

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «ПРОТОТИП КРИПТОВАЛЮТНОГО ПЛАТІЖНОГО ШЛЮЗУ ДЛЯ
E-COMMERCE»

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

_____ Ярослав БАЛАБАНОВ

Виконав:
здобувач вищої освіти
група ІСДМ-62

Ярослав БАЛАБАНОВ

Керівник:
*науковий ступінь,
вчене звання*

Каміла СТОРЧАК
д.т.н., професор

Рецензент:
*науковий ступінь,
вчене звання*

Ім'я, ПРІЗВИЩЕ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Магістр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІСТ

_____ Каміла СТОРЧАК

«_____» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Балабанову Ярославу Анатолійовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Прототип криптовалютного платіжного шлюзу для e-commerce»

керівник кваліфікаційної роботи Каміла СТОРЧАК, д.т.н., професор

(ім'я, ПРИЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «30» жовтня 2025 року № 467.

2. Строк подання кваліфікаційної роботи: 26 грудня 2025 року.

3. Вихідні дані до кваліфікаційної роботи: Ethereum, BNB, USDT, Bitcoin, USD Coin;

бібліотека Axios, фреймворк Spring Boot;

технології HTML, CSS, JavaScript, backend-технології;

HTTP-запити;

науково-технічна література з питань, пов'язаних з криптовалютами технологіями.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 1. Аналіз сучасних криптоплатіжних рішень та технологій.
 2. Дослідження та обґрунтування підходів до створення криптовалютного платіжного рішення для e-commerce.
 3. Frontend-частина криптовалютного платіжного шлюзу.
 4. Backend-частина платіжного шлюзу на Java Spring.
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання: 30 жовтня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	30.10 – 05.11.25	
2	Дослідження сучасних криптоплатіжних рішень та технологій	06.11 – 10.11.25	
3	Визначення технологічних стеків для побудови платіжних систем	11.11 – 17.11.25	
4	Розробка структури SPA-додатку	18.11 – 21.11.25	
5	Проектування REST API	24.11 – 08.12.25	
6	Аналіз вимог до безпеки та надійності криптовалютних транзакцій	09.12 – 12.12.25	
7	Оформлення роботи: вступ, висновки, реферат	15.12 – 19.12.25	
8	Розробка демонстраційних матеріалів	22.12 – 25.12.25	

Здобувач вищої освіти

(підпис)

Ярослав БАЛАБАНОВ

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Каміла СТОРЧАК

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 78 стор., 35 рис., 6 табл., 30 джерел.

Мета роботи - розробка концептуальної моделі прототипу криптовалютного платіжного шлюзу для e-commerce, який забезпечує обробку криптовалютних транзакцій.

Об'єкт дослідження – процес обробки криптовалютних транзакцій у системах електронної комерції.

Предмет дослідження – методи, технології та механізми реалізації криптовалютного платіжного шлюзу для e-commerce.

Короткий зміст роботи: Проведено аналіз сучасних криптоплатіжних рішень та технологій, аналіз вимог до безпеки та надійності криптовалютних транзакцій. Здійснено порівняння технологічних стеків платіжних систем. Обгрунтовано вибір бази даних PostgreSQL для збереження даних про користувачів. Розроблено концептуальну модель прототипу криптовалютного платіжного шлюзу, яка може бути використана як основа для подальшого розвитку повнофункціонального платіжного шлюзу, розширення підтримки різних блокчейн-мереж та інтеграції з реальними e-commerce платформами.

КЛЮЧОВІ СЛОВА: КРИПТОВАЛЮТНА ТРАНЗАКЦІЯ, ТЕХНОЛОГІЧНИЙ СТЕК, E-COMMERCE, КРИПТОВАЛЮТНИЙ ПЛАТІЖНИЙ ШЛЮЗ, ТЕХНОЛОГІЯ, АЛГОРИТМ, СЕРВЕР, ПРОТОКОЛ, REST API, ІНФОРМАЦІЙНА СИСТЕМА, ВАЛІДАЦІЯ ДАНИХ.

ABSTRACT

Text part of the master`s qualification work: 78 pages, 35 pictures, 6 table, 30 sources.

The purpose of the work is to develop a conceptual model of a prototype cryptocurrency payment gateway for e-commerce, which provides processing of cryptocurrency transactions.

Object of research is the process of processing cryptocurrency transactions in e-commerce systems.

Subject of research is methods, technologies, and mechanisms for implementing a cryptocurrency payment gateway for e-commerce.

Summary of the work: Analysis of modern crypto payment solutions and technologies, an analysis of requirements for security and reliability of cryptocurrency transactions was carried out. A comparison of technological stacks of payment systems was carried out. The choice of the PostgreSQL database for storing user data was justified. A conceptual model of a prototype of a cryptocurrency payment gateway was developed, which can be used as a basis for further development of a full-featured payment gateway, expansion of support for various blockchain networks and integration with real e-commerce platforms.

KEYWORDS: CRYPTOCURRENCY TRANSACTION, TECHNOLOGY STACK, E-COMMERCE, CRYPTOCURRENCY PAYMENT GATEWAY, TECHNOLOGY, ALGORITHM, SERVER, PROTOCOL, REST API, INFORMATION SYSTEM, DATA VALIDATION.

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ КРИПТОПЛАТІЖНИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ.....	12
1.1 Стан розвитку криптовалютних платежів у сфері e-commerce.....	12
1.2 Огляд архітектур платіжних шлюзів.....	16
1.3 Технологічні стеки для побудови платіжних систем.....	23
1.4 Аналіз вимог до безпеки та надійності криптовалютних транзакцій.....	28
РОЗДІЛ 2. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ПІДХОДІВ ДО СТВОРЕННЯ КРИПТОВАЛЮТНОГО ПЛАТІЖНОГО РІШЕННЯ ДЛЯ E-COMMERCE.....	31
2.1 Функціональні та нефункціональні вимоги до розробки прототипу криптовалютного платіжного шлюзу.....	31
2.2 Визначення характеристик платіжного рішення для електронної комерції.....	33
2.3 Вибір криптоплатіжного шлюзу для онлайн-бізнесу.....	38
2.4 Процедура розробки криптовалютного платіжного інструменту.....	41
РОЗДІЛ 3. FRONTEND-ЧАСТИНА КРИПТОВАЛЮТНОГО ПЛАТІЖНОГО ШЛЮЗУ.....	48
3.1 Дослідження ринку криптовалютних платіжних систем.....	48
3.2 Структура SPA-додатку.....	55
3.3 Механізм відправлення запитів до REST API.....	58
РОЗДІЛ 4. BACKEND-ЧАСТИНА ПЛАТІЖНОГО ШЛЮЗУ НА JAVA SPRING.....	63
4.1 Проєктування REST API.....	63
4.2 Використання Axios як інструменту для роботи з RESTful API.....	66
4.3 Особливості використання Java Spring фреймворку для розробки backend-частини платформи.....	69
4.4 База даних на PostgreSQL.....	73
4.5 Використання Web3J для налаштування платежів.....	76

ВИСНОВКИ.....	79
ПЕРЕЛІК ПОСИЛАНЬ.....	81
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	84

ВСТУП

Актуальність теми. Стрімкий розвиток цифрових технологій та глобалізація економічних процесів створюють умови для появи нових фінансових інструментів, зокрема криптовалют. Сьогодні криптовалюти набувають дедалі більшого поширення у світі, пропонуючи можливості для миттєвих міжнародних платежів, децентралізованих фінансових сервісів (DeFi), захисту заощаджень від інфляції та створення нових цифрових сервісів у фінансах, освіті, мистецтві та соціальних проєктах.

Попри численні переваги, криптовалюти характеризуються високою волатильністю, відсутністю єдиного регулювання та певними ризиками для користувачів, зокрема можливістю шахрайства та втрати коштів. У різних країнах світу підходи до легалізації та регулювання криптовалют залишаються неоднозначними, що ускладнює їхнє широке впровадження.

У зв'язку з цим виникає потреба у розробці ефективних, безпечних та надійних рішень для обробки криптовалютних платежів у сфері e-commerce. Магістерська кваліфікаційна робота, присвячена створенню прототипу криптовалютного платіжного шлюзу, є актуальною, оскільки поєднує дослідження сучасних технологій блокчейну з практичним впровадженням безпечного інструменту для електронної комерції.

Метою роботи є розробка концептуальної моделі прототипу криптовалютного платіжного шлюзу для e-commerce, який забезпечує обробку криптовалютних транзакцій.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- проаналізувати сучасний стан розвитку криптовалютних платежів у сфері e-commerce та існуючі рішення платіжних шлюзів;
- дослідити архітектури платіжних систем та технологічні стеки, що використовуються для обробки криптовалютних транзакцій;
- розробити концептуальну модель прототипу криптовалютного платіжного шлюзу, що забезпечує інтеграцію з e-commerce платформами;

- надати рекомендації щодо практичного використання прототипу криптовалютного платіжного шлюзу.

Об'єкт дослідження – процес обробки криптовалютних транзакцій у системах електронної комерції.

Предметом дослідження є методи, технології та механізми реалізації криптовалютного платіжного шлюзу для e-commerce.

Методи дослідження: аналіз літературних та інформаційних джерел; системний аналіз; метод порівняння та узагальнення; моделювання.

Наукова новизна отриманих результатів: розроблено концептуальну модель прототипу криптовалютного платіжного шлюзу для e-commerce, що поєднує сучасні технології блокчейну, механізми безпеки транзакцій та оптимальні архітектурні рішення для інтеграції цифрових валют у платіжні системи.

Практична значущість кваліфікаційної магістерської роботи. Концептуальна модель прототипу криптовалютного платіжного шлюзу дозволяє наочно відобразити архітектуру системи, послідовність обробки транзакцій та механізми забезпечення безпеки. Вона буде слугувати основою для подальшої розробки реального програмного продукту, тестування функціональних та технічних рішень, а також адаптації для інтеграції у різні платформи e-commerce.

Апробація результатів та публікації:

Балабанов Я. А. «Аналіз вимог до безпеки та надійності криптовалютних транзакцій». Тези доповіді на III Всеукраїнській науково-технічній конференції «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу». – Київ, 18 листопада 2025 року.

1 АНАЛІЗ СУЧАСНИХ КРИПТОПЛАТІЖНИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ

1.1 Стан розвитку криптовалютних платежів у сфері e-commerce

Криптовалюти - це цифрові активи, які застосовують криптографічні методи для забезпечення безпеки та функціонують у децентралізованих мережах (блокчейн). Вони дозволяють здійснювати peer-to-peer платежі без посередників і виконують роль платіжного засобу, інструменту збереження вартості та об'єкта інвестицій. Незважаючи на високу волатильність і відсутність єдиного регулювання, їх інноваційність та глобальний характер роблять криптовалюти популярними для транзакцій, накопичень та спекулятивних операцій.

Криптовалюти не перебувають під контролем жодного центрального органу, такого як банк або уряд, а керуються мережею користувачів. Транзакції захищені складними криптографічними алгоритмами, що робить їх практично неможливими для підробки і базуються на технології блокчейн - розподіленому та незмінному реєстрі всіх операцій. Переведення коштів прив'язане до адрес гаманців, а не до особистих даних користувачів, хоча повна анонімність не забезпечується. Криптовалюти дозволяють проводити транснаціональні платежі швидко та з меншими комісіями порівняно з традиційними платіжними системами.

Ми звикли користуватися традиційними грошима - гривнями, доларами, євро: розраховуємося ними, відкладаємо, отримуємо зарплату. Всі ці валюти мають спільну характеристику - їх випускає держава або центральний банк. Криптовалюта також є грошовим засобом, проте вона цифрова та незалежна від державних установ і банків. Її не друкують на папері і не карбують у вигляді монет; вона існує виключно в електронній формі у спеціальній мережі, відомій як блокчейн. Система блокчейн має такі переваги, як прозорість та безпека. Всі операції прозорі, тому будь-хто може переконатися у дійсності транзакції. Модифікувати інформацію у вже існуючих блоках майже неможливо - це схоже на спробу одночасно переписати сторінки у мільйонах копій одного зошита [1,2].

Отже, блокчейн дає змогу здійснювати обмін цінностями без участі посередників, таких як банки, державні установи або компанії.

Біткоїн став першою та найвідомішою криптовалютою (рис. 1.1). Його було створено у 2009 році під псевдонімом Сатоші Накамото. Ідея була простою, але революційною - створити цифрові гроші, які функціонують на основі довіри між користувачами, а не під контролем банків.



Рис. 1.1 Криптовалюта біткоїн

На сьогодні існують тисячі інших криптовалют, серед яких:

- Ethereum (Ефіріум) - платформа для створення «розумних контрактів» та децентралізованих додатків;
- BNB (Binance Coin) використовується для сплати комісій на біржі Binance;
- USDT (Tether) - стейблкоїн, вартість якого прив'язана до долара США;
- Solana, Tron, Cardano, Polygon - нові мережі, що підтримують ігри, NFT та фінансові сервіси.

Важливо розуміти, що криптовалюта - це не лише «гроші для інвестицій», а й технологія, яка відкриває можливості для створення нових видів цифрових сервісів.

Досліджуючи особливості криптовалюти, було визначено, для чого її актуально використовувати. Криптовалюту доцільно використовувати для миттєвих міжнародних переказів. Криптовалюта дозволяє надсилати кошти у будь-яку країну світу за кілька хвилин, без участі банків і очікування у чергах. Також криптовалюту доцільно використовувати для захисту від інфляції та нових

фінансових сервісів (DeFi). У деяких країнах криптовалюту використовують для збереження заощаджень у періоди швидкого знецінення національної валюти. Децентралізовані фінанси дозволяють користувачам позичати або обмінювати активи напряму один з одним, без банківських посередників. Блокчейн-технології дають змогу підтверджувати право власності на цифрові об'єкти, такі як картини, музика або документи. Також актуальним є використання криптовалюти в освітніх та соціальних проєктах адже криптотехнології застосовуються для збору благодійних коштів, створення систем електронного голосування та прозорого моніторингу державних витрат [7].

Для використання криптовалюти необхідно мати криптогаманець - спеціальний додаток або пристрій, що зберігає приватні ключі, які виконують роль пароля для доступу до наших коштів. Існують наступні види гаманців:

- онлайн-гаманці (наприклад, Metamask, Trust Wallet), які є зручними у використанні, але потребують обережного поводження;
- апаратні гаманці (Ledger, Trezor) - фізичні пристрої, схожі на флешку, що забезпечують високий рівень безпеки;
- паперові гаманці є найпростішим варіантом, коли ключі просто роздруковані на папері.

У більшості країн криптовалюти не заборонені, проте не мають статусу офіційних грошей. В Україні закон «Про віртуальні активи» визначає криптовалюту як нематеріальне благо, яким можна володіти, передавати та обмінювати. Водночас вона не є платіжним засобом, тому розраховуватися нею в магазинах напряму неможливо. Проте держава поступово впроваджує регуляторні правила для забезпечення безпеки ринку та захисту користувачів.

Технологія криптовалют розвивається надзвичайно швидко. Держави експериментують із власними цифровими валютами (CBDC), компанії впроваджують блокчейн у логістику та документообіг, а університети відкривають курси з цифрових активів. Можливо, вже через кілька років криптогаманець стане таким же звичним, як банківська картка. Проте навіть

сьогодні важливо розуміти принципи цих технологій, щоб користуватися ними безпечно та свідомо.

Криптовалюта – це цифрові гроші, що функціонують без участі банків і держав; елемент технології блокчейн, що забезпечує прозорість і безпеку транзакцій; основа для створення нових рішень у фінансах, освіті, мистецтві та соціальних проєктах. Розуміння принципів роботи криптовалюти допомагає не лише уникати шахрайства, а й краще орієнтуватися у цифровій економіці.

У рамках досліджень, проведених у магістерській кваліфікаційній роботі, були зібрані дані про ціну та ринкову капіталізацію найпопулярніших криптовалют станом на листопад 2025 зокрема для Bitcoin (BTC), Ethereum (ETH), Tether (USDT), Binance Coin (BNB), XRP та USD Coin (USDC) (табл. 1.1).

Таблиця 1.1

Ціна та капіталізація найбільш популярних криптовалют світу станом на листопад 2025 року

№ п/п	Криптовалюта	Приблизна ціна (USD)	Ринкова капіталізація (USD)
1	Bitcoin (BTC)	87342	1.743 трлн.
2	Ethereum (ETH)	3322 (на момент сьогоднішнього індексу)	За даними листопада – близько 338 млрд.
3	Tether (USDT)	1	184.6 млрд.
4	Binance Coin (BNB)	902	115.1 млрд.
5	XRP (XRP)	2+ (точна ціна коливається)	121 – 133 млрд.
6	USD Coin (USDC)	1 (стейблкоїн)	77 – 80 млрд.

Як видно з проведеного дослідження, BTC залишається беззаперечним лідером за капіталізацією - на нього припадає значна частина ринку. ETH як «робоча конструкторська платформа» для смарт-контрактів теж утримує стабільно високу капіталізацію і має високу ліквідність - важливий для DeFi, NFT, dApp. USDT і USDC - стейблкоїни, що використовують як засіб обміну або «простір безпеки» у періоди волатильності ринку. BNB - токен екосистеми біржі, який має свою нішу, не так «цифрове золото», як утилітарну цінність. XRP є одним з лідерів серед

альткоїнів із великою капіталізацією, його часто використовують для платежів/транзакцій, що може бути цікавим для e-commerce та платіжних систем.

Стрімкий розвиток економічних процесів став поштовхом до появи нового виду фінансового активу - криптовалюти. Ігнорувати такі світові зміни вже неможливо. Проаналізувавши історію виникнення та сучасні темпи розвитку криптовалюти, слід зазначити, що у умовах глобалізації цифрові валюти набувають дедалі більшого поширення. Сьогодні існує 941 вид криптовалюти, першою та найвідомішою з яких є Bitcoin. Ставлення країн до використання криптовалюти різняться: одні визнають її як засіб платежу або прирівнюють до електронних грошей, інші ж категорично відмовляються від легалізації через потенційні ризики шахрайства, хакерських атак та можливих махінацій, пов'язаних із відмиванням коштів.

Основними перевагами цифрової валюти є відкритий код алгоритму, що дозволяє добувати її кожному бажаючому, відсутність єдиного цифрового банку та контролю над транзакціями, а також децентралізований характер, який робить криптовалюту практично неможливою для підробки [12, 14].

До основних недоліків криптовалюти належить мінливість вартості через специфіку використання та відсутність регуляторних механізмів, які гарантували б збереження електронних криптогаманців. Незважаючи на численні переваги та широку розповсюдженість, перспективи криптовалют залишаються доволі неоднозначними. Слід зазначити, що поки що у світі немає єдиної стратегії щодо подальшого розвитку цього виду валюти, однак її впровадження та становлення у різних країнах триває і відбувається дедалі активніше.

1.2 Огляд архітектур платіжних шлюзів

Платіжний шлюз є ключовим програмно-технічним компонентом сучасних платіжних систем, що забезпечує взаємодію між клієнтською частиною електронної комерційної платформи, серверною логікою обробки платежів та зовнішніми фінансовими або блокчейн-інфраструктурами. Його основним

завданням є безпечно приймання, перевірка, передавання та підтвердження платіжних транзакцій з подальшим інформуванням сторін про результати виконання операцій.

У контексті електронної комерції платіжний шлюз виконує функцію посередника між покупцем та платіжною мережею (рис. 1.2). Він приймає платіжні дані від клієнта, ініціює транзакцію у відповідній платіжній системі та повертає результат операції до інтернет-магазину. У випадку криптовалютних платежів роль шлюзу додатково включає взаємодію з блокчейн-мережею, генерацію криптовалютних адрес, моніторинг стану транзакцій та обробку підтверджень у розподіленому реєстрі. Таким чином, платіжний шлюз виступає не лише як передавальний елемент, але і як логічний центр обробки платежів, що забезпечує цілісність, надійність та безпеку фінансових операцій.

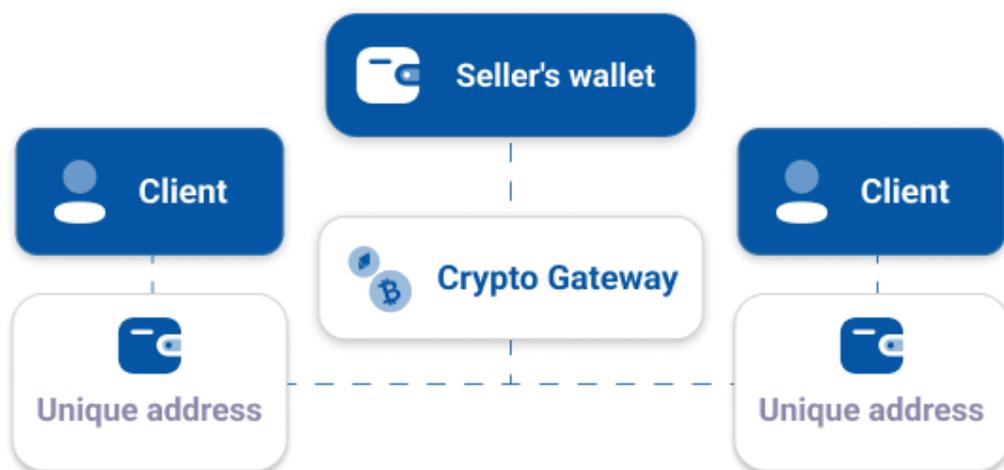


Рис. 1.2 Crypto Gateway

Станом до 2026 року кількість торговців (мерчантів), які приймають криптоплатежі, значно зростає. Дослідження за 2025 рік показують, що тисячі підприємств по всьому світу вже інтегрували криптовалютні методи оплати. Водночас загальна частка криптоплатежів серед усіх онлайн-транзакцій залишається невеликою - за різними оцінками, поки що це менше 1% онлайн-продажів. Крипторозрахунки мають потенціал зростання: не домінують, але займають стабільно зростаючу позицію, особливо серед певних груп бізнесів і споживачів. Також дедалі більше інтернет-магазинів та продавців підключають

криптовалютні платежі й цьому є причини: глобальна доступність, нижчі комісії та зменшення витрат, швидкість та зручність, попит з боку частини аудиторії, наявність інструментів для простішої інтеграції.

Криптовалюти не прив'язані до конкретних країн або банків, що дає змогу приймати платежі від клієнтів з різних куточків світу без потреби у банківській інфраструктурі. Криптовалютні транзакції часто дешевші за традиційні карткові або банківські перекази, особливо у випадках cross-border (між країнами) платежів. Для покупця достатньо гаманця та кількох кліків, що зручно у порівнянні із заповненням банківських реквізитів, перевітками та переказами. Серед користувачів криптовалют є ті, хто вважає оплату криптою зручною та безпечною і готові використовувати її для онлайн-покупок. Платіжний шлюз та сервіси (комерційні провайдери, плагіни для e-commerce) дозволяють спростити бізнесу підключення криптоплатежів без глибоких технічних знань. Це поступово підвищуватиме значимість криптовалют як варіанту оплати для онлайн-торгівлі.

Провівши дослідження, було визначено, що криптовалюти в e-commerce використовують:

- Інтернет-магазини з міжнародною клієнтурою, щоб легко приймати платежі з різних країн, уникаючи валютних і банківських обмежень;
- бізнеси, які орієнтовані на «крипто-аудиторію» - молодих, цифрово підкованих, зацікавлених у Web3, технологіях, які вже тримають криптоактиви;
- магазини або сервіси в регіонах, де банківська/платіжна інфраструктура обмежена або непостійна, так як криптовалюта дає альтернативу;
- платформи з високим рівнем автономії - для малого та середнього бізнесу, фрилансерів, digital-сервісів, де швидкість і мінімальні комісії мають велике значення.

Але є й обмеження, чому криптовалюта ще не перебуває у масовому застосуванні. Наприклад, курс криптовалют може суттєво коливатися. Якщо магазин приймає «як є» без конвертації, це дає валютні ризики. Багато користувачів і бізнесів побоюються шахрайств, не розуміючи, як це працює. Також у багатьох країнах правовий статус криптоплатежів все ще не чіткий, що

створює бар'єри для масового прийняття. Для деяких бізнесів налаштування шлюзу, безпека, конвертація є складними технічними задачами. Причиною обмеження впровадження криптотехнологій є те, що не у всіх є криптогаманець, не всі хочуть або вміють користуватися ним, адже для багатьох зручніше звичні картки або цифрові гаманці. Також є сумніви щодо масштабованості та стабільності - іноді навантаження, підтвердження транзакцій, комісії у мережі блокчейн можуть створювати затримки або додаткові витрати. Саме такі чинники стримують те, щоб криптовалюта стала повсякденним методом для масового e-commerce станом на сьогоднішній день [18].

Дослідивши тенденції ринку криптотехнологій за 2024-2025 рік, можна зробити наступні висновки. Зростає частка стейблкоїнів та «on-chain» рішень як спосіб зробити криптоплатежі більш стабільними та придатними для комерції. Багато великих платформ та процесорів платежів пропонують «crypto-ready» інструменти, що полегшує підключення криптоплатежів для бізнесу. Поступово змінюється сприйняття: криптовалюту вже сприймають не тільки як інвестицію або «спекулятивний актив», але і як робочий інструмент оплати, особливо для міжнародної торгівлі, цифрових послуг, cross-border e-commerce. Очікується, що з розвитком технологій (smart-контракти, blockchain-інфраструктура), а також з розвитком нормативно-правових баз, частка криптоплатежів буде зростати.

Отже, криптовалютні платежі є вже не просто експериментом, а вони стають повноцінною частиною e-commerce-екосистеми. Хоча поки частка таких платежів відносно невелика, тенденції 2024–2025 років демонструють стабільне зростання. Для магазинів з міжнародним охопленням або тих, хто орієнтується на крипто-аудиторію, прийняття криптовалюти може бути конкурентною перевагою.

Типова архітектура платіжного шлюзу у сфері e-commerce складається з таких основних компонентів:

- *клієнтський рівень (frontend)* – веб-інтерфейс або мобільний застосунок, через який користувач ініціює платіж;
- *серверна частина (backend)* – обробляє платіжні запити, керує бізнес-логікою та взаємодіє з іншими компонентами системи;

- *платіжний шлюз* – програмний модуль або сервіс, який відповідає за створення та контроль транзакцій;
- *зовнішня платіжна або блокчейн-інфраструктура* – традиційні платіжні системи або блокчейн-мережі, де безпосередньо відбувається передавання коштів;
- *системи моніторингу та повідомлень* – механізми відстеження стану платежів і інформування учасників транзакції.

У криптовалютних шлюзах додатково використовуються вузли блокчейна або сторонні API-провайдери для взаємодії з розподіленими мережами (рис. 1.3).

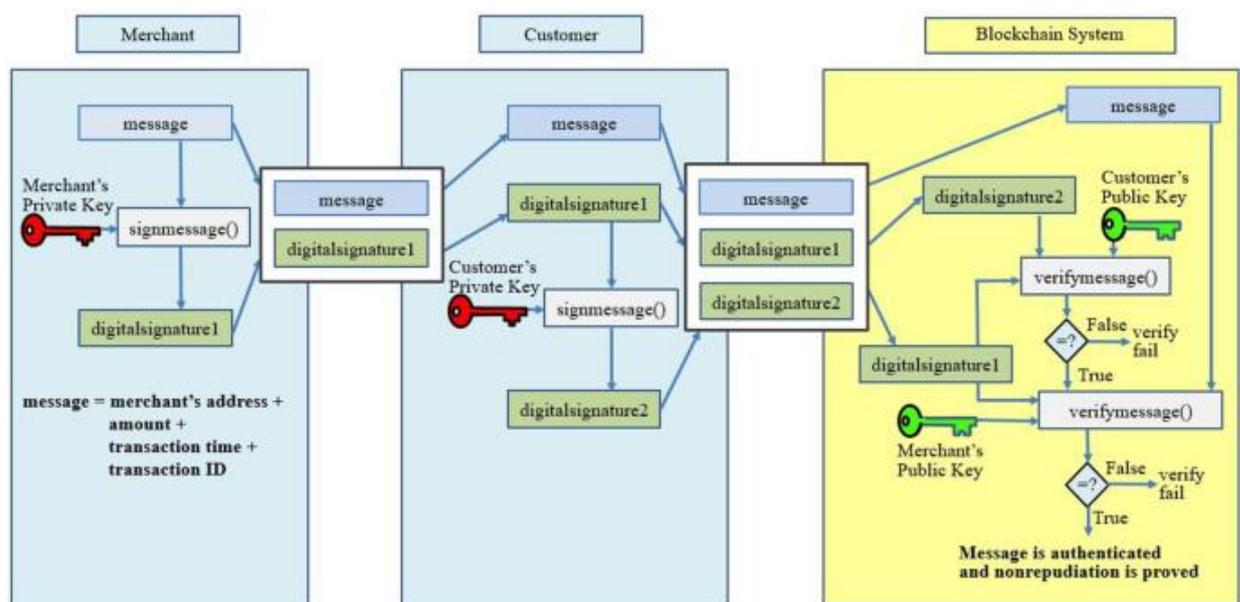


Рис. 1.3 Блок-схема підсистем автентифікації блокчейну

Централізовані архітектури платіжних шлюзів

Централізована архітектура передбачає наявність єдиного керуючого сервісу, який контролює всі етапи обробки платежів. У такій моделі платіжний шлюз відповідає за генерацію платіжних реквізитів, зберігання тимчасових даних, моніторинг транзакцій та обробку підтверджень. Перевагами централізованої архітектури є: відносна простота реалізації; зручність адміністрування та оновлення; централізований контроль безпеки. Недоліками є: наявність єдиної точки відмови; залежність від доступності центрального сервера; обмежена масштабованість у випадку значного зростання навантаження. Незважаючи на

зазначені недоліки, централізовані архітектури залишаються найбільш розповсюдженим підходом у комерційних платіжних системах, зокрема на початкових етапах впровадження.

Децентралізовані та гібридні архітектури

Децентралізований підхід передбачає мінімізацію ролі центрального сервера та активне використання можливостей блокчейн-мережі. У таких системах підтвердження транзакцій відбувається безпосередньо у розподіленому реєстрі, а платіжний шлюз виконує переважно функції ініціалізації та спостереження за станом операції.

Гібридні архітектури поєднують елементи централізованих і децентралізованих моделей. Центральний сервер використовується для управління бізнес-логікою, обробки запитів клієнтів та інтеграції з e-commerce-платформою, тоді як фактичне виконання транзакцій і їх перевірка відбуваються у блокчейн-мережі.

Гібридний підхід вважається найбільш практичним для криптовалютних платіжних шлюзів у сфері електронної комерції, оскільки забезпечує баланс між контрольованістю, безпекою та масштабованістю.

API-орієнтована архітектура платіжних шлюзів

Сучасні платіжні шлюзи будуються за принципами API-орієнтованої архітектури. Взаємодія між e-commerce-платформою та платіжним шлюзом здійснюється через REST або GraphQL API, що дозволяє легко інтегрувати шлюз у різноманітні системи.

Основними функціями API є: створення платіжних запитів; генерація адрес або інвойсів; перевірка статусу транзакцій; отримання повідомлень про підтвердження платежів (webhooks).

Такий підхід підвищує гнучкість системи та спрощує її масштабування.

В рамках магістерської кваліфікаційної роботи визначені архітектурні вимоги до криптовалютних платіжних шлюзів.

Визначено, що при проектуванні архітектури криптоплатіжного шлюзу необхідно враховувати:

- масштабованість - здатність обробляти велику кількість транзакцій;
- відмовостійкість - коректну роботу у випадку часткових збоїв системи;
- безпеку - захист приватних ключів, каналів передавання даних та API-інтерфейсів;
- гнучкість інтеграції - можливість підключення до різних e-commerce-платформ;
- надійність підтвердження транзакцій - коректне відстеження стану операцій у блокчейні.

Саме ці вимоги значною мірою визначатимуть вибір архітектурної моделі для реалізації платіжного шлюзу.

Слід зазначити, що архітектура платіжного шлюзу є визначальним фактором ефективності та безпеки платіжної системи в електронній комерції. Сучасні криптовалютні платіжні шлюзи доцільно організовувати на гібридних, API-орієнтованих архітектурах, які дозволять поєднати централізовану бізнес-логіку з децентралізованою природою блокчейн-технологій. Аналіз зазначених архітектурних підходів є основою для подальшої розробки прототипу криптовалютного платіжного шлюзу.

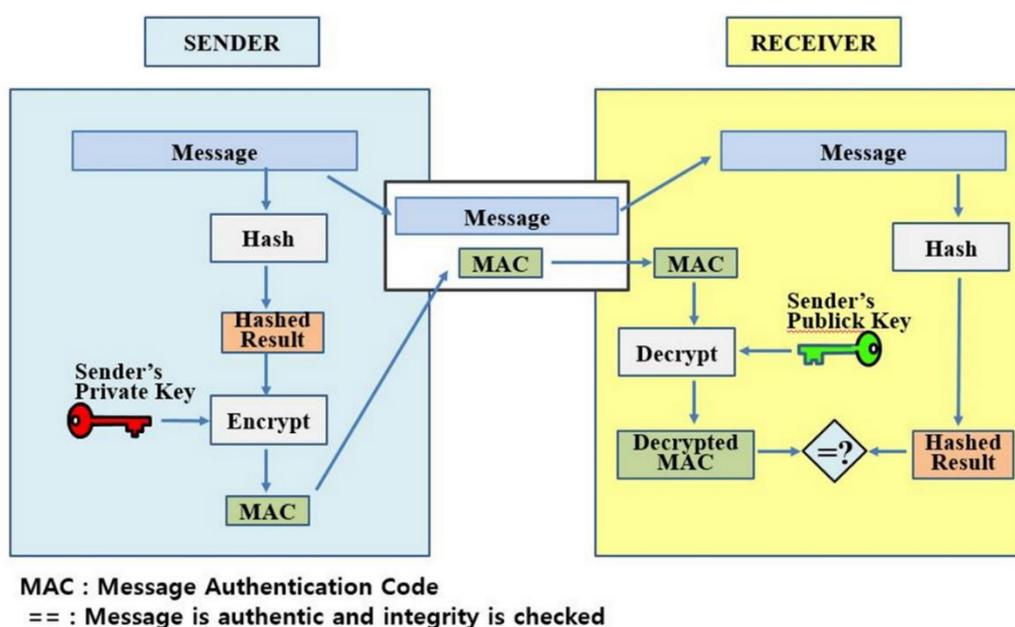


Рис. 1.4 Діаграма автентифікації повідомлень

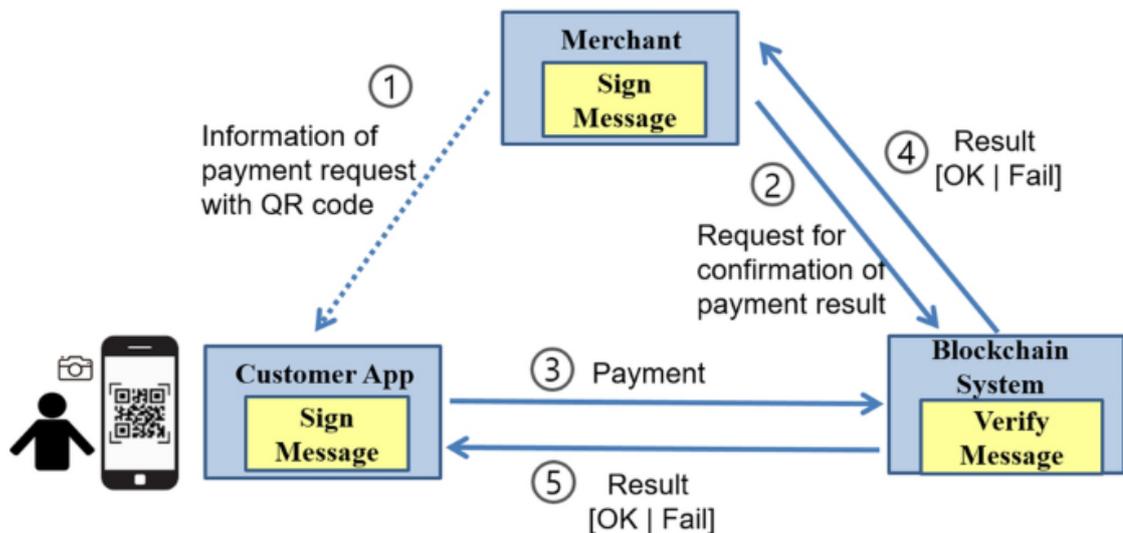


Рис. 1.5 Структура автентифікації

1.3 Технологічні стеки для побудови платіжних систем

Розробка сучасних платіжних систем, зокрема криптовалютних платіжних шлюзів для електронної комерції, вимагає використання комплексного технологічного стеку, який забезпечує надійну обробку транзакцій, масштабованість, безпеку та зручну інтеграцію з зовнішніми системами. Технологічний стек охоплює сукупність програмних і апаратних засобів, що використовуються на різних рівнях архітектури платіжної системи - від користувацького інтерфейсу до взаємодії з блокчейн-мережами.

Технологічний стек платіжної системи включає наступні рівні:

- клієнтський рівень (Frontend) відповідає за взаємодію з користувачем;
- серверний рівень (Backend) реалізує бізнес-логіку та керує платіжними операціями;
- рівень зберігання даних (Data Layer) забезпечує збереження транзакційної та службової інформації;
- інтеграційний рівень (Integration Layer) відповідає за обмін даними із зовнішніми сервісами;
- блокчейн-рівень (Blockchain Layer) забезпечує взаємодію з криптовалютними мережами (рис. 1.6).

Такий підхід дозволяє чітко розмежувати відповідальність компонентів системи та спрощує її масштабування.

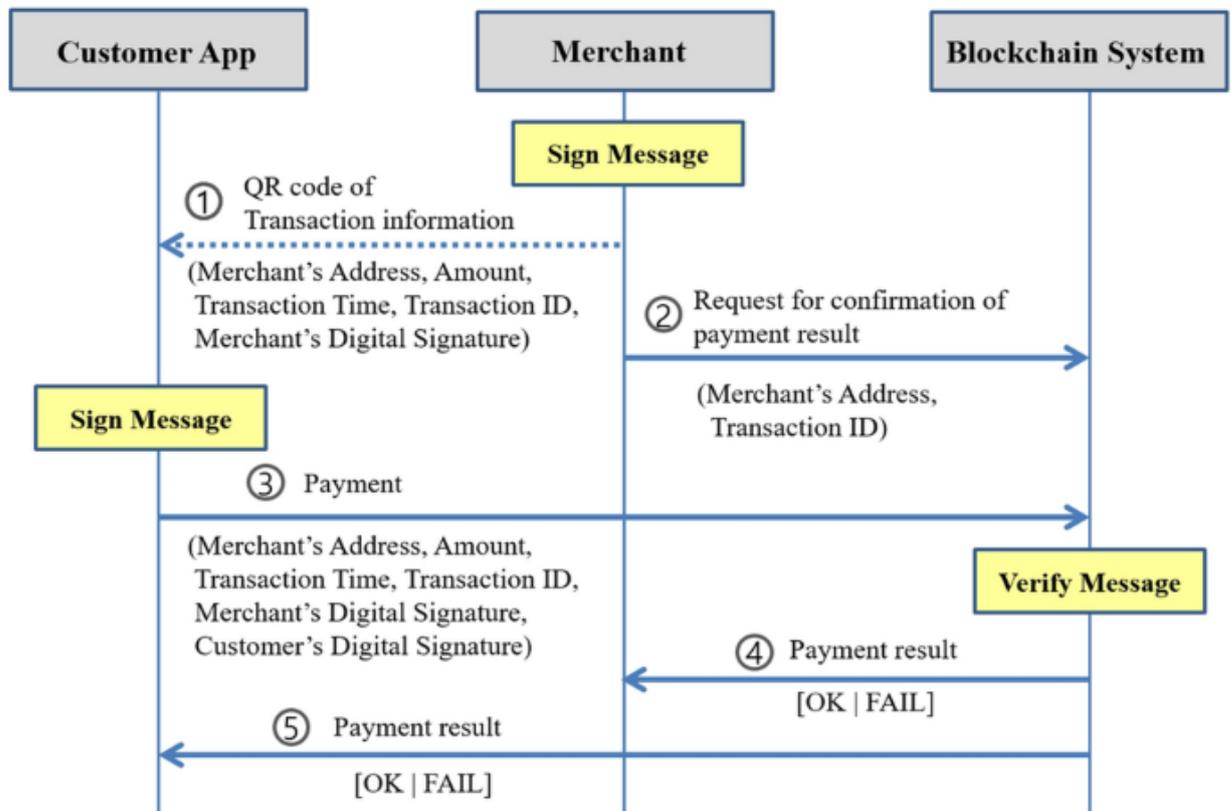


Рис. 1.6 Архітектура платіжної системи електронної комерції блокчейн

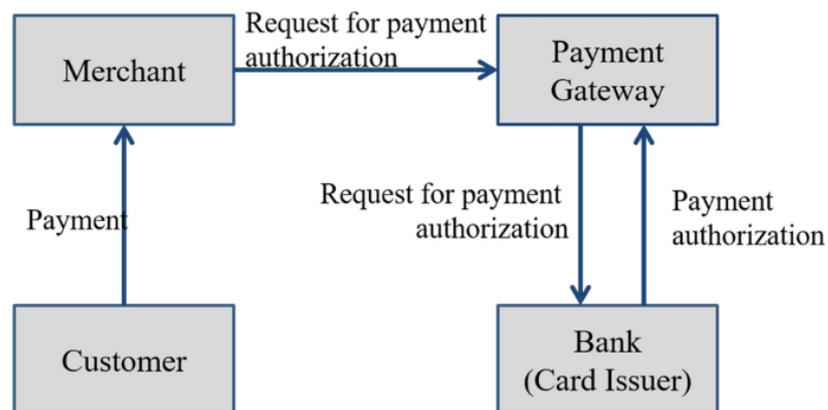


Рис. 1.7 Модель оплати кредитною картою

Frontend-компонент платіжної системи відповідає за взаємодію користувача з e-commerce-платформою на етапі ініціалізації платежу. Основними вимогами до клієнтського рівня є швидкодія, надійність та зручність користування.

Найпоширенішими технологіями є: HTML, CSS, JavaScript - базові технології створення веб-інтерфейсу; JavaScript-фреймворки та бібліотеки (React, Vue, Angular), що дозволяють створювати динамічні та інтерактивні інтерфейси; використання адаптивного дизайну для коректної роботи на різних пристроях. Frontend зазвичай взаємодіє з платіжним шлюзом через HTTP-запити до API, не виконуючи жодних критично важливих обчислень, що підвищує загальний рівень безпеки системи.

Backend-технології. Серверний рівень є основою платіжної системи та відповідає за реалізацію бізнес-логіки. До основних функцій backend-компонента належать: створення та обробка платіжних запитів; керування станами транзакцій; взаємодія з блокчейн-мережами та зовнішніми сервісами; забезпечення автентифікації та авторизації. Для реалізації серверної логіки доцільно застосовувати такі технології: Node.js - завдяки асинхронній обробці запитів і високій продуктивності; Python - простота розробки та велика кількість бібліотек; Java, C# - для корпоративних та високонавантажених систем.

Вибір конкретної мови програмування та фреймворку залежить від вимог до швидкодії, масштабованості та досвіду команди розробки.

Системи зберігання даних. Платіжні системи працюють із великою кількістю транзакційних даних, які необхідно зберігати безпечно та надійно. Для цього доцільно використовувати: реляційні бази даних (PostgreSQL, MySQL) - для збереження транзакцій, користувачів, станів платежів; NoSQL-рішення (MongoDB, Redis) - для кешування або швидкого доступу до тимчасових даних; механізми резервного копіювання та реплікації для підвищення відмовостійкості.

Правильний вибір системи зберігання даних є критично важливим для стабільної роботи платіжного шлюзу.

Інтеграційні технології та API. Сучасні платіжні системи реалізуються з використанням API-орієнтованих підходів. Взаємодія між e-commerce-платформою та платіжним шлюзом здійснюється через: REST API або GraphQL; механізми webhooks для отримання асинхронних повідомлень про стан платежів; стандартизовані формати обміну даними (JSON). Застосування API дозволяє

легко інтегрувати платіжну систему з різними сайтами, мобільними застосунками та зовнішніми сервісами.

Блокчейн-технології та Web3-інструменти. Одним із ключових компонентів криптовалютної платіжної системи є взаємодія з блокчейн-мережами. Для цього доцільно використовувати: JSON-RPC або WebSocket API для обміну даними з блокчейн-вузлами; Web3-бібліотеки (наприклад, для роботи з Ethereum-подібними мережами); сторонні провайдери блокчейн-інфраструктури або власні вузли. Ці інструменти дозволяють створювати адреси для оплати, відстежувати транзакції та отримувати підтвердження їх виконання.

Контейнеризація та хмарні сервіси. Для розгортання платіжних систем широко застосовуються: контейнерні технології (Docker); оркестратори (Kubernetes) для керування навантаженням; хмарні платформи, що дозволяють забезпечити масштабованість і високу доступність сервісу. Використання таких технологій підвищує гнучкість інфраструктури та спрощує підтримку системи.

В рамках магістерської кваліфікаційної роботи визначені критерії вибору технологічного стеку. Визначено, що при виборі технологічного стеку для платіжної системи необхідно враховувати: вимоги до безпеки та відповідності стандартам; необхідний рівень масштабованості; складність інтеграції з e-commerce-платформами; доступність інструментів і документації; можливість подальшого розвитку системи.

Також в рамках дипломної роботи було здійснене порівняння технологічних стеків для побудови платіжних систем, у тому числі криптовалютних шлюзів. Результати порівняння визначені у табл. 1.2.

Таблиця 1.2

Порівняння технологічних стеків платіжних систем

№ п/п	Критерій	Стек 1: Node.js+React+ PostgreSQL	Стек 2: Python (Django/FastAPI)+Vue+ PostgreSQL	Стек 3: Java (Spring) + Angular + Oracle/MySQL
1	Тип системи	Сучасні web- та сурпто-платіжні сервіси	Гнучкі аналітичні та фінансові системи	Корпоративні платіжні рішення
2	Backend	Node.js (Express, NestJS)	Python (Django/FastAPI)	Java (Spring Boot)

Продовження таблиці 1.2

№ п/п	Критерій	Стек 1: Node.js+React+ PostgreSQL	Стек 2: Python (Django/FastAPI)+Vue+ PostgreSQL	Стек 3: Java (Spring) + Angular + Oracle/MySQL
3	Frontend	React	Vue.js	Angular
4	База даних	PostgreSQL, Redis	PostgreSQL, MongoDB	Oracle/MySQL
5	API-архітектура	REST, GraphQL	REST	REST
6	Підтримка blockchain	Web3.js, ethers.js	Web3.py	Web3j
7	Продуктивність	Висока при I/O навантаженні	Середня-висока	Дуже висока
8	Масштабованість	Висока (горизонтальна)	Висока	Висока
9	Складність реалізації	Середня	Низька-середня	Висока
10	Безпека	Висока (JWT, OAuth, TLS)	Висока	Дуже висока
11	Типові сфери застосування	Стартапи, крипто-шлюзи, e- commerce	Фінтех-сервіси, API	Банківські платформи
12	Переваги	Асинхронність, швидкість, велика спільнота	Простота, швидка розробка	Надійність, стабільність
13	Недоліки	Складність для початківців	Нижча продуктивність при великому навантаженні	Висока вартість розробки

Таблиця 1.3

Інфраструктурні та blockchain-технології

№ п/п	Компонент	Варіант	Призначення
1	Контейнеризація	Docker	Ізоляція сервісів
2	Оркестрація	Kubernetes	Масштабування та балансування
3	Хмарні сервіси	AWS/GCP/Azure	Висока доступність
4	Blockchain-доступ	Власний Node/RPC-провайдер	Взаємодія з мережею
5	Стейблкоїни	USDT, USDC	Зменшення волатильності
6	Асинхронність	Message queues	Обробка транзакцій

Визначено, що стек Node.js+React є найбільш популярним для криптовалютних платіжних шлюзів завдяки простій інтеграції з Web3 та високій швидкодії. Python-стек зручний для швидкої розробки прототипу та досліджень. Java-рішення оптимальні для великих, регульованих фінансових систем, але складніші для дипломного проєкту. Для прототипу криптоплатіжного шлюзу в e-commerce найкраще підходить API-орієнтований стек з асинхронною обробкою.

Порівняльний аналіз технологічних стеків показав, що оптимальним для реалізації прототипу криптовалютного платіжного шлюзу є використання сучасного web-орієнтованого стеку з асинхронною обробкою запитів, API-архітектурою та підтримкою інтеграції з блокчейн-мережами.

1.4 Аналіз вимог до безпеки та надійності криптовалютних транзакцій

Криптовалютні транзакції є основним механізмом передавання цифрової вартості у блокчейн-мережах. На відміну від традиційних платіжних систем, вони функціонують у децентралізованому середовищі, де відсутній центральний орган контролю. Це зумовлює підвищені вимоги до безпеки, оскільки будь-яка помилка або компрометація ключів може призвести до незворотної втрати коштів.

Основними вимогами до безпеки криптовалютних транзакцій є: автентичність; цілісність; конфіденційність (часткова); незаперечність; стійкість до зовнішніх і внутрішніх атак.

В основі безпеки криптовалютних транзакцій лежить криптографія з відкритим ключем. Кожен користувач володіє парою ключів: приватний ключ, який використовується для підписання транзакцій; публічний ключ або адреса, який використовується для перевірки підпису.

Цифровий підпис гарантує: автентичність відправника; цілісність даних транзакції; неможливість відмови від авторства (non-repudiation).

Втрата або компрометація приватного ключа призводить до повної втрати контролю над активами, що робить захист ключів критичним аспектом безпеки.

Криптовалюти широко використовують криптографічні хеш-функції (SHA-256, Кессак-256). Вони забезпечують: формування унікального ідентифікатора транзакції; зв'язування блоків у ланцюг (blockchain); захист від підміни даних. Зміна будь-якого байта транзакції призводить до зміни хешу, що робить фальсифікацію практично неможливою без контролю над значною частиною мережі.

Надійність криптовалютних транзакцій значною мірою залежить від безпеки гаманців. Основні вимоги: зберігання приватних ключів у зашифрованому вигляді; використання апаратних гаманців; багатофакторна автентифікація (2FA); захист від фішингових атак. У платіжних шлюзах для e-commerce зазвичай використовуються: hot wallets - для оперативних платежів; cold wallets - для зберігання основного балансу.

Механізм мультипідпису передбачає, що для виконання транзакції потрібно підтвердження кількох ключів. Це значно підвищує стійкість до компрометації одного ключа; внутрішню безпеку платіжних систем; контроль за доступом до коштів.

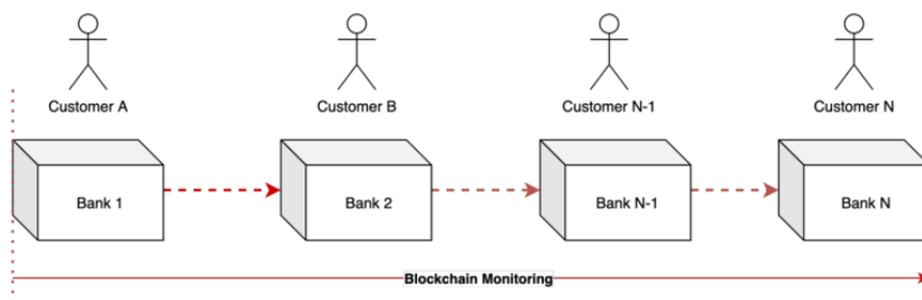


Рис. 1.8 Моніторинг криптовалютних транзакцій у блокчейн мережі

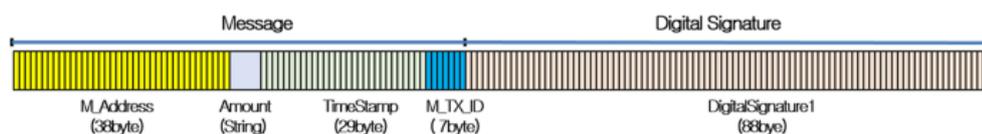


Рис. 1.9 Формат повідомлення продавця та цифровий підпис

Надійність криптовалютних транзакцій безпосередньо пов'язана з алгоритмом консенсусу, який визначає правила підтвердження блоків: Proof of Work (PoW); Proof of Stake (PoS); Delegated Proof of Stake (DPoS). Ці механізми забезпечують

захист від подвійних витрат, узгодженість стану реєстру, стійкість до маніпуляцій.

Подвійне списання (Double Spending) є однією з ключових загроз у цифрових платіжних системах. У криптовалютах воно запобігається за рахунок підтвердження транзакцій у блоках; очікування певної кількості підтверджень; перевірки UTXO або стану облікового запису. Для e-commerce-систем часто застосовуються частково підтвержені платежі з подальшою фіналізацією.

Платіжні шлюзи взаємодіють з блокчейн-мережею та клієнтами через API. Основними вимогами є використання TLS, автентифікація запитів, обмеження частоти запитів, захист від DDoS-атак. Невразливість API є ключовою умовою стабільної роботи платіжної системи.

Для забезпечення надійності та масштабованості застосовується асинхронна обробка черги повідомлень, повторна валідація транзакцій, логування усіх подій. Це зменшує ризик втрати даних при пікових навантаженнях або відмовах.

Криптовалютні платіжні системи повинні відповідати вимогам високої доступності, а саме резервування серверів; горизонтальне масштабування; моніторинг стану вузлів; автоматичне відновлення після збоїв. Використання хмарних інфраструктур дозволяє зменшити ризик простоїв і забезпечити стабільну обробку платежів.

Для e-commerce криптоплатежів також важливими є KYC/AML-політики, логування транзакцій; аудит безпеки, дотримання стандартів інформаційної безпеки. Поєднання технічних та організаційних заходів є необхідною умовою довготривалої надійності платіжного шлюзу.

Проаналізувавши вимоги до безпеки та надійності криптовалютних транзакцій, слід зазначити, що вони базуються на поєднанні криптографічних алгоритмів, мережових механізмів захисту, відмовостійкої архітектури та організаційних заходів. Для побудови криптовалютного платіжного шлюзу в e-commerce необхідно комплексно враховувати всі перелічені вимоги, оскільки порушення будь-якого з компонентів може призвести до фінансових втрат і зниження довіри користувачів.

2 ДОСЛІДЖЕННЯ ТА ОБГРУНТУВАННЯ ПІДХОДІВ ДО СТВОРЕННЯ КРИПТОВАЛЮТНОГО ПЛАТІЖНОГО РІШЕННЯ ДЛЯ E-COMMERCE

2.1 Функціональні та нефункціональні вимоги до розробки прототипу криптовалютного платіжного шлюзу

Визначення функціональних та нефункціональних вимог у магістерській кваліфікаційній роботі має критичне значення, бо воно визначає, що система має робити і якою має бути. Функціональні вимоги визначають, що система має робити (обробка платежів, генерація адрес, вивід коштів, інтеграція з API e-commerce). Нефункціональні вимоги визначають як система має працювати (швидкість транзакцій, безпека, масштабованість, надійність, відповідність регуляторним нормам).

Функціональні вимоги описують конкретні дії, які система повинна виконувати. Також функціональні вимоги дають чітке уявлення, які функції реалізовувати першочергово, дозволяють перевірити, чи система робить те, що від неї очікують користувачі, служать основою для планування розробки та тестування (QA). Для криптоплатіжного шлюзу – це прийом платежів у BTC, ETH, генерація адрес для транзакцій, виведення коштів на фіатний рахунок.

Нефункціональні вимоги описують якість системи, а не конкретні дії, забезпечують надійність, безпеку, швидкодію, масштабованість системи, а також допомагають уникнути проблем, пов'язаних із великим навантаженням або регуляторними обмеженнями та служать критерієм успіху для оцінки готової системи. Для криптоплатіжного шлюзу транзакції мають підтверджуватися за кілька секунд, дані користувачів шифруються, платформа повинна мати здатність обробляти одночасно 1000+ запитів без збоїв.

Функціональні вимоги визначають конкретні дії, які система має виконувати:

а) прийом платежів у криптовалюті (підтримка основних криптовалют: Bitcoin (BTC), Ethereum (ETH), USDT, USDC; можливість прийому платежів на різних блокчейнах);

б) генерацію унікальних адрес для платежів (створення адреси для кожного замовлення або користувача; можливість відстеження транзакцій за кожною адресою);

в) конвертацію криптовалюти у фіат або інші цифрові активи (миттєву конвертацію за поточним курсом; підтримку популярних валют для виведення: USD, EUR, GBP);

г) інтеграцію з e-commerce платформами (використання API для взаємодії з онлайн-магазинами; плагіни для популярних CMS (WooCommerce, OpenCart, Shopify));

д) обробку транзакцій (відправку запитів до RESTful API; отримання відповідей сервера та оновлення стану операцій);

е) управління користувачами та мерчантами (реєстрацію та аутентифікацію користувачів; перегляд історії платежів, балансів та статусів замовлень);

є) адміністративні функції (можливість відстеження помилок; модерацію транзакцій у випадку проблем).

Нефункціональні вимоги - це характеристики якості та обмеження системи. Визначимо ці характеристики:

а) безпека (шифрування даних (HTTPS, SSL/TLS); двофакторна аутентифікація (2FA); захист від шахрайських транзакцій та чарджбеків);

б) продуктивність (обробка великої кількості одночасних транзакцій; мінімальна затримка при перевірці та підтвердженні платежів);

в) надійність і відмовостійкість (використання механізмів відновлення після збоїв; логи всіх транзакцій для аудиту та відновлення даних);

г) масштабованість (можливість додавання нових криптовалют та блокчейнів; збільшення кількості користувачів та магазинів без зниження продуктивності);

д) зручність користування (інтуїтивно зрозумілий інтерфейс для користувачів та адміністраторів; адаптивний дизайн для мобільних пристроїв та десктопів);

е) сумісність (підтримка різних браузерів та платформ; інтеграція з REST API та базою даних PostgreSQL);

є) регуляторні вимоги (відповідність стандартам AML/KYC; можливість роботи згідно з ліцензіями MSB, EMI, VASP, FCA).

2.2 Визначення характеристик платіжного рішення для електронної комерції

Для використання криптовалюти в електронній комерції спочатку потрібно виконати інтеграцію криптовалютного платіжного шлюзу з платформою електронної комерції. Після впровадження відповідного програмного рішення покупці отримують можливість обирати криптовалюту як форму розрахунку на етапі оформлення покупки. Система автоматично генерує унікальну адресу для здійснення платежу, а після підтвердження транзакції отримані кошти зараховуються на гаманець продавця. У рамках дипломної роботи визначені характеристики платіжного рішення у криптовалюті для електронної комерції.

Мультивалютна підтримка. Система має підтримувати кілька криптовалют, забезпечуючи користувачам широкий вибір способів оплати. Зокрема, розроблений криптовалютний платіжний шлюз повинен надавати можливість здійснення розрахунків у сфері електронної комерції з використанням цифрових активів (рис. 2.1). Крім того, мультивалютна підтримка передбачає коректну роботу з різними блокчейн-мережами, зокрема обробку транзакцій, формування адрес для платежів та відстеження їх статусів у режимі реального часу. Це дозволяє підвищити зручність використання платіжного шлюзу для міжнародних клієнтів, зменшити залежність від окремих платіжних інструментів і забезпечити гнучкість системи в умовах динамічного розвитку ринку криптовалют.

Повна інтеграція. Платіжне рішення повинно бути легко інтегроване з веб-сайтом або платформою електронної комерції. Запропоноване рішення повинне бути встановлене на сервери й процес має бути запущений на виконання протягом одного дня. При цьому інтеграція має здійснюватися за допомогою стандартизованого REST API або готових плагінів, що мінімізує потребу у додаткових доопрацюваннях з боку мерчанта.

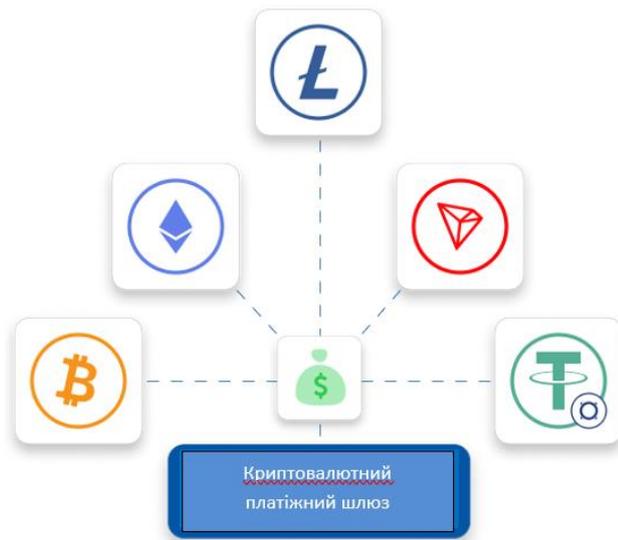


Рис. 2.1 Мультивалютна підтримка

Безпечні транзакції. Рішення повинне забезпечувати безпечний процес оплати для мерчантов та покупців. Технологія криптовалютного платіжного шлюзу повинна забезпечувати повну анонімність угоди сторін. Має бути забезпечене виявлення транзакцій, які надходять з підозрілих рахунків, і розміщення їх у карантин (рис. 2.2).

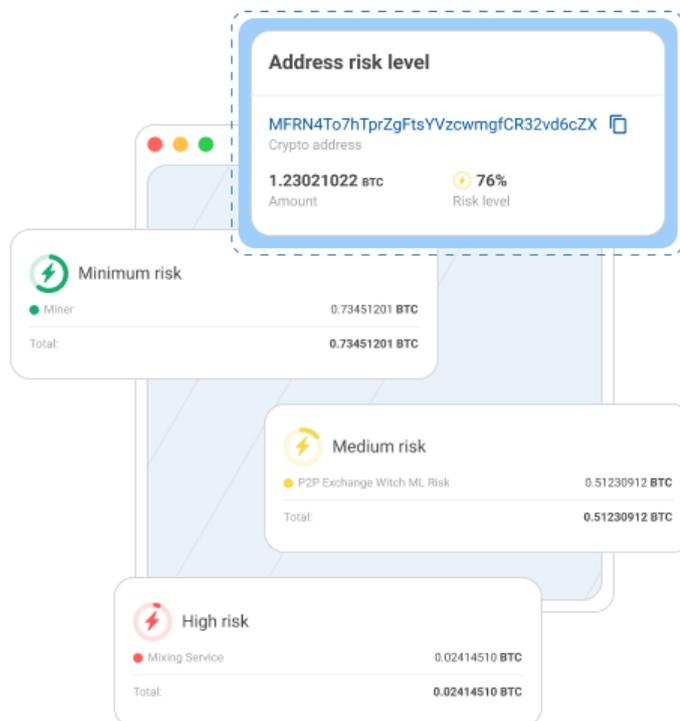


Рис. 2.2 Безпечні транзакції

Миттєве підтвердження платежу. Рішення повинне забезпечувати миттєве підтвердження платежа покупцю та мерчанту. Необхідно уникати затримок в обробці замовлень. Підтвердження криптовалютного платіжного шлюзу повинне відбуватися миттєво за транзакцією (рис. 2.3).

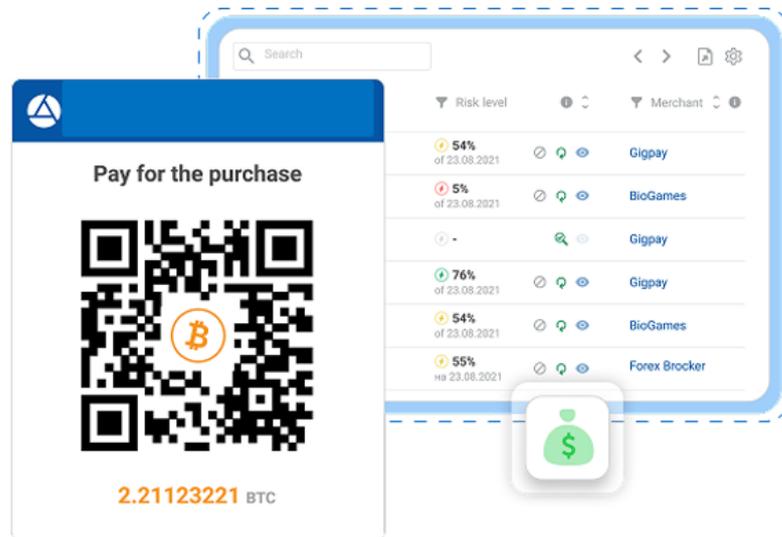


Рис. 2.3 Миттєве підтвердження платежу

Низькі комісії за транзакції. Запропоноване рішення має забезпечувати мінімальні транзакційні комісії, що робить його економічно вигідним як для клієнтів, так і для мерчантів. Зокрема, у системі передбачено інструмент автоматичного виведення коштів у періоди знижених комісій у блокчейн-мережі.

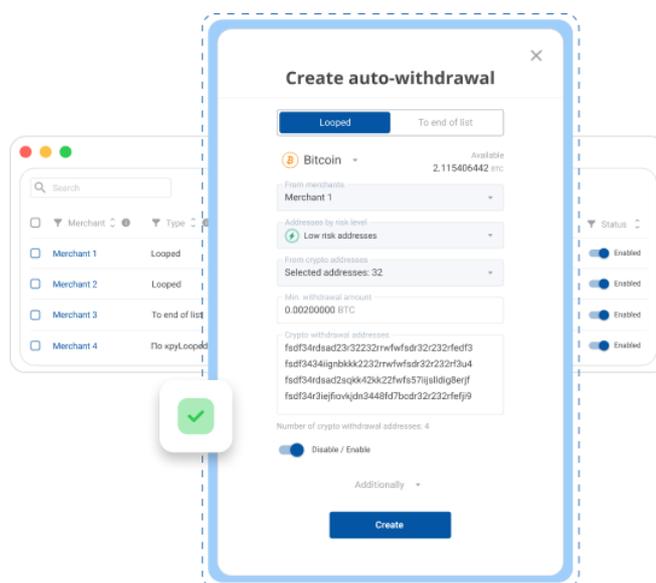


Рис. 2.4 Низькі комісії за транзакції

Зручний інтерфейс. Рішення повинне мати зручний інтерфейс, в якому легко орієнтуватися. Запропонований криптовалютний платіжний шлюз має простий інтерфейс, який є зручним для користувача. В результаті функціональність обліку шлюзу є простою для розуміння та простою у використанні (рис. 2.5).

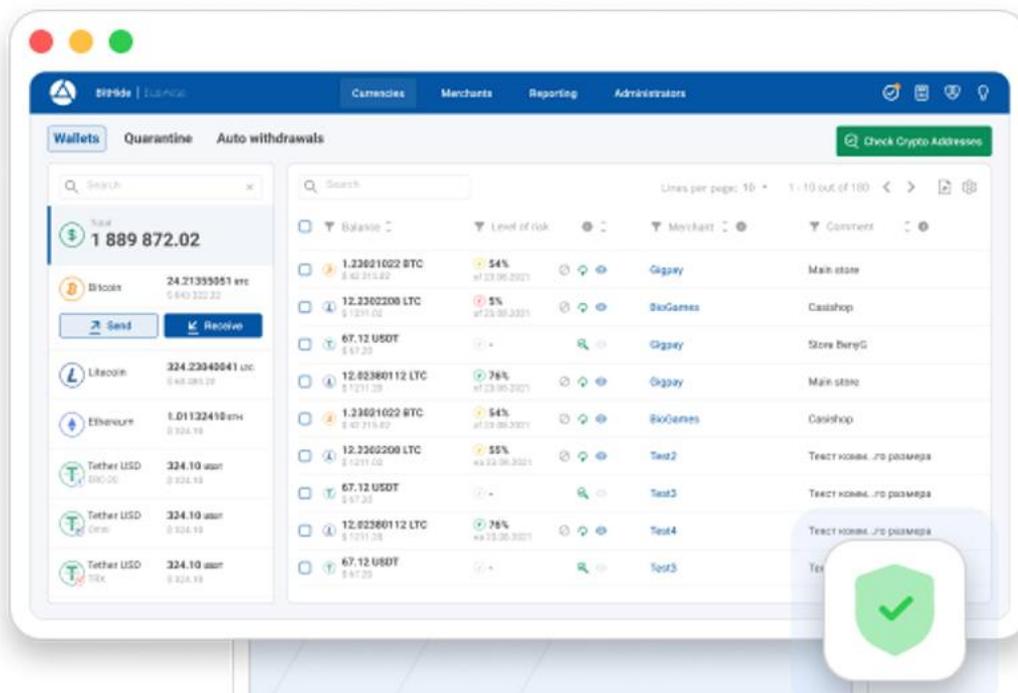


Рис. 2.5 Інтерфейс криптовалютного платіжного шлюзу

Служба підтримки. Рішення повинне забезпечувати надійну підтримку клієнтів для вирішення проблем, які виникають у процесі. У випадку проблем з оплатою клієнт швидко отримує допомогу (рис. 2.6).

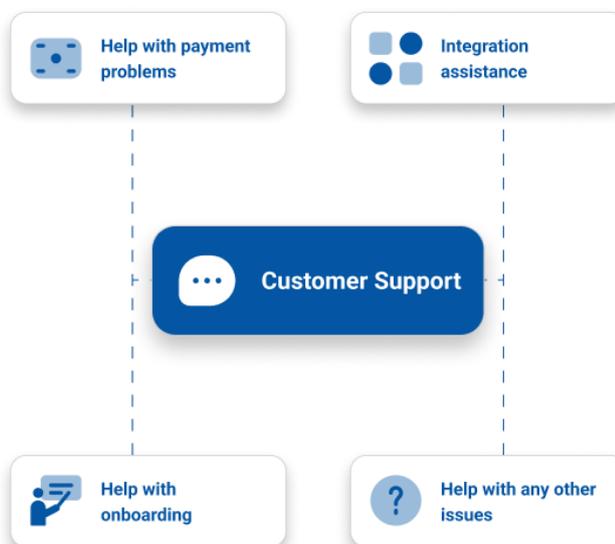


Рис. 2.6 Служба підтримки

Станом на сьогоднішній день тенденція криптовалютної електронної комерції знаходиться на підйомі. Визначимо основні переваги цього способу оплати, а це:

- більш низька комісія за транзакції адже традиційні методи, такі як кредитні карти коштують дорожче;
- більш швидкі транзакції. Криптовалютні транзакції оброблюються швидше, ніж традиційні способи оплати. Обробка банківських переказів може зайняти декілька днів;
- підвищена безпека – криптовалюта пропонує підвищену безпеку. Це виконується за допомогою шифрування та децентралізованих мереж;
- глобальна доступність, так як криптовалюти можна використовувати для транзакцій у всьому світі;
- покращена конфіденційність клієнтів. Криптовалюти пропонують підвищену конфіденційність та анонімність. Це пов'язано з тим, що транзакції не вимагають персональної інформації, такої як дані кредитної карти;
- розширення можливостей продажів. Приймаючи криптовалюти, підприємства електронної комерції можуть вийти на новий ринок технічно підкованих клієнтів, які вважають за краще використовувати цифрові валюти для транзакцій.

Рис. 2.7 відображає процес прийому платежів у криптовалюті.



Рис. 2.7 Процес прийому платежів у криптовалюті

Для того, щоб приймати платежі у криптовалюти, необхідно:

1) Обрати провайдера криптоплатежів. Потрібно провести попередній аналіз і підібрати сервіс, який відповідатиме вимогам бізнесу. Пріоритетом може бути високий рівень безпеки.

2) Створити криптовалютний гаманець. Для приймання оплат необхідно налаштувати відповідний гаманець. У сервісі криптовалютного платіжного шлюзу можна створювати необмежену кількість дочірніх гаманців, що зручно для приймання платежів з кількох проєктів або сервісів.

3) Інтегрувати платіжний шлюз. Бажано, щоб підключення криптошлюзу не вимагало складних технічних робіт. За потреби можна звернутися до технічної підтримки, яка допоможе виконати інтеграцію в найкоротші терміни без залучення сторонніх фахівців.

4) Відобразити способи оплати. Потрібно додати можливість оплати криптовалютою на сторінці оформлення замовлення, зокрема посилання на гаманець.

5) Приймати платежі. Після здійснення клієнтом криптоплатежу система буде автоматично обробляти транзакцію та зараховувати кошти на гаманець продавця.

Щоб інтегрувати криптовалютні платежі на свій веб-сайт, необхідно вибрати шлюз, який підтримує криптовалюту і створити обліковий запис. Потім стає можливим згенерувати кнопку оплати або ключ API, щоб додати їх на сторінку оформлення замовлення веб-сайту, щоб клієнти могли сплачувати покупки криптовалютою. Як тільки клієнт виконає оплату, платіжний шлюз буде обробляти транзакцію. Нарешті, він внесе кошти у вказаний криптовалютний гаманець. Потрібно, щоб криптовалютний платіжний шлюз було легко включати у поточні процеси.

2.3 Вибір криптоплатіжного шлюзу для онлайн-бізнесу

Глобальний ринок електронної комерції стрімко зростає, а криптовалютні розрахунки вже давно вийшли за межі вузької ніші й перетворилися на важливу

складову конкурентоспроможного онлайн-бізнесу. Вони дозволяють зменшити комісійні витрати та залучати покупців з усього світу, забезпечуючи швидкий, надійний і універсальний прийом платежів. Водночас постає питання: як серед великої кількості рішень обрати оптимальний криптовалютний платіжний шлюз, наприклад, для власного інтернет-магазину.



Рис. 2.8 Crypto Payment Gateway

Криптовалютний платіжний шлюз - це онлайн-сервіс, що забезпечує приймання платежів у криптовалюті (Bitcoin, Ethereum, Tron, USDT, USDC та інші) з можливістю автоматичної конвертації у традиційні валюти (USD, EUR, GBP). За принципом роботи він подібний до традиційних систем інтернет-еквайрингу, однак працює на основі блокчейн-технологій, що дозволяє здійснювати платежі швидше, з меншими комісіями та підвищеним рівнем захисту.

Криптоплатіжний шлюз є ефективним для онлайн-бізнесу з наступних причин:

- низькі комісії – дешевше карток та банківських переказів;
- глобальне охоплення – прийом платежів без обмежень по країнам;
- безпека – відсутність чарджбеків, захист від шахрайства;
- швидкість – транзакції закінчуються за хвилини, навіть під час міжнародних розрахунків.

Визначимо основні критерії вибору криптовалютного платіжного шлюзу.

а) *Підтримувані криптовалюти та мережі.* Чим більший перелік доступних валют і блокчейн-мереж, тим універсальніше рішення. Оптимальним є підтримка

Bitcoin, Ethereum, стейблкоїнів USDT/USDC, а також мережі Tron, яка забезпечує швидкі й недорогі перекази.

б) *Механізми виплат.* Одні платіжні шлюзи працюють виключно за схемою crypto-to-crypto, інші надають можливість оперативної конвертації у фіатні валюти. Перевагою буде, якщо платформа криптовалютного платіжного шлюзу буде поєднувати обидва підходи.

в) *Рівень комісій.* Середні комісійні витрати становлять приблизно 0,5–2% з кожної транзакції, додатково можуть нараховуватися збори за виведення коштів, обмін та мережеві операції. Перед впровадженням сервісу доцільно ознайомитися з тарифами та розрахунковими інструментами провайдера.

г) *Варіанти інтеграції.* Сучасний криптоплатіжний шлюз має підтримувати: плагіни для WooCommerce; плагіни для OpenCart; індивідуальні рішення під специфічні бізнес-потреби. Бажано, щоб криптовалютний шлюз надавав всі перелічені можливості.

д) *Безпека та регуляторна відповідність.* Обов'язковими вимогами є використання криптографічного захисту, двофакторної автентифікації, а також дотримання процедур KYC/AML для мінімізації ризиків.

е) *Швидкодія системи.* Важливим показником є оперативне підтвердження транзакцій і оновлення їх статусів у режимі реального часу.

є) *Робота з high-risk сегментами.* Перевагою буде підтримка високоризикових напрямів бізнесу. Шлюз має бути орієнтованим на такі сегменти ринку та пропонувати адаптивний і гнучкий процес підключення.

Помилки, яких потрібно уникати:

- а) Вибір провайдера тільки по ціні.
- б) Ігнорування вимог регуляторів.
- в) Відсутність тестів інтеграції.
- г) Слабка або повільна підтримка.

Кращий криптовалютний шлюз – це баланс швидкості, безпеки, надійності, відповідності вимогам та зручності для користувачів (рис. 2.9).

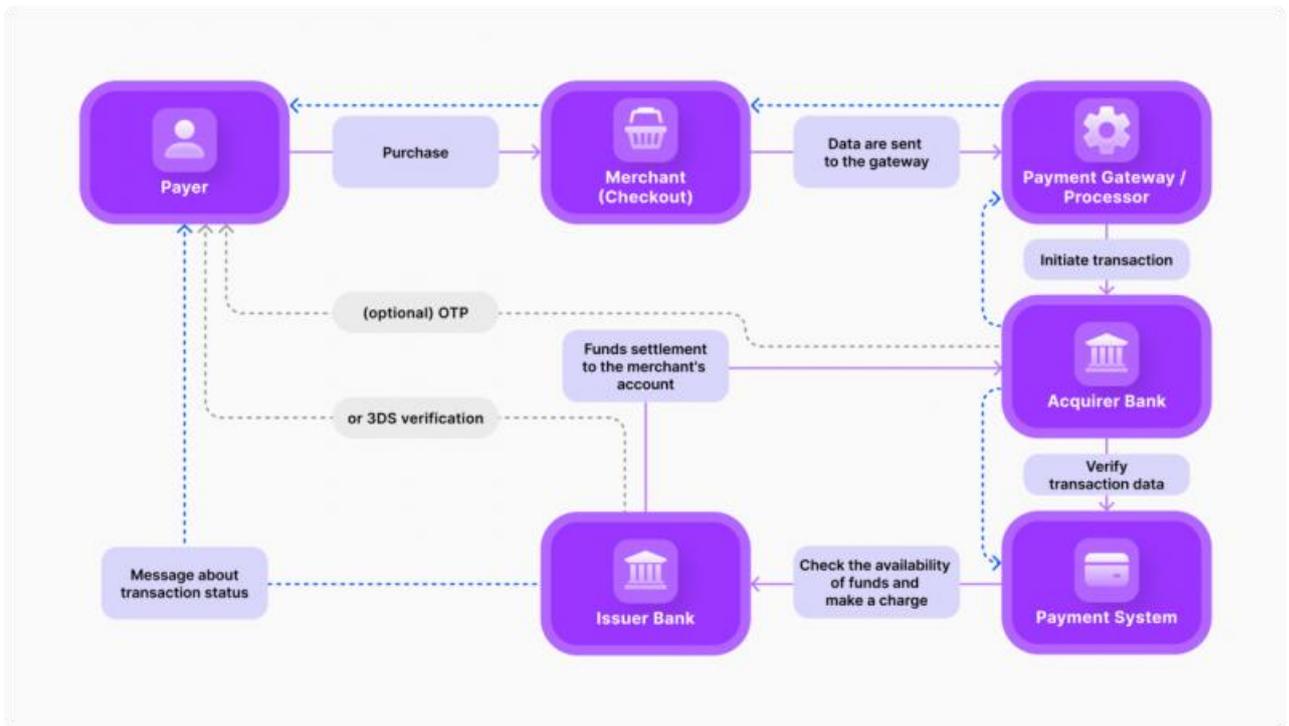


Рис. 2.9 Процесинг платіжної операції

2.4 Процедура розробки криптовалютного платіжного інструменту

Криптовалютний платіжний шлюз (Crypto Payment Gateway) - це інструмент, що дає змогу компаніям приймати оплату в різних цифрових валютах, за потреби автоматично конвертувати їх у фіатні гроші або інші криптоактиви, а також здійснювати подальший вивід коштів на банківські рахунки або криптогаманці. Подібні рішення стають особливо актуальними в галузях, де традиційні платіжні системи не забезпечують достатньої швидкості, безпеки або гнучкості. Йдеться про e-commerce, SaaS, індустрію онлайн-ігор та міжнародні фінансові операції. У цих секторах використання криптовалют дає суттєві переваги - мінімальні комісії, швидке проведення платежів і можливість здійснювати міжнародні транзакції без участі банків або інших посередників.

У сучасних умовах стрімкого розвитку цифрових технологій, коли криптовалюти та інші цифрові активи відіграють дедалі важливішу роль, криптовалютні платіжні шлюзи стають ключовим елементом, що забезпечує оперативні, захищені та економічно вигідні фінансові операції.

Основні функції криптовалютного платіжного шлюзу:

- створення індивідуальних криптовалютних адрес для отримання платежів;
- можливість автоматичної конвертації цифрових активів у фіатні валюти (за потреби);
- інтеграцію з онлайн- та офлайн-бізнесами через API, плагіни або індивідуальні рішення;
- гарантування безпечної обробки транзакцій і зниження ризиків шахрайства;
- підтримку широкого спектру блокчейнів та стейблкоїнів для підвищення зручності користувачів.

Останній звіт Forbes щодо криптоплатежів показує, що у світі вже приблизно 270 000 торговців приймають біткоїни. До цього переліку входять такі відомі компанії, як Subway, Starbucks, BMW та Microsoft (рис. 2.10).

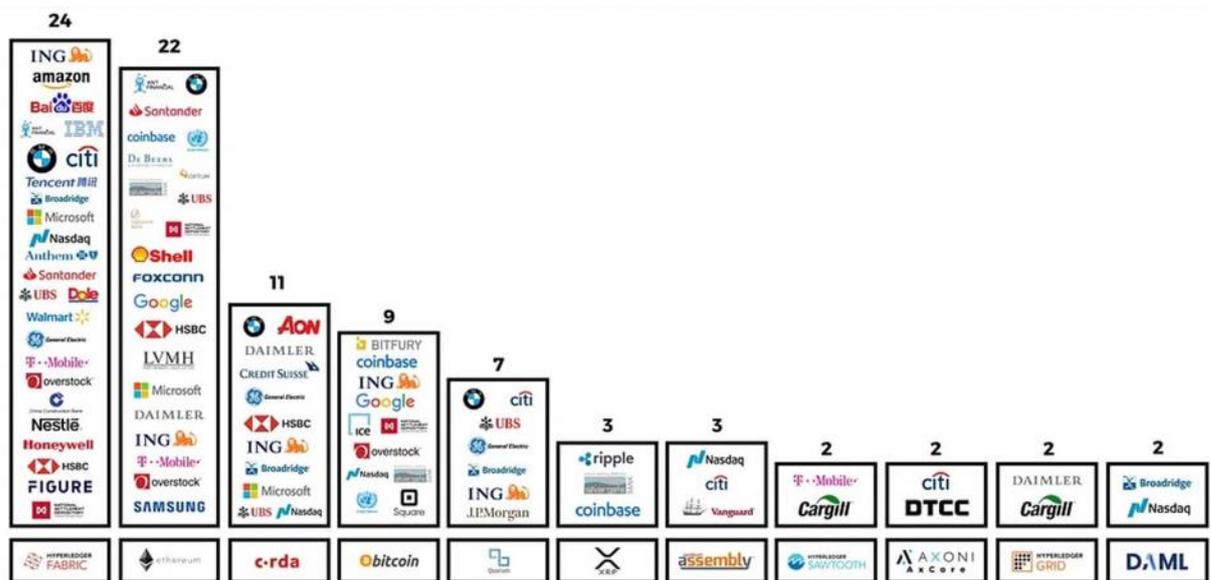


Рис. 2.10 Платформи для розробки, представлені в рейтингу Forbes Blockchain 50

Поява криптоплатіжних шлюзів у бізнес-середовищі суттєво змінила підхід до використання та прийому криптовалюти. Такі системи стали ключовою ланкою, що поєднує компанії з їхніми еквайєрами, зокрема банківськими установами. Вони забезпечують оперативну та надійну перевірку даних клієнтів, їхнє шифрування та захист, що дозволяє мінімізувати ризики витоку або несанкціонованого доступу до інформації [9].

Криптовалютні платіжні шлюзи функціонують як онлайн-платформи або цифрові системи, що застосовують технологію блокчейну для здійснення переказу цифрових активів з однієї адреси на іншу. Окрім цього, такі рішення надають можливість обмінювати криптовалюту на фіатні кошти - електронний аналог традиційних грошей, які використовуються у різних країнах світу.

Криптовалютні платіжні сервіси набувають дедалі більшої популярності завдяки здатності робити процес приймання та переказу цифрових активів простішим і ефективнішим. Такі шлюзи оснащені спеціалізованими програмними механізмами, що забезпечують безперебійну конвертацію криптовалюту у фіатні кошти та дають змогу автоматично переказувати отримані суми на гаманці продавців. Зазвичай вони підтримують широкий спектр цифрових валют, серед яких BTC, ETH, LTC, XRP, BCH, а також стейблкоїни - криптоактиви зі стабільною вартістю. На думку багатьох фахівців, саме стейблкоїни можуть стати найнадійнішим інструментом оплати в майбутньому (рис. 2.11).

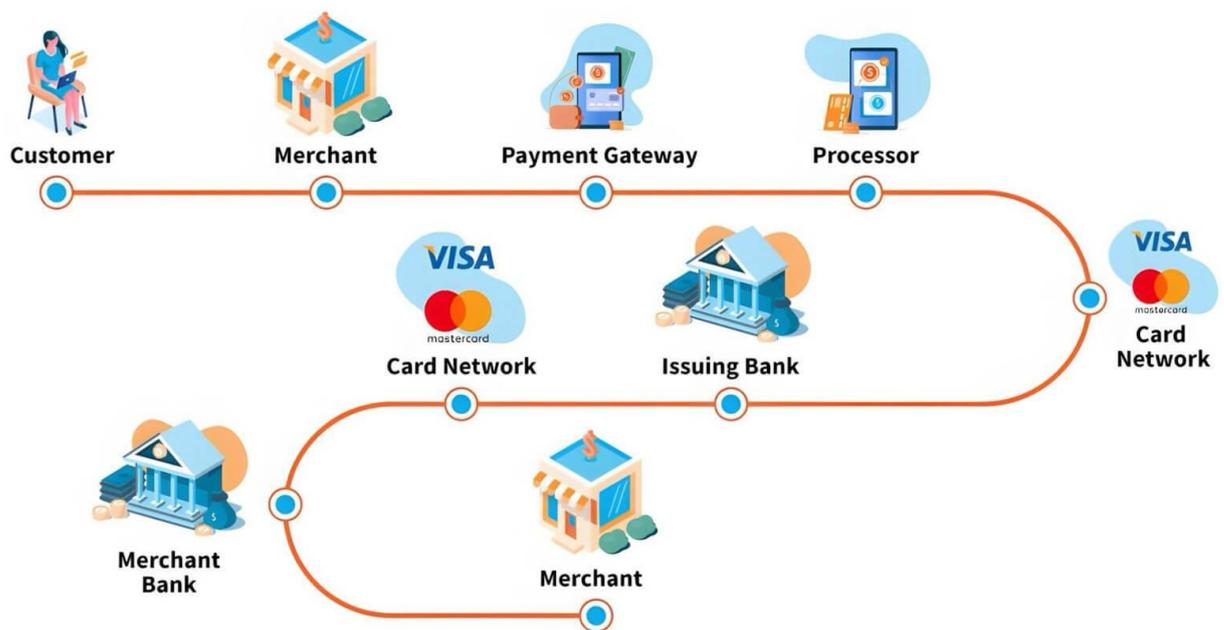


Рис. 2.11 Маршрутизація платежу в електронній комерції

Також важливо зазначити, що криптовалютний платіжний шлюз є ключовим компонентом платіжної системи та забезпечує її узгоджену роботу. Оскільки як звичайні, так і криптовалютні транзакції складаються з низки технічно складних етапів - перевірки, обробки, підтвердження та валідації - саме криптоплатіжний

шлюз відповідає за узгоджену взаємодію всіх елементів, задіяних у процесі платежу. Система функціонує як замкнений цикл, у якому кожна транзакція послідовно проходить усі необхідні стадії контролю та виконання.

Під час створення прототипу криптовалютного платіжного шлюзу важливо враховувати багаторівневу структуру його архітектури. З одного боку, потрібно побудувати функціональну платформу, здатну приймати платежі на різних блокчейн-мережах. З іншого - забезпечити просту й швидку інтеграцію з торговими майданчиками, адже основними користувачами продукту будуть сайти, що не спеціалізуються на роботі з криптовалютами. Тому, розробляючи блокчейн-проекти, необхідно усвідомлювати можливі труднощі та технічні особливості, які виникають під час їх практичної реалізації.

Визначимо ключові етапи створення власного криптошлюзу. На рис. 2.12 представлено кабінет користувача криптошлюзу.

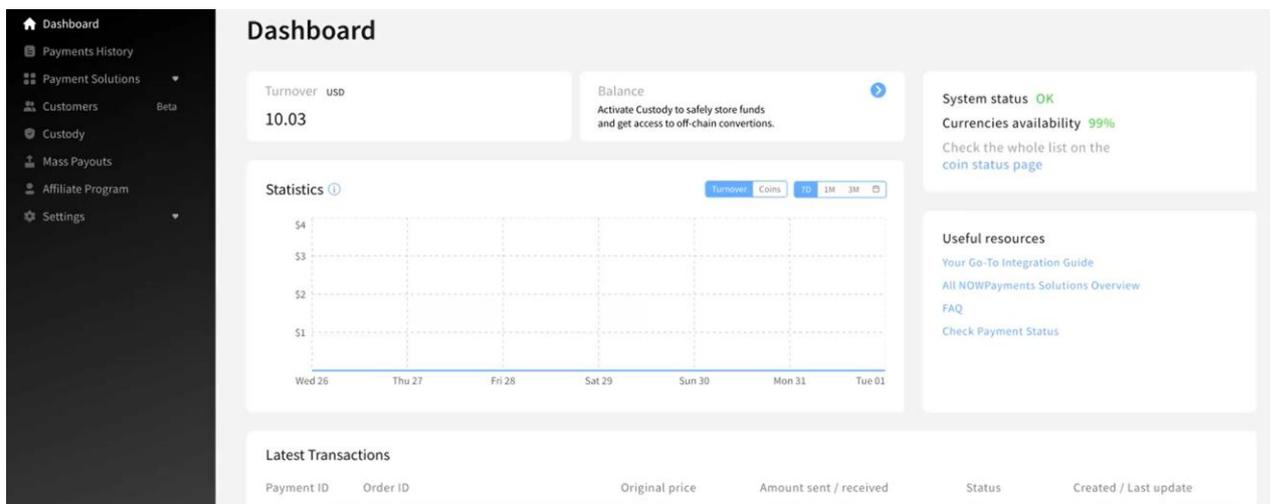


Рис. 2.12 Кабінет користувача криптошлюзу

Основним етапом розробки є визначення *архітектури проекту*. На цьому етапі формується структура системи, прораховуються шляхи проходження транзакцій користувачем і визначаються функціональні можливості торговця. Особливо важливо детально описати або наочно візуалізувати (за допомогою графічних схем) усі основні етапи взаємодії користувача з криптоплатіжним шлюзом. Це дозволяє чітко зрозуміти роботу всієї системи та мінімізувати можливі помилки й

логічні колізії у подальшій реалізації. Тобто першочерговим завданням є розробка архітектури криптовалютного платіжного шлюзу.

UI/UX-дизайн є стандартною процедурою створення візуального оформлення кожної сторінки. Для розробки макетів та їх перевірки доцільно використовувати Figma. На цьому етапі важливо врахувати всі деталі, щоб уникнути проблем під час інтеграції функціональних елементів, які могли бути пропущені. Саме тому наявність детально описаної архітектури, підготовленої на попередньому етапі, має критичне значення: вона дозволяє точно дотримуватися дизайн-макету без прогалин і невідповідностей (рис. 2.13).

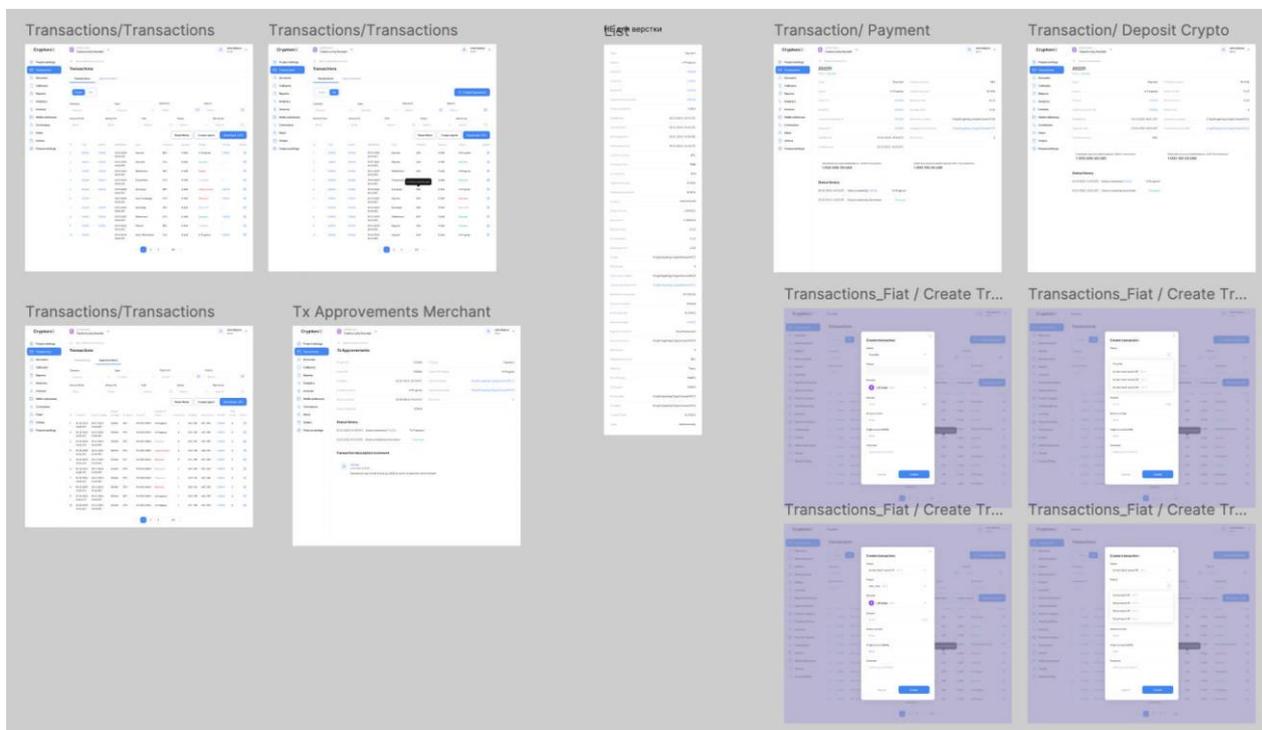


Рис. 2.13 UI/UX-дизайн

Розробка *Backend* та *Frontend* є найскладнішим і водночас ключовим етапом створення криптоплатіжного шлюзу. Він потребує максимального залучення команди та займає найбільше часу. Процес програмування координується менеджером проекту, який організовує безперервне розподілення завдань між розробниками. Такий підхід дозволяє мінімізувати затримки та забезпечити здачу

проєкту у визначені терміни. Особливу увагу слід приділити технологічному стеку, що зазвичай застосовується для розробки криптоплатіжних шлюзів:

- Backend: PHP, Node.js;
- Frontend: React.js.

Встановлення блокчейн-вузла. Криптоплатіжний шлюз функціонує на основі блокчейн-вузлів, що потребує значного часу на налаштування, особливо якщо передбачено інтеграцію декількох криптовалют для обробки платежів. Команда розробників повинна підготувати сервер, розгорнути вузол та реалізувати функціонал для введення, виведення та переказу коштів. Важливою перевагою є можливість використовувати один блокчейн для всіх криптовалют, що підтримуються його мережею. Тривалість реалізації залежить від кількості та складності блокчейн-вузлів.

Після реалізації ключових функцій взаємодії з блокчейн-вузлом проводиться *тестування* у тестовій мережі. QA-фахівці можуть здійснювати необмежену кількість транзакцій із тестовими засобами. Особлива увага приділяється перевірці функцій введення та виведення, адже будь-які збої в їх роботі можуть призвести до втрати коштів користувачів і завдати значної шкоди бізнесу. Орієнтовний термін тестування становить близько одного тижня (рис. 2.14).

	MAINNET	TESTNET
Purpose	Functional Blockchain	Testing Environment
Transactions	Real Transactions Stored on Actual Blockchain	Fake Transactions
Coins	Posses Real Value	No Monetary Value
Transaction Cost	Paid Using Native Coins	No Cost (Free)
Transaction Frequency	High	Low
Mining	Economic Incentive to Earn Real Coins	No Economic Benefit

Рис. 2.14 Тестування транзакцій

Наступний етап передбачає комплексне *тестування* всієї *платформи*. Зазвичай на цьому етапі виявляється велика кількість помилок, тому важливо забезпечити позитивну динаміку, щоб кількість багів зменшувалася з кожною ітерацією. Тривалість повного тестування залежить від якості коду та складності криптоплатіжного шлюзу. Крім того, на цьому етапі усуваються всі критичні помилки.

Коли платформа успішно пройшла тестування і за звітами QA демонструє стабільну роботу, відбувається її запуск в основній мережі. Проте перехід у робочий режим може проходити не завжди без проблем, тому надзвичайно важливо додатково протестувати всі транзакції з використанням реальних засобів.

Після завершення успішного тестування, *платформа запускається* у реальному режимі роботи. Під час взаємодії з реальними користувачами можуть виникати незначні помилки, які оперативно усуває служба технічної підтримки.

Після успішного запуску проєкту у реальному режимі роботи часто виникає потреба масштабувати бізнес і додавати нові функції. Усі пропозиції обговорюються з бізнес-аналітиком, який допомагає скласти *новий план розробки*. Водночас перед початком реалізації важливо чітко визначити необхідність кожної функції, оцінити її вартість та терміни впровадження. Після погодження всіх деталей розпочинається новий цикл розробки.

3 FRONTEND-ЧАСТИНА КРИПТОВАЛЮТНОГО ПЛАТІЖНОГО ШЛЮЗУ

3.1 Дослідження ринку криптовалютних платіжних систем

На ринках США та Європи представлені різні криптовалютні платіжні рішення, кожне з яких має свої сильні та слабкі сторони. *BitPay* - один із перших і найбільших провайдерів. *Coinbase Commerce* інтегрується з популярною біржею *Coinbase*, зручний для бізнесу, проте вимагає верифікації та підтримує лише певні криптовалюти. *CoinPayments* надає підтримку понад 2000 криптовалют, проте користувачі інколи скаржаться на затримку виплат. *NOWPayments* не потребує KYC, пропонує автоматичну конвертацію у фіат, але менш відомий серед великих компаній.

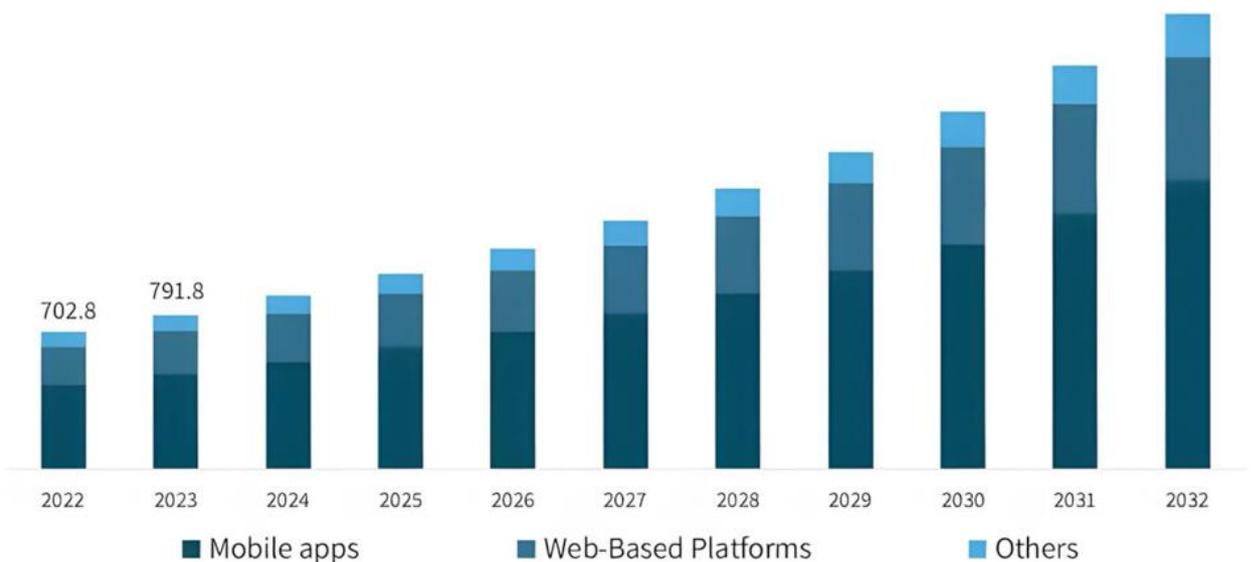


Рис. 3.1 Розмір ринку криптовалютних платіжних додатків за платформами, 2022–2032 рр. (млн. дол. США)

Криптовалютні платіжні системи функціонують за бізнес-моделлю, схожою на традиційні еквайрингові сервіси, але адаптованою до особливостей цифрових активів. Основні джерела доходу включають:

- 1 - комісію за транзакції;
- 2 - прибуток від конвертації криптовалют;

- 3 - абонентську плату за користування сервісом;
- 4 - додаткові послуги (AML/KYC, фіатні шлюзи, кастомізацію);
- 5 - white-label рішення для бізнесу.

1. Основним джерелом прибутку криптоплатіжних шлюзів є комісія з кожної транзакції. *Фіксована комісія* за обробку платежу зазвичай становить від 0.5% до 1.5%. *Гнучкі комісії* можуть змінюватися залежно від обсягу платежів клієнта або обраного тарифного плану. *Додаткові збори* можуть застосовуватися за прискорене підтвердження транзакцій або перекази через певні блокчейн-мережі. Наприклад, BitPay стягує 1% комісії за кожну транзакцію, що значно менше порівняно з традиційними платіжними системами, такими як Visa або Mastercard (2–5%).

2. Багато криптоплатіжних шлюзів надають можливість автоматичної конвертації криптовалюти у фіатні гроші (наприклад, долари або євро) або обміну однієї криптовалюти на іншу. *Спред на курсі* - це різниця між ціною купівлі та продажу активу, зазвичай становить 0.5–2%. *Динамічне ціноутворення* означає, що вартість обміну може змінюватися залежно від ліквідності ринку та попиту. Наприклад, CoinPayments отримує 0.5% прибутку на конвертації криптовалюти, тоді як деякі сервіси встановлюють більшу маржу при автоматичному обміні.

3. Деякі криптоплатіжні шлюзи пропонують передплатні або преміальні тарифні плани, що включають знижену комісію за транзакції, розширені API-інструменти та пріоритетну технічну підтримку. Наприклад, white-label рішення для компаній можуть коштувати від 500\$ до 5000\$ на місяць за ліцензію.

4. Криптоплатіжні шлюзи можуть отримувати прибуток від надання додаткових послуг:

- AML/KYC-модулі для відповідності регуляторним вимогам (вартість перевірки користувача зазвичай від 1\$ до 3\$);
- підключення банківських рахунків для виведення коштів;
- індивідуальна кастомізація платформи для великих клієнтів.

Наприклад, NOWPayments пропонує кастомні рішення та інтеграції, ціна яких стартує від 10000\$.

5. Деякі компанії пропонують white-label криптоплатіжні шлюзи, які можна використовувати під власним брендом. Це вигідно для тих, хто хоче швидко вийти на ринок, але не має технічних ресурсів для розробки з нуля. Ліцензія на використання зазвичай коштує від 20000\$ до 50000\$, а місячна підтримка та оновлення - від 2000\$ до 10000\$ залежно від складності. Наприклад, OpenNode надає white-label рішення для мерчантів і стягує відсоток від обороту.

Завдяки цим механізмам платіжні шлюзи забезпечують стабільний грошовий потік, водночас стикаючись з викликами, пов'язаними з регуляторними вимогами, волатильністю ринку та високими стандартами безпеки.

Розробка криптовалютного платіжного шлюзу є складним технологічним процесом, що вимагає високого рівня експертизи у блокчейні, безпеці та фінтех-інфраструктурі. Вибір підходу залежить від бюджету, цілей криптобізнесу та потреби у гнучкому налаштуванні системи. Власна розробка підходить великим компаніям, тоді як аутсорсинг дозволяє швидко запускати продукт з мінімальними витратами.

Етапами розробки криптоплатіжного шлюзу є:

- зрозуміти ринок;
- створити план процесу розробки прототипу криптовалютного платіжного шлюзу;
- проєктування та тестування.

На рис. 3.2 приведені етапи розробки криптоплатіжного шлюзу.



Рис. 3.2 Етапи розробки криптоплатіжного шлюзу

Якщо розглядати можливість *власної розробки* платіжного шлюзу, слід розуміти, що це забезпечить повний контроль над системою, але й потребуватиме витрат. Для створення шлюзу необхідно розумітися на блокчейн-технологіях, backend-розробці, володіти достатнім рівнем знань по безпеці та DevOps. Процес включає проєктування архітектури, розробку криптогаманців, інтеграцію з блокчейнами, створення API та механізмів обробки платежів, підключення фіатних шлюзів і забезпечення відповідності регуляторним вимогам. Перевагою такого способу розробки є: можливість гнучкого налаштування функціоналу; незалежність від сторонніх рішень; довгострокова економія на комісіях та ліцензіях. До недоліків слід віднести: тривалий час розробки (від 12 місяців), а також складність отримання ліцензій та дотримання регуляторних вимог.

Більш швидким та економним варіантом є *аутсорсинг розробки*. Спеціалізовані компанії використовують готові модулі та напрацювання, що дозволяє скоротити час і витрати на розробку. Процес розробки займає приблизно 4–6 місяців, а вартість коливається в залежності від обсягу функціоналу. Перевагою є можливість швидкого розгортання рішення, нижчі витрати, можливість доступу до експертних рішень. Недоліками такого способу розробки є: обмежені можливості кастомізації, залежність від зовнішніх розробників.

Вартість розробки криптовалютного платіжного шлюзу визначається функціоналом, рівнем безпеки, кількістю підтримуваних криптовалют та інтеграцією з фіатними платіжними системами. Умовно проєкти можна класифікувати на три категорії: базове рішення, рішення середнього рівня та повноцінна платіжна платформа.

Базовий варіант шлюзу включає мінімальний набір функцій для прийому та обробки криптовалютних платежів. Розробка триває приблизно 2–4 місяці та зазвичай реалізується на базі готових рішень (white-label платформ). Вартість такого рішення: 20000\$ - 40000\$. До функціоналу базового рішення слід віднести: підтримку кількох криптовалют (BTC, ETH, USDT), генерацію адрес для прийому платежів, просту панель управління для мерчантів, API для інтеграції з вебсайтами та магазинами, базовий рівень безпеки (шифрування, 2FA). До

обмежень варто віднести відсутність підтримки фіатних платежів, обмежені можливості кастомізації, відсутність розширених інструментів моніторингу та AML. Цей варіант підходить для невеликих онлайн-магазинів, які хочуть швидко впровадити прийом криптовалют.

Варіант рішення середнього рівня розробки криптовалютного платіжного шлюзу орієнтований на компанії, що прагнуть запропонувати клієнтам більш зручний та безпечний сервіс. Така розробка може тривати від 4 до 6 місяців і вимагати складнішої архітектури. В ході проведених досліджень у магістерській кваліфікаційній роботі визначено функціонал для рішення середнього рівня:

- підтримка 5–10 криптовалют;
- фіатні шлюзи (SWIFT, SEPA, картки Visa/Mastercard);
- вбудовані механізми автоматичної конвертації;
- API з розширеними можливостями для інтеграції;
- AML/KYC-модулі для дотримання регуляторних вимог;
- забезпечення дворівневої системи безпеки (холодні гаманці, мультипідпис)

Цей варіант підходить для криптовалютних стартапів, онлайн-платформ та сервісів, що працюють з цифровими активами. Вартість середнього рівня рішення може сягати 40000\$ - 80000\$.

Функціонал рішення для великих проєктів здатний забезпечити підтримку десятків криптовалют та токенів; мати гнучку систему налаштування платіжних потоків та гнучку систему комісій; забезпечувати підключення партнерів банків та кастомних шлюзів (фіатних); мати повну інтеграцію з біржами та DeFi-протоколами; мати розширені механізми моніторингу транзакцій та захист від несанкціонованого доступу; мати масштабовану інфраструктуру з можливістю підтримки високого навантаження. Вартість розгортання великого проєкту буде в межах 80000\$ - 240000\$ адже це повноцінна платіжна платформа з великими можливостями та високою системою захисту. Для розробки великого проєкту криптовалютного платіжного шлюзу необхідно буде 6 – 12 місяців. Дана розробка потребуватиме складних технічних рішень.

Окрім безпосередньої розробки криптоплатіжного шлюзу, бізнесу слід враховувати юридичні та операційні витрати, такі як: ліцензування (MSB у США, ЕМІ в ЄС) - від 10000\$ до 50000\$; інфраструктура (сервери, хостинг, безпека) - від 5000\$ на рік; підтримка та оновлення – 10-20% від вартості розробки щорічно.

Вибір конкретного рішення залежить від бізнес-плану. Базові варіанти рішень дозволяють швидко запустити продукт на ринку, але мають обмежений функціонал. Рішення середнього рівня забезпечують оптимальний баланс між ціною та можливостями, тоді як масштабні платформи підходять для компаній, які планують значний розвиток і розширення.

Криптовалютні платіжні шлюзи підлягають фінансовому регулюванню в більшості країн. Для законної діяльності необхідно отримати відповідні ліцензії та дотримуватися вимог AML/KYC. При запуску бізнесу важливо обрати правильну юрисдикцію для реєстрації компанії, оскільки податкові ставки, вимоги до звітності та можливості взаємодії з клієнтами залежать від країни реєстрації.

Популярні юрисдикції для реєстрації криптобізнесу:

- США (штати Вайомінг, Делавер) - високий рівень довіри, проте складне регулювання;
- Естонія, Литва - доступні ліцензії та зручні умови роботи з ЄС;
- Сінгапур, Гонконг - вигідні податкові умови для міжнародного бізнесу;
- ОАЕ (Дубай, Абу-Дабі) - 0% податок на дохід та сприятливе криптовалютне середовище.

Одержання необхідних ліцензій є необхідним етапом процесу, що гарантує законність діяльності та підвищує довіру користувачів.

Основними ліцензіями є:

- MSB (Money Services Business) – США (через FinCEN);
- ЕМІ (Electronic Money Institution) – ЄС (через країни ЄС, наприклад, Чехію, Литву, Естонію);
- FCA Crypto License – Великобританія;
- PSP (Payment Service Provider) – для здійснення фіатних платежів;

- VASP (Virtual Asset Service Provider) – глобальна ліцензія для операцій з криптовалютою.

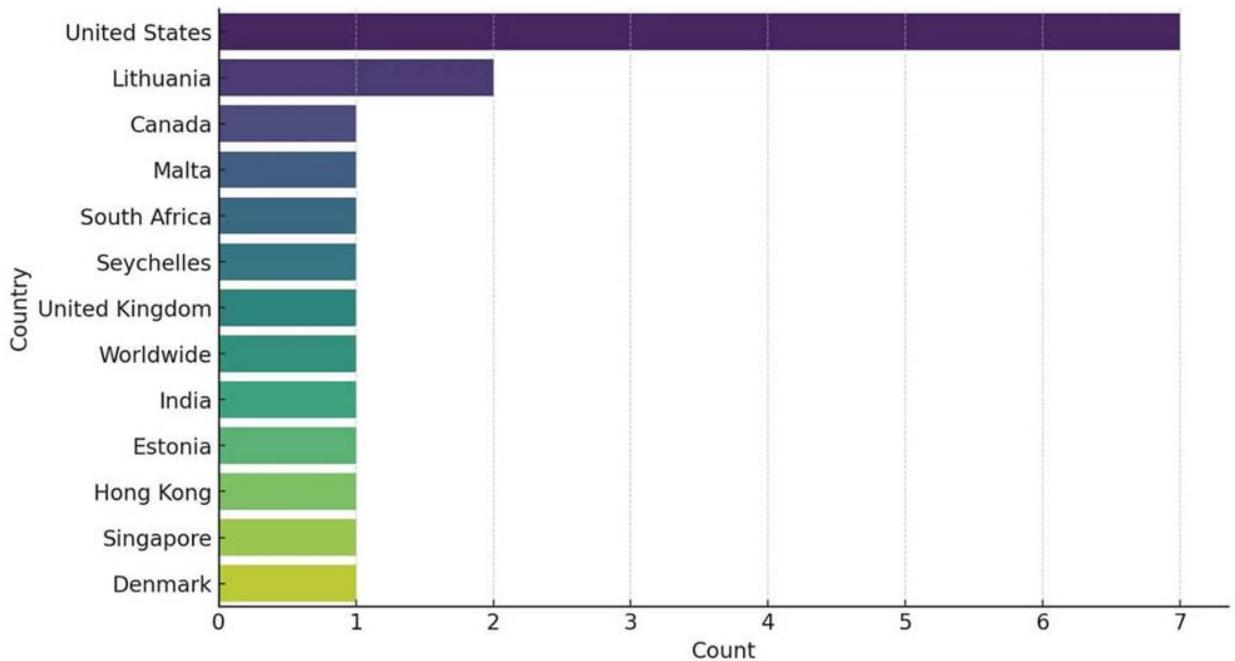


Рис. 3.3 Розподіл платіжних систем по країнам

Криптовалютні платіжні шлюзи у сучасному світі стають важливою складовою фінансової екосистеми, надаючи бізнесу інструменти та можливості для безпечного та ефективного прийому, обробки платежів у криптовалюті. Із зростанням популярності цифрових валют і розширенням аудиторії користувачів, такі рішення стають особливо актуальними для р2р-обмінників та інтернет-магазинів. Вони забезпечують компаніям зручність і економію, а також відкривають нові можливості для розвитку бізнесу у сфері криптовалютних операцій.

Проте створення та впровадження таких платіжних шлюзів вимагає глибоких знань у сфері блокчейн-технологій, фінансового регулювання та питань безпеки. Правильний вибір стратегії розробки від аутсорсингу до власного рішення разом із врахуванням усіх юридичних аспектів є надважливими умовами для успішної реалізації.

3.2 Структура SPA-додатку

Розробка frontend-частини криптовалютного платіжного шлюзу - це створення користувацького інтерфейсу, через який клієнти та мерчанти взаємодіють із платіжною системою. На початку потрібно зрозуміти, яким чином буде відбуватися користування системою. Клієнт вводить e-mail; вказує суму платежу, отримує адресу/QR-код для оплати, бачить статус транзакції (очікується, підтверджено, помилка). Мерчант (продавець) переглядає список платежів, бачить статуси та історію операцій. Це формує логіку інтерфейсу та перелік екранів.

Для прототипу криптоплатіжного шлюзу оптимальним є SPA (Single Page Application). SPA-додаток (Single Page Application) - це вебдодаток, у якому вся взаємодія користувача відбувається на одній сторінці браузера без повного перезавантаження сторінки. У SPA-додатку браузер завантажує сторінку один раз, далі інтерфейс оновлюється динамічно, дані підвантажуються через API (JSON).

SPA-додаток має наступні ознаки: одна HTML-сторінка; клієнтська логіка виконується у браузері; обмін із сервером через REST API; швидка реакція інтерфейсу; чітке розділення frontend і backend. Зазвичай SPA можна створювати за допомогою: Vue.js, Angular або React. У контексті криптовалютного платіжного шлюзу SPA - це клієнтська частина системи, з якою взаємодіють користувач та адміністратор. Користувач вводить e-mail, вводить суму платежу, ініціює оплату, а також бачить статус транзакції (очікується/підтверджено/успішно/помилка). Адміністратор (мерчант) переглядає платежі, бачить статуси транзакцій, контролює баланс та отримує повідомлення про успішні платежі.

У рамках магістерської кваліфікаційної роботи у прототипі SPA відповідає за відображення інтерфейсу, взаємодію з backend, роботу у реальному часі. При організації взаємодії з backend SPA надсилає HTTP-запити до REST API, отримує відповіді у форматі JSON, оновлює інтерфейс відповідно до стану транзакції. SPA добре інтегрується з REST API, легко масштабується під e-commerce.

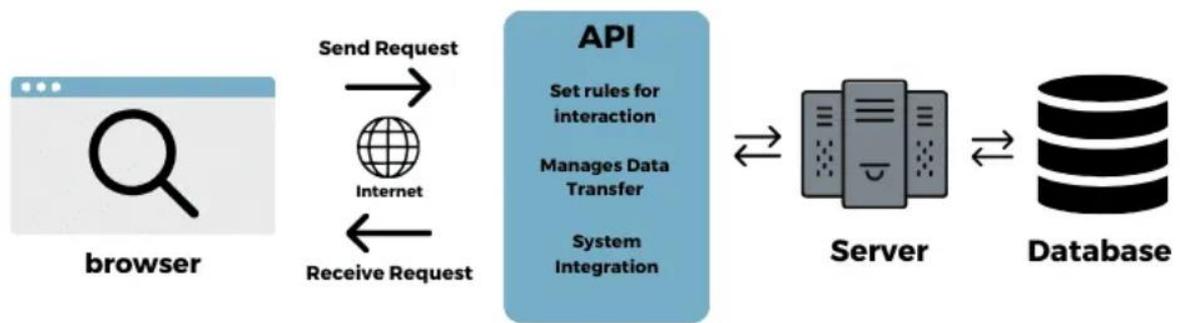


Рис. 3.4 Інтеграція з API

Актуальними фреймворками для побудови SPA-додатків є Vue.js та Angular. Vue.js є сучасним JavaScript-фреймворком для створення користувацьких інтерфейсів, зокрема SPA-додатків (Single Page Application). Vue.js дозволяє швидко й зручно створювати веб-інтерфейси, які динамічно оновлюються без перезавантаження сторінки, взаємодіють з бекендом через REST API, мають чітку структуру та добре масштабуються. Визначимо основні характеристики Vue.js. До них віднесемо:

- компонентний підхід (інтерфейс будується з незалежних компонентів (форма оплати, статус транзакції));
- реактивність (дані та інтерфейс автоматично синхронізуються: якщо змінюється стан транзакції - UI оновлюється сам);
- SPA-архітектура (весь додаток працює в межах однієї HTML-сторінки, а переходи між екранами відбуваються без перезавантаження);
- зручна робота з REST API (Vue легко інтегрується з бекендом через HTTP-запити);
- підтримка сучасних інструментів (Vue CLI, Vite, router, state-management - все необхідне для реального проєкту).

У магістерській кваліфікаційній роботі Vue.js може використовуватись для створення SPA-інтерфейсу платіжної форми, введення даних (ID замовлення, сума поповнення, e-mail), відправлення запитів до REST API бекенду, відображення статусу транзакції (очікується, підтверджено, помилка), імітації реального процесу оплати в e-commerce. Важливим є те, що Vue не працює з

блокчейном напряму, він лише взаємодіє з бекендом, який уже реалізує логіку платежів [21].

Vue.js чітко демонструє SPA + REST API архітектуру, а також підходить для прототипу криптовалютного платіжного шлюзу. Vue.js - це прогресивний JavaScript-фреймворк для розробки односторінкових веб-додатків, який забезпечує створення динамічного та реактивного користувацького інтерфейсу для взаємодії з REST API криптовалютного платіжного шлюзу.

Іншим фреймворком, який можна використати для побудови динамічних, масштабованих SPA-додатків є Angular - це комплексний frontend-фреймворк на основі TypeScript, який дозволяє створювати повноцінні клієнтські вебдодатки з чіткою архітектурою та розвинутою екосистемою. Хоча Angular краще демонструє інженерний підхід і масштабованість, для прототипу у дипломній роботі простіше використовувати Vue.

Використати Vue CLI для створення нового проєкту:

```
vue create crypto-payment-gateway
```

Далі необхідно вибрати стандартну конфігурацію з Babel, Router, Vuex (якщо потрібен стан) та Linter.

SPA має основну сторінку та сторінку оформлення платежу:

```
const routes = [  
  { path: '/', component: Home },  
  { path: '/checkout', component: Checkout }  
]
```

Для реалізації форми із полями: ID, суми поповнення, e-mail необхідно створити компонент *PaymentForm.vue*. В якості полів використано: *transactionId* (ID транзакції), *amount* (сума поповнення), *e-mail* (e-mail користувача). Використано двостороннє зв'язування через *v-model*:

```
<input v-model="transactionId" placeholder="Transaction ID" />  
<input v-model="amount" type="number" placeholder="Amount" />  
<input v-model="email" type="email" placeholder="Email" />
```

Для валідації полів необхідно перевірити, що всі поля заповнені, перевірити формат e-mail, перевірити, що сума > 0.

Кнопка для підтвердження платежу:

```
<button @click="submitPayment">Pay</button>
```

SPA-додаток у межах криптовалютного платіжного шлюзу виконує роль клієнтського інтерфейсу, що забезпечує взаємодію користувача з системою без перезавантаження сторінок. Він реалізує введення платіжних даних, ініціацію транзакцій та відображення їхнього стану шляхом обміну даними з серверною частиною через REST API.

3.3 Механізм відправлення запитів до REST API

Frontend не працює з блокчейном напряму, а взаємодіє з бекендом через REST API. Користувач натискає кнопку «Оплатити». Фронтенд формує HTTP-запит (POST). Дані надсилаються на сервер: сума; e-mail; ID замовлення. Сервер повертає відповідь: адресу гаманця; статус операції.

RESTful API (Representational State Transfer API) - це архітектурний стиль для створення веб-сервісів, який дозволяє клієнту (наприклад, веб-додатку або мобільному додатку) взаємодіяти з сервером через стандартні HTTP-запити. RESTful API – це набір правил та принципів, які визначають, як клієнт може отримувати, додавати, оновлювати або видаляти дані на сервері. Визначимо основні особливості RESTful API:

- використання HTTP-методів: GET (отримати дані), POST (створити нові дані), PUT/PATCH (оновити дані), DELETE (видалити дані).
- ресурси: всі дані представляються у вигляді ресурсів з унікальними URL-адресами;
- безстанова взаємодія (stateless): кожен запит містить усю необхідну інформацію для його обробки, сервер не зберігає стан клієнта між запитами;
- можливість використання різних форматів даних, наприклад, таких як JSON, XML.

Структура RESTful API приведена на рис. 3.5.

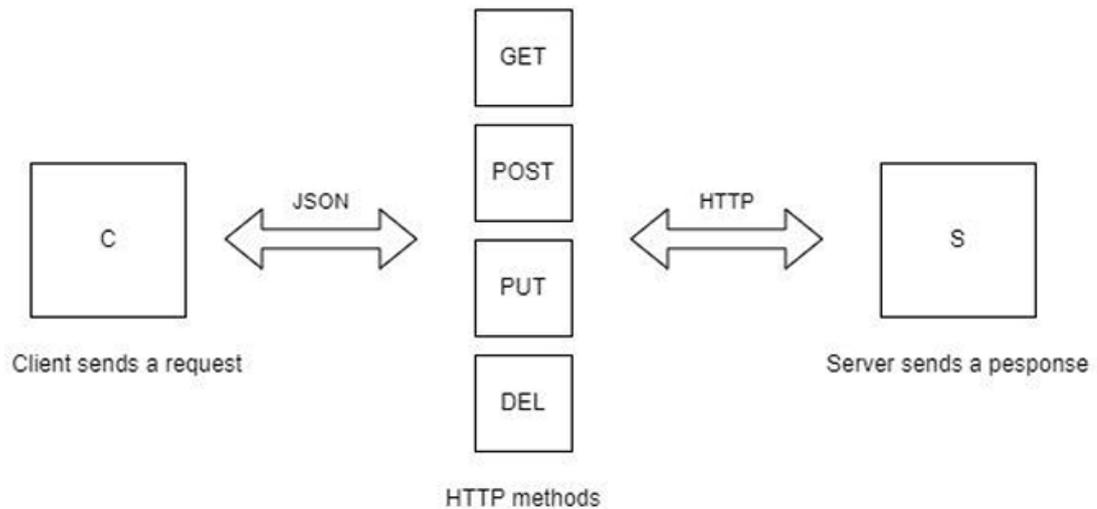


Рис. 3.5 Структура RESTful API

Ідемпотентність при реалізації RESTful API - це принцип, за яким багаторазове виконання одного й того самого запиту призводить до одного й того самого результату для стану системи, незалежно від кількості повторів. Якщо клієнт випадково надішле запит двічі або більше, то система не створить дублікати; не змінить дані повторно; збереже коректний і передбачуваний стан. Це особливо важливо для фінансових систем, зокрема криптовалютних платіжних шлюзів, де повторний запит не повинен призводити до подвійного списання коштів.

Ідемпотентні запити можна безпечно повторювати у разі втрати з'єднання, тайм-ауту, повторної відправки клієнтом. Це особливо важливо для фінансових систем і платіжних шлюзів. У криптовалютному платіжному шлюзі ідемпотентність запобігає подвійному списанню коштів, гарантує коректну обробку платежів, забезпечує надійність при збої мережі.

REST спирається на стандартні методи протоколу HTTP для роботи з ресурсами, а в RESTful API ці методи є ключовим інструментом організації обміну даними між клієнтською та серверною частинами системи.

Формат протоколу HTTP представлений на рис. 3.6.

Запит

```
GET / HTTP/1.0  
Host: example.com
```

Відповідь

```
HTTP/1.1 200 OK  
Date: Wed, 26 Sep 2022 16:30:39 GMT  
Server: nginx  
Content-Language: en  
Content-Type: text/plain; charset=utf-8  
Content-Length: 128  
Connection: close
```

Text data

Рис. 3.6 Формат протоколу HTTP

Основні методи HTTP, що застосовуються в REST: GET, POST, PUT, PATCH, DELETE, також є методи HEAD та OPTIONS.

Призначення методу GET - отримати дані з сервера без їх зміни. Це ідемпотентний метод, тобто багаторазові запити не змінюють стан ресурсу. Використання: запит інформації про користувача, список транзакцій, баланс гаманця.

Метод POST створює новий ресурс на сервері або виконує обробку даних. Повторні запити можуть створювати дублікати ресурсів. Використання: створення нового користувача, ініціація платежу, додавання нового запису транзакції.

Метод PUT повністю оновлює існуючий ресурс або створює його, якщо він не існує. Особливість PUT-методу полягає в ідемпотентності - повторні запити не змінюють результат, якщо дані однакові. Використання: оновлення даних користувача, зміна суми транзакції, редагування налаштувань гаманця.

Призначення PATCH-методу полягає у частковому оновленні ресурсів, тобто зміні лише певних полів. Використання: оновлення лише e-mail або статусу транзакції без зміни інших даних.

DELETE - видаляє ресурс на сервері. Повторні запити видалення одного і того ж ресурсу не викликають помилок і не змінюють результат. Використання: видалення користувача, відміна транзакції, видалення старого запису з бази даних.

Метод HEAD у HTTP схожий на GET, проте не повертає тіло відповіді, а лише заголовки. Він застосовується для перевірки існування ресурсу та отримання інформації про нього без завантаження повного обсягу даних.

HTTP-метод OPTIONS використовується для визначення можливостей ресурсу, зокрема методів, які підтримуються HTTP, і допустимих заголовків.

Категорії статус кодів HTTP-відповідей: 1xx – інформаційні повідомлення, 2xx – успішна операція, 3xx – переадресація, 4xx – помилки клієнта, 5xx – помилки сервера.

Рис. 3.7 відображає основні статус коди HTTP.

Код	Значення	Пояснення
200	OK	The requested action was successful.
201	Created	A new resource was created.
202	Accepted	The request was received, but no modification has been made yet.
204	No Content	The request was successful, but the response has no content.
400	Bad Request	The request was malformed.
401	Unauthorized	The client is not authorized to perform the requested action.
404	Not Found	The requested resource was not found.
415	Unsupported Media Type	The request data format is not supported by the server.
422	Unprocessable Entity	The request data was properly formatted but contained invalid or missing data.
500	Internal Server Error	The server threw an error when processing the request.

Рис. 3.7 основні статус коди HTTP

Frontend повинен коректно реагувати на відповіді бекенду. Якщо платіж створено - показати адресу або QR-код; якщо помилка - показати повідомлення; якщо статус змінюється - оновити інформацію на екрані. Це реалізує інформування користувача, прозорість процесу оплати.

Frontend криптовалютного шлюзу реалізується як SPA-додаток; забезпечує взаємодію користувача з платіжною системою; працює через REST API; відображає процес та статуси платежів; є ключовим елементом прототипу для e-commerce.

Механізм відправлення запитів до REST API складається з наступних етапів:

1) Створення сервісу для API-запитів (api.js)

```
import axios from 'axios';
```

```
const API_URL = 'https://your-backend-url.com/api';
```

```
export function createTransaction(data) {  
  return axios.post(`${API_URL}/transactions`, data);  
}
```

2) Виклик API із компонента

```
methods: {  
  async submitPayment() {  
    try {  
      const response = await createTransaction({  
        id: this.transactionId,  
        amount: this.amount,  
        email: this.email  
      });  
      this.$emit('transaction-success', response.data);  
    } catch (error) {  
      this.$emit('transaction-error', error.response.data);  
    }  
  }  
}
```

4 BACKEND-ЧАСТИНА ПЛАТІЖНОГО ШЛЮЗУ НА JAVA SPRING

4.1 Проєктування REST API

Проєктування REST API - це процес визначення структури, правил та логіки взаємодії між клієнтською частиною (SPA-додатком) і серверною частиною системи через HTTP-протокол. У контексті криптовалютного платіжного шлюзу для e-commerce REST API виконує роль центрального комунікаційного механізму, який приймає запити від frontend-додатку, передає дані в базу даних, взаємодіє з блокчейн-мережами або платіжною логікою, повертає клієнту результат операції (успіх, помилка, статус платежу).

Проєктування API є критично важливим, оскільки забезпечує узгоджену архітектуру системи, спрощує масштабування та підтримку, підвищує безпеку фінансових операцій, дозволяє інтегрувати платіжний шлюз із зовнішніми e-commerce платформами.

REST API базується на таких ключових принципах:

- клієнт–серверна архітектура, де frontend і backend є незалежними компонентами;
- відсутність стану, коли кожен HTTP-запит містить усю необхідну інформацію, сервер не зберігає стан сесії;
- орієнтація на ресурси - всі дані розглядаються як ресурси, доступ до яких здійснюється через URL;
- використання стандартних HTTP-методів GET, POST, PUT, DELETE, PATCH;
- уніфікований інтерфейс - єдині правила і формат обміну даними (JSON).

XML	JSON	YAML
<pre><Servers> <Server> <name>Server1</name> <owner>John</owner> <created>123456</created> <status>active</status> </Server> </Servers></pre>	<pre>{ Servers: [{ name: Server1, owner: John, created: 123456, status: active }] }</pre>	<pre>Servers: - name: Server1 owner: John created: 123456 status: active</pre>

Рис. 4.1 Формати обміну даними

На етапі проєктування визначаються основні ресурси системи. Для прототипу криптовалютного платіжного шлюзу такими ресурсами є: користувачі (users); платежі (payments); транзакції (transactions); баланси (balances).

Приклад REST-ресурсів:

/api/users

/api/payments

/api/transactions

/api/balances

Далі визначається, які операції можна виконувати над кожним ресурсом. Для ресурсу *payments* операції зведені до табл. 4.1.

Таблиця 4.1

Операції для ресурсу *payments*

№ п/п	HTTP-метод	Endpoint	Опис
1	POST	<i>/api/payments</i>	Створення нового платежу
2	GET	<i>/api/payments/{id}</i>	Отримання інформації про платіж
3	GET	<i>/api/payments</i>	Список усіх платежів
4	PUT	<i>/api/payments/{id}</i>	Оновлення статусу платежу
5	DELETE	<i>/api/payments/{id}</i>	Видалення платежу (опціонально)

REST API використовує JSON як стандартний формат передавання даних.

Запит (створення платежу):

```
{  
  "order_id": "ORD-1001",  
  "amount": 250.00,  
  "currency": "USDT",  
  "email": "customer@example.com"  
}
```

Відповідь сервера:

```

{
  "payment_id": "PAY-5678",
  "status": "pending",
  "created_at": "2025-05-10T14:30:00"
}

```

Правильне використання кодів відповіді є важливою частиною проектування. HTTP-статуси у REST API приведені у табл. 4.2.

Таблиця 4.2

№ п/п	Код	Значення
1	200 OK	Запит виконано успішно
2	201 Created	Ресурс створено
3	400 Bad Request	Помилка у запиті
4	401 Unauthorized	Немає доступу
5	404 Not Found	Ресурс не знайдено
6	500 Internal Server Error	Помилка сервера

На основі досліджень, проведених у попередньому розділі роботи, варто зазначити, що для фінансових операцій ідемпотентність є критично важливою, а ідемпотентний запит - це запит, який при повторному виконанні не змінює результат. У платіжному шлюзі GET, PUT, DELETE – ідемпотентні запити; POST - зазвичай не ідемпотентний запит (створює нову транзакцію). Для платежів часто використовують idempotency key, щоб уникнути подвійних списань.

Реалізація REST API:

```

const express = require('express');
const app = express();

app.use(express.json());

app.post('/api/payments', (req, res) => {
  const { order_id, amount, currency, email } = req.body;

  res.status(201).json({

```

```

    payment_id: 'PAY-12345',
    status: 'pending',
    order_id,
    amount,
    currency
  });
});

app.get('/api/payments/:id', (req, res) => {
  res.json({
    payment_id: req.params.id,
    status: 'completed'
  });
});

app.listen(3000, () => {
  console.log('REST API server running on port 3000');
});

```

У SPA-додатку (у роботі використано Vue.js): frontend надсилає HTTP-запити через `fetch` або `axios`; REST API обробляє бізнес-логіку; база даних зберігає фінансові та користувацькі дані; клієнт отримує результат без перезавантаження сторінки. Проектування REST API визначає логіку взаємодії між компонентами системи, забезпечує безпеку, масштабованість і зручність інтеграції з e-commerce платформами.

4.2 Використання Axios як інструменту для роботи з RESTful API

Axios - це JavaScript-бібліотека, призначена для надсилання HTTP-запитів у веб- та мобільних застосунках. Вона забезпечує зручну взаємодію з RESTful API, дозволяючи клієнтським додаткам обмінюватися даними із сервером. Axios

підтримує виконання різних типів HTTP-запитів, зокрема GET, POST, PUT, DELETE та інших, які використовуються для доступу до ресурсів і виконання операцій у межах RESTful API [20].

Axios автоматично опрацьовує відповіді, що надходять від сервера, та надає зручний механізм доступу до даних, отриманих через RESTful API. Бібліотека підтримує роботу з різними форматами даних, зокрема JSON, XML і текстовими відповідями, що значно спрощує їх подальшу обробку та використання у додатку.

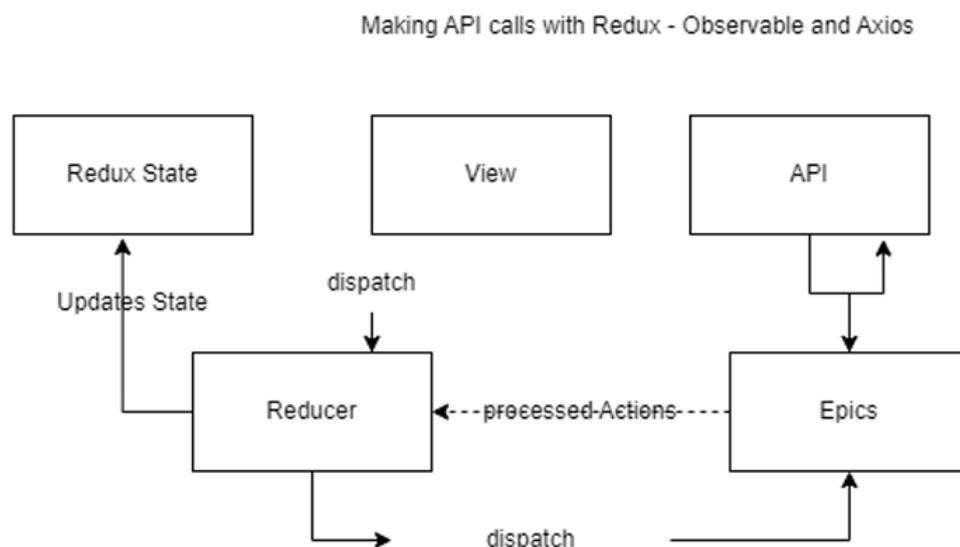


Рис. 4.2 Використання Axios

Axios надає можливість гнучко конфігурувати HTTP-запити, зокрема задавати заголовки, параметри запити (query parameters), тіло запити та інші налаштування. Це дає змогу чітко контролювати, які саме дані передаються на сервер і з якими параметрами виконується запит.

Axios надає зручні механізми для опрацювання помилок, що можуть виникати під час виконання HTTP-запитів. Бібліотека дозволяє аналізувати коди статусу відповіді, коректно реагувати на помилки та реалізовувати необхідну логіку обробки для різних результатів виконання запити.

Міжсайтовий обмін ресурсами (CORS): Axios надає можливість коректно працювати з обмеженнями CORS, які можуть виникати під час звернення до RESTful API, розміщеного на іншому домені. За допомогою бібліотеки можна

налаштовувати необхідні заголовки та параметри запитів, що забезпечує коректну та безпечну взаємодію клієнтського додатка із сервером. На рис. 4.3 приведено приклад застосування Axios для того, щоб виконувати відправлення запитів на backend-частину криптовалютного платіжного шлюзу.

```
export default {
  data() {
    return {
      user: {},
      wallet: {},
    };
  },
  mounted() {
    this.fetchUser();
  },
  methods: {
    async fetchUser() {
      try {
        const response = await api.get('url: "/profile"');
        this.user = response.data;
        this.wallet = response.data.wallet;
      } catch (error) {
        console.error("Error fetching messages:", error);
      }
    },
  },
};
</script>
```

Рис. 4.3 Використання Axios для відправлення запиту

Отже, визначимо можливості, завдяки яким Axios забезпечує ефективну інтеграцію з RESTful API:

- 1) просте надсилання HTTP-запитів (GET, POST, PUT, DELETE);
- 2) автоматична обробка відповідей та доступ до даних у форматах JSON, XML або тексті;
- 3) гнучке налаштування параметрів запиту, заголовків і даних;
- 4) інструменти для обробки помилок, включаючи перевірку статусів і логіку реагування;
- 5) підтримка CORS для безпечної взаємодії з ресурсами на інших доменах.

Загалом, Axios робить взаємодію з RESTful API дуже зручною і надає широкий набір інструментів для роботи з віддаленими ресурсами, що робить його

популярним вибором серед багатьох JavaScript-розробників завдяки високому рівню функціональності та надійності.

Організація взаємодії REST API з frontend та backend-частинами проєкту представлена на рис. 4.4.

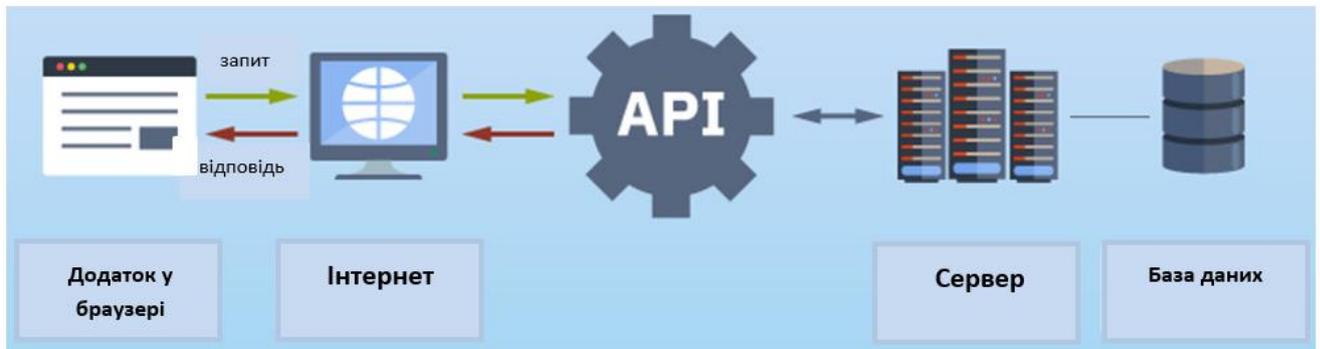


Рис. 4.4 Організація взаємодії REST API з frontend та backend-частинами проєкту

4.3 Особливості використання Java Spring фреймворку для розробки backend-частини платформи

Java Spring - це потужний фреймворк для розробки корпоративних та веб-додатків на мові Java, який спрощує створення складних програмних систем завдяки модульній архітектурі та готовим рішенням для стандартних завдань.

Java Spring пропонує широкий набір інструментів і готових рішень, які значно полегшують процес створення серверних додатків. Фреймворк містить велику кількість модулів, що дають змогу швидко та зручно розширювати функціональні можливості додатків. Spring базується на принципах інверсії керування та впровадження залежностей, що підвищує читабельність коду й спрощує його тестування.

Завдяки підтримці RESTful API, Spring є зручним інструментом для створення серверної логіки вебдодатків. Java Spring представляє собою комплексну екосистему розробки, що містить велику кількість бібліотек, спрямованих на спрощення створення та підтримки різних функціональних можливостей додатків [14, 25].

Основні бібліотеки та залежності, які використовуються у Java Spring:

- *Spring Boot* - спеціальний фреймворк у екосистемі Spring, який спрощує створення та запуск Java-додатків, особливо веб-сервісів і REST API;
- *Spring MVC (Model-View-Controller)* - це модуль фреймворку Spring, який призначений для розробки веб-додатків та REST API за архітектурним патерном MVC;
- *Spring Data* - модуль фреймворку Spring, який спрощує роботу з базами даних та іншими сховищами даних, автоматизуючи більшість рутинних операцій доступу до даних;
- *Spring Security* - це модуль фреймворку Spring, призначений для забезпечення безпеки Java-додатків, включаючи веб-додатки та REST API. Він дозволяє реалізувати автентифікацію, авторизацію та захист від типових загроз;
- *Spring Cloud* - набір інструментів та бібліотек для розробки розподілених та хмарних (cloud-native) Java-додатків у рамках екосистеми Spring;
- *Spring Integration* - модуль фреймворку Spring, який дозволяє будувати інтеграційні рішення та системи для обміну повідомленнями між різними компонентами у Java-додатках;
- *Spring Batch* - модуль фреймворку Spring, призначений для розробки додатків пакетної обробки даних (batch processing);
- *Spring WebFlux* - модуль фреймворку Spring для створення реактивних веб-додатків та REST API, який підтримує асинхронну та неблокуючу обробку запитів;
- *Spring AOP (Aspect-Oriented Programming)* - модуль фреймворку Spring, який дозволяє реалізовувати аспектно-орієнтоване програмування;
- *Spring Test* - модуль фреймворку Spring, який забезпечує потужні інструменти для тестування Spring-додатків, включаючи веб-додатки, REST API та бізнес-логіку;
- *Spring Boot Starter* - це набір готових залежностей (dependency packages) у Spring Boot, який швидко додає необхідний функціонал у проєкт без складної конфігурації;

- *Hibernate* - фреймворк для Java, який реалізує ORM (Object-Relational Mapping), тобто дозволяє працювати з реляційними базами даних через об'єкти Java, замість того щоб писати прямі SQL-запити. Хоча *Hibernate* є окремою бібліотекою для роботи з реляційними базами даних, її часто інтегрують із *Java Spring* для забезпечення доступу до даних.

Ці бібліотеки та залежності сприяють розробці ефективних і розширюваних додатків на базі *Java Spring*, забезпечуючи широкий набір інструментів для роботи з веб-додатками, даними, безпекою, мікросервісами та іншими аспектами системи (рис. 4.5).

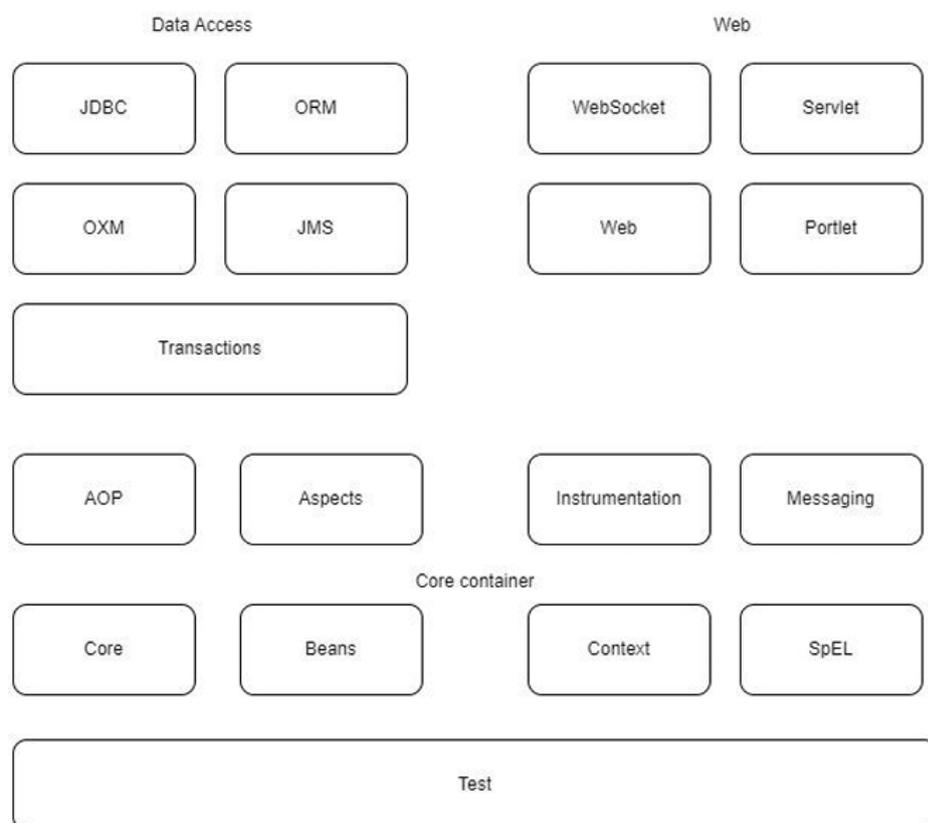


Рис. 4.5 Spring архітектура

У результаті досліджень, проведених в магістерській кваліфікаційній роботі, було визначено, що на розробку сучасних Java-додатків орієнтовані фреймворки *Spring 2.3 M3*, *Micronaut 2.0 M2* та *Quarkus 1.3.1*. *Quarkus* і *Micronaut* позиціонуються як високопродуктивні та легковагові фреймворки для мікросервісів, і cloud-native додатків. *Spring 2.3* є більш класичним, з великою

екосистемою, але може бути важчим для стартапів або мікросервісної архітектури. Micronaut та Quarkus розроблені з урахуванням менш витратних ресурсів і швидкого старту, що важливо для контейнеризації (Docker) і Kubernetes. Spring Boot має великий стек, що може впливати на час старту сервісів.

Порівняльна характеристика фреймворків приведена у табл. 4.3.

Таблиця 4.3

Порівняльна характеристика фреймворків для розробки Java-додатків

METRIC	MICRONAUT 2.0 M2	QUARKUS 1.3.1	SPRING 2.3 M3
Compile Time	1.48s	1.45s	1.33s
Test Time /mvn test	4.3s	5.8s	7.2s
Start Time Dev Mode	420ms	866ms (1)	920ms
Start Time Production java -jar myjar.jar	510ms	655ms	1.24s
Time to First Response	960ms	890ms	1.85s
Requests Per Second (2)	79k req/sec	75k req/sec	75k req/sec
Request Per Second - Xmx18m	50k req/sec	46k req/sec	46k req/sec
Memory Consumption After Load Test (- Xmx128m) (4)	290MB	390MB	480MB
Memory Consumption After Load	249MB	340MB	430MB

4.4 База даних на PostgreSQL

База даних відіграє визначальну роль у розробці прототипу криптоплатіжного шлюзу для e-commerce, оскільки забезпечує централізоване зберігання, обробку та узгодження інформації, необхідної для здійснення онлайн-платежів із використанням криптовалют.

Насамперед база даних використовується для збереження даних про користувачів, замовлення та платіжні транзакції, що є критично важливим для функціонування електронної комерції. Вона дозволяє пов'язувати конкретний платіж із замовленням, покупцем та продавцем, а також зберігати статуси платежів на різних етапах їх виконання - від ініціалізації до остаточного підтвердження у блокчейн-мережі [16].

База даних забезпечує надійність та цілісність інформації, що особливо важливо в умовах асинхронного підтвердження криптовалютних транзакцій. Завдяки підтримці транзакційності та механізмів контролю цілісності даних, система може коректно обробляти помилки, уникати дублювання платежів і гарантувати узгодженість фінансових даних між компонентами платіжного шлюзу.

У контексті e-commerce база даних також виконує роль посередника між блокчейном і веб-додатком. Вона зберігає технічні дані транзакцій (хеші, адреси гаманців, часові мітки), отримані через бібліотеки взаємодії з блокчейном, що дозволяє frontend-частині швидко отримувати актуальну інформацію про стан платежу без прямого звернення до блокчейн-мережі.

Окрім цього, база даних створює основу для масштабування та подальшого розвитку платіжної системи, зокрема реалізації аналітики, формування звітів, ведення історії платежів та інтеграції з іншими сервісами електронної комерції.

Архітектурні принципи та характерні особливості SQL і NoSQL баз даних різняться з огляду на відмінні підходи, які вони використовують для зберігання та обробки інформації.

Реляційна модель даних: SQL-бази даних ґрунтуються на реляційній моделі, у якій інформація зберігається у вигляді таблиць із чітко визначеними стовпцями та встановленими зв'язками між ними, що забезпечує впорядкованість, структурованість і логічну узгодженість даних. SQL-бази даних потребують схеми даних, у якій структура таблиць і типи даних задаються заздалегідь та є фіксованими, а внесення змін до такої схеми зазвичай є складним і трудомістким процесом. Для роботи з SQL-базами даних застосовується мова запитів SQL, що дозволяє виконувати як вибірку, так і модифікацію даних за допомогою складних запитів (рис. 4.6).

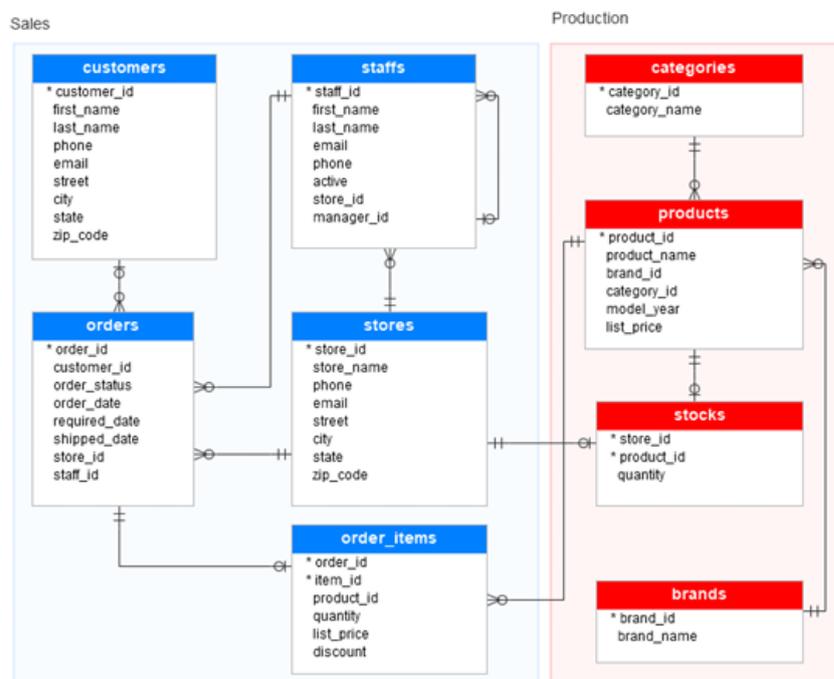


Рис. 4.6 SQL DB

NoSQL-бази даних включають в себе різні моделі зберігання інформації: документальні (наприклад, MongoDB), стовпчасті (наприклад, Apache Cassandra), ключ-значення (наприклад, Redis) та графові (наприклад, Neo4j), що дає можливість обирати оптимальну модель відповідно до типу та структури даних. NoSQL-бази даних підтримують зберігання даних без жорстко визначеної схеми або з динамічною схемою, що дозволяє змінювати структуру даних у процесі роботи без суттєвих обмежень. NoSQL-бази даних характеризуються високою продуктивністю та можливістю горизонтального масштабування, що робить їх

ефективними для обробки великих обсягів даних і високого навантаження (рис. 4.7).

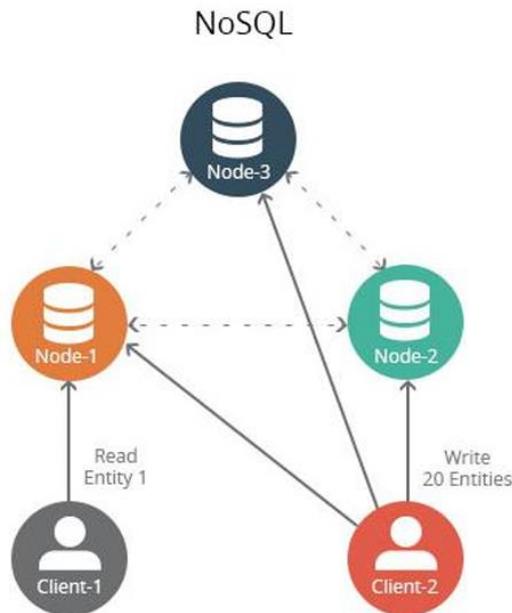


Рис. 4.7 NoSQL DB

У магістерській роботі буде використовуватися PostgreSQL - це SQL-база даних, тобто реляційна система управління базами даних (RDBMS). Варто зазначити, що PostgreSQL також підтримує деякі можливості, властиві NoSQL, а саме зберігання JSON/JSONB документів, роботу з ключ-значення через hstore, підтримку графових структур через розширення. Тому PostgreSQL часто називають гібридною базою, але її основна модель - реляційна (SQL).

На рис. 4.8 представлені порівняльні характеристики тестування різних баз даних.

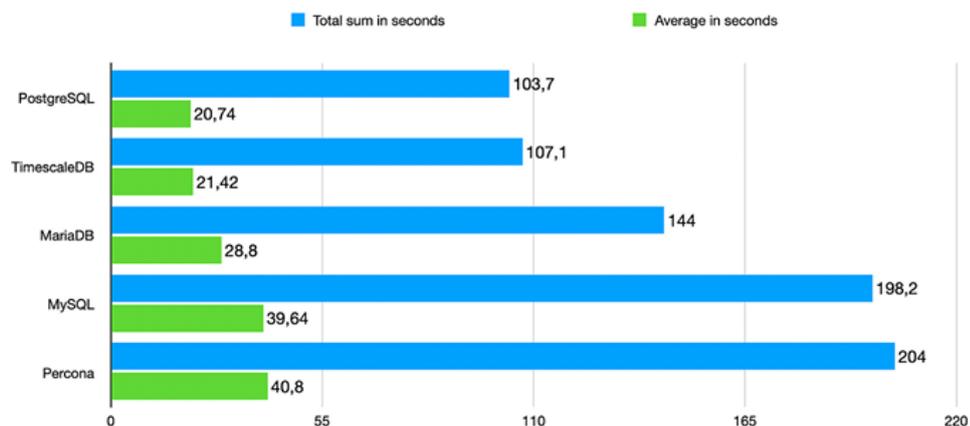


Рис. 4.8 Порівняльний аналіз баз даних

Таким чином, у прототипі криптоплатіжного шлюзу для e-commerce база даних виступає ключовим елементом, що забезпечує стабільність, надійність і функціональну повноту всієї платіжної платформи.

4.5 Використання Web3J для налаштування платежів

Web3J - це Java-бібліотека для взаємодії з Ethereum-блокчейном, яка дозволяє створювати, відправляти та відстежувати транзакції, а також працювати із смарт-контрактами. Web3J дозволяє встановлювати з'єднання з Ethereum-нодами через RPC, WebSocket або IPC, що забезпечує можливість відправки та отримання транзакцій у мережі Ethereum (рис. 4.9).

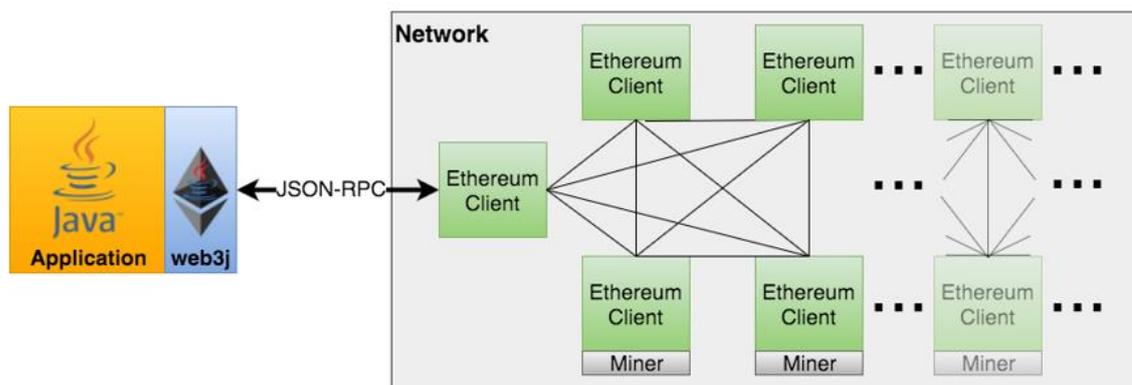


Рис. 4.9 З'єднання з Ethereum-нодами

Через Web3J можна генерувати транзакції на переведення криптовалюти з одного гаманця на інший. Бібліотека дозволяє підписувати транзакції приватними ключами користувачів або сервісними ключами - це забезпечує безпеку та аутентичність платежів.

Визначимо переваги використання Web3J:

- інтеграція з Java/Java Spring: повна сумісність з backend-частиною платіжного шлюзу;
- автоматизація процесів: відправка та перевірка транзакцій без ручного втручання;

- безпека та надійність: підпис транзакцій і перевірка блокчейном забезпечують захищеність платежів;
- підтримка смарт-контрактів: дозволяє реалізовувати складну логіку платежів, наприклад, умовні перекази або escrow-сервіси.

Визначимо використання Web3J у криптоплатіжному шлюзі для e-commerce. Web3J підключається до backend на Java Spring для створення і відправки транзакцій при оформленні замовлення. Дозволяє оновлювати статус платежів у базі PostgreSQL на основі підтверджень у блокчейні. Забезпечує інтеграцію з REST API e-commerce платформ для синхронізації платежів і замовлень. Забезпечує використання смарт-контрактів для автоматизації платежів та розподілу коштів між продавцем і платформою.

У криптовалютному платіжному шлюзі Web3J виступає як шар комунікації з блокчейном, завдяки якому ваша система може обробляти криптовалютні транзакції напряму.

Концептуальна модель прототипу платіжного шлюзу представлена на рис. 4.10. Схема процесу криптоплатежів, яка показує послідовність кроків транзакції: від ініціації платежу користувачем до підтвердження у блокчейні і оновлення статусу у системі, фокус на процесі обробки платежу представлена на рис. 4.11. Схема процесу криптоплатежів використовується, щоб зрозуміти логіку транзакцій і потоки даних між компонентами.

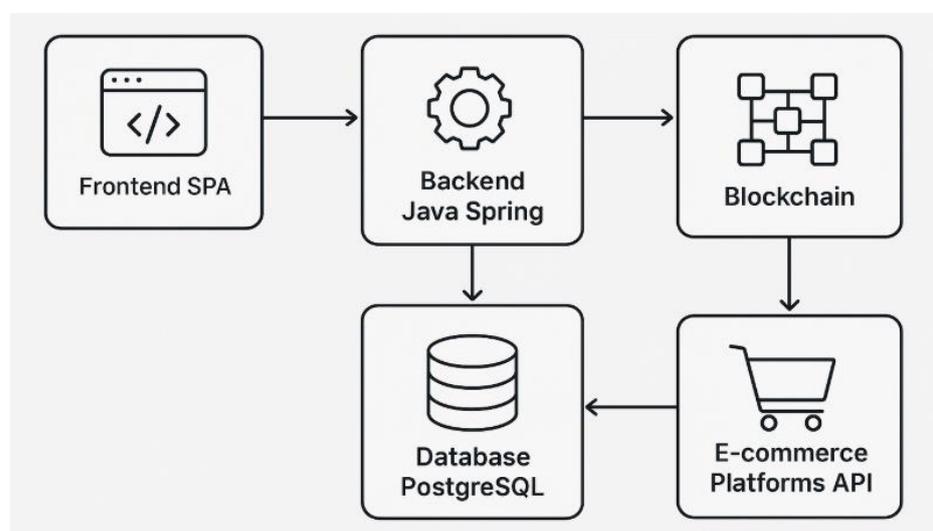


Рис. 4.10 Концептуальна модель прототипу платіжного шлюзу

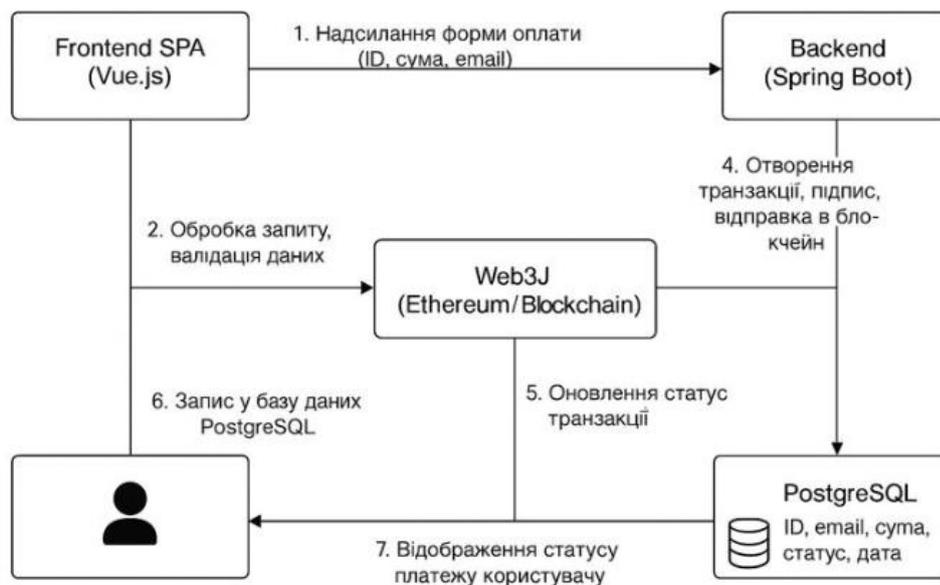


Рис. 4.11 Схема процесу криптоплатежів

Використання Web3J для налаштування платежів дозволить інтегрувати Ethereum-блокчейн у платформу та реалізувати обробку криптовалютних транзакцій у реальному часі. Використання Web3J забезпечить взаємодію із смарт-контрактами, підписання транзакцій та отримання підтверджень від мережі, що є критично важливим для надійності та безпеки платіжного шлюзу.

ВИСНОВКИ

Використання цифрових валют стає все більш актуальним та перспективним напрямком у комерційній діяльності. Сучасні дослідження показують значне зростання обсягів транзакцій із криптовалютами, що пов'язано з підвищеною інтеграцією блокчейн-технологій, глобалізацією ринку та потребою користувачів у швидких, зручних та безпечних механізмах оплати. Аналіз показав, що цифрові валюти забезпечують високу ступінь прозорості та дозволяють зменшити транзакційні витрати, що робить їх конкурентоспроможними у порівнянні з традиційними платіжними інструментами.

Огляд архітектур платіжних шлюзів дозволив визначити основні компоненти та модулі сучасних криптоплатіжних систем. Встановлено, що ефективна архітектура повинна включати інтеграцію з блокчейн-мережами, механізми обробки транзакцій, системи контролю безпеки та модулі аналітики. Важливим є забезпечення масштабованості та надійності платіжного шлюзу, що дозволяє адаптувати систему під різні обсяги та потреби бізнесу.

Технологічні стеки для побудови платіжних систем були проаналізовані з точки зору їх здатності забезпечувати безпеку, швидкість обробки транзакцій та інтеграцію з різними сервісами. Дослідження показало, що найбільш ефективними є стеки, які поєднують мікросервісну архітектуру, надійні бібліотеки для роботи з блокчейном (наприклад, Web3j для Ethereum) та сучасні засоби шифрування даних. Такий підхід дозволяє побудувати гнучку та надійну платіжну систему, здатну працювати у різних сценаріях e-commerce.

Дослідження та обґрунтування підходів до створення криптовалютного платіжного рішення для e-commerce дозволило визначити функціональні та нефункціональні вимоги до прототипу платіжного шлюзу. Функціонально система повинна забезпечувати прийом різних криптовалют, обробку транзакцій у реальному часі, інтеграцію з існуючими e-commerce платформами та надання звітності. Нефункціональні вимоги включають масштабованість, надійність, безпеку та зручність користувацького інтерфейсу.

Встановлено, що система повинна підтримувати стандарти безпеки, мати швидкий час обробки транзакцій, забезпечувати сумісність з різними платформами та надавати механізми аналітики для бізнесу. Також важливою є можливість інтеграції з існуючими процесами обліку та управління фінансами.

Вибір криптоплатіжного шлюзу для онлайн-бізнесу ґрунтувався на порівняльному аналізі існуючих рішень. Було обрано підхід, який поєднує зручність інтеграції, високу надійність, підтримку популярних криптовалют та наявність документації і прикладів реалізації (наприклад, через Web3j для взаємодії з Ethereum). Це забезпечує можливість швидкого запуску та ефективного використання платіжного шлюзу у реальному бізнес-середовищі.

Структура SPA-додатку (Single Page Application) була спроектована з урахуванням потреб користувача та принципів зручності інтерфейсу. Встановлено, що використання SPA дозволяє забезпечити високу інтерактивність, мінімізувати затримки під час обробки запитів та полегшити масштабування фронтенд-частини. Розробка передбачає логічне розділення компонентів, що підвищує підтримуваність та дозволяє легко інтегрувати додаткові функції у майбутньому. Використання RESTful підходу забезпечує стандартизовану взаємодію між frontend та backend, спрощує тестування та підтримку системи, а також створює основу для майбутніх інтеграцій із сторонніми платформами. Spring забезпечує високу стабільність роботи серверної частини та спрощує реалізацію REST API для обробки криптовалютних транзакцій.

Процедура розробки криптовалютного платіжного інструменту включає послідовність етапів: визначення вимог, проектування архітектури, вибір технологічного стеку, імплементацію та тестування, забезпечення безпеки та оптимізацію процесів. Такий системний підхід дозволяє створити ефективний прототип платіжного шлюзу, який відповідає сучасним вимогам ринку електронної комерції та забезпечує безпечне проведення криптовалютних транзакцій.

ПЕРЕЛІК ПОСИЛАНЬ

1. <https://www.geeksforgeeks.org/http-headers-keep-alive/>
2. <https://www.odinschool.com/blog/big-data/top-big-data-skills-to-future-proof-your-career-become-a-data-engineer-now>
3. <https://techreviewer.co/blog/postgresql-vs-mysql-which-relational-database-is-better>
4. <https://talentgrid.io/spring-boot-interview-questions/>
5. <https://www.knowledgefactory.net/2022/02/spring-boot-postgresql-vuejs-crud-example.html>
6. <https://www.ryadel.com/en/code-first-model-first-database-first-vs-comparison-orm-asp-net-core-entity-framework-ef-data/>
7. <https://getdevdone.com/blog/should-you-use-bootstrap-for-wordpress-theme-development.html>
8. <https://www.abtasty.com/ci-cd/>
9. <https://www.edx.org/es/learn/software-development/ibm-test-and-behavior-driven-development-tdd-bdd>
11. Ковалева Т. Становлення та шляхи удосконалення обліку криптовалют. Проблеми і перспективи розвитку підприємництва. 2020. № 24. С. 90–100.
12. Financial measures to curb speculation in cryptocurrency trading. Press Release: 23 January 2018.
13. Qassim A. Cryptocurrency: Global Accounting Leaders Seek Guidance. Bloomberg BNA. 2020.
14. Міжнародні стандарти фінансової звітності. Концептуальна основа фінансової звітності. URL: <http://www.minfin.gov.ua/news/view/mizhnarodni-standarty-finansovoizvitnostiversiia-perekladu-ukrainskoiiu-movoiu-rik?category=bjudzhet>
15. Сторчак К.П., Ткаленко О.М., Бурда Ю.О., Гарник Д.О. «Дослідження проблеми оптимізації продуктивності в мові програмування Java». Стаття у

загальногалузевому науково-виробничому журналі «Зв'язок», м.Київ - №5(165), 2023. – С.15-20.

16. D. Corradini, A. Zampieri, M. Pasqua, M. Ceccato Empirical Comparison of Black-box Test Case Generation Tools for RESTful APIs 2021 IEEE 21st International Working Conference on Source Code Analysis and Manipulation (SCAM) (2021), pp. 226-236.

17. A. Ehsan, M.A.M.E. Abuhaliqa, C. Catal, D Mishra RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions Appl. Sci., 12 (2022), p. 4369, 10.3390/app12094369

18. Andy Neumann, Nuno Laranjeiro, Jorge Bernardino An Analysis of Public REST Web Service APIs IEEE Transactions on Services Computing, 14 (4) (2021), pp. 957-970n.

19. Dessì D., Osborne F., Recupero D.R., Buscaldi D., Motta E., Sack H. AI-KG: an automatically generated knowledge graph of artificial intelligence ISWC 2020, Lecture notes in computer science, vol.12507, Springer (2020), pp. 127-143.

20. Carvalho T. Programming and mapping strategies for embedded computing runtime adaptability (Ph.D. thesis) University of Porto, Faculty of Engineering, Porto, Portugal (2019)

21. LuoK. Enhanced grey wolf optimizer with a model for dynamically estimating the location of the prey Appl. Soft Comput. (2019)

22. KambojV.K. et al. An intensify Harris Hawks optimizer for numerical and engineering optimization problems Appl. Soft Comput. (2020)

23. A new artificial bee colony algorithm employing intelligent forager forwarding strategies Appl. Soft Comput. (2020)

24. Бурда Ю.О. «Дослідження технології побудови ефективного додатку з використанням restful api». Тези доповіді на I всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу». – Київ, 28 листопада 2023 р.

25. Li Y and Jiang Z (2019). Assessing and optimizing the performance impact of the just-in-time configuration parameters - a case study on PyPy, *Empirical Software Engineering*, 24:4, (2323-2363), Online publication date: 1-Aug-2019
26. Scott Oaks " Java Performance, 2nd Edition" – Addison-Wesley, O'Reilly Media, Inc. 2020 – 714p.
27. Matt Weisfeld, " The Object-Oriented Thought Process, 5th Edition" Addison Wesley Professional 2019- 323 p.
28. A.C. Carniel et al. A generic and efficient framework for flash-aware spatial indexing *Inf. Syst.* (2019)
29. Z. He et al. GeoBeam: a distributed computing framework for spatial data *Comput. Geosci.* (2019)
30. Garcia K.D., Carvalho T., Mendes-Moreira J., Cardoso J.M., de Carvalho A.C.P.L.F. A study on hyperparameter configuration for human activity recognition Álvarez F. Martínez, Lora A. Troncoso, Muñoz J.A. Sáez, Quintián H., Corchado E. (Eds.), 14th International conference on soft computing models in industrial and environmental applications, (SOCO 2019), Springer International Publishing (2019), pp. 47-56.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ
(Презентація)

1. **Проектування REST API для backend-частини платформи** дозволило визначити чіткі ендпоінти для обробки транзакцій, управління користувачами та інтеграції з зовнішніми сервісами.
2. **Використання Axios як інструменту для роботи з RESTful API** забезпечило ефективну та безпечну передачу даних між SPA-додатком і серверною частиною. Впровадження Axios дозволяє здійснювати асинхронні запити, обробляти помилки та керувати токенами аутентифікації, що підвищує надійність роботи платформи та покращує взаємодію користувача з інтерфейсом.
3. **Особливості використання Java Spring фреймворку для розробки backend-частини платформи** демонструють переваги цього рішення: підтримка масштабованості, модульність, зручність інтеграції з базами даних та можливість реалізації комплексної бізнес-логіки.
4. **База даних на PostgreSQL** була обрана як надійний та масштабований механізм зберігання інформації. Розробка структури бази даних дозволила забезпечити ефективне зберігання даних про користувачів, транзакції та інші важливі об'єкти платформи, а також підтримку високого навантаження при роботі з великим обсягом операцій. Використання реляційної моделі даних сприяє простому аналізу та забезпечує цілісність інформації.
5. **Використання Web3J для налаштування платежів** дозволило інтегрувати Ethereum-блокчейн у платформу та реалізувати обробку криптовалютних транзакцій у реальному часі. Використання Web3J забезпечує взаємодію з смарт-контрактами, підписання транзакцій та отримання підтверджень від мережі, що є критично важливим для надійності та безпеки платіжного шлюзу.
6. **Загальна інтеграція frontend та backend-частин платформи** забезпечує створення комплексного рішення для прийому криптовалютних платежів у сфері e-commerce. Синергія між SPA-додатком, REST API, Java Spring, PostgreSQL та Web3J дозволяє побудувати систему, яка є зручною для користувача, безпечною та ефективною при обробці транзакцій.

Поетапна розробка прототипу криптовалютного платіжного шлюзу

1. Аналіз ринку та вивчення вимог

На початковому етапі необхідно провести детальний аналіз сучасних криптоплатіжних систем, визначити їх переваги та недоліки, а також виявити потреби цільової аудиторії. Паралельно формуються **функціональні та нефункціональні вимоги**, які включають перелік підтримуваних криптовалют, швидкість обробки транзакцій, безпеку системи, інтеграцію з e-commerce платформами та вимоги до користувацького інтерфейсу.

2. Проектування архітектури системи

На цьому етапі визначається структура платіжного шлюзу, розробляється схема взаємодії між компонентами. Важливим є розділення на **frontend- та backend-частини**, а також визначення модулів обробки транзакцій, взаємодії з базою даних і зовнішніми сервісами (наприклад, блокчейн-мережами). Архітектура повинна забезпечувати **масштабованість, надійність та безпеку** системи.

3. Розробка frontend-частини (інтерфейсу користувача)

Фронтенд прототипу створюється як **Single Page Application (SPA)** для забезпечення високої інтерактивності та швидкості роботи. Реалізується логіка відправки запитів до REST API, обробка відповідей сервера, відображення інформації про баланс та статус транзакцій. На цьому етапі також проєктуються інтуїтивно зрозумілі елементи інтерфейсу, які забезпечують зручність використання платіжного шлюзу.

4. Розробка backend-частини

Backend створюється з використанням **Java Spring**, що забезпечує модульність та надійність платформи. Розробляються REST API для обробки запитів від frontend, управління користувачами, обробки транзакцій та взаємодії з базою даних. Впроваджується логіка аутентифікації, обробки

помилки та контролю безпеки. На цьому етапі також підключається **Web3J** для інтеграції з блокчейном та реалізації криптовалютних платежів.

5. **Проектування та інтеграція бази даних**

Вибирається **PostgreSQL** як надійна реляційна база даних для зберігання інформації про користувачів, транзакції, статуси платежів та інші важливі дані. Створюється структура бази даних, налаштовуються зв'язки між таблицями, забезпечується цілісність та безпека даних. База даних інтегрується з backend-частиною через ORM або прямі SQL-запити.

6. **Інтеграція з блокчейном і смарт-контрактами**

Використання **Web3J** дозволяє налаштувати зв'язок платформи з Ethereum-блокчейном, підписувати транзакції, надсилати платежі та отримувати підтвердження від мережі. На цьому етапі забезпечується взаємодія прототипу з смарт-контрактами та тестування коректності проведення криптовалютних операцій.

7. **Тестування та оптимізація**

Після інтеграції всіх компонентів проводиться **тестування прототипу**: перевіряється коректність обробки транзакцій, швидкість відповіді API, робота інтерфейсу користувача, стійкість системи під навантаженням, безпека обробки даних та транзакцій. За результатами тестування вносяться необхідні виправлення та оптимізації.

8. **Документування та підготовка до демонстрації**

На завершальному етапі створюється **технічна документація**, яка описує архітектуру, структуру бази даних, REST API, механізм інтеграції з блокчейном, а також інструкції для користувачів і розробників. Прототип готовий до демонстрації та подальшого впровадження у реальному середовищі.

Доповідь дипломника: «Розробка прототипу криптовалютного платіжного шлюзу для e-commerce»

1. Вступ (Слайд 1–2)

Текст

доповіді:

«Доброго дня! Мене звати [ПІБ студента], і я представляю дипломну роботу на тему «Розробка прототипу криптовалютного платіжного шлюзу для e-commerce». Актуальність дослідження зумовлена стрімким розвитком криптовалют та збільшенням кількості онлайн-покупок, що потребують швидких, надійних та безпечних механізмів оплати. Метою роботи було створення прототипу платіжного шлюзу, який інтегрується з популярними e-commerce платформами та забезпечує обробку криптовалютних транзакцій.»

Рисунки для демонстрації:

- Графік зростання обсягів криптовалютних платежів у e-commerce.
 - Схематичне зображення сучасного ринку криптоплатежів (Рис. 1).
-

2. Аналіз існуючих рішень (Слайд 3–5)

Текст

доповіді:

«У першому розділі проведено аналіз сучасних криптоплатіжних рішень та технологій. Було розглянуто стан розвитку криптовалютних платежів у e-commerce, архітектури платіжних шлюзів та технологічні стеки для їх побудови. Також ми дослідили вимоги до безпеки та надійності транзакцій, зокрема захист приватних ключів, мультипідписні гаманці та аудитні журнали.»

Рисунки для демонстрації:

- Схема архітектури типового криптоплатіжного шлюзу (Рис. 2).
 - Таблиця з порівнянням технологічних стеків (Рис. 3).
-

3. Проектування та обґрунтування прототипу (Слайд 6–8)

Текст

доповіді:

«Другий розділ присвячений дослідженню та обґрунтуванню підходів до створення прототипу. Було визначено функціональні та нефункціональні вимоги:

підтримка кількох криптовалют, інтеграція з платформами e-commerce, висока швидкість обробки транзакцій, безпека та зручність інтерфейсу. Також було обрано оптимальний криптоплатіжний шлюз для онлайн-бізнесу та описано процедуру розробки прототипу.»

Рисунки для демонстрації:

- Блок-схема етапів розробки прототипу (Рис. 4).
 - Таблиця функціональних та нефункціональних вимог (Рис. 5).
-

4. Розробка frontend-частини (Слайд 9–11)

Текст

доповіді:

«У третьому розділі розроблено frontend-частину прототипу. Використано концепцію SPA-додатку, що забезпечує високу інтерактивність та швидкість роботи. Реалізовано механізм відправки запитів до REST API, обробку відповідей сервера та відображення інформації про баланс і транзакції. Це дозволяє користувачеві швидко та зручно взаємодіяти з платіжним шлюзом.»

Рисунки для демонстрації:

- Структура SPA-додатку (Рис. 6).
 - Схема взаємодії frontend з REST API (Рис. 7).
 - Приклад інтерфейсу користувача (скріншот) (Рис. 8).
-

5. Розробка backend-частини (Слайд 12–15)

Текст

доповіді:

«Четвертий розділ присвячено backend-частині прототипу, розробленій на Java Spring. Було спроектовано REST API для обробки транзакцій та управління користувачами, налаштовано базу даних PostgreSQL для зберігання інформації про користувачів та транзакції. Інтеграція з Ethereum-блокчейном реалізована за допомогою Web3J, що дозволяє обробляти криптовалютні платежі та взаємодіяти зі смарт-контрактами.»

Рисунки для демонстрації:

- Архітектура backend-частини (Рис. 9).

- Схема взаємодії REST API з базою даних та блокчейном (Рис. 10).
 - Приклад обробки транзакції через Web3J (Рис. 11).
-

6. Тестування та результати (Слайд 16–17)

Текст

доповіді:

«Прототип пройшов комплексне тестування: перевірялися коректність обробки транзакцій, швидкість відповіді REST API, стабільність роботи під навантаженням та безпека збереження даних. Результати показали, що розроблений платіжний шлюз працює надійно, відповідає вимогам безпеки та забезпечує зручну інтеграцію з e-commerce платформами.»

Рисунки для демонстрації:

- Діаграма тестування транзакцій (Рис. 12).
 - Таблиця результатів перевірки безпеки та продуктивності (Рис. 13).
-

7. Висновки (Слайд 18)

Текст

доповіді:

«Отже, у ході роботи був розроблений прототип криптовалютного платіжного шлюзу для e-commerce, який поєднує зручний frontend, надійний backend на Java Spring, базу даних PostgreSQL та інтеграцію з Ethereum через Web3J. Система забезпечує безпечне, швидке та ефективне проведення криптовалютних транзакцій. Дослідження підтвердило актуальність та перспективність використання таких рішень у сучасному онлайн-бізнесі.»

ВИСНОВКИ

Використання цифрових валют стає все більш актуальним та перспективним напрямком у комерційній діяльності. Сучасні дослідження показують значне зростання обсягів транзакцій із криптовалютами, що пов'язано з підвищеною інтеграцією блокчейн-технологій, глобалізацією ринку та потребою користувачів у швидких, зручних та безпечних механізмах оплати. Аналіз показав, що цифрові валюти забезпечують високу ступінь прозорості та дозволяють зменшити

транзакційні витрати, що робить їх конкурентоспроможними у порівнянні з традиційними платіжними інструментами.

Огляд архітектур платіжних шлюзів дозволив визначити основні компоненти та модулі сучасних криптоплатіжних систем. Встановлено, що ефективна архітектура повинна включати інтеграцію з блокчейн-мережами, механізми обробки транзакцій, системи контролю безпеки та модулі аналітики. Важливим є забезпечення масштабованості та надійності платіжного шлюзу, що дозволяє адаптувати систему під різні обсяги та потреби бізнесу.

Технологічні стеки для побудови платіжних систем були проаналізовані з точки зору їх здатності забезпечувати безпеку, швидкість обробки транзакцій та інтеграцію з різними сервісами. Дослідження показало, що найбільш ефективними є стеки, які поєднують мікросервісну архітектуру, надійні бібліотеки для роботи з блокчейном (наприклад, Web3j для Ethereum) та сучасні засоби шифрування даних. Такий підхід дозволяє побудувати гнучку та надійну платіжну систему, здатну працювати у різних сценаріях e-commerce.

Дослідження та обґрунтування підходів до створення криптовалютного платіжного рішення для e-commerce дозволило визначити функціональні та нефункціональні вимоги до прототипу платіжного шлюзу. Функціонально система повинна забезпечувати прийом різних криптовалют, обробку транзакцій у реальному часі, інтеграцію з існуючими e-commerce платформами та надання звітності. Нефункціональні вимоги включають масштабованість, надійність, безпеку та зручність користувацького інтерфейсу.

Встановлено, що система повинна підтримувати стандарти безпеки, мати швидкий час обробки транзакцій, забезпечувати сумісність з різними платформами та надавати механізми аналітики для бізнесу. Також важливою є можливість інтеграції з існуючими процесами обліку та управління фінансами.

Вибір криптоплатіжного шлюзу для онлайн-бізнесу ґрунтувався на порівняльному аналізі існуючих рішень. Було обрано підхід, який поєднує зручність інтеграції, високу надійність, підтримку популярних криптовалют та наявність документації і прикладів реалізації (наприклад, через Web3j для

взаємодії з Ethereum). Це забезпечує можливість швидкого запуску та ефективного використання платіжного шлюзу у реальному бізнес-середовищі.

1. **Процедура розробки криптовалютного платіжного інструменту** включає послідовність етапів: визначення вимог, проектування архітектури, вибір технологічного стеку, імплементацію та тестування, забезпечення безпеки та оптимізацію процесів. Такий системний підхід дозволяє створити ефективний прототип платіжного шлюзу, який відповідає сучасним вимогам ринку електронної комерції та забезпечує безпечне проведення криптовалютних транзакцій.

Висновки за Розділ 3 та Розділ 4

7. **Дослідження ринку криптовалютних платіжних систем** показало, що сучасний ринок характеризується високим рівнем конкуренції та динамічним розвитком технологій. Основні тенденції включають інтеграцію платіжних шлюзів із популярними e-commerce платформами, підтримку широкого спектру криптовалют, а також розвиток децентралізованих фінансових інструментів (DeFi). Аналіз існуючих рішень дозволив визначити ключові фактори успіху: зручність для користувача, безпека транзакцій, швидкість обробки платежів та можливість масштабування.
8. **Структура SPA-додатку (Single Page Application)** була спроектована з урахуванням потреб користувача та принципів зручності інтерфейсу. Встановлено, що використання SPA дозволяє забезпечити високу інтерактивність, мінімізувати затримки під час обробки запитів та полегшити масштабування фронтенд-частини. Розробка передбачає логічне розділення компонентів, що підвищує підтримуваність та дозволяє легко інтегрувати додаткові функції у майбутньому.
9. **Механізм відправлення запитів до REST API** було реалізовано з використанням сучасних підходів для забезпечення надійності та безпеки передачі даних. Було проаналізовано способи аутентифікації, обробки

помилки та асинхронної взаємодії з сервером, що дозволяє забезпечити стабільну роботу платіжного шлюзу та підтримку високого навантаження.

10. **Проектування REST API для backend-частини платформи** дозволило визначити чіткі ендпоінти для обробки транзакцій, управління користувачами та інтеграції з зовнішніми сервісами. Використання RESTful підходу забезпечує стандартизовану взаємодію між frontend та backend, спрощує тестування та підтримку системи, а також створює основу для майбутніх інтеграцій із сторонніми платформами.
11. **Використання Axios як інструменту для роботи з RESTful API** забезпечило ефективну та безпечну передачу даних між SPA-додатком і серверною частиною. Впровадження Axios дозволяє здійснювати асинхронні запити, обробляти помилки та керувати токенами аутентифікації, що підвищує надійність роботи платформи та покращує взаємодію користувача з інтерфейсом.
12. **Особливості використання Java Spring фреймворку для розробки backend-частини платформи** демонструють переваги цього рішення: підтримка масштабованості, модульність, зручність інтеграції з базами даних та можливість реалізації комплексної бізнес-логіки. Spring забезпечує високу стабільність роботи серверної частини та спрощує реалізацію REST API для обробки криптовалютних транзакцій.
13. **База даних на PostgreSQL** була обрана як надійний та масштабований механізм зберігання інформації. Розробка структури бази даних дозволила забезпечити ефективне зберігання даних про користувачів, транзакції та інші важливі об'єкти платформи, а також підтримку високого навантаження при роботі з великим обсягом операцій. Використання реляційної моделі даних сприяє простому аналізу та забезпечує цілісність інформації.
14. **Використання Web3J для налаштування платежів** дозволило інтегрувати Ethereum-блокчейн у платформу та реалізувати обробку криптовалютних транзакцій у реальному часі. Використання Web3J забезпечує взаємодію з смарт-контрактами, підписання транзакцій та

отримання підтверджень від мережі, що є критично важливим для надійності та безпеки платіжного шлюзу.

15. Загальна інтеграція frontend та backend-частин платформи забезпечує створення комплексного рішення для прийому криптовалютних платежів у сфері e-commerce. Синергія між SPA-додатком, REST API, Java Spring, PostgreSQL та Web3J дозволяє побудувати систему, яка є зручною для користувача, безпечною та ефективною при обробці транзакцій.