був запит на отримання актуальних курсів валют для конкретного міста. Результат успішного виконання такого запиту наведено на рисунку 3.6.

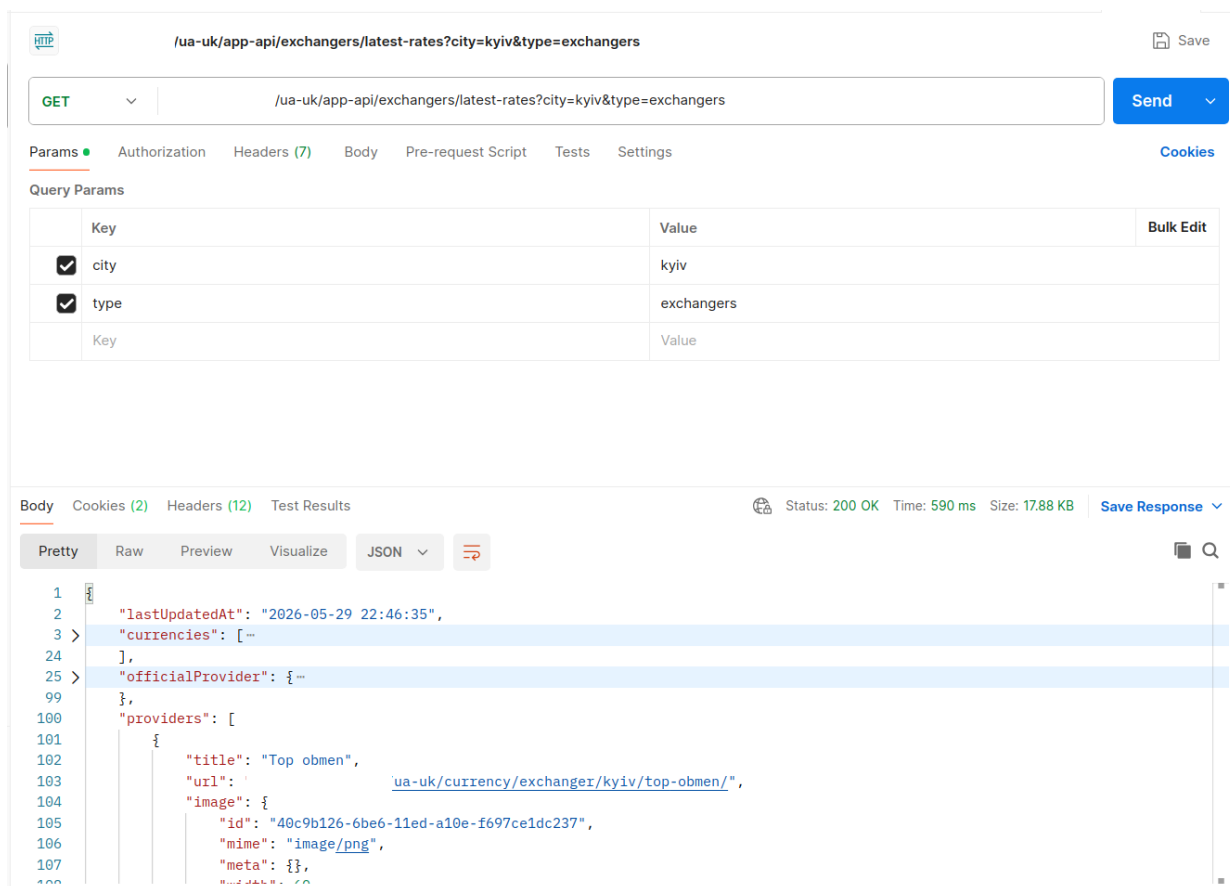


Рисунок 3.6 – Успішне виконання запиту отримання курсів валют у Postman

На рисунку продемонстровано результат успішного звернення до програмного інтерфейсу системи із передачею коректних параметрів запиту. Сервер успішно обробив запит та повернув код відповіді 200 (OK), що свідчить про правильне функціонування механізму отримання актуальних курсів валют та передачі даних клієнту.

Для перевірки механізму валідації також було виконано запит із некоректним значенням параметра міста. Реакцію системи на цей запит наведено на рисунку 3.7.

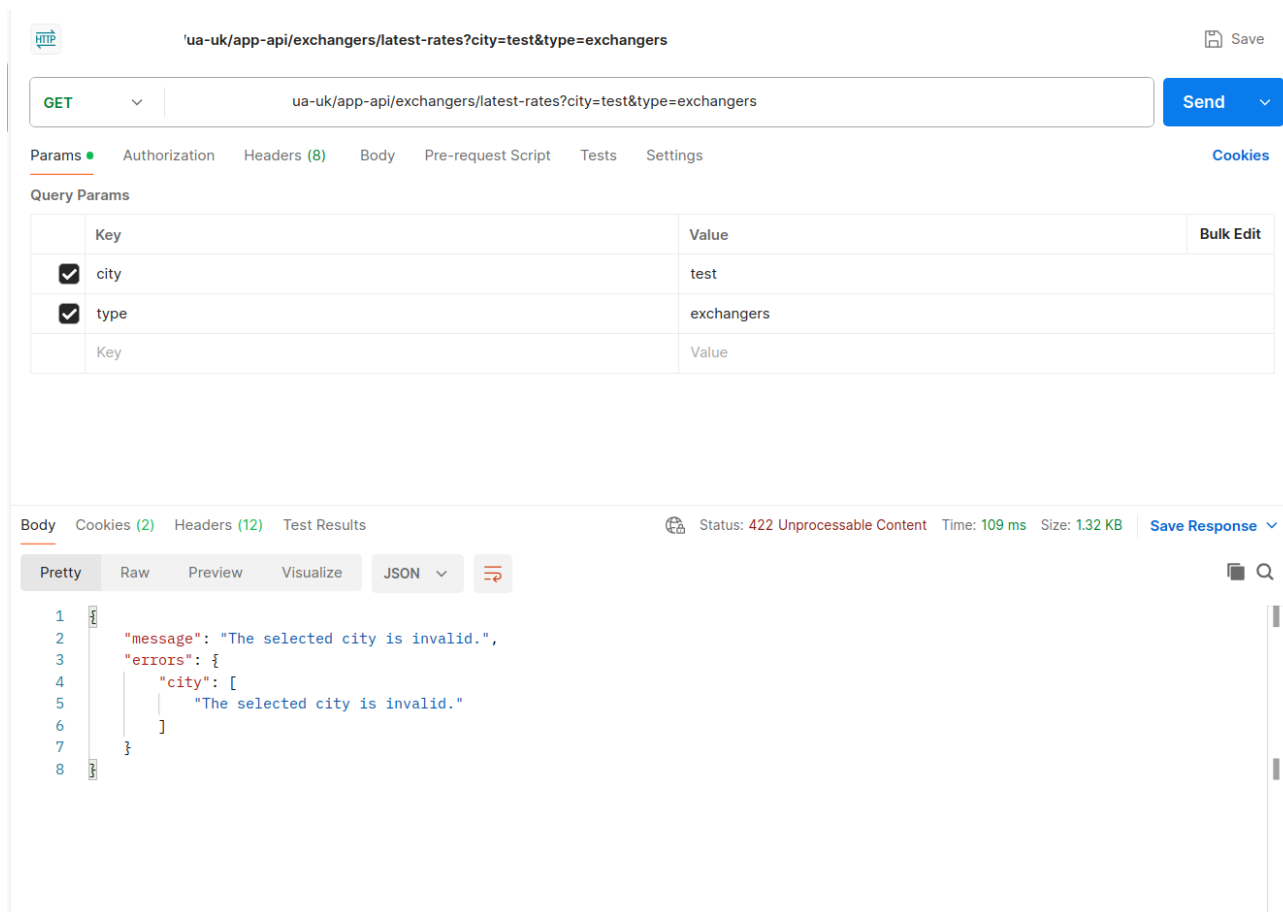


Рисунок 3.7 – Результат обробки некоректного запиту з помилковим параметром

Отримана відповідь демонструє що система коректно визначила помилкове значення параметра `city` та повернула код відповіді 422 (Unprocessable Content) із повідомленням про помилку валідації, що підтверджує правильність реалізації механізмів перевірки вхідних даних та захисту API від некоректних запитів.

Також, під час тестування серверної частини було виявлено помилку, пов'язану з обробкою параметра дати у URL-адресі. У разі передачі некоректного значення дати система намагалася виконати його перетворення без попередньої перевірки, що призводило до виникнення внутрішньої помилки сервера. На рисунку 3.8 наведено результат передачі некоректного значення параметра дати.

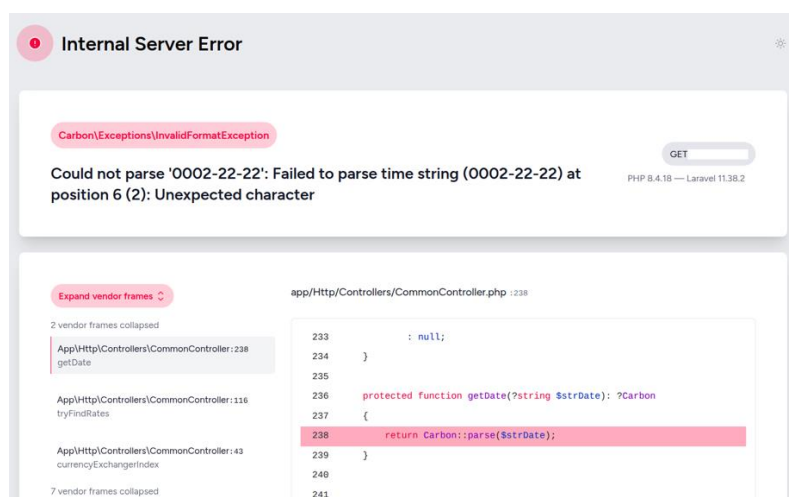


Рисунок 3.8 – Виникнення помилки під час обробки некоректного значення дати

Отриманий результат показав, що через відсутність додаткової перевірки система генерувала виняток під час спроби перетворення переданого значення у формат дати, що призводило до появи помилки 500 (Internal Server Error).

Для усунення виявленого недоліку було реалізовано додаткову перевірку вхідного параметра та механізм його валідації перед подальшою обробкою. Після внесення відповідних змін система перестала генерувати внутрішню помилку та почала коректно обробляти подібні ситуації. Результат роботи системи після внесення змін наведено на рисунку 3.9.



Рисунок 3.9 – Коректна обробка некоректного значення дати після виправлення

Після впровадження додаткової перевірки система почала повертати стандартну сторінку помилки 404 (Not Found), що є коректною поведінкою для

випадків звернення до неіснуючих або недопустимих ресурсів. Таке рішення дозволило підвищити стабільність роботи серверної частини та покращити обробку помилкових запитів користувачів.

Окрему увагу під час тестування було приділено адаптивності користувацького інтерфейсу на мобільних пристроях. У процесі перевірки роботи таблиці курсів валют було виявлено проблему з відображенням назв обмінних пунктів та міст. Якщо назви містили велику кількість символів, вони виходили за межі доступної області та частково перекривали інші елементи інтерфейсу, зокрема значення курсів валют, що ускладнювало перегляд інформації та негативно впливало на зручність використання системи на мобільних пристроях.

Для усунення проблеми було внесено зміни до адаптивної верстки та реалізовано механізм скорочення довгих назв із використанням багатокрапки. Після внесення змін відображення таблиці стало коректним навіть для записів із довгими назвами. У результаті оптимізації інтерфейсу назви обмінних пунктів більше не перекривають інші елементи таблиці, а інформація про курси валют залишається повністю доступною для перегляду користувачем. Відображення таблиці курсів валют до та після внесення змін наведено на рисунку 3.10.

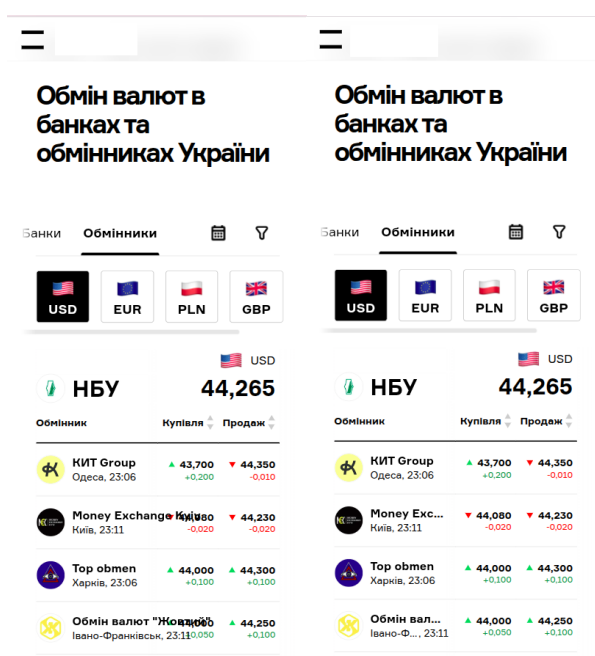


Рисунок 3.10 – Відображення таблиці курсів валют до та після внесення змін

У результаті проведеного тестування було перевірено основні функціональні можливості розробленого сервісу, коректність роботи API, механізмів валідації даних та адаптивності користувацького інтерфейсу. Під час тестування було виявлено та усунуто помилки, пов'язані з обробкою параметрів URL та відображенням даних на мобільних пристроях. Отримані результати підтверджують працездатність та готовність системи до подальшої експлуатації.

3.3 Розробка бази даних

Коректне функціонування сервісу для пошуку оптимальних точок обміну валют потребує зберігання структурованої інформації про країни, міста, валюти, банки, обмінні пункти, їхні фізичні розташування та курси валют. У розроблюваній системі для зберігання основних довідкових даних використовується MySQL, а для накопичення історичних значень курсів валют та виконання аналітичних запитів застосовується ClickHouse. Такий підхід дозволяє розділити оперативні дані та історичні записи, що значно спрощує роботу із системою та підвищує ефективність обробки великих обсягів інформації.

Основними сутностями бази даних є країни, міста, провайдери курсів валют, розташування обмінних пунктів, валюти та записи про валютні курси. Таблиці країн і міст використовуються для географічної прив'язки курсів валют, таблиця провайдерів зберігає інформацію про банки та обмінники, а таблиця розташувань містить координати, адресу, контактні дані та графік роботи окремих точок обміну. Дані про валютні курси пов'язуються з конкретним провайдером, містом та валютною парою.

Для демонстрації структури бази даних та взаємозв'язків між основними сутностями сервісу побудовано ER-діаграму, наведену на рисунку 3.11.

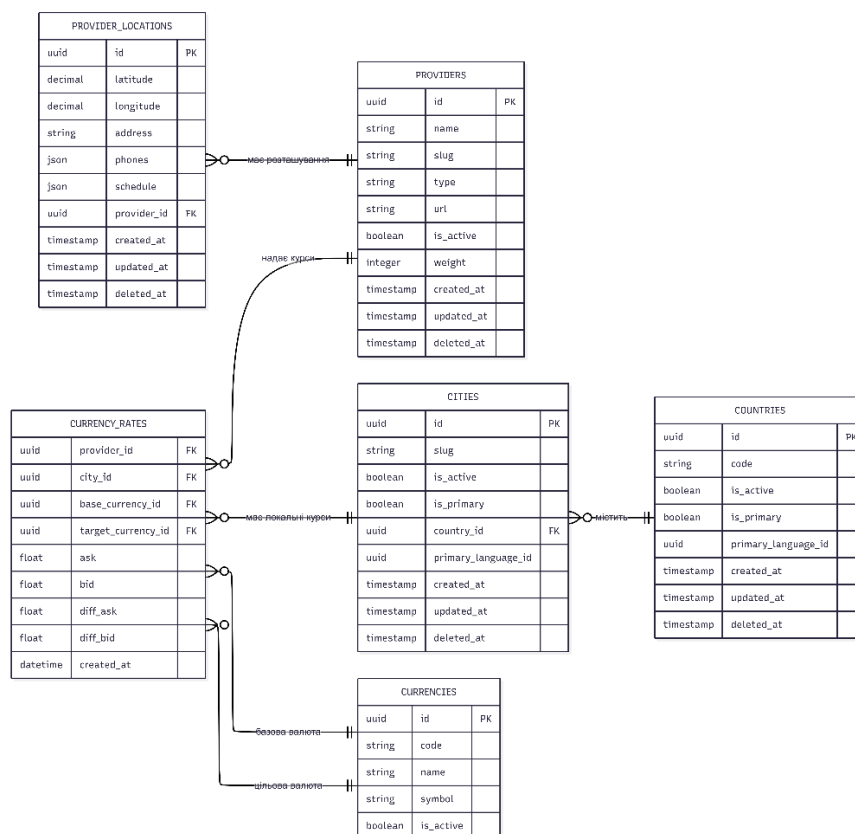


Рисунок 3.11 – ER-діаграма бази даних сервісу

На наведеній ER-діаграмі відображено основні зв'язки між таблицями, які використовуються для роботи сервісу. Країна може містити багато міст, а провайдер може мати декілька фізичних розташувань. Записи про курси валют пов'язуються з провайдером, містом, базовою та цільовою валютою. Використання такої структури дозволяє зберігати курси валют як для загальних банківських значень, так і для локальних обмінних пунктів у конкретних містах.

Реалізація наведеної структури бази даних у системі здійснюється за допомогою механізму міграцій [15]. Міграції дозволяють описувати структуру таблиць у вигляді програмного коду, виконувати створення та зміну таблиць, а також забезпечувати контроль версій схеми бази даних. Фрагмент міграції створення однієї з основних таблиць providers, призначеної для зберігання інформації про банки та обмінники, наведено у лістингу 3.9.

Лістинг 3.9 – Фрагмент міграції створення таблиці провайдерів

```
Schema::create('providers', function (Blueprint $table) {
```

```

$table->uuid('id')->primary();
$table->string('name');
$table->string('slug');
$table->string('type');
$table->string('url')->nullable();
$table->boolean('is_active');
$table->unsignedInteger('weight')->default(0);
$table->timestamps();
$table->softDeletes();
});

```

кінець лістингу 3.9

У наведеному фрагменті міграції створюється таблиця `providers`, у якій зберігаються основні відомості про джерела валютних курсів. Поле `id` використовується як унікальний ідентифікатор запису, поле `name` містить назву банку або обмінника, поле `slug` застосовується для формування адрес сторінок у вебзастосунку, а поле `type` дозволяє розрізнити тип провайдера, наприклад банк або обмінний пункт. Також тут додатково передбачено збереження URL-адреси, статусу активності, ваги для сортування, часових міток та підтримку м'якого видалення записів.

Оскільки одним із завдань системи є відображення фізичних точок обміну валют на карті, для провайдерів передбачено окрему таблицю розташувань, фрагмент якої наведено в лістингу 3.10. Вона дозволяє зберігати координати, адресу, телефони та графік роботи конкретного відділення або обмінного пункту.

Лістинг 3.10 – Фрагмент таблиці розташувань обмінних пунктів

```

Schema::create('provider_locations', function (Blueprint $table) {
    $table->uuid('id')->primary();
    $table->decimal('latitude', 18, 9);
    $table->decimal('longitude', 18, 9);
    $table->string('address')->nullable();
    $table->json('phones')->nullable();
    $table->json('schedule')->nullable();
    $table->uuid('provider_id')->index();
    $table->timestamps();
    $table->softDeletes();

    $table->foreign('provider_id')
        ->references('id')
        ->on('providers');
});

```

кінець лістингу 3.10

У наведеній таблиці `provider_locations` кожен запис пов'язаний із конкретним провайдером через поле `provider_id`. Наявність зовнішнього ключа забезпечує цілісність даних та не дозволяє створювати розташування для неіснуючого банку або обмінного пункту. Поля `latitude` і `longitude` використовуються для відображення точки на карті, а поля `address`, `phones` і `schedule` зберігають додаткову інформацію, необхідну користувачеві під час вибору місця для обміну валют.

Окрему важливу роль у системі відіграє зберігання історичних значень курсів валют. Для цього використовується ClickHouse, оскільки ця система оптимізована для миттєвої роботи з великими обсягами даних. На відміну від довідкової інформації, яка зберігається в MySQL, історія курсів валют постійно накопичується внаслідок регулярного оновлення даних із різних джерел. Структура таблиці для зберігання історичних курсів наведена у лістингу 3.11.

Лістинг 3.11 – Створення таблиці історичних курсів валют у ClickHouse

```
CREATE TABLE currency_rates
(
    provider_id          UUID,
    city_id              Nullable(UUID),
    base_currency_id     UUID,
    target_currency_id   UUID,
    ask                  Float64,
    bid                  Float64,
    diff_ask             Float64,
    diff_bid             Float64,
    latest_diff_ask      Float64,
    latest_diff_bid      Float64,
    created_at           DateTime DEFAULT now()
) ENGINE MergeTree()
  PARTITION BY toYYYYMM(created_at)
  ORDER BY (provider_id, city_id, base_currency_id, target_currency_id,
created_at)
  SETTINGS allow_nullable_key = 1;
```

кінець лістингу 3.11

У наведеній таблиці `currency_rates` реалізований в ClickHouse фіксуються ідентифікатор провайдера, місто, базова та цільова валюта, значення купівлі й продажу, різниця між поточними та попередніми значеннями, а також дата створення запису. Для таблиці використовується рушій MergeTree. Розділення

даних за місяцем через `PARTITION BY toYYYYMM(created_at)` дає змогу оптимізувати роботу з історичними записами, а сортування за провайдером, містом, валютною парою та датою пришвидшує виконання аналітичних запитів.

Для пришвидшення аналізу історичних даних розроблено агреговану таблицю `currency_rates_aggr_daily`, яка зберігає узагальнені значення курсів за день. Її наповнення виконується за допомогою матеріалізованого представлення, що автоматично формує агреговані записи на основі даних із таблиці `currency_rates`. Такий підхід дозволяє зменшити навантаження під час побудови графіків та отримання статистичної інформації за тривалі проміжки часу.

Формування запитів до бази даних у системі здійснюється як за допомогою міграцій, так і через механізми ORM та query builder [16]. Наприклад, під час отримання актуальних курсів валют система формує запит із фільтрацією за базовою та цільовою валютою, містом або типом провайдера. Для історичних даних застосовуються запити до ClickHouse, які дозволяють отримувати вибірки за періодами та будувати графіки динаміки зміни курсів валют.

Таким чином, розроблена структура бази даних забезпечує зберігання основної довідкової інформації, географічну прив'язку точок обміну, накопичення актуальних і історичних курсів валют, а також підтримку аналітичної обробки даних. Поєднання MySQL і ClickHouse дозволило розділити різні типи даних відповідно до їхнього призначення та забезпечити ефективну роботу сервісу при зростанні обсягів інформації.

3.4 SEO інформаційно-комп'ютерної системи

Одним із важливих аспектів розробки вебзастосунку є його оптимізація для пошукових систем. Сервіс для пошуку оптимальних точок обміну валют потребує особливої уваги до SEO, оскільки значна частина користувачів знаходить подібні ресурси через пошукові запити, пов'язані з курсами валют, обмінними пунктами та фінансовою інформацією. Тому під час розробки системи було реалізовано комплекс заходів, спрямованих на покращення

індексації сторінок пошуковими системами, підвищення видимості ресурсу у результатах пошуку та забезпечення швидкого завантаження сторінок.

Для спрощення керування пошуковою оптимізацією в системі реалізовано механізм шаблонів метаданих, який дозволяє автоматично формувати значення метатегів для великої кількості сторінок на основі попередньо визначених шаблонів та змінних параметрів, що забезпечує унікальність метаданих для різних сторінок без необхідності їх ручного налаштування [17].

Приклад налаштування шаблону метаданих для сторінок з таблицею курсів валют наведено на рисунку 3.12.

Мета шаблони / Сторінка з таблицею курсів Зберегти Україна Українська

Ключ	Значення
[%siteName%]	Назва сайту
[%year%]	Поточний рік
[%date%]	Дата
[%country%]	Країна
[%countryGenitive%]	Країна (в родовому відмінку)
[%countryDative%]	Країна (в давальному відмінку)
[%countryAccusative%]	Країна (в знахідному відмінку)
[%countryAblative%]	Країна (в родовому відмінку)
[%countryLocative%]	Країна (в місцевому відмінку)

Meta Title

+ Додати умову

Обмін валют в Україні на [%date%] – актуальні курси та їх динаміка | [%siteName%]

+ Додати варіант

Meta Description

+ Додати умову

Дізнайтеся актуальні курси валют станом на сьогодні в Україні ☒ Порівняйте пропозиції банків та обмінників, готівковий і безготівковий міжбанк курс різних валют світу від [%siteName%]

+ Додати варіант

Рисунок 3.12 – Налаштування шаблонів метаданих

У шаблонах використовуються змінні, значення яких автоматично підставляються під час формування сторінки. Зокрема може використовуватись назва сайту, дата, країна та інші параметри сформовані на основі відображеного контенту, що дозволяє створювати інформативні заголовки та описи сторінок, які

повністю відповідають їхньому змісту та покращують відображення результатів у пошукових системах.

Окрім автоматичного формування метаданих за шаблонами, у системі реалізовано можливість їх індивідуального налаштування для окремих/конкретних сторінок за URL-адресою. Це може використовуватись у випадках, коли необхідно задати унікальний заголовок або опис для конкретної сторінки незалежно від загального шаблону. Приклад налаштування метаданих для окремої сторінки наведено на рисунку 3.13.

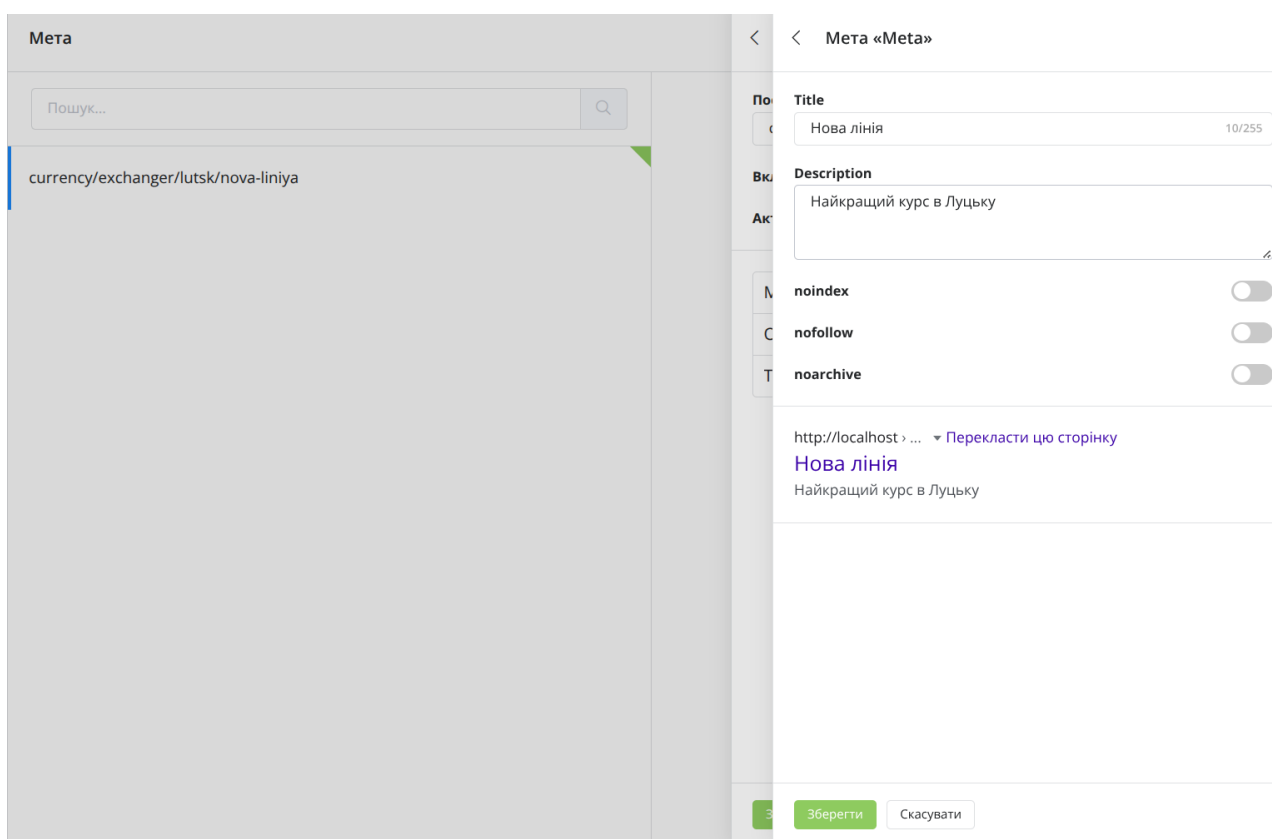


Рисунок 3.13 – Налаштування метаданих для окремого/конкретного URL

Використання двох рівнів налаштування SEO дозволяє поєднати переваги автоматизації та гнучкого керування: більшість сторінок отримує метадані автоматично на основі шаблонів, тоді як для найбільш важливих сторінок можуть створюватися індивідуальні налаштування. Додатково забезпечено формування зрозумілих та оптимізованих для пошукових систем URL-адрес із використанням текстових ідентифікаторів міст та обмінних пунктів, що робить посилання

інформативними як для користувачів, так і для пошукових систем та позитивно впливає на індексацію сторінок і їхню релевантність для пошукових запитів.

Важливим фактором пошукової оптимізації також є швидкість завантаження вебресурсу. Для її забезпечення в системі використовується серверний рендеринг сторінок, оптимізація запитів до бази даних, кешування та завантаження лише необхідної інформації. Ефективність реалізованого програмного рішення була перевірена за допомогою сервісу Google PageSpeed Insights [18], результати роботи якого наведено на рисунку 3.14.

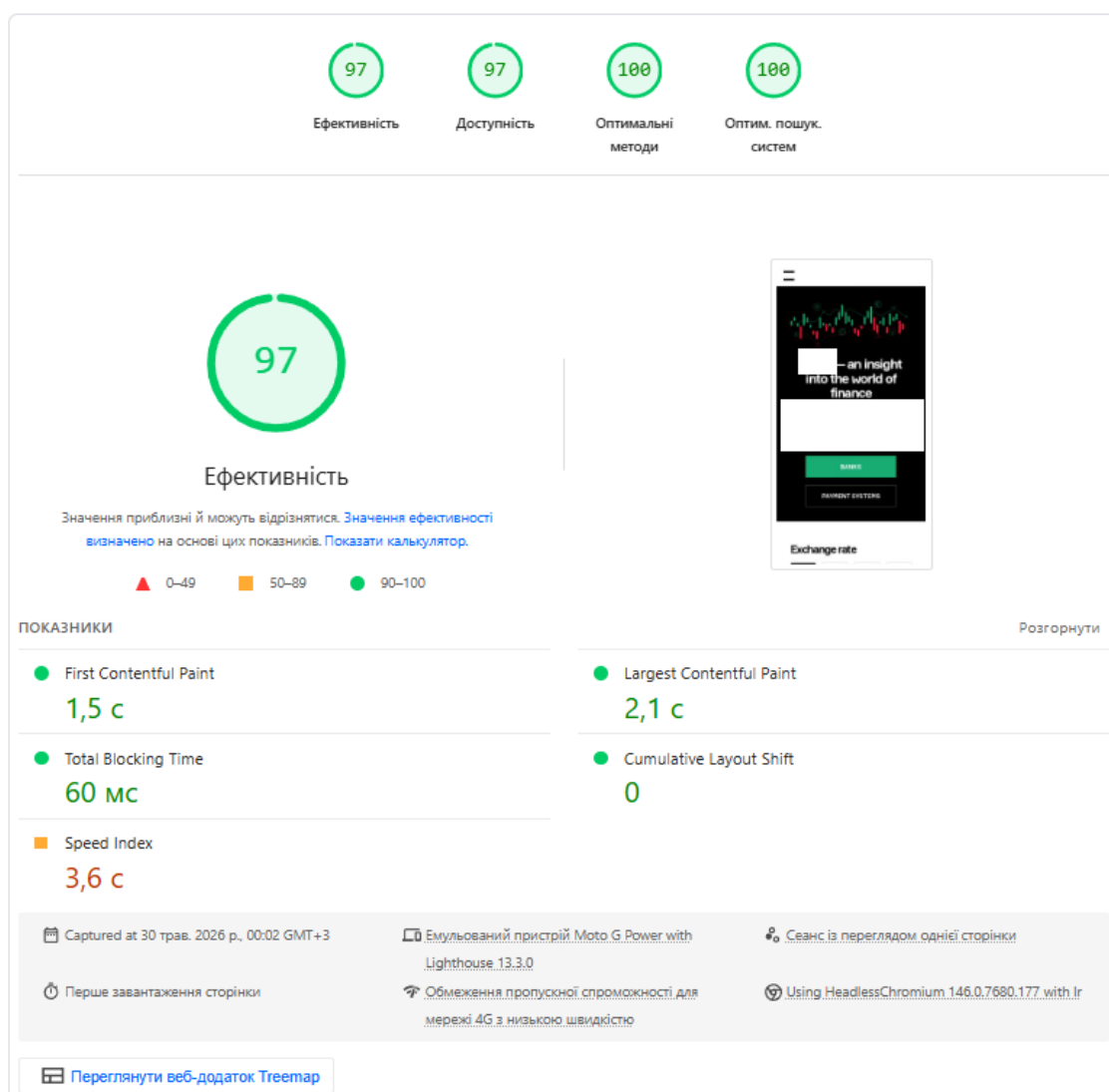


Рисунок 3.14 – Результат аналізу сторінок сервісу від Google PageSpeed Insights

Згідно з отриманими результатами сервіс демонструє високі показники продуктивності, доступності, відповідності сучасним вебстандартам та вимогам

пошукових систем, що свідчить про ефективність використаних технічних рішень та позитивно впливає як на користувацький досвід, так і на позиції ресурсу в результатах пошуку.

Отже, у розробленому сервісі реалізовано комплекс засобів пошукової оптимізації, який включає автоматичне та індивідуальне налаштування метаданих, використання оптимізованих під пошукові системи URL-адрес і забезпечення високої швидкодії сторінок, що сприяє ефективній індексації ресурсу та підвищує його видимість у результатах пошуку.

3.5 Розробка документації для програмного продукту

Важливим етапом створення програмного продукту є підготовка документації, яка забезпечить зручне використання системи кінцевими користувачами та спростить її подальшу підтримку і модернізацію. У документації наведено рекомендації щодо використання системи, визначено вимоги до програмного середовища та описано порядок локального розгортання проєкту для потреб подальшої розробки.

Оскільки розроблений сервіс реалізовано у вигляді вебзастосунку, доступ до його функціональних можливостей здійснюється через веббраузер, де для коректної роботи достатньо використовувати сучасний браузер із підтримкою актуальних вебстандартів, JavaScript та технологій обробки асинхронних запитів, а також мати доступ до мережі Інтернет. Вебзастосунок адаптований для роботи як на персональних комп'ютерах, так і на мобільних пристроях, що забезпечує зручний доступ до інформації про курси валют незалежно від типу використовуваного пристрою.

Під час використання сервісу користувач може переглядати актуальні курси валют банків та обмінних пунктів, здійснювати пошук інформації за містами, аналізувати динаміку зміни курсів валют за різні проміжки часу та переглядати детальну інформацію про окремі точки обміну. Інтерфейс системи побудований

таким чином, щоб основні функції були доступні без додаткового навчання або ознайомлення зі спеціалізованою документацією.

Оскільки система є вебрішенням, її кінцеві користувачі працюють із уже розгорнутим серверним середовищем, тому процедура встановлення програмного продукту для них відсутня. Водночас для подальшої модернізації або внесення змін до функціональних можливостей сервісу може виникнути потреба у локальному розгортанні проєкту розробниками. У такому випадку необхідно встановити програмні залежності серверної та клієнтської частин застосунку, налаштувати параметри підключення до баз даних та виконати підготовку середовища з використанням команд, наведених у лістингу 3.12.

Лістинг 3.12 – Основні команди для розгортання проєкту

```
composer install  
npm install  
php artisan migrate  
npm run build  
php artisan serve
```

кінець лістингу 3.12

Після встановлення необхідних залежностей виконується налаштування конфігураційного файлу, у якому визначаються параметри підключення до MySQL, ClickHouse та інших сервісів, що використовуються системою. Далі виконується запуск серверної частини та збірка клієнтської частини сервісу, після чого вебзастосунок стає доступним для тестування та подальшої розробки.

Таким чином, сформована документація щодо використання програмного продукту, визначає мінімальні вимоги до середовища його експлуатації та описує загальний порядок локального розгортання системи для потреб подальшої підтримки та розвитку сервісу для пошуку оптимальних точок обміну валют.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра досягнуто поставленої мети, яка полягала у розробці сервісу для пошуку оптимальних точок обміну валют з використанням Laravel, React, MySQL, ClickHouse, Redis та Elasticsearch. Розроблене програмне забезпечення забезпечує автоматизований збір, обробку, зберігання та представлення інформації про валютні курси, що відповідає сучасним тенденціям розвитку веборієнтованих інформаційних систем і сервісів аналітичної обробки даних.

У процесі виконання роботи проведено аналіз предметної області, сучасних технологій розробки вебсистем та існуючих сервісів моніторингу валютних курсів. На основі проведеного аналізу визначено функціональні вимоги до програмного продукту та обґрунтовано доцільність створення сервісу для пошуку оптимальних точок обміну валют.

Спроектовано архітектуру інформаційної системи на основі клієнт-серверного підходу із використанням Laravel для реалізації серверної частини та React для створення користувацького інтерфейсу. Обрана архітектура забезпечує масштабованість програмного забезпечення та підтримує незалежний розвиток окремих функціональних модулів.

Розроблено структуру зберігання даних із використанням MySQL для роботи з основними сутностями системи та ClickHouse для зберігання й обробки історичних даних про валютні курси. Запропонована структура забезпечує ефективне зберігання інформації та підтримує виконання аналітичних запитів до великих обсягів даних.

Реалізовано механізм автоматизованого збору та оновлення інформації про валютні курси із використанням програмних модулів парсингу та серверних засобів Laravel. Розроблене рішення забезпечує отримання актуальних даних із зовнішніх джерел, їх нормалізацію та подальше використання в межах системи.

Розроблено вебінтерфейс користувача на основі React, який забезпечує перегляд актуальних курсів валют, пошук інформації про обмінні пункти,

фільтрацію даних за містом та перегляд детальної інформації про доступні точки обміну валют. Реалізований інтерфейс адаптовано для використання як на персональних комп'ютерах, так і на мобільних пристроях.

Реалізовано засоби аналізу історичних даних та візуалізації динаміки зміни валютних курсів із використанням інтерактивних графіків. Це дозволяє користувачам аналізувати тенденції зміни курсів валют за різні проміжки часу та отримувати додаткову інформацію для прийняття фінансових рішень.

Проведено тестування та налагодження розробленої системи, у результаті чого усунено виявлені недоліки та перевірено коректність роботи основних функціональних модулів. Також підготовлено документацію щодо використання та локального розгортання програмного продукту.

Робота виконана в межах досліджень у сфері веброзробки, проектування інформаційних систем та обробки даних із використанням сучасних підходів до побудови клієнт-серверних застосунків, організації сховищ даних і реалізації механізмів пошуку та аналітичної обробки інформації.

Практичне значення роботи полягає у створенні програмного продукту для моніторингу валютного ринку, пошуку оптимальних умов обміну валют та аналізу динаміки зміни курсів. Результати роботи можуть бути використані користувачами, фінансовими установами, аналітичними сервісами та іншими організаціями, діяльність яких пов'язана з обробкою фінансової інформації.

Подальший розвиток роботи може бути пов'язаний із розширенням переліку джерел отримання даних, впровадженням додаткових аналітичних інструментів, використанням методів прогнозування валютних курсів та розширенням функціональних можливостей сервісу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Qi Y. Fintech and the Digital Transformation of Financial Services. Highlights in Business, Economics and Management. 2023. Vol. 8. P. 322-327. URL: <https://doi.org/10.54097/hbem.v8i.7225> (дата звернення: 02.03.2026).
2. Putra F. P. E., et al. Trends and Best Practices in API-Based Web Development Using Laravel and React. Brilliance: Research of Artificial Intelligence. 2025. Vol. 5. № 1. P. 223-233. URL: <https://doi.org/10.47709/brilliance.v5i1.5971> (дата звернення: 02.03.2026).
3. Schulze R., et al. ClickHouse – Lightning Fast Analytics for Everyone. Proceedings of the VLDB Endowment. 2024. Vol. 17. № 12. P. 3731-3744. URL: <https://doi.org/10.14778/3685800.3685802> (дата звернення: 02.03.2026).
4. Installation. Laravel 13.x – The clean stack for Artisans and agents. Laravel. URL: <https://laravel.com/docs/13.x> (дата звернення: 13.04.2026).
5. React. URL: <https://react.dev/> (дата звернення: 13.04.2026).
6. MySQL Documentation. URL: <https://dev.mysql.com/doc/> (дата звернення: 15.04.2026).
7. ClickHouse Docs. Fast Open-Source OLAP DBMS – ClickHouse. URL: <https://clickhouse.com/docs> (дата звернення: 20.04.2026).
8. Redis Docs. URL: <https://redis.io/docs/latest/> (дата звернення: 22.04.2026).
9. Elasticsearch. Elasticsearch Reference. Elastic – The Search AI Company. Elastic. URL: <https://www.elastic.co/docs/reference/elasticsearch> (дата звернення: 22.04.2026).
10. Guzzle, PHP HTTP client – Guzzle Documentation. URL: <https://docs.guzzlephp.org/en/stable/> (дата звернення: 22.04.2026).
11. RTK Query Overview. Redux Toolkit. URL: <https://redux-toolkit.js.org/rtk-query/overview> (дата звернення: 22.04.2026).
12. React-chartjs-2. URL: <https://react-chartjs-2.js.org/> (дата звернення: 02.06.2026).

13. Postman Docs. URL: <https://learning.postman.com/> (дата звернення: 25.04.2026).

14. Send API requests and get response data in Postman. Postman Docs. URL: <https://learning.postman.com/docs/use/send-requests/requests/> (дата звернення: 25.04.2026).

15. Database: Migrations. Laravel 13.x – The clean stack for Artisans and agents. Laravel. URL: <https://laravel.com/docs/13.x/migrations> (дата звернення: 01.05.2026).

16. Eloquent: Getting Started. Laravel 13.x - The clean stack for Artisans and agents. Laravel. URL: <https://laravel.com/docs/13.x/eloquent> (дата звернення: 01.05.2026).

17. How to Write Meta Descriptions. Google Search Central. Documentation. Google for Developers. URL: <https://developers.google.com/search/docs/appearance/snippet> (дата звернення: 04.05.2026).

18. Lighthouse performance scoring. Chrome for Developers. URL: <https://developer.chrome.com/docs/lighthouse/performance/performance-scoring> (дата звернення: 05.05.2026).