

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Інтеграція платіжних систем для фіатних і
криптовалютних оплат на платформі створення
оголошень»**

Виконав студент групи ПЗм-22-1
КИЯШЕНКО Анатолій Сергійович

Керівник роботи:
ЧИЖМОТРЯ Олексій Володимирович

Рецензент:
МОРОЗОВ Андрій Васильович

Житомир – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
 ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
 ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

зав. кафедри інженерії

програмного забезпечення

Тетяна ВАКАЛЮК

«18» вересня 2023 р.

ЗАВДАННЯ на кваліфікаційну роботу

Здобувач вищої освіти: **КИЯШЕНКО Анатолій Сергійович**

Керівник роботи: **ЧИЖМОТРЕ Олексій Володимирович**

Тема роботи: **«Інтеграція платіжних систем для фіатних і криптовалютних оплат на платформі створення оголошень»,**

затверджена Наказом закладу вищої освіти від **«18» вересня 2023 р. № 479/с**

Вихідні дані для роботи: об'єктом дослідження є платіжні системи, які дозволяють здійснювати оплату як криптовалютами, так і фіатними грошима.

Предметом дослідження є процеси інтеграції криптовалют і фіатних грошей у платіжні системи, їх взаємодія, порівняння та вивчення можливостей та перспектив їх застосування для оплати на веб-сайтах.

Консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Чижмотря О.В.	22.09.2023р.	22.09.2023р.
2	Чижмотря О.В.	22.09.2023р.	22.09.2023р.
3	Чижмотря О.В.	22.09.2023р.	22.09.2023р.

РЕФЕРАТ

Кваліфікаційна робота присвячена інтеграції платіжних систем в платформу онлайн оголошень для прийому фіатних та криптовалютних оплат, а також впливу наявності цих систем на конверсію сайту.

Кваліфікаційна робота складається з програмного комплексу платформи створення онлайн оголошень та пояснювальної записки.

Пояснювальна записка складається з вступу, трьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 68 сторінках друкованого тексту.

Пояснювальна записка має 7 сторінок додатків. Список використаних джерел містить 31 найменування і займає 3 сторінки. В роботі наведено 51 рисунок. Загальний обсяг роботи – 77 сторінок.

У першому розділі було проаналізовано задачу, засоби та методи її вирішення. Виконаний аналіз існуючих платіжних систем. Поставлено технічне завдання та вимоги до програми.

У другому розділі проведено проектування програмного продукту на логічному рівні, описані алгоритми взаємодії окремих модулів системи та створені діаграми. Наведено опис програмного коду реалізації.

У третьому розділі наведено інтерфейс реалізованої системи, її порядок роботи. Проведено тестування реалізованих алгоритмів та прийому оплат.

Висновок містить в собі результати виконаної роботи розробки платформи створення онлайн оголошень з інтегрованими платіжними системами.

У додатку представлені вимоги та лістинг коду розробленого програмного продукту.

Ключові слова: ВЕБ-ДОДАТОК, ВЕБ-САЙТ, ПЛАТІЖНІ СИСТЕМИ, КРИПТОВАЛЮТА, СЕРВЕР, БАЗИ ДАНИХ, ДАНІ.

ABSTRACT

The qualification work is dedicated to the integration of payment systems for accepting fiat and cryptocurrency payments, as well as the impact of the presence of these systems on the conversion of the website.

The qualification work consists of a software complex of the online advertising creation platform and an explanatory note.

The explanatory note consists of an introduction, three chapters, conclusions, a list of used literature, and an appendix.

The textual part is presented on 68 pages of printed text.

The explanatory note has 7 pages of attachments. The list of used sources contains 31 titles and occupies 3 page. The work includes 51 figures. The total volume of the work is 77 pages.

In the first chapter, the task, means, and methods of its solution were analyzed. An analysis of existing payment systems was carried out. Technical tasks and requirements for the program were set.

In the second chapter, the design of the software product at the logical level was carried out, the algorithms of interaction of individual system modules were described, and diagrams were created. A description of the implementation's program code is provided.

In the third chapter, the interface of the implemented system and its workflow are presented. Testing of the implemented algorithms and payment acceptance was conducted.

The conclusion contains the results of the work done on developing the platform for creating online advertisements with integrated payment systems.

The appendix presents the requirements and the listing of the code of the developed software product.

Key words: WEB APPLICATION, WEBSITE, PAYMENT SYSTEMS, CRYPTOCURRENCIES, SERVER, DATABASES, DATA.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	12
1.1.Обґрунтування вибору інструментальних засобів	12
1.2.Аналіз існуючих платіжних систем для оплати фіатними та крипто валютами	14
1.3.Аналіз вимог та постановка задачі.....	18
Висновки до розділу 1	21
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ВПЛИВУ КРИПТОВАЛЮТНОЇ ОПЛАТИ НА ПРИВАБЛИВІСТЬ САЙТУ ТА ПРОЕКТУВАННЯ ПЛАТФОРМИ СТВОРЕННЯ ОНЛАЙН ОГолошень	23
2.1.Аналіз існуючих платіжних систем для оплати фіатними та крипто валютами	23
2.2.Визначення варіантів використання та об'єктно-орієнтованої структури системи.....	31
2.3.Проектування та реалізація алгоритмів роботи системи.....	34
2.4.Інтеграція платіжних систем для прийому банківських карток та криптовалюти	42
Висновки до розділу 2.	43
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ ТА ІНТЕГРАЦІЯ ПЛАТІЖНИХ СИСТЕМ	45
3.1.Порядок встановлення та налаштування параметрів системи.....	45
3.2.Структура інтерфейсу платформи створення онлайн оголошень	46
3.3.Тестування інтеграції платіжних систем для оплати гривнею та крипто валютами	53
3.4.Тестування роботи програмного комплексу	56
Висновки до розділу 3	63

ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66
ДОДАТКИ.....	69

ВСТУП

Актуальність теми. Сучасний світ фінансів зазнає значних трансформацій, особливо у контексті зростаючого впливу криптовалют та технологій блокчейну. Інтеграція цих нових фінансових інструментів із традиційними платіжними системами відкриває нові горизонти можливостей, але також ставить перед нами ряд викликів.

Однією з найбільш значущих переваг криптовалют є сприяння фінансовій інклюзії. Криптовалюти та блокчейн-технології можуть надати доступ до фінансових послуг мільйонам людей, які не мають можливості користуватися традиційними банківськими послугами. Це особливо важливо для жителів віддалених або малорозвинених регіонів. Також, криптовалюти можуть знижувати вартість трансграничних переказів, спрощуючи процеси глобальної торгівлі та особистих переказів [2].

Інтеграція криптовалют з традиційними платіжними системами відкриває перед бізнесом нові можливості. Вона сприяє доступу до нових ринків, зниженню витрат на транзакції та підвищенню ефективності міжнародних платежів. Крім того, блокчейн може забезпечити прозорість та безпеку в фінансових операціях, що є ключовим фактором для залучення інвестицій та підтримки довіри клієнтів.

Технології блокчейну та криптографії постійно розвиваються, відкриваючи нові можливості для створення ефективних, безпечних та прозорих платіжних систем. Ці технології відіграють ключову роль у формуванні майбутньої фінансової інфраструктури, забезпечуючи стійкість та адаптивність до постійно змінюваних умов ринку.

Однак, поряд з можливостями, криптовалюти ставлять перед нами ряд викликів, особливо у сфері безпеки та регулювання. Існує необхідність розробки та впровадження стандартів безпеки, щоб захистити користувачів від шахрайства та забезпечити стабільність криптовалютних систем. Також,

потрібно адаптувати існуючі регуляторні рамки для врахування особливостей криптовалют, зберігаючи їх переваги, такі як децентралізація та ефективність.

Криптовалюти та блокчейн також можуть викликати значні соціально-економічні зміни. Вони мають потенціал радикально змінити спосіб, яким ми ведемо бізнес, управляємо фінансами та здійснюємо економічні транзакції. Це вимагає подальших досліджень та розвитку, щоб максимально реалізувати їх потенціал та мінімізувати можливі ризики.

Сучасний розвиток цифрових технологій і зростаюче прийняття криптовалют у якості платіжного засобу ставлять перед нами завдання інтеграції різних платіжних систем у одну єдину ефективну платформу. Ця робота зосереджена на аналізі та розробці методів інтеграції платіжних систем, які підтримують як криптовалюти, так і фіатні гроші, для платформи онлайн оголошень.

На сьогодні цифровий ринок просто кишить різними цифровими монетами і токенами, що беруть активну участь у багатьох фінансових транзакціях. Bitcoin, Litecoin, Ripple, Tether, Dogecoin та багато інших криптовалют набирають все більшої популярності серед користувачів практично у всьому світі. Якщо раніше більшість людей ставилися до криптовалют з великим побоюванням та зневагою, то зараз ситуація кардинально змінилася. Немає країни, в якій немає хоч одного володаря криптовалют. З моменту свого випуску варіанти для їх використання були дуже обмежені, проте тепер вони широко поширені і застосовуються в багатьох сферах діяльності людини, в тому числі і в онлайн-торгівлі або для оплати послуг [3].

Оплата криптовалютами в онлайн платформі оголошень може бути важливою конкурентною перевагою, ось кілька причин, чому:

1. Розширення аудиторії та географічного охоплення:

Приймаючи платежі у криптовалютах, відкривається ринок для тих, хто використовує криптовалюту, а це глобальна аудиторія. Люди з усього світу

можуть легко здійснювати оплату без обмежень, пов'язаних із конвертацією валют.

2. Прискорення транзакцій та низькі комісії:

Криптовалютні транзакції зазвичай швидші та мають нижчі комісії порівняно з традиційними банківськими платежами. Це забезпечить користувачам швидкий та вигідний спосіб оплати [2].

3. Безпека та анонімність:

Криптовалютні платежі забезпечують високий рівень безпеки та анонімності. Користувачі можуть використовувати криптовалютні гаманці без необхідності розкривати особисту фінансову інформацію.

4. Інноваційний імідж:

Прийняття криптовалют в якості платіжного засобу свідчить про інноваційний підхід та готовність йти в ногу з сучасними тенденціями. Це сприяє позитивному сприйняттю вашого проєкту та залучає увагу нових користувачів.

5. Сприяння розвитку криптоспільноти:

Шляхом включення криптовалют до платіжної системи ви сприяєте розвитку криптоспільноти та сприяєте прийняттю криптовалют в реальному світі.

6. Уникнення фінансових обмежень:

Деякі користувачі можуть знаходитися в країнах, де доступ до традиційних фінансових послуг обмежений або незручний. Прийняття криптовалют дозволяє їм здійснювати оплату без таких обмежень.

Загалом, можливість оплати криптовалютами відкриває нові можливості та привертає аудиторію, яка цінує інновації та гнучкість у платіжних опціях.

Мета дослідження. Метою дослідження є глибоке вивчення та аналіз інтеграції платіжних систем для оплати криптовалютами та фіатними грошима, а також оцінка їхніх перспектив використання в сучасному економічному просторі.

Завдання дослідження:

1. Аналіз існуючих платіжних систем, що підтримують оплату криптовалютами та фіатними грошима.
2. Вивчення технологічних аспектів інтеграції криптовалют і фіатних грошей у платіжні системи.
3. Проведення дослідження впливу криптовалютної оплати на конверсію веб-сайту.
4. Створення платформи для онлайн оголошень, що буде містити платні функції.
5. Інтегрувати платіжні системи для поповнення балансу гривнею та криптовалютами.

Об'єкт дослідження. Об'єктом дослідження є платіжні системи, які дозволяють здійснювати оплату як криптовалютами, так і фіатними грошима.

Предмет дослідження. Предметом дослідження є процеси інтеграції криптовалют і фіатних грошей у платіжні системи, їх взаємодія, порівняння та вивчення можливостей та перспектив їх застосування для оплати на веб-сайтах.

Методи дослідження:

1. Аналіз літератури:

Вивчення існуючих наукових праць, статей, книг, журнальних публікацій, які стосуються криптовалют, фіатних грошей, блокчейну, та платіжних систем. Це допоможе сформувати теоретичну базу для дослідження.

2. Компаративний аналіз (порівняльний аналіз):

Порівняння різних платіжних систем, що використовують криптовалюти та фіатні гроші, з метою виявлення їхніх переваг та недоліків, а також оцінки ефективності інтеграції.

3. Кейс-стаді (аналіз випадків):

Дослідження конкретних випадків використання платіжних систем у різних секторах, що допоможе зрозуміти практичні аспекти їх функціонування та інтеграції.

4. Емпіричні дослідження:

Збір та аналіз первинних даних через опитування, інтерв'ю, анкетування серед фахівців, користувачів платіжних систем та інших зацікавлених сторін.

Наукова новизна. Дослідження вносить вклад у розвиток аналітичних підходів та методик для оцінки процесів інтеграції криптовалютних та фіатних платіжних систем. Це охоплює як технічні аспекти, так і економічні, регуляторні та соціальні виміри. Проаналізовано вплив криптовалют на традиційні фінансові платіжні системи, що є порівняно новим напрямком досліджень. Основне зосередження йде на соціально-економічних аспектах інтеграції, включаючи вплив на фінансову інклюзію, зміни у споживчому поведінці та на глобальні економічні процеси. Дослідження інтегрує знання та методології з різних дисциплін, включаючи економіку, фінанси, технології та право, що дозволяє комплексно підійти до вивчення теми та виявити нові зв'язки та взаємодії.

Практичне значення.

- Результати дослідження допоможуть вдосконалити процеси міжнародних платежів, знижуючи транзакційні витрати та спрощуючи фінансові операції на глобальному рівні та забезпечують цінну інформацію для підприємств та інвесторів, які планують адаптувати свої бізнес-моделі під нові фінансові технології.
- Розробка та впровадження інтегрованих платіжних рішень може допомогти забезпечити більш широкий доступ до фінансових послуг, особливо в недостатньо обслуговуваних або віддалених регіонах.

Апробація результатів роботи. Матеріали кваліфікаційної роботи доповідалися на III Всеукраїнській науково-практичній Інтернет-конференції «СУЧАСНІ КОМП'ЮТЕРНІ ТА ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ», 02-19 грудня 2022 р [19, 20].

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Обґрунтування вибору інструментальних засобів

Серверна частина проекту розроблятиметься на PHP, високорівневій скриптовій мові програмування. Основні переваги PHP включають швидкість розробки, простоту синтаксису, велику спільноту розробників, постійний розвиток, а також наявність численних модулів та фреймворків. З іншого боку, серед недоліків відзначаються відсутність багатопотоковості і обмеженість застосування лише до веб-додатків.

PHP є популярним вибором у сфері веб-розробок. Коли PHP-код обробляється на сервері, він перетворюється в HTML, який потім відправляється до клієнта. На відміну від JavaScript, клієнт не бачить PHP-коду, а отримує лише готовий HTML, що збільшує безпеку, але може знижувати інтерактивність веб-сторінок. У цьому проекті PHP використовується як серверна мова, тоді як фронтенд розробляється на React. Замість генерації HTML-сторінок PHP буде повертати дані у форматі JSON.

React — це відкрита JavaScript бібліотека, призначена для створення користувацьких інтерфейсів. Її головне завдання — вирішувати проблеми оновлення вмісту веб-сторінок при розробці односторінкових застосунків. Розробляється компаніями Facebook, Instagram та спільнотою незалежних розробників [31].

React дозволяє розробляти великі веб-застосунки, які ефективно працюють з даними, що змінюються у часі, і це можливо без необхідності перезавантаження сторінки. Ціль React — забезпечити швидкість, простоту і масштабованість. React фокусується на користувацькому інтерфейсі застосунків, відповідаючи за розділ "представлення" у шаблоні MVC. Він може бути інтегрований з іншими JavaScript бібліотеками або фреймворками, такими як AngularJS. У моєму проекті, який базується на шаблоні MVC, React буде ідеальним вибором для розробки інтерфейсів [10].

У проєкті також застосовуватиметься фреймворк Symfony для PHP, що підтримує шаблон MVC. Symfony — це відкритий фреймворк для веб-застосунків, який включає багато пакунків, використовуваних іншими фреймворками [28]. Наприклад, Symfony Monolog для логування застосовується у Magento. Оскільки серверна частина проєкту буде працювати як REST API, буде використаний фреймворк Api Platform, базований на Symfony [6][27]. Api Platform оброблятиме і повертатиме дані у форматі JSON. React відповідатиме за динамічне та асинхронне відображення цих даних користувачу, без необхідності перезавантаження сторінки в браузері.

Основною базою даних буде виступати MySQL. MySQL це реляційна система керування базою даних, що є однією з найпопулярніших в світі СКБД. Вона дозволяє зберігати велику кількість даних та створювати важкі вибірки, при цьому швидко їх виконує.

Для запису статистичних даних, таких як логи та тіла запитів від платіжних шлюзів, буде використана MongoDB. MongoDB це так звана NoSQL база даних, що дозволяє швидко записувати нові дані. Проте вибірки проводяться дещо повільніше, ніж у SQL баз даних. Саме тому MongoDB вибрана для записів таких даних, що будуть часто записуватись, проте не дуже часто переглядатись [19].

Проєкт буде розгорнуто за допомогою Docker. Docker — програмне забезпечення для автоматизації розгортання та керування програмами в середовищах з підтримкою контейнеризації, контейнеризатор додатків [1]. Дозволяє "упакувати" додаток з усім його оточенням і залежностями в контейнер, який може бути розгорнутий на будь-якій Linux-системі за допомогою контрольних груп в ядрі, а також надає набір команд для управління цими контейнерами [20].

В якості середовища розробки виступить PhpStorm. PhpStorm слугує середовищем розробки і є комерційною, крос-платформною інтеграційною системою для PHP. Це інтелектуальний редактор, який підтримує PHP, HTML, і

JavaScript, надаючи функції аналізу коду, запобігання помилкам і автоматизованого рефакторингу для PHP і JavaScript. PhpStorm має функцію автодоповнення коду, сумісну з PHP версій 5.3 до 8.1. Включає в себе повноцінний SQL-редактор із можливістю редагування результатів запитів. Включення WebStorm гарантує ефективність як для PHP, так і для розробки на React.

1.2. Аналіз існуючих платіжних систем для оплати фіатними та крипто валютами

Для можливості оплати криптовалютою необхідно інтегрувати сервіс, що дозволяє приймати такі платежі.

Перший сервіс, який я вирішив розглянути, є NowPayments. NOWPayments - це сервіс, який дозволяє всім бажаючим приймати криптовалюту на своїх веб-сайтах, інтернет-магазинах та акаунтах у соціальних мережах. Це некастодіальний сервіс, що означає, що він ніяк не зберігає ваші кошти. NOWPayments підтримує понад 50 криптовалют і пропонує низькі комісії за транзакції.

NOWPayments була заснована в 2019 році командою, яка створила сервіс обміну криптовалют ChangeNOW. Їх місія полягає в тому, щоб надати криптовалютний платіжний шлюз, який легко використовувати та вбудовувати в існуючі веб-сайти. Вони вважають, що кожен повинен мати можливість приймати криптовалютні платежі або криптовалютні пожертви для своїх інтернет-магазинів, благодійних організацій або будь-якого іншого бізнесу.

Вони надають простий, легкий в інтеграції сервіс, і єдиний крок, необхідний для прийому криптоплатежів - це скопіювати фрагмент HTML-коду. Транзакції прив'язані до облікового запису NOWPayments і ключа API. Вони пропонують гнучкі рішення, наприклад, приймати платежі в різних монетах або конвертувати всі отримані криптоактиви у наявні криптовалюти.

Автоматичний обмін здійснюється за допомогою власного сервісу обміну криптовалют ChangeNOW [27].

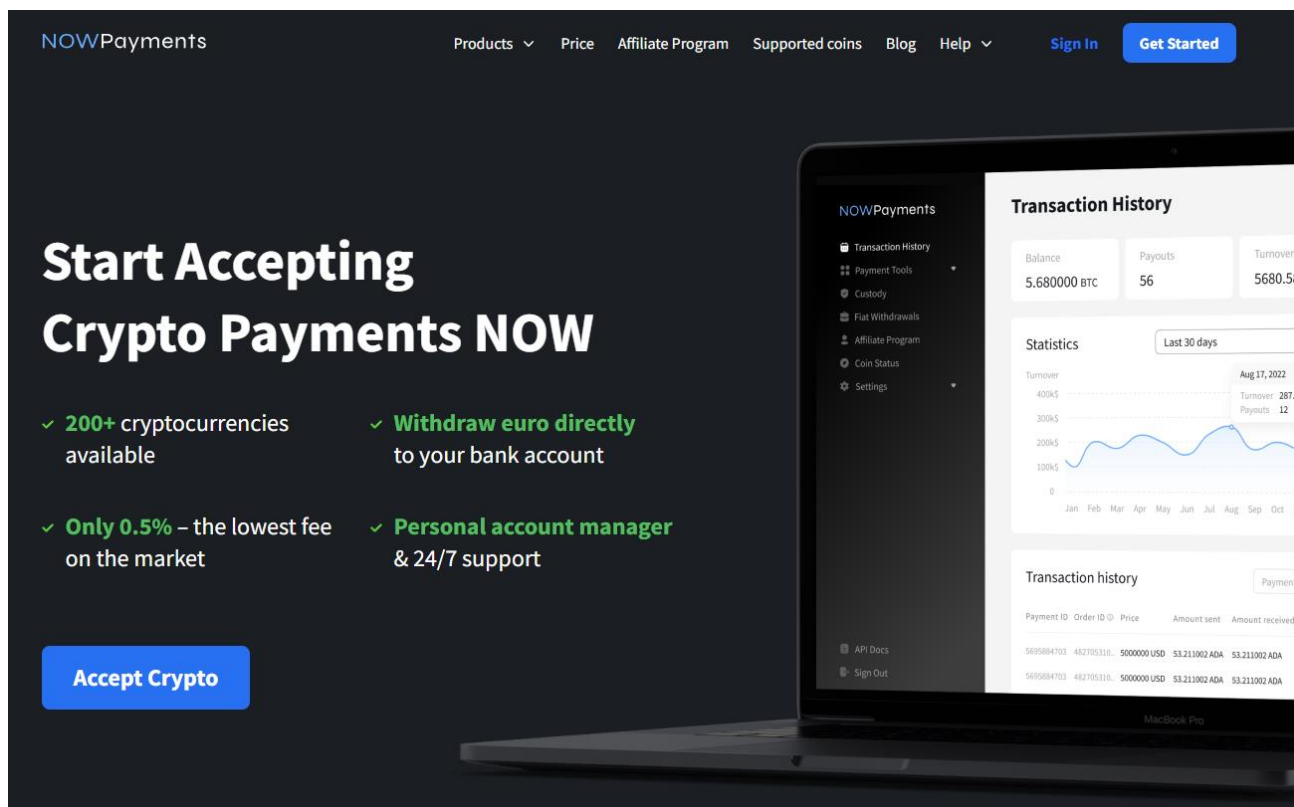


Рис. 1.1. Головна сторінка сайту NowPayments

Другим сервісом є Асептсойн. Асептсойн — це передовий сервіс для прийому платежів у криптовалюті, що дає можливість продавцю товарів або послуг в Інтернеті отримувати як оплату BTC, USDT, MATIC, HZM, BAND, XRP, DASH, DOGE, ETH, SHIBA, CHZ, XMR, TRX, BNB, XLM, SNX та багато інших криптовалют.

Платформа Асептсойн не пов'язана зі збереженням, зберіганням або використанням ваших цифрових активів. Сервіс виступає в ролі деякого посередника, який сприяє швидкому прийому оплати за покупки ваших клієнтів і виведенні цих коштів у потрібній і зручній для користувача валюті, будь то фіат (долари, євро, гривні, фунти тощо) або криптовалюти.

Крім платежів у крипті, через Асептсойн також можливо проводити масові платежі, наприклад, для виплат зарплати своїм співробітникам, оплати комісій, дивідендів тощо. Такі платежі в рази заощадають час і виключають

помилки, які часто виникають при проведенні кількох десятків, а то й сотень транзакцій.

Місія Асептсоеін — надати власникам сайтів простий у використанні та інтегрований у застосуванні сервіс для прийому криптовалюти на веб-сайтах власників, а також проведення масових виплат (батчів). Поліпшення та автоматизація бізнесу можливі, якщо звернутися за допомогою до таких сервісів, як наш.

У платформи існує «Партнерська програма», що допоможе не тільки вигідно співпрацювати обом сторонам, а й пасивно заробляти кошти нашим клієнтам.

Вони також пропонують корисні плагіни для прийому криптовалютних платежів в інтернет-магазинах електронної комерції, таких як Magento2, OpenCart, WooCommerce, Shopify та інших. За допомогою простого плагіна можна додати можливість оплати криптовалютою для оформлення замовлення у інтернет-магазині [28].

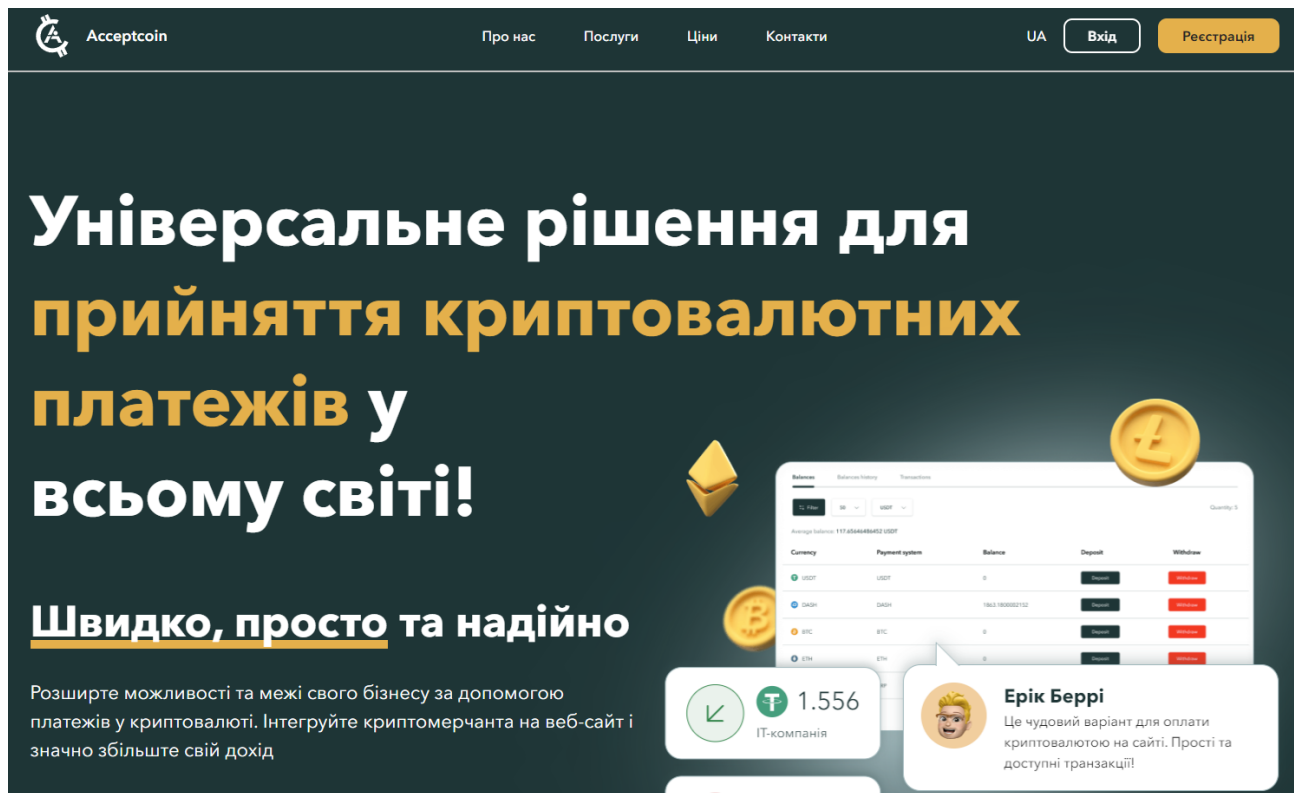


Рис. 1.2. Головна сторінка сайту Асептсоеін

Також розглянемо криптовалютний платіжний шлюз ExSoft.



Integrations

Crypto license ▾

About ▾

Support

Pricing

News

Get started

Need a ready-made **crypto** business? Use Exsoft's on-box solution!

If you want to have your own business in the cryptosphere, then take advantage of Exsoft's out-of-the-box services. Exsoft is your one-stop shop providing turn-key cryptocurrency exchanger creation.

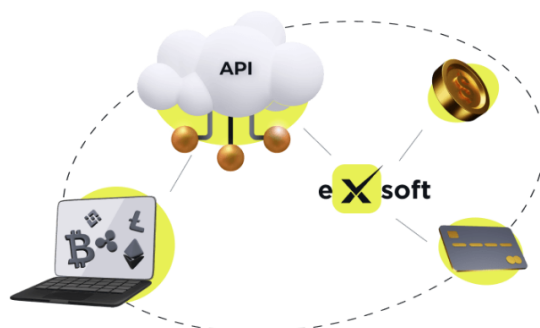
[Read more](#)


Рис. 1.3. Головна сторінка сайту Exsoft

Криптовалютний платіжний шлюз Exsoft – це спеціальний сервіс, який дозволяє здійснювати платежі за допомогою криптовалюти. Він інтегрується з обмінником криптовалют і дозволяє користувачам здійснювати платежі в криптовалюті при покупці товарів або послуг.

Платіжний шлюз у криптовалюті працює так само, як і звичайний платіжний шлюз, але замість обробки традиційних платіжних платежів він обробляє платежі в криптовалюті. Він може підтримувати різні криптовалюти та блокчейни, такі як Bitcoin, Ethereum, Litecoin та інші.

Щоб здійснити платіж через платіжний шлюз криптовалюти, користувач зазвичай повинен вибрати відповідний спосіб оплати, вказати адресу гаманця та надіслати транзакцію на вказану адресу. Після цього платіжний шлюз виконує необхідні перевірки та підтвердження, а потім переказує кошти на адресу, вказану продавцем.

Основні переваги використання криптовалютного платіжного шлюзу включають швидкість і низькі комісії за транзакції, можливість здійснювати міжнародні платежі без звернення до банківської системи, а також безпеку та прозорість блокчейну.

Для підключення платіжного шлюзу криптовалюти можуть знадобитися спеціальні знання з програмування та крипто валюти [29].

Для інтеграції я обрав Exsoft. Цей сервіс володіє не тільки прийомом криптовалюти, а і звичних фіатних валют. Підключення подібних систем вимагає цілу низку верифікацій та документів. Я отримав доступ до Exsoft, що є суттєвою перевагою. Також цей криптовалютний шлюз володіє усіма функціями, що потрібні для інтеграції оплати. Тут є більше 50 криптовалют, низькі комісії, а також можливість отримати поточний курс фіатних та криптовалют.

1.3. Аналіз вимог та постановка задачі

Даний документ описує високо рівневі вимоги, якими повинна володіти платформа онлайн оголошень, щоб мати можливість безперервно виконувати покладені на неї функції, тим самим відповідно задовольнивши всі зазначені стандарти, специфікації та інші формальні документи. Зазначені умови, яких повинен дотримуватись користувач для роботи з даною інформаційною системою та вказані основні ділові переваги даної системи. Приведені можливості системи, обмеження та показники якості.

Бізнес-вимоги

1. **Основні цілі:** проект створюється з метою надання можливості розміщення онлайн оголошень для покупки, продажу або обміну товарів та послуг.
2. **Консультації:** надання консультацій зовнішнім користувачам щодо користування сайтом засобами спілкування через онлайн чат зі службою підтримки.
3. **Представлення проекту:** проект буде реалізовано у вигляді сайту, що містить структуровану інформацію по даній тематичній області.
4. **Адміністрація та адміністративна панель:** надається можливість керування інформацією, що розміщена на сайті певній групі користувачів (адміністратори, служба підтримки).

5. Платні послуги: можливість підняти своє оголошення чи надати статус VIP на обмежений термін.
6. Оплата послуг як карткою, так і криптовалютою: прийом гривні та популярних криптовалют, а саме USDT, BTC, ETH та TRX.

Вимоги користувачів

1. Зовнішні користувачі:

- двосторонній зв'язок зі службою підтримки в онлайн режимі;
- отримання інформації відносно активних оголошень (на сайті);
- перегляд рейтингу (оцінка інших користувачів) власника оголошення (на сайті);
- покупець має можливість залишити коментар та оцінити надані послуги на після покупки;
- перегляд останніх новин сайту: інформація щодо нововведених функцій інформаційної системи доступних користувачеві, інформація щодо останніх найбільших виконаних домовленостей, що відбулися за участі даної інформаційної системи;
- здійснення розрахунків за платні послуги;
- можливість розрахунку карткою та криптовалютою.

2. внутрішні користувачі – адміністратор, модератор, служба підтримки:

- премодерація оголошень;
- ручна перевірка переписки покупця та продавця;
- можливість блокування та м'якого видалення користувача;
- двосторонній зв'язок з користувачами, що потребують допомоги в онлайн режимі.

3. Характеристика об'єкта комп'ютеризації:

Користувач покупець буде мати можливість знайти товар чи послугу, що його цікавить за допомогою каталогу чи пошуку з фільтрацією, вибрати активне оголошення, зв'язатися з продавцем через онлайн чат та заключити угоду на покупку. Користувач продавець буде мати можливість виставити

оголошення зі своїм товаром чи послугою, яку він хоче надати користувачу покупцю, редагувати свої оголошення та користуватися платними послугами сайту для покращення ранжування своїх оголошень. Користувачі також можуть продивитися поширення запитання (FAQ) та зв'язатися зі службою підтримки онлайн.

Функціональні вимоги

1. Авторизація користувачів в системі: В системі повинна бути представлена можливість реєстрації користувача та присвоєння йому відповідної ролі (користувач, служба підтримки, адміністратор).
2. Робота з оголошеннями: Користувач має мати можливість створювати, редагувати та видаляти оголошення, шукати їх за певними критеріями, додавати до обраних.
3. Можливість збереження інформації: Система повинна зберігати інформацію і надавати можливість користувачу керувати нею.
4. Створення умов онлайн спілкування користувачів: Система повинна надавати можливість користувачам спілкуватися в режимі реального часу.
5. Модерація: Оголошення на сайті має з'являтися тільки після проходження модерації. Повідомлення між користувачами мають мати автоматичну модерацію на перевірку посилань чи номерів карт. Має бути можливість перевірити переписку вручну.
6. Видалення користувача: Користувач може запросити видалення його з системи. Має бути можливість м'якого видалення. (Повне видалення буде неможливе через велику кількість зв'язків).
7. Платні послуги: Має бути можливість скористатися платними послугами для покращення ранжування оголошень.
8. Платіжні методи: Користувач може вибрати різні платіжні методи для поповнення балансу, а саме поповнення гривневою картою, та криптовалютами.

Нефункціональні вимоги

1. Сприйняття

- Час, потрібний для навчання інструментами роботи з інформаційною системою для звичайних користувачів – 1 години, а для досвідчених – має бути інтуїтивно зрозумілий відразу.
- Час відповіді системи для звичайних запитів не повинен перевищувати 3 секунд, а для більш складних запитів – 5 сек.
- Інтерфейс представлення ІС повинен бути інтуїтивно зручним для користувача та не вимагати від нього додаткової підготовки.

2. Надійність

- Доступність – час, потрібний для обслуговування системи не повинен перевищувати 10% від загального часу роботи.
- Середній час безперервної роботи – 20 робочих днів.
- Максимальна норма помилок та дефектів в роботі системи – 1 помилка на 1000 запитів користувача.

3. Продуктивність

Система повинна підтримувати мінімум 100 одночасно працюючих користувачів, пов'язаних з спільною базою даних.

4. Можливість експлуатації

- Масштабування – система повинна мати можливість збільшувати потужності (продуктивність), зі збільшенням користувачів таким чином, щоб це аж ні як негативно не відобразилося на її роботі.
- Оновлення версій – Оновлення версій повинно здійснюватися в автоматичному режимі залежно від вподобань користувачів та розширення списку пропонуємих послуг.

Висновки до розділу 1

Перший розділ роботи "Аналіз предметної області та обґрунтування актуальності теми роботи" зосереджений на первинному аналізі та постановці вимог і задач для дослідження та майбутнього програмного комплексу.

Перший підрозділ зосереджений на обґрунтуванні вибору інструментальних засобів, які необхідні для розробки та аналізу ефективних платіжних систем. Тут було розглянуто різні технологічні інструменти та їх придатність для вирішення поставлених задач.

В підрозділі 2 представлено детальний аналіз існуючих платіжних систем, що пропонують оплату як фіатними, так і крипто валютами. Було розглянуто ключові переваги та недоліки існуючих систем, а також їх придатність для інтеграції.

В третьому підрозділі було сформовано постановку задачі, виходячи з проведеного аналізу. Визначено основні цілі та завдання, які повинні бути вирішені в ході дослідження, і встановлено критерії для оцінки ефективності розроблюваних рішень.

Таким чином, перший розділ закладає міцний фундамент для подальшого дослідження, надаючи чітке розуміння актуальності теми, необхідності інструментів та важливості аналізу існуючих систем для розробки нових і ефективних платіжних рішень.

РОЗДІЛ 2 ДОСЛІДЖЕННЯ ВПЛИВУ КРИПТОВАЛЮТНОЇ ОПЛАТИ НА ПРИВАБЛИВІСТЬ САЙТУ ТА ПРОЕКТУВАННЯ ПЛАТФОРМИ СТВОРЕННЯ ОНЛАЙН ОГОЛОШЕНЬ

2.1. Аналіз існуючих платіжних систем для оплати фіатними та крипто валютами

Згідно з статистичними даними за 2022 рік, 10% населення світу володіє певною формою крипто валюти [28].

OVER 420+ MILLION CRYPTO OWNERS WORLDWIDE



Рис. 2.1. Карта кількості користувачів у світі

Нігерія – 19,4% власників криптовалюти. Незважаючи на невдалий експеримент із впровадження CBDC у Нігерії, криптоплатформи повідомляють, що принаймні 33,4 мільйона нігерійців займаються криптовалютою, як офіційно, так і неофіційно. Незважаючи на те, що Центральний банк Нігерії обмежив криптовалютні транзакції, це не вплинуло на загальну тенденцію зростання користувачів [4].

Індія – 15% власників крипто валюти. Маючи 115 мільйонів власників криптовалюти, Індія має один із найвидатніших потенціалів для впровадження криптовалюти. Як і у випадку з Нігерією, негативне ставлення уряду не впливає на швидке зростання користувачів криптовалюти.

США – 8,3% власників крипто валюти. Близько 27 мільйонів американців є власниками криптовалюти, що становить 8,3% від загального населення країни. Біткойн є найпопулярнішою криптовалютою в США, лише за 2021 рік вона зросла на 16,7%.

В'єтнам – 6% власників крипто валюти. Згідно з дослідженням Triple A, 6 мільйонів в'єтнамців володіють криптовалютними активами. В'єтнамці мають власне ставлення до криптовалюти, і очікується, що кількість користувачів криптовалюти значно збільшиться до кінця 2023 року.

Пакистан – 4,1% власників крипто валют. Пакистан замикає п'ятірку країн із найвищим відсотком власників криптовалюти, який становить 9 мільйонів пакистанців і 4,1% населення країни. KuCoin, одна з найбільших криптобірж, стверджувала, що серед криптовалютного населення Пакистану переважно домінують люди віком від 18 до 60 років.

Підприємства та корпорації все частіше використовують криптовалюту як форму оплати своїх товарів і послуг. Згідно з опитуванням Deloitte у 2022 році, 75% роздрібних торговців планують почати приймати криптовалюту протягом наступних двох років. Серед компаній, що вже приймають криптовалюту, - AMC, AT&T, CheapAir, Dish, ExpressVPN, Microsoft, Newegg.com, Overstock.com, PayPal, Rakuten, Starbucks, Tesla, Time Magazine, Twitch.

Нещодавно були опубліковані звіти основних світових бірж за підсумками 2022 року, які містять відповідну інформацію про зростання їх бази користувачів.

- у грудні 2021 року у Coinbase було 73 мільйони підтверджених користувачів, що зросло на 48% до 108 мільйонів за наступні дванадцять місяців. На даний момент Coinbase є другою за величиною криптобіржею в світі за кількістю користувачів із щоденним обсягом торгів до 2,1 мільярда доларів.
- За словами генерального директора Binance, платформа досягла 120 мільйонів користувачів до 2022 року, що в 4 рази перевищило її базу

користувачів порівняно зі звітами 2021 року. Щоденний обсяг торгів Binance становить понад 20,1 мільярда доларів, що робить її найбільшою біржею криптовалют у світі.

- База користувачів KuCoin зросла майже втричі з 10 мільйонів до 27 мільйонів у період з грудня 2021 року по грудень 2022 року. З щоденним обсягом торгів у 699 мільйонів доларів KuCoin наразі є 27-ю найбільшою біржею в світі за обсягом торгів.

Криптовалютна біржа — це лише спосіб торгівлі криптовалютами активами. Для того, щоб залучитися до ринку криптовалют і проводити регулярні операції з криптовалютою, малому та середньому бізнесу потрібен платіжний шлюз криптовалюти. Ці шлюзи дозволяють продавцям приймати платежі в криптовалюті безпосередньо від клієнтів і надають низку корисних інструментів для більш ефективного керування транзакціями та відстеження потоків доходів [3].

За даними coinmap.org, у світі наразі понад 30 000 компаній приймають криптовалюту в якості оплати. Судячи з карти на рисунку 2.2 в Києві є більше 50 підприємств з такою послугою і навіть одна в Житомирі. Це вказує на те, що біткойн стає все більш популярним, і компанії визнають його важливість як варіанта оплати [29].

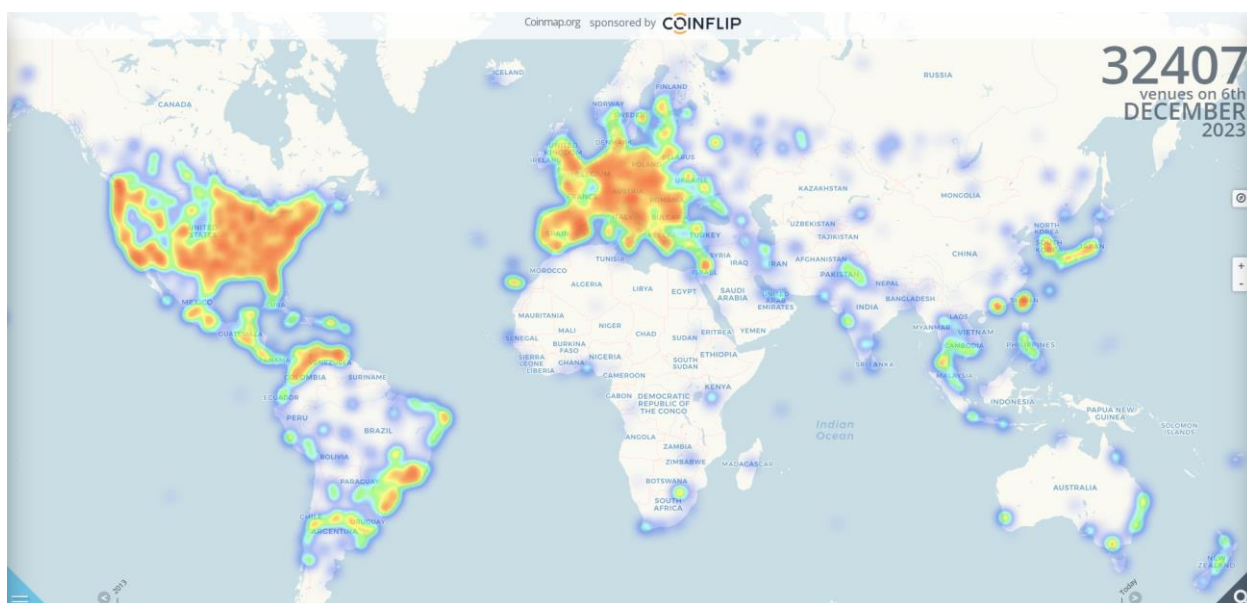


Рис. 2.2. Карта підприємств, що приймають криптовалюту

Для дослідження впливу криптовалютної оплати на привабливість сайту я провів опитування, в якому прийняло участь 116 респондентів з України. Опитування проводилося не лише серед студентів-одногрупників, а також розповсюджувалося в публічних каналах Telegram, серед колег, батьків та їх знайомих, щоб захопити різну за демографією аудиторію. Більшість опитуваних, 61%, мають вік від 18 до 24. Близько 15% мають вік від 25 до 35, 10% – молодше 18, решта старше 35 або відмовилися вказувати вік.

Перше запитання «Ставлення до криптовалюти» дозволяє дізнатися загальне представлення та активність користування такими активами. Результати представлені на рисунку 2.3.

Ставлення до криптовалюти:

116 відповідей

 Копіювати



Рис. 2.3. Ставлення до криптовалюти

Згідно з результатами цього запитання 10.3% опитуваних активно користується криптовалютами. Загалом майже 77% респондентів знає про крипто валюту і лише 23.3% мало знайомий або не знайомий з криптовалютами.

Наступне запитання дозволяє оцінити відношення опитуваних до сайту, що приймає криптовалюту в якості оплати. Результати представлені на рисунку 2.4.

Вплив на сприйняття сайту при наявності опції криптовалютної оплати:



116 відповідей

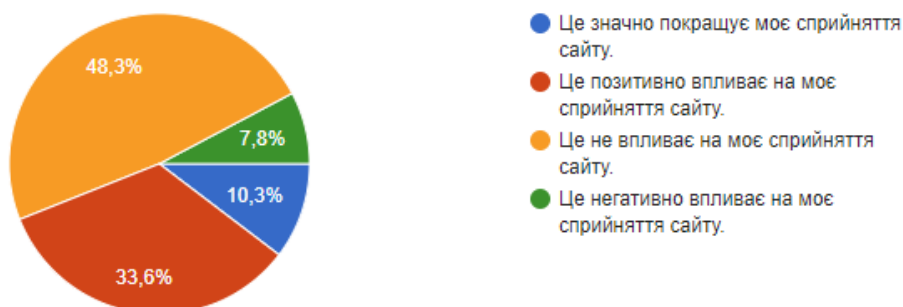


Рис. 2.4. Вплив на сприйняття сайту

Майже 8% опитуваних негативно сприймуть прийом криптовалютної оплати, оскільки все ще є думка, що криптовалюти це щось незрозуміле та нелегальне. Проте цей пункт перебиває 10,3%, чий відношення значно покращиться при такій опції, а також 33,6% що сприймуть це позитивно, що в сумі дає 43,9%. Також відношення 48,3% респондентів не зміниться. З цього можна зробити висновок, що прийом криптовалютної оплати дасть можливість привабити більше клієнтів до програмного продукту.

Наступним запитанням я оцінив вірогідність використання криптовалюти в якості основного методу оплати.

Вірогідність використання криптовалютної оплати на сайті:



116 відповідей



Рис. 2.5. Вірогідність використання оплати криптовалютою

68% опитуваних скористається таким методом оплати, з них 7.8% надасть перевагу перед оплатою звичайними фіатними валютами. Проте 31.9%, що складає майже третину, не скористається криптовалютною оплатою. Це означає, що криптовалюта не має бути єдиним способом оплати, а стояти поруч з класичними способами оплати, як прийом карток чи готівки, оскільки більшість людей все ж не використовує криптовалюти, інакше веб-сайт може втратити багато потенційних клієнтів.

Далі йде запитання з можливістю вказати декілька відповідей. В цьому запитанні оцінюється що людям подобається в криптовалютах і які переваги вони в цьому бачать.

Переваги криптовалютної оплати на вашу думку:

 Копіювати

116 відповідей

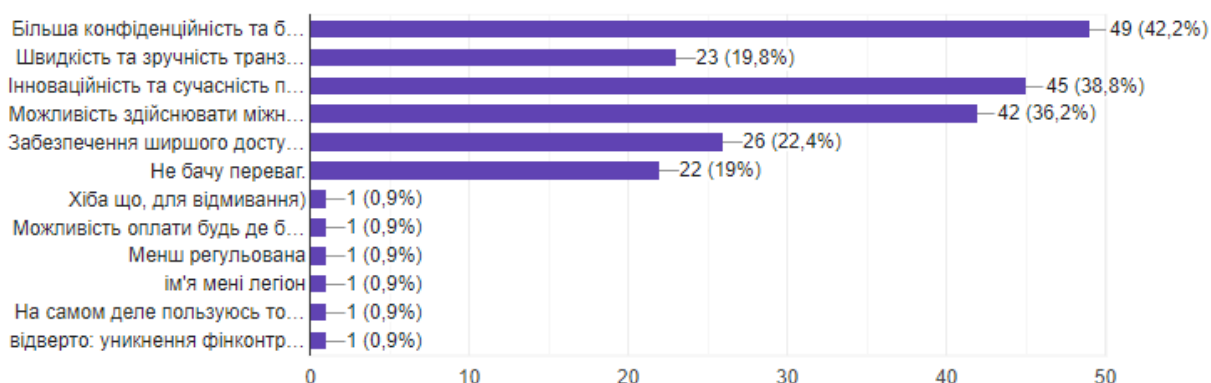


Рис. 2.6 Переваги криптовалютної оплати на думку опитаних

Найпопулярніші відповіді це – більша конфіденційність та безпека, інноваційність та сучасність підходу та можливість здійснювати міжнародні платежі в "універсальній" валюті з меншими комісіями. Людям подобається, що оплата криптовалютою анонімна і дізнатися хто відправляв кошти неможливо. Проте можна дізнатися звідки ці гроші надійшли і відхилити їх прийом, якщо криптовалюта «брудна», що може бути корисне для власника бізнесу. Це називається перевірка AML. Також криптовалюти це все ще досить нова технологія і людям подобається коли сайт містить останні технологічні рішення. Це може бути сигналом для людини, що все інше також реалізовано сучасно і безпечно. Останній і найцікавіший для власнику бізнесу варіант це

можливість здійснювати міжнародні платежі в "універсальній" валюті. Більшість сайтів приймають картки тієї країни в якій вони працюють. Проте через відкритість кордонів і глобалізацію в країні може бути багато іноземців які мають картки тільки своєї країни. Неможливість прийому платежів від цих людей просто відкидає цих потенційних клієнтів, які підуть до конкурентів. Також останнім часом спостерігається ріст доставки з магазинів закордоном, таких як NpShopping і подібних, тож клієнтів з-за кордону зараз ще більше. Зробити прийом будь-яких карток практично неможливо, міжнародні платежі зазвичай мають велику комісію, а криптовалюта не прив'язана до країни і комісія завжди єдина. Тож прийом криптовалюти може бути найпростішим та найвигіднішим способом забезпечити іноземців можливістю оплати [2].

В передостанньому запитанні необхідно вказати які існують бар'єри для використання криптовалютою оплати на думку опитуваних.

Бар'єри для використання криптовалютою оплати:

 Копіювати

116 відповідей

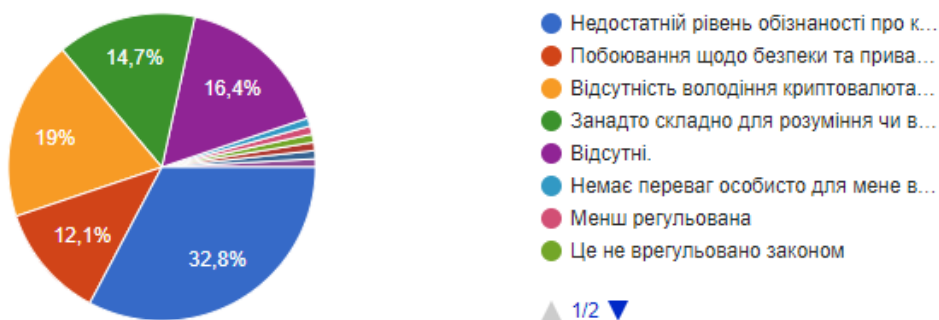


Рис. 2.7. Бар'єри для використання криптовалютою оплати

Майже третина вказала «Недостатній рівень обізнаності про криптовалюту» як основний бар'єр для використання. Також для 14,7% криптовалюти занадто складні для розуміння і 19% просто не володіє криптовалютами. Це запитання, як і попереднє містило варіант «Інше», де можна було написати свою відповідь. Декілька респондентів вказали цей варіант і написали приблизно однакову причину – криптовалюта не регульована законом. Хоч децентралізація і відсутність регуляції є однією з

позитивних особливостей криптовалюти, для повноцінного виходу в легальне поле це необхідність. Коли такі закони будуть розроблені та прийняті потенційно може з'явитися більше можливості зручного придбання криптовалюти, можливо навіть в банках, до яких ми звикли і такі бар'єри як «Недостатній рівень обізнаності про криптовалюти» чи їх складність для їх розуміння також зникнуть.

Останнє запитання «Чи здійснювали ви оплату криптовалютами? Якщо так, то чому і на яких ресурсах?» не має варіантів, лише поле для розгорнутої відповіді. Воно дозволило дізнатися, де респонденти здійснювали оплату криптовалютою і що стало причиною такого рішення. Зазвичай були вказані купівлі товарів закордоном як цифрових так і фізичних чи арбітраж трафіку, а причиною неможливість оплати карткою через відсутність прийому гривні на цих сайтах. Сюди можна зарахувати оплату приватним особам. Якщо вести бізнес з іноземцями процес оплати може бути довгий і дорогий, через великі комісії на міжнародні платежі. Були ще варіанти конфіденційної купівлі, при якій оплата карткою неможлива, адже розкриває особистість людини, що оплачує. Також люди використовували криптобіржі, як Binance, для здійснення переказів, та криптообмінники, щоб отримати фіатні гроші за крипту.

Це дослідження дозволило побачити зацікавленість людей в криптовалютах, показало можливість розширення клієнтської бази сайту при наявності криптовалютної оплати. З цього можна зробити висновок, що прийом криптовалюти має бути обов'язковою опцією, якщо ви плануєте залучити клієнтів з-за кордону. Навіть якщо ні, це буде хорошим варіантом, оскільки є багато людей зацікавлених в конфіденційності, а також яким просто подобається, що сайт відповідає їх уявленням про сучасні технології. Проте криптовалютна оплата небажана в якості єдиного варіанту і повинна бути поруч з прийомом оплати карткою, якщо на те немає інших причин, як конфіденційність, інакше бізнес ризикує лишитися багатьох клієнтів, що не володіють криптовалютою і не мають достатнього рівня обізнаності для цього.

2.2. Визначення варіантів використання та об'єктно-орієнтованої структури системи.

Виходячи з умов, зазначених у підрозділі 1.4, додаток буде мати 4 актора. Перший – це неавторизований, або анонімний користувач. Такий користувач має можливість переглядати оголошення та звертатися до служби підтримки через онлайн чат, але для доступу до розширених функцій необхідно пройти реєстрацію та авторизацію на сайті.

Авторизований користувач може виступати у двох ролях: клієнт або адміністратор. Клієнт має змогу не тільки переглядати оголошення, але й зв'язуватися з власниками оголошень через внутрішній чат, створювати власні оголошення та керувати ними (редагувати чи видаляти). Вони також мають особистий рахунок на сайті, який дозволяє користуватися платними функціями, такими як надання оголошенню VIP-статусу. VIP-оголошення відображаються на початку списку згідно з застосованими фільтрами. Клієнти можуть поповнювати баланс через Visa/MasterCard або криптовалюту.

Служба підтримки – це окремий тип користувача, що не авторизується на сайті. Вони мають схожі можливості з неавторизованими користувачами, але мають доступ до спеціалізованого сервісу онлайн чату для відповідей на запитання користувачів. У випадках, таких як підозра у шахрайстві, служба підтримки може звернутися до адміністратора.

Адміністратор – це авторизований користувач з адміністративними правами. Вони не можуть створювати оголошення чи використовувати внутрішній чат для спілкування з клієнтами, але мають доступ до всіх оголошень, включно з видаленими, та можливість їх видаляти. Адміністратори також мають доступ до списку користувачів, де вони можуть переглядати їхні оголошення та повідомлення. В разі доказів шахрайства, адміністратор може заблокувати або видалити користувача.

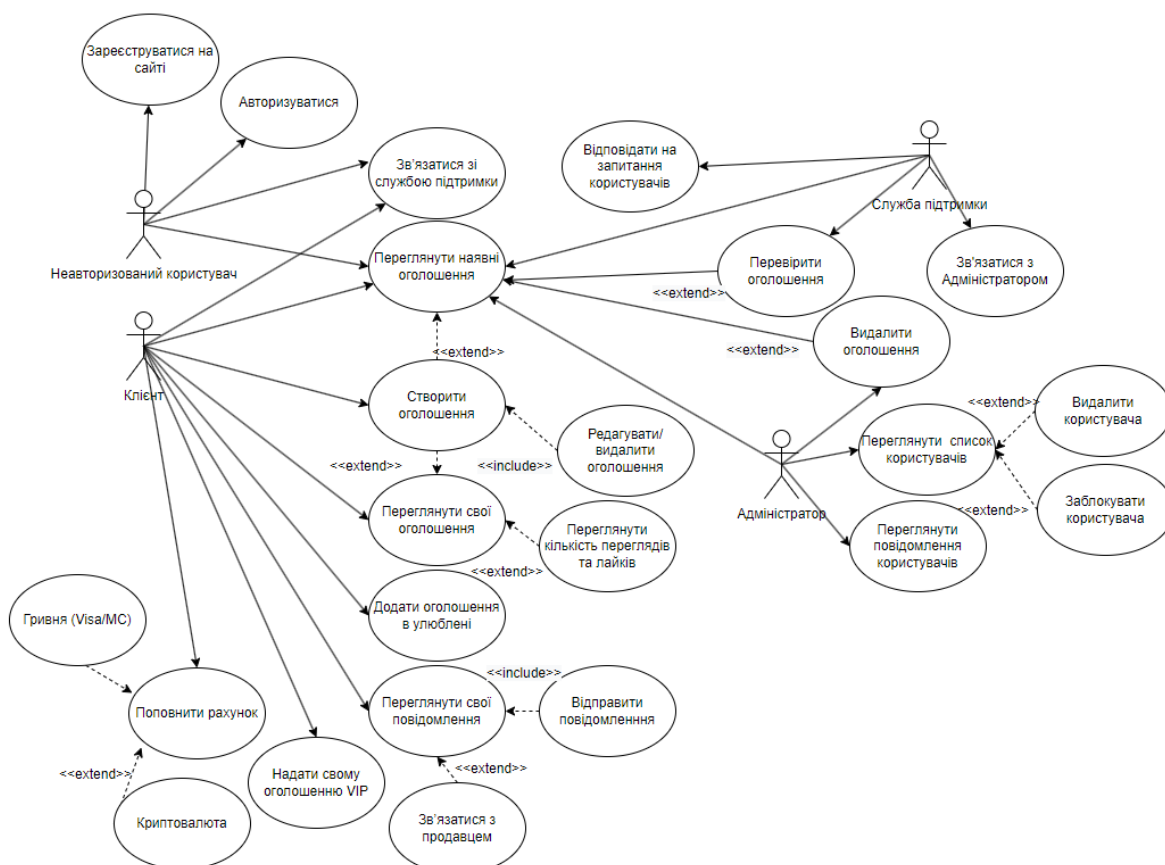


Рис. 2.8. Діаграма використання

На рисунку 2.3 зображена діаграма класів, що будуть реалізовані.

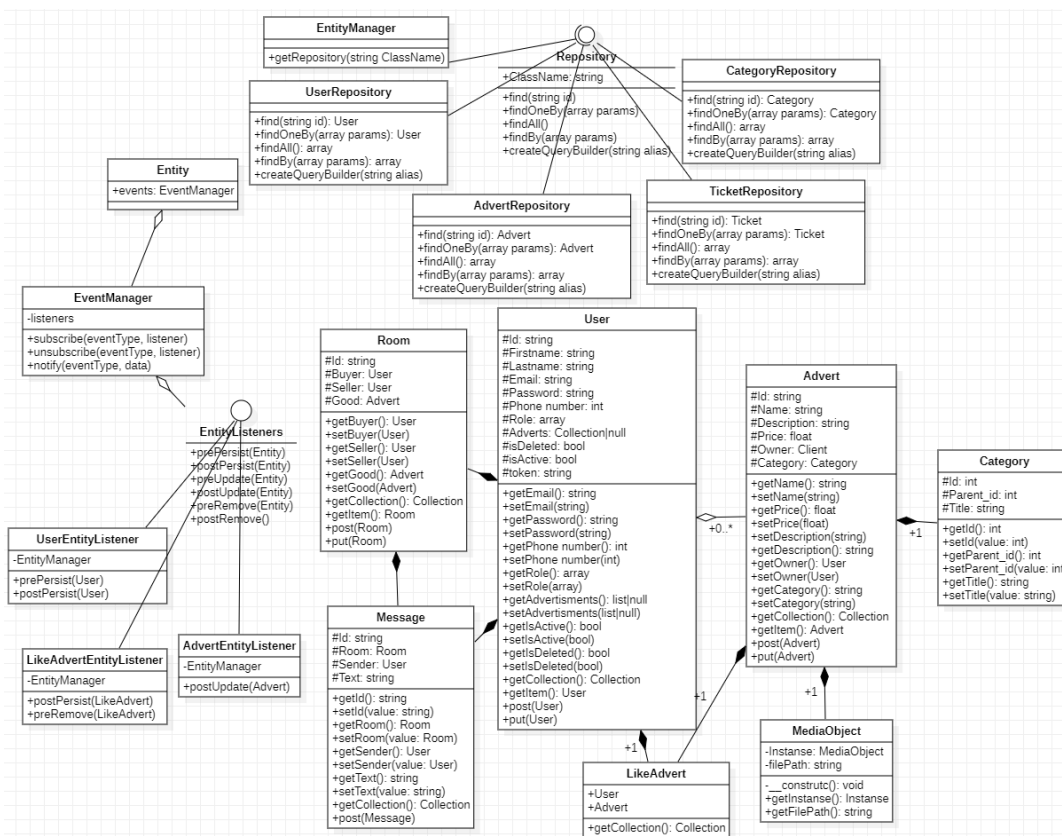


Рис. 2.9. Діаграма класів

Найбільший клас на діаграмі – User. Це користувач сайту. Це може бути як клієнт, так і адміністратор. Цей клас має поле Role, що визначає роль користувача. В залежності від ролі, користувачу доступні різні можливості. Клієнт може створювати та редагувати оголошення, писати повідомлення іншим користувачам. Адміністратор може переглядати список користувачів, читати їх повідомлення та блокувати їх.

Клас Advertisement – це оголошення, що розміщене на сайті. Вони містять в собі ім'я, опис, ціну, користувача, якому належить це оголошення, та категорію.

Клас Room представляє з себе кімнату повідомлень користувачів. Він зберігає в собі покупця, продавця, та оголошення до якого кімната відноситься. Самі повідомлення представляються класом Message.

Клас Category – це категорія. Цей клас має ієрархічну структуру. Поле parent_id в цьому класі посилається на id батьківської категорії. Коренева категорія має значення null для цього поля. Таку структуру можна легко обходити рекурсивним методом для відображення дерева категорій.

Ці класи не містять функцій для роботи з базою даних. Вони містять поля і методи що потрібні для сереалізації та десереалізації, наприклад getCollection, getItem, post та put, що представляють з себе відповідні типи запитів GET, POST та PUT. Оскільки backend частина реалізована як API, вона отримує та віддає дані у текстовому вигляді, а саме в json або xml форматі.

Для вибірок реалізовано патерн Репозиторій [9]. Він має методи find для пошуку по полю id, findBy для пошуку з параметрами, findOneBy для пошуку одного об'єкту з параметрами та findAll, щоб вибрати всі. Також в ньому присутній метод createQueryBuilder, що дозволяє створити об'єкт класу QueryBuilder за допомогою якого можна побудувати власні запити [31].

Також на схемі видно реалізацію патерну Спостерігач. Реалізовано три спостерігача, для сутності Advert, для сутності User та для сутності LikeAdvert.

Цей патерн відслідковує зміни в класі Advert, для того, щоб сповіщувати клієнтів про зниження ціни оголошення, якщо вони додали це оголошення в улюблені та підписаний на створення нової сутності User, для відправки повідомлення з підтвердженням реєстрації, та шифрування паролю перед записом в базу даних. Слухач сутності LikeAdvert допомагає зберігати правильне число уподобань кожного оголошення, оскільки набагато ефективніше зберігати просто число, ніж рахувати його при кожному запиті кількості уподобань Оголошення [31].

2.3. Проектування та реалізація алгоритмів роботи системи

Основною базою даних виступає MySQL. Вона містить 9 таблиць: таблиця користувачів, таблиця оголошень, таблиця категорій, таблиця медіа-об'єктів, таблиця улюблених оголошень, таблиця кімнат для повідомлень, таблиця повідомлень та таблиці налаштувань. Виходячи з цього, всі дані будуть зберігатись у базі даних з наступними таблицями.

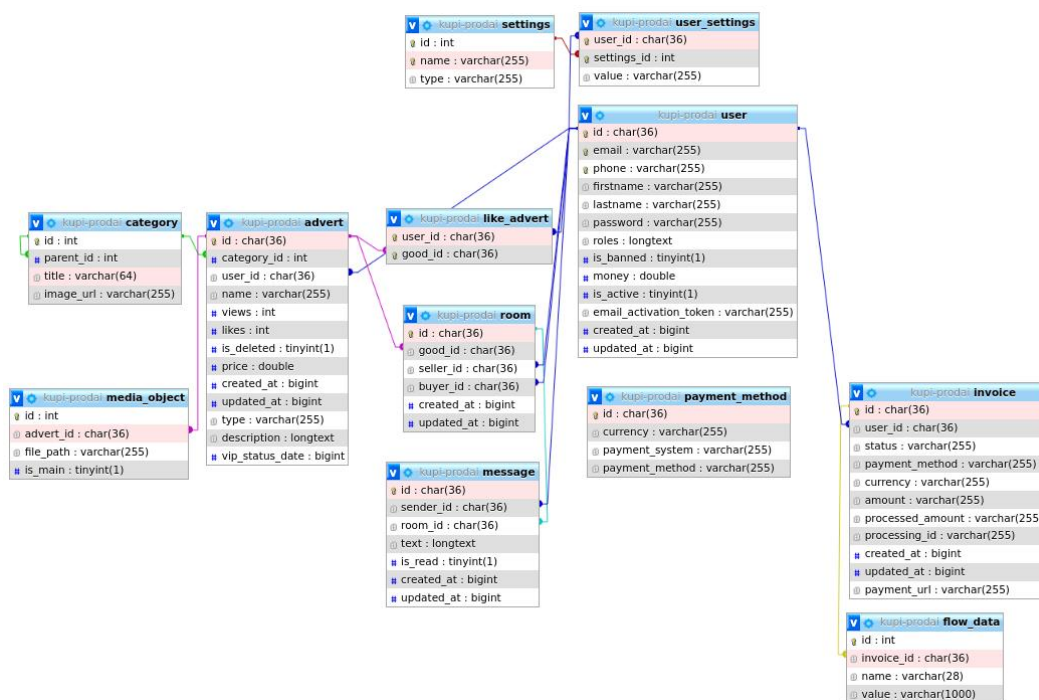


Рис. 2.10. Схема бази даних системи

Система включатиме модуль для автентифікації та авторизації користувачів. Публічно доступні будуть лише оголошення та опція зв'язку зі службою підтримки. Для створення, редагування оголошень або взаємодії з іншими користувачами, необхідно пройти процес автентифікації, який детально описано на діаграмі активності (рисунок 2.11). Спочатку система перевіряє, чи користувач зареєстрований. Якщо ні, він отримує можливість перейти до форми реєстрації, де необхідно вказати ім'я, прізвище, електронну адресу та пароль. Після цього користувачу потрібно підтвердити свою електронну пошту, щоб отримати доступ до форми входу. На формі входу вводяться логін та пароль. Також передбачена функція скидання пароля, при використанні якої на електронну пошту користувача надсилається посилання з токеном для зміни пароля. Цей токен буде дійсним протягом обмеженого часу - 5 хвилин. Перейшовши за посиланням, користувач має можливість встановити новий пароль.

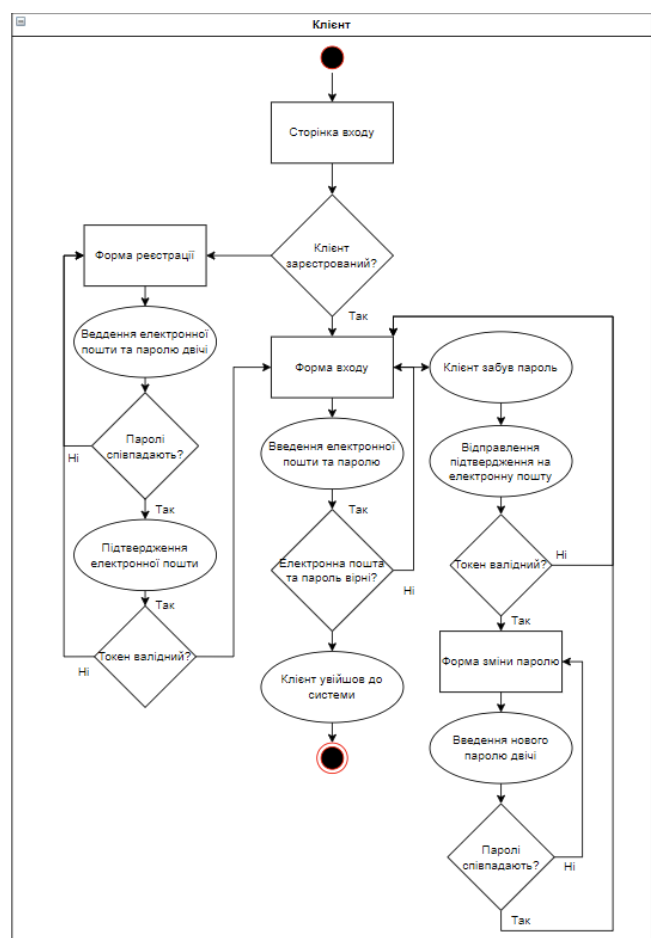


Рис. 2.11. Діаграма активності процесу автентифікації

В системі буде присутня модерація. Кожне оголошення перед тим як потрапити в публічний доступ буде перевірено вручну модератором. Схема перевірки зображена на діаграмі активності рисунку 2.6.

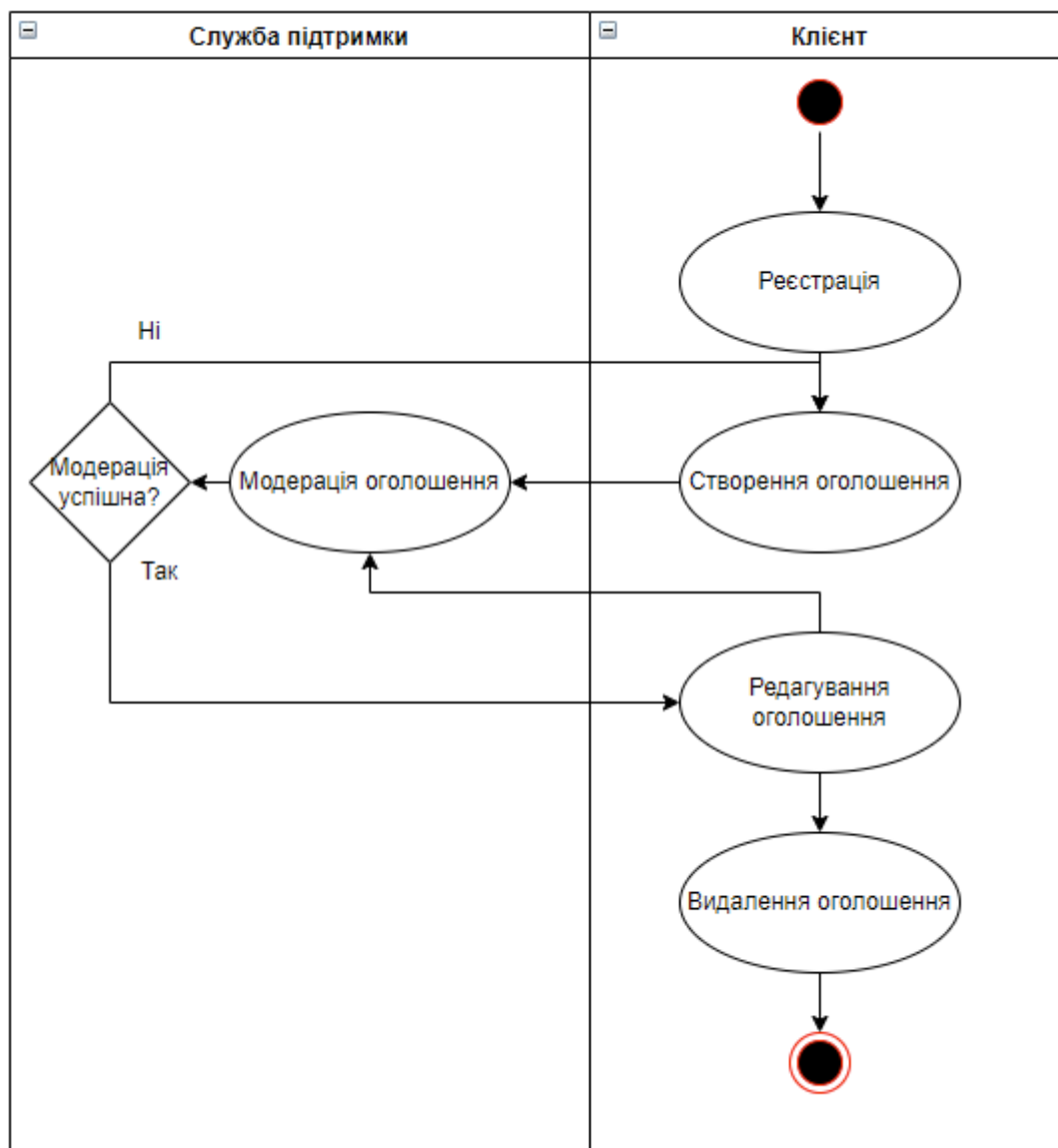


Рис. 2.12. Діаграма активності процесу створення заявки

В системі буде можливість зв'язатися зі службою підтримки. Служба підтримки буде займатися питаннями по роботі сайті, проте, якщо запитання клієнту пов'язане з шахрайством, заявка з запитанням клієнта буде передана адміністратору, оскільки адміністратор має право переглядати повідомлення між клієнтами та блокувати їх. Схема роботи служби підтримки зображена на діаграмі активності рисунку 2.7.

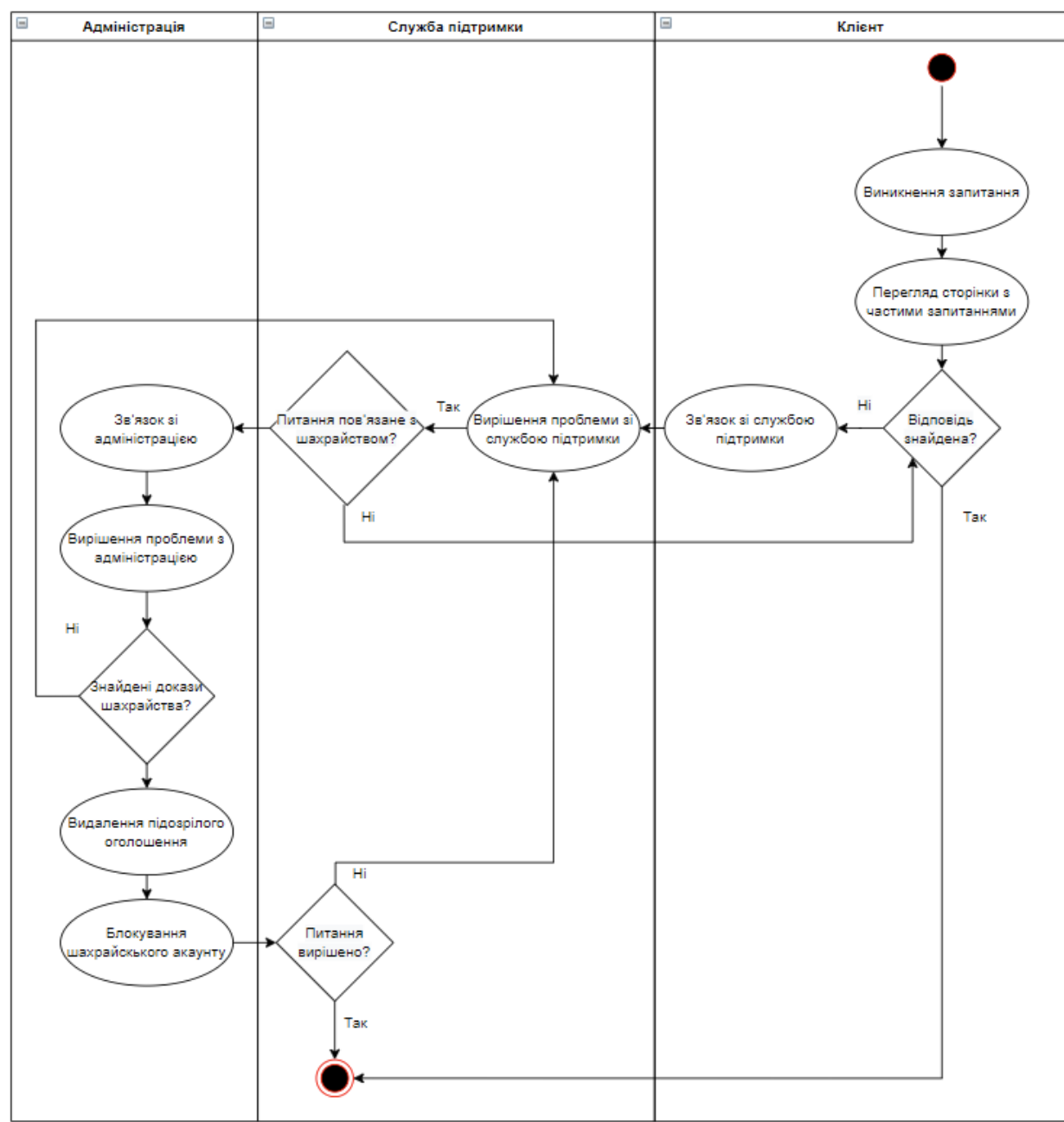


Рис. 2.13. Діаграма активності процесу обробки запитання клієнта

Кошти для оплати платних послуг на сайті списуються з балансу клієнта. Для поповнення балансу буде інтегровано платіжні системи для прийому гривні та криптовалют. Діаграма послідовності поповнення балансу за допомогою криптовалюти зображено на рисунку 2.14.

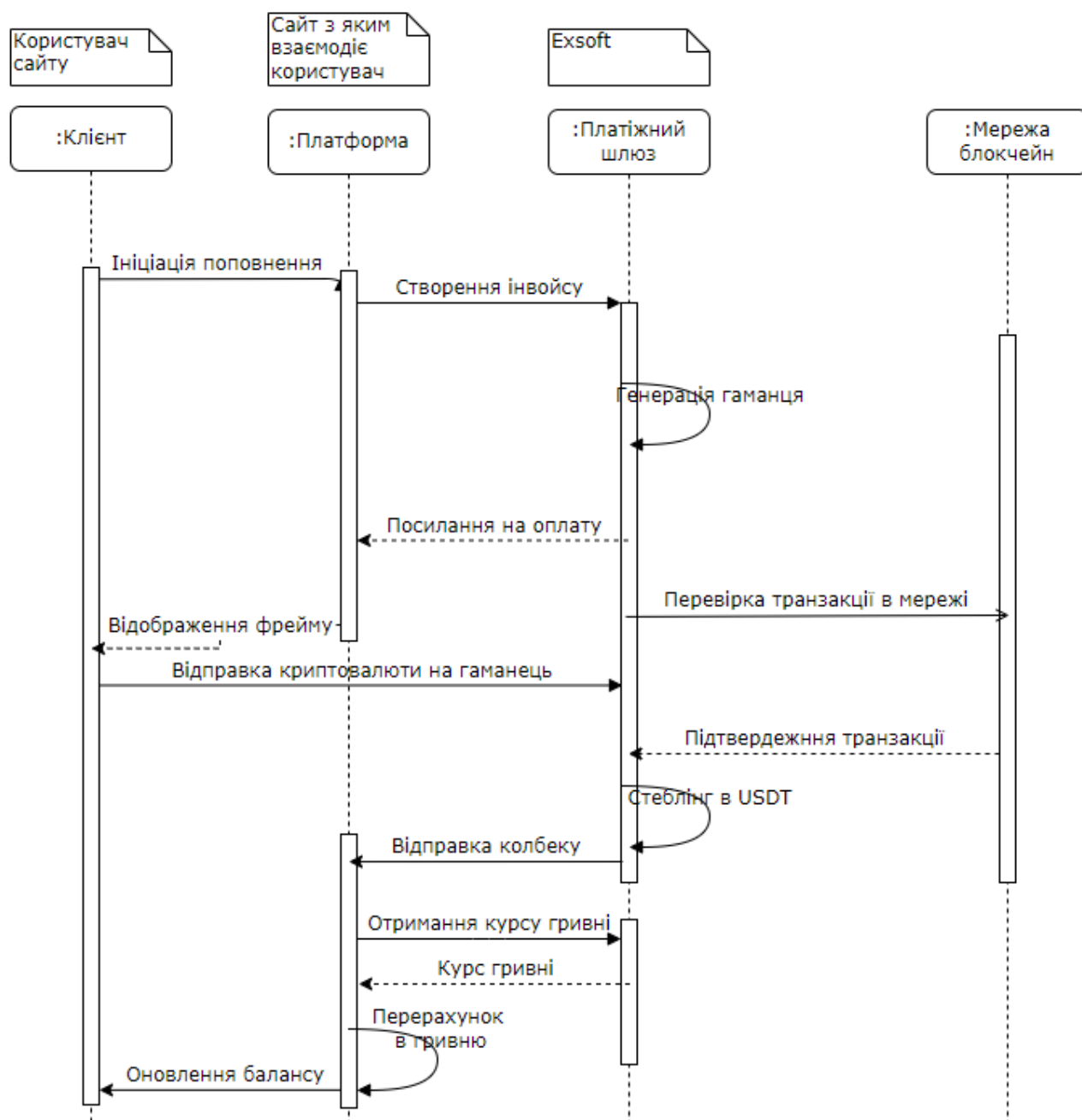


Рис. 2.14. Діаграма послідовності поповнення балансу за допомогою криптовалюти

Спочатку клієнт ініціює поповнення свого балансу. Для цього він обирає потрібну криптовалюту і пише суму, яку бажає перерахувати. Платформа робить запит на створення інвойсу на оплату на платіжному шлюзі. Шлюз генерує гаманець, і надсилає посилання на оплату платформі. З цього моменту шлюз асинхронно перевіряє статус транзакції в мережі блокчейн.

Після отримання посилання на оплату платформа відображає фрейм оплати користувачу, на якому знаходиться гаманець, на який потрібно перерахувати кошти, і сума у вибраній валюті. Клієнт відправляє гроші на вказаний гаманець.

Після отримання необхідної кількості підтверджень з мережі блокчейн, інвойс на шлюзі отримує статус «Виконаний» та обмінює отриману криптовалюту на стейблкоїн USDT. Стейблкоїн – це тип криптовалюти, стабільність вартості якої підтримується шляхом прив'язки до іншого стабільного активу, як-от традиційна валюта або дорогоцінні метали. Найпоширенішими активами для прив'язки стейблкоїнів є долар США, євро та золото. USDT прив'язаний до американського долару за курсом один до одного. Процес перетворення нестабільної криптовалюти в стабільну називають стейблінгом. Стейблінг – це зручний варіант уникнути волатильності криптовалют, без ризику зниження прибутку.

Далі шлюз відправляє запит до платформи, що іменують колбек (callback), в якому знаходиться статус, а також оброблена сума. Це сума, що була зарахована на баланс мерчанта на шлюзі, з якої вже були зняті всі комісії. При тіло колбеку містить статус, та оброблену суму. Якщо статус «Оброблено» платформа робить запит на отримання поточного курсу до шлюзу. Далі оброблена сума перетворюється на гривню за отриманим курсом і нараховується клієнту на баланс, оскільки на платформі онлайн оголошень баланс в гривні.

Додаток буде побудований на принципах Rest API. Це означає, що користувач взаємодіє лише з зовнішньою частиною сайту, що рендериться на його стороні. Сервер лише віддає та приймає дані у json або xml форматах.

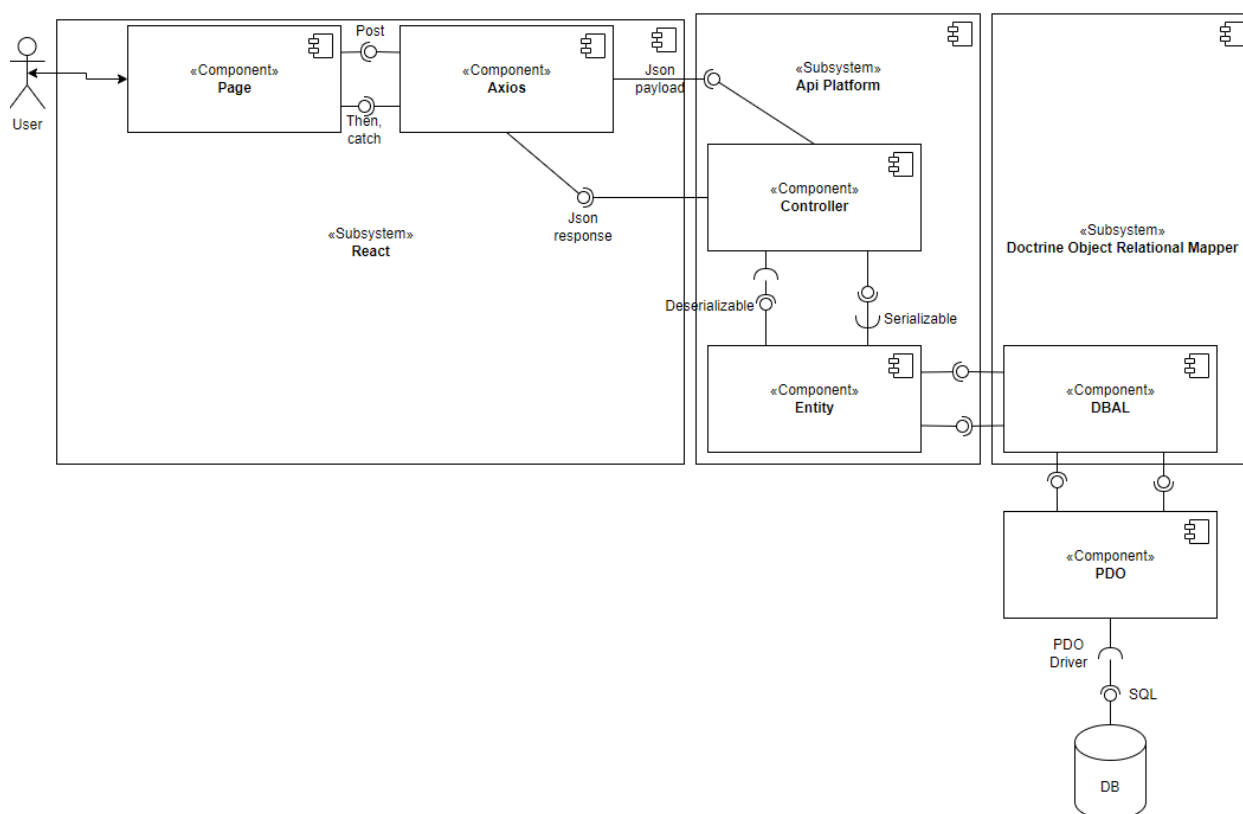


Рис. 2.15. Діаграма компонентів системи

Користувачі взаємодіють з компонентами React, які рендеряться на клієнтській стороні. Під час рендерингу динамічних сторінок відбувається відправлення асинхронних запитів на сервер за допомогою Axios, що є асинхронною бібліотекою. Axios має методи `then`, `catch` та `finally`, які активуються залежно від відповіді сервера.

Метод `then` викликається, коли сервер повертає відповідь без помилок, тобто зі статусами 2xx. У цьому методі можна задати стани (states) в React, сигналізуючи про необхідність перерендерингу сторінки без візуального оновлення для користувача. Метод `catch` застосовується для перехоплення помилок від сервера, дозволяючи їх обробити та представити користувачу у зрозумілому форматі. Метод `finally` запускається після `then` або `catch` у будь-якому випадку, дозволяючи виконати загальні дії, що не потребують повторення в обох попередніх методах [31].

Запит потім передається на контролер, який займається обробкою API-запитів, направляючи їх до відповідного класу сутності, з якою працює Axios.

При обробці сутності відбувається валідація даних. Це можуть бути стандартні валідатори, такі як перевірка типу даних, виявлення пустоти тощо. Є також можливість створення користувацьких валідаторів. Наприклад, при введенні номера карти стандартна перевірка типу даних прийме будь-яке числове значення, але користувацький валідатор може включати власний регулярний вираз для більш специфічної валідації [13].

При проходженні валідації відбувається процес десериалізації даних в об'єкт. Процес обробки даних включає два етапи. Перший етап - це декодування даних з формату JSON або XML, які отримано від клієнта, в масив. На другому етапі цей масив перетворюється на об'єкт запитуваної сутності.

Після цього настає робота з Doctrine. Doctrine - це об'єктно-реляційний проєктор, який працює на основі абстрактного шару доступу до бази даних (DBAL). Однією з ключових особливостей Doctrine є можливість формувати запити до бази даних за допомогою його власної об'єктно-орієнтованої мови запитів, DQL (Doctrine Query Language), яка натхненна HQL (Hibernate Query Language). Хоча Doctrine також підтримує написання запитів на стандартному SQL, використання DQL часто виявляється більш зручним і безпечним варіантом. Реалізація додатку веб-орієнтованої платформи створення онлайн оголошень [29].

Серверна частина системи виключно обробляє та передає дані у форматі JSON. Для надання динамічності веб-сторінок була використана клієнтська частина, побудована на основі React. React відповідає за рендеринг сторінок та вставляння в них даних, отриманих від сервера. Це дозволяє змінювати вміст сторінки без її повторного завантаження, що ефективно в ситуаціях, коли змінюються дані, але сама структура сторінки залишається незмінною, наприклад, при пагінації або фільтрації контенту [18]. Для отримання даних від сервера React використовує Axios - HTTP-клієнт для Node.js та React, що здійснює асинхронні запити через XMLHttpRequests або AJAX [11].

У додатку також присутні сторінки, які потребують відображення даних у реальному часі, наприклад, чат, де нові повідомлення мають з'являтися без перезавантаження сторінки. У цьому випадку використання Axios неефективне, оскільки точний момент для виконання запиту невідомий. Для реалізації цієї функціональності застосовується веб-сокет Mercure. Це відкритий протокол, спеціально розроблений для додавання потокових та асинхронних можливостей до API веб-додатків [30].

2.4. Інтеграція платіжних систем для прийому банківських карток та криптовалют

Інтеграція платіжної системи для прийому банківських карток та криптовалюти в веб-додаток або мобільний застосунок включає кілька ключових етапів:

1. Вибір Платіжного Провайдера: Першим кроком є вибір надійного платіжного провайдера, який підтримує транзакції криптовалютами. Важливо врахувати репутацію провайдера, підтримку необхідних криптовалют, комісійні ставки, безпеку, та легкість інтеграції. В першому розділі було розглянуто декілька сервісів, що надають можливість оплати крипто валютою. Також вказано, що для інтеграції обрано криптовалютний шлюз Exsoft.

2. Вивчення API та Документації: Після вибору провайдера необхідно ознайомитися з його API та документацією. Це дозволить розуміти, як взаємодіяти з платіжною системою, які функції доступні, та як забезпечити безпеку транзакцій.

3. Реєстрація та Отримання API Ключів: Для початку роботи з API, необхідно зареєструватися у системі платіжного провайдера та отримати API ключі. Ці ключі використовуються для автентифікації та авторизації запитів до платіжного API. До Exsoft можна підключитися двома методами, як користувач та як проєкт. Підключення як користувач відбувається на сайті шлюзу, а

підключення як проєкт створене для інтеграції в додаток. Для такого підключення необхідно вказати ідентифікатор свого проєкту.

4. Інтеграція API з Веб-додатком/Мобільним Застосунком: Розробники інтегрують API платіжного провайдера з додатком. Це включає налаштування з'єднань, виконання тестових платежів, налаштування обробки помилок і забезпечення безпеки транзакцій. Для цього тіла запитів, що приходять від платіжного шлюзу зберігаються в MongoDB.

5. Тестування та Валідація: Провести ретельне тестування, щоб переконатися, що інтеграція працює коректно. Це включає тестування процесу оплати, обробки помилок, і зворотніх викликів (callbacks) для підтвердження транзакцій.

6. Запуск та Моніторинг: Після успішного тестування та перевірки безпеки, платіжну систему можна впроваджувати в експлуатацію. Важливо постійно моніторити транзакції та систему для виявлення будь-яких непередбачених проблем.

Висновки до розділу 2.

Цей розділ роботи був присвячений детальному аналізу та розробці платформи для онлайн оголошень з інтеграцією криптовалютних платіжних систем. Основна увага була зосереджена на тому, як включення опції оплати криптовалютою може підвищити привабливість вебсайту та загальну ефективність платформи.

У підрозділі 1 було досліджено потенційні перспективи використання криптовалюти як платіжного засобу. Було встановлено, що інтеграція криптовалюти може значно розширити ринкові можливості та залучити нових користувачів, які шукають інноваційні та безпечні способи оплати.

У підрозділі 2 були визначені варіанти використання криптовалюти в системі та розроблена об'єктно-орієнтована структура для її імплементації. Така структура гарантує гнучкість та масштабованість платформи, дозволяючи легко адаптуватися до змін у криптовалютному секторі.

Підрозділ 3 був присвячений проектуванню та реалізації алгоритмів для забезпечення ефективної роботи системи. Було розроблено алгоритми, які дозволяють автоматизувати процеси транзакцій, забезпечуючи високий рівень безпеки та прозорості.

В останньому підрозділі було розглянуто інтеграцію платіжних систем для прийому банківських карток та криптовалюти. Це дозволяє користувачам вибирати найзручніший спосіб оплати, що, в свою чергу, збільшує задоволеність користувачів та приваблює більшу аудиторію.

Таким чином, дослідження та розробка платформи онлайн оголошень з інтеграцією криптовалютної оплати виявилися важливими у визначенні нових можливостей для бізнесу, пропонуючи інноваційні, безпечні та гнучкі рішення для сучасного цифрового ринку.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ ТА ІНТЕГРАЦІЯ ПЛАТІЖНИХ СИСТЕМ

3.1. Порядок встановлення та налаштування параметрів системи

Проект буде розгорнуто за допомогою Docker. Docker — програмне забезпечення для автоматизації розгортання та керування програмами в середовищах з підтримкою контейнеризації, контейнеризатор додатків. Дозволяє "упакувати" додаток з усім його оточенням і залежностями в контейнер, який може бути розгорнутий на будь-якій Linux-системі за допомогою контрольних груп в ядрі, а також надає набір команд для управління цими контейнерами [20].

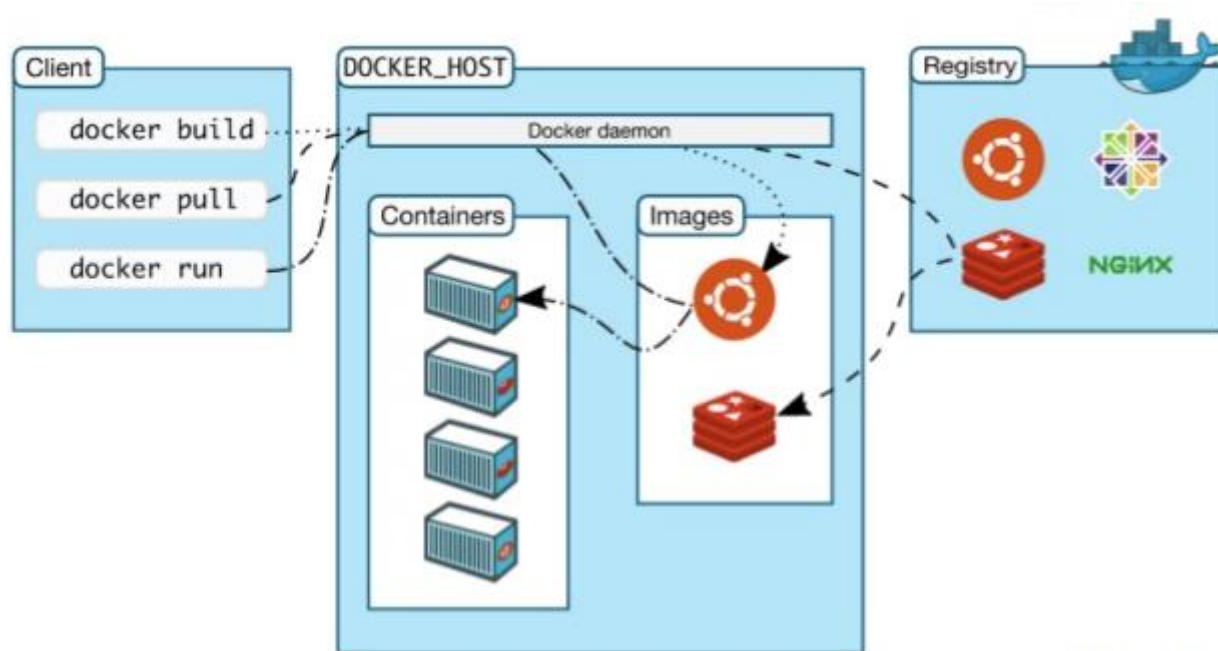


Рис. 3.1. Схема роботи Docker

Наступна схема показує контейнери, що будуть запущені в додатку. В якості віртуальної ОС буде виступати Arpine Linux, що створений спеціально для мережевих шлюзів. Веб сервером виступає Nginx, основною базою даних є MySQL. Для статистичних записів, наприклад логів буде використовуватись MongoDB. Також тут планується зберігання тіл запитів, що надходять від платіжних шлюзів. Це допоможе вирішити проблеми з транзакціями, якщо такі

будуть. Для відсилання даних про оновлення використовується хаб Mercure. Клієнтська частина працює на React.

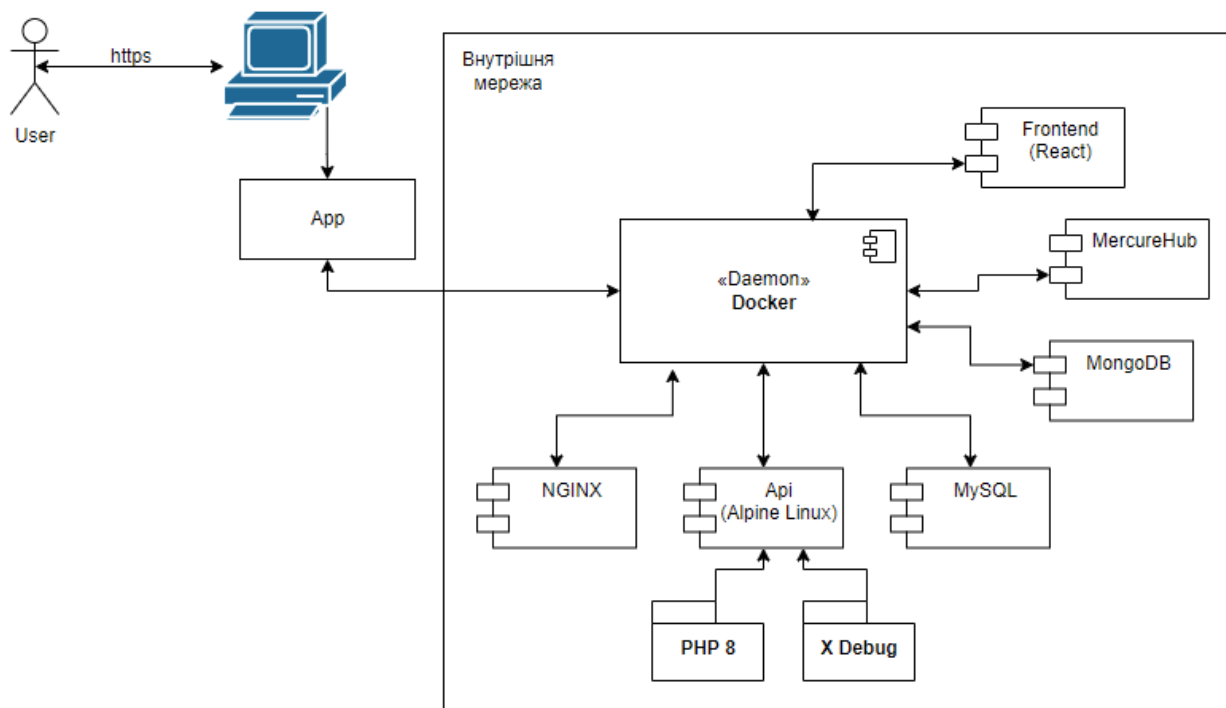


Рис. 3.2. Діаграма розгортання

Усі контейнери пов'язані в одну внутрішню мережу, для можливості спілкування між ними. Керує контейнерами Docker, що працює як демон. Демон, або daemon, це програма у Unix системах, що працює в фоновому режимі, без прямої взаємодії з користувачем [1].

3.2. Структура інтерфейсу платформи створення онлайн оголошень

Під час виконання роботи була створена платформа для розміщення онлайн оголошень. Головна сторінка сайту оснащена навігаційною панеллю, пошуковим полем та переліком категорій. На навігаційній панелі розміщено три кнопки: перша веде на домашню сторінку і представлена у вигляді логотипу сайту, інші дві призначені для входу на сайт та реєстрації. У нижній частині кожної сторінки розташований футер, що включає статичні розділи, такі як "Про нас", "Підтримка", та інші, а також контактну інформацію.

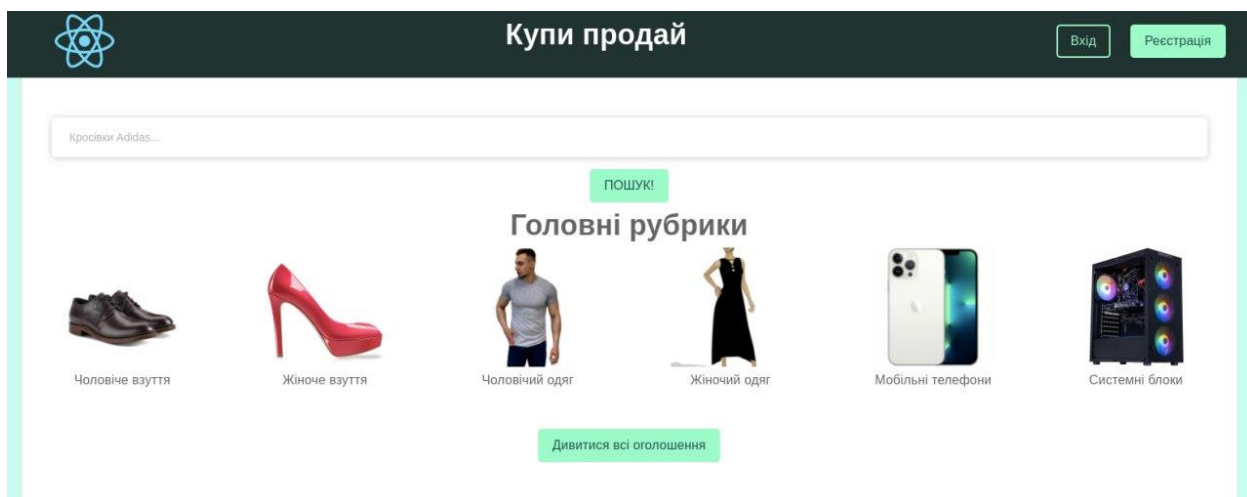


Рис. 3.3. Головна сторінка додатку

Форма входу дає можливість авторизуватися за допомогою логіну та паролю. Також форма містить посилання на скидання паролю.

Рис. 3.4. Форма входу

Щоб увійти на сайт, користувач має спочатку пройти процес реєстрації. На сторінці реєстрації знаходиться форма, де потрібно заповнити поля з ім'ям, прізвищем, електронною адресою, номером мобільного телефону та паролем. Важливо, щоб електронна адреса та номер телефону були унікальними, оскільки система не дозволяє створювати кілька акаунтів з однаковими даними. Для запобігання автоматизованій реєстрації ботами внизу форми

реєстрації розміщено капчу. Після заповнення форми на вказану електронну адресу відправляється лист з посиланням для активації акаунта, без якої авторизація на сайті неможлива.

Реєстрація

Ім'я:

Прізвище:

Email:

Телефон:

Пароль:

Повторіть пароль:

This reCAPTCHA is for testing purposes only. Please report to the site admin if you are seeing this.

☐ Я не робот


reCAPTCHA
Конфіденційність - Умови використання

Зареєструватися

Рис. 3.5. Форма реєстрації

На сторінці з оголошеннями представлений перелік оголошень. Кожне оголошення в списку включає основне зображення, назву, ціну, категорію та дату розміщення. Для отримання більш детальної інформації можна перейти на сторінку конкретного оголошення. Оголошення можна фільтрувати за різними критеріями, такими як категорія, назва, ціна та дата публікації. Є можливість сортувати оголошення за датою публікації та ціною. За замовчуванням на сайті

спочатку показуються найновіші оголошення. Сторінки оголошень організовані через пагінацію. Перші три позиції у списку займають VIP-оголошення, які відбираються випадково на основі заданих фільтрів та відображаються на початку кожної сторінки. Автентифіковані користувачі мають доступ до посилань на свої власні та улюблені оголошення через навігаційну панель.

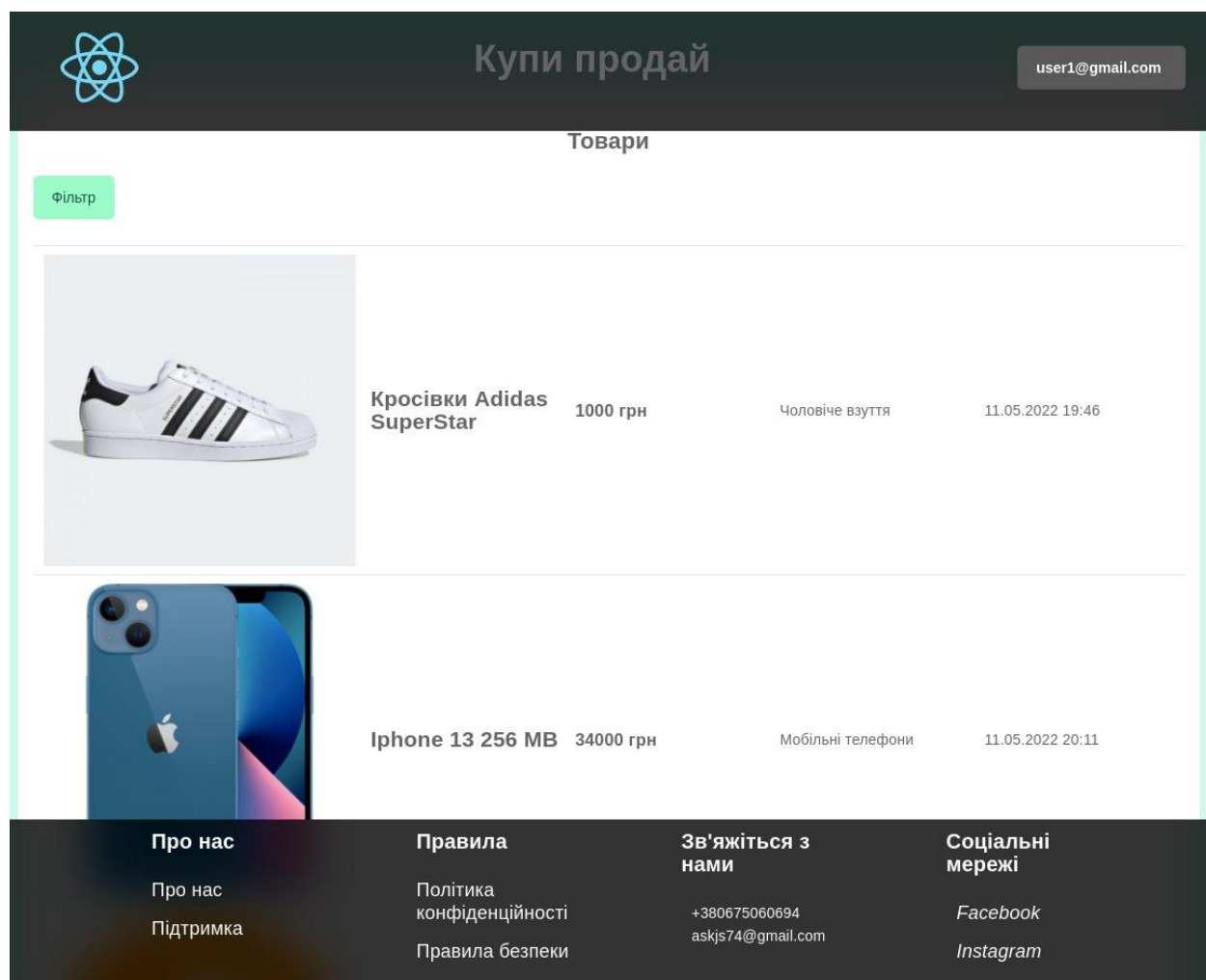


Рис. 3.6. Список оголошень

Сторінка "Мої оголошення" має схожий до головної сторінки оголошень формат, проте вона відображає виключно оголошення, розміщені самим користувачем. Кожне оголошення тут супроводжується додатковою інформацією, наприклад, кількістю лайків і переглядів. На цій сторінці також розміщено інформацію про баланс клієнта з опцією поповнення, що необхідно для користування платними послугами сайту. Крім того, користувачі мають

можливість створювати нові оголошення, які будуть доступні для перегляду всіма відвідувачами сайту. Для цього на сторінці розміщено кнопку "Створити оголошення".

Купи продай

askjs74@gmail.com

Назад

Назва:

Наприклад: Кросівки adidas

Опис *

Ціна:

Ціна в гривнях

Виберіть категорію

Зображення:

Вибрати файли | Файл не вибрано

Оголошення буде закріплено за цією категорією

Зберегти

Про нас	Правила	Зв'яжіться з нами	Соціальні мережі
Про нас	Політика конфіденційності	+380675060694	Facebook
Підтримка	Правила безпеки	askjs74@gmail.com	Instagram

Рис. 3.7. Форма створення оголошення

Кожне оголошення на сайті має окрему сторінку для детального перегляду. Якщо користувач переглядає своє власне оголошення, на цій сторінці стають доступними дві додаткові опції: кнопки "Видалити" та "Редагувати". За допомогою кнопки "Видалити" можна назавжди видалити оголошення, а кнопка "Редагувати" дозволяє внести зміни в інформацію, яка міститься в оголошенні.

Назад

1 of 4

Назва:

Кросівки Adidas SuperStar

Опис *

силуету Superstar. Оновлений шкіряний верх з пізнанням гумовим миском-черепашкою та трьома контрастними смужками для сильних та стильних. Шкіряний верх Класичний прогумований мис-черепашка Підкладка із синтетичних матеріалів Зручна та функціональна устілка OrthoLite з антимікробним покриттям

Ціна:

1000

Додати зображення:

Вибрати файли Файл не вибрано

Видалити зображення:

1, 2 ...

Головне зображення:

Рис. 3.8. Форма редагування оголошення

На сторінці, де представлено деталі оголошення, яке належить іншому користувачу, існує можливість зв'язатися з власником оголошення через чат. Відмінність такого чату від звичних меседжерів полягає в тому, що він прив'язується безпосередньо до конкретного оголошення, а не до власника. Таким чином, один користувач може мати кілька окремих чатів з тією ж самою людиною, але кожен чат буде відноситися до різних оголошень.

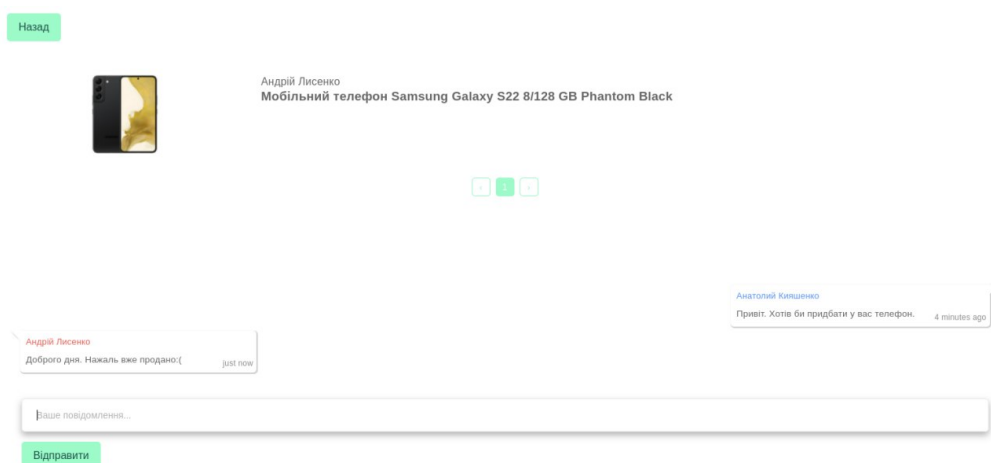


Рис. 3.9. Чат з продавцем

Користувач, який має роль "Адміністратор", отримує розширені права взаємодії з усіма оголошеннями на сайті, включаючи можливість доступу до

видалених оголошень. Це стало можливим завдяки впровадженню системи "м'якого видалення", яка дозволяє зберігати всю інформацію про оголошення навіть після їх видалення. Адміністратор також має можливість переглядати список усіх користувачів, зареєстрованих на сайті.

Користувачі					
Фільтр					
id	Email	Ім'я	Прізвище	Роль	Дата створення
2b3bbb5c-3958-4118-a46e-f716d8625094	askjs74@gmail.com	Анатолий	Кияшенко	ROLE_CLIENT	08.05.2022 18:48
7862fc7f-3cdf-4ca1-bd9c-0868f78ff389	askjs75@gmail.com	Анатолий	Кияшенко	ROLE_ADMIN	10.05.2022 22:15
db6ca415-f81a-4162-bb07-2449e6eeb1a6	user1@gmail.com	User	User	ROLE_CLIENT	08.05.2022 18:56

Рис. 3.10. Список всіх користувачів

На сторінці детального перегляду користувача адміністратор може заблокувати клієнта. Також ця сторінка має всі оголошення користувача і перехід на список чатів з цим користувачем.

Назад

Заблокувати

Повідомлення користувача

ID #1e3fb193-aa99-4080-8f30-469df42cc101

Email askjs74@gmail.com

Ім'я: Анатолий

Прізвище: Кияшенко

Роль: ROLE_CLIENT

Дата створення: 01.01.1970 03:00

Оголошення користувача

Фільтр

	id	Назва	Категорія	Ціна	Дата створення	Дата оновлення	Переглянули	Додали до улюблених
	19cb531c-6950-4e8f-9d4c-11d2c8ac03bb	Кросівки Adidas Runfalcon 2.0	Чоловіче взуття	1399	07.06.2022 21:43	07.06.2022 22:39	3	0
	91a163b6-c365-47f7-8a23-bd17075358a8	Кросівки Nike Defyallday	Чоловіче взуття	1999	07.06.2022 21:42	07.06.2022 22:39	2	0
	360ba318-b4bd-4db4-9b4f-993476bf9e50	Кросівки New Balance 393	Чоловіче взуття	1449	07.06.2022 21:41	07.06.2022 21:41	0	Маєте питання? 🗨️

Рис. 3.11. Сторінка детального перегляду користувача

3.3. Тестування інтеграції платіжних систем для оплати гривнею та крипто валютами

У кожного клієнта є свій баланс, з якого відбувається оплата за платні послуги в системі. Баланс можна переглянути на сторінці зі своїми оголошеннями, в правому верхньому куті сторінки.

	Id	Назва	Категорія	Ціна	Дата створення	Дата оновлення	Переглянули	Додали до улюблених
	19cb531c-6950-4e8f-9d4c-11d2c8ac03bb	Кросівки Adidas Runfalcon 2.0	Чоловіче взуття	1399	07.06.2022 21:43	07.06.2022 22:39	3	0
	91a163b6-c365-47f7-8a23-bd17075358a8	Кросівки Nike Defyallday	Чоловіче взуття	1999	07.06.2022 21:42	07.06.2022 22:39	2	0
	360ba318-b4bd-4db4-9b4f-993476bf9e50	Кросівки New Balance 393	Чоловіче взуття	1449	07.06.2022 21:41	07.06.2022 21:41	0	0
	ace85116-ec02-42e1-811a-5e3b0ebfe04c	Кросівки Puma X-Ray 2 Square	Чоловіче взуття	500	07.06.2022 21:40	13.06.2022 17:25	20	0
	f4c40b2d-c2ac-423c-b9b3-bfcf22736181	Iphone 13 256 GB	Мобільні телефони	33100	11.05.2022 20:59	05.12.2023 13:58	21	2

Рис. 3.12. Перегляд балансу

Під балансом знаходиться кнопка «Поповнити рахунок». При натисканні відбудеться перехід на сторінку поповнення.

Сума:

Введіть суму у вибраній валюті

Після натискання на кнопку "Перейти до оплати" ви будете перенаправлені на сайт платіжної системи, де зможете здійснити оплату.

Перейти до оплати

Рис. 3.13. Сторінка поповнення балансу

Сторінка поповнення містить платіжні методи у вигляді картинок, та поле для вводу суми. Серед платіжних методів є Visa/MasterCard – поповнення карткою

в гривні та 4 популярних криптовалюти: Tether (USDT) Ethereum (ETH) Bitcoin (BTC) Tron (TRX)

Сума вводиться у вибраній валюті. Якщо обрано не гривню, буде відображено приблизну суму, що буде зараховано на баланс, оскільки баланс на платформі онлайн оголошень у гривні.

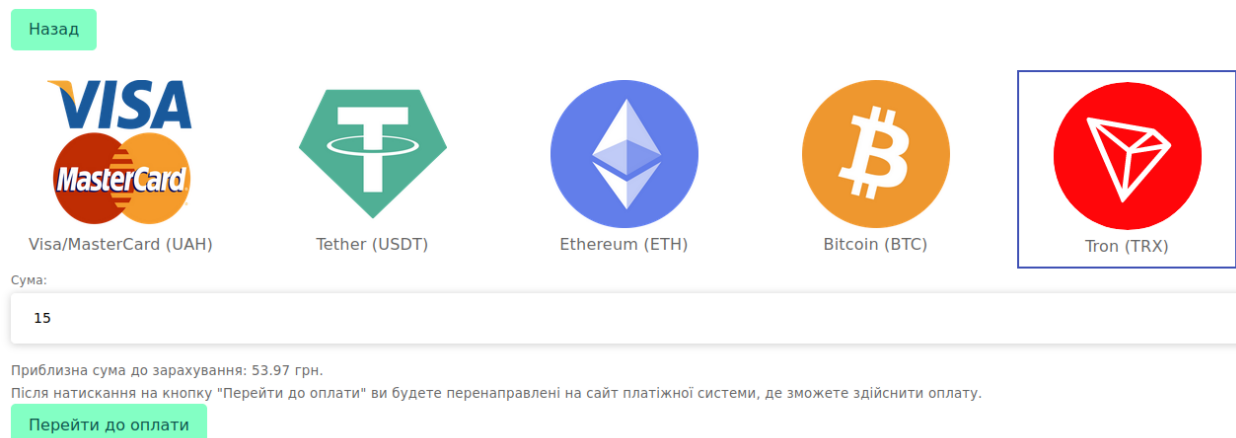


Рис. 3.14. Приблизна сума до зарахування

Сума приблизна, оскільки остаточна сума буде розрахована за курсом на момент зарахування.

При виборі платіжного методу Visa/MasterCard і переходу до оплати буде створено інвойс на оплату картою. Платіжний шлюз у відповідь відправить посилання для оплати і клієнту буде відображено сторінку з формою вводу картки (cardgate), з якої буде списано гроші.



Рис. 3.15. Сторінка з формою вводу картки

При виборі криптовалюти процес аналогічний. Сторінка, що буде бачити користувач в цьому випадку – це фрейм платіжного шлюзу Exsoft, на якому відображено суму та валюту, адресу гаманця і qr код з гаманцем.



Рис. 3.16. Фрейм при виборі оплати криптовалютою

Після відправки коштів на вказаний гаманець статус оплати в Exsoft буде змінено і фрейм оновиться.

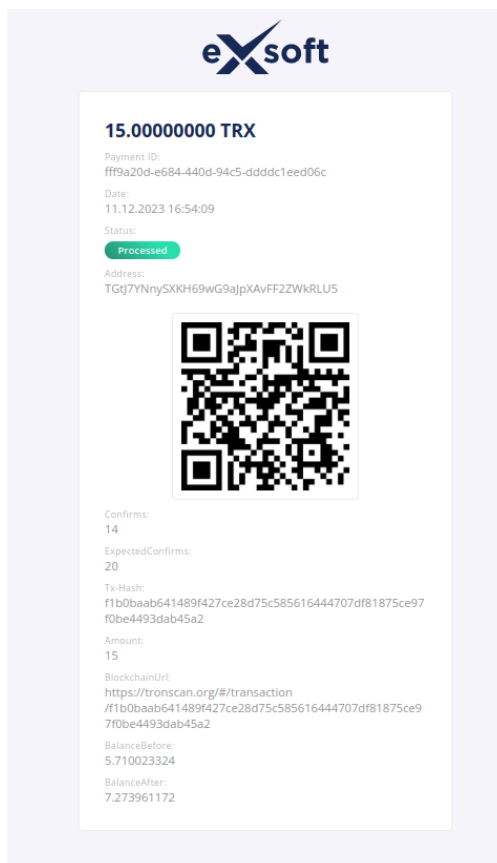


Рис. 3.17. Фрейм з підтвердженням платежем

Також в цей момент буде відправлено запит на платформу онлайн оголошень з даними про цю оплату. Статус оплати буде змінено і буде вирахована сума, що буде зарахована на баланс.

id	Статус	Сума	Зараховано на баланс	Дата створення
57e4436d-3e94-4930-a286-b1c9374479f7	PROCESSED	15 TRX	58.6 грн	11.12.2023 16:54

Рис. 3.18. Оплата зі статусом «Оброблено» в історії платежів

Далі ця сума буде додана на баланс користувача. Сума відрізняється в більшу від тієї, що була вказана на сторінці створення оплати, оскільки, як було сказано вище, вона обраховується за курсом на момент зарахування, а курс оновлюється раз у хвилину. В даному випадку клієнту було зараховано більше ніж він очікував, але може бути і навпаки.

**У вас на рахунку
116.9 грн.**

Поповнити рахунок

Рис. 3.19. Поповнений рахунок

3.4. Тестування роботи програмного комплексу

Робота з веб-сайтом починається з авторизації. Щоб авторизуватися користувач має бути зареєстрованим. Форма реєстрації містить певний набір перевірок. Перш за все це Капча, що може запобігти реєстрації ботів. У випадку непроходження Капчі користувачу буде виведено повідомлення.

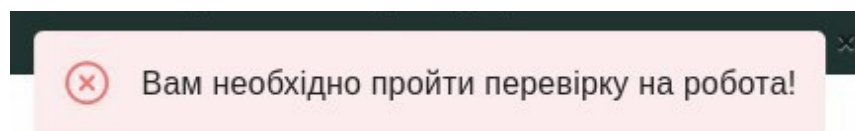


Рис. 3.20. Помилка про непроходження Капчі

Електронна пошта та телефон клієнта мають бути унікальними, в разі спроби реєстрації на вже наявну пошту буде видана помилка.

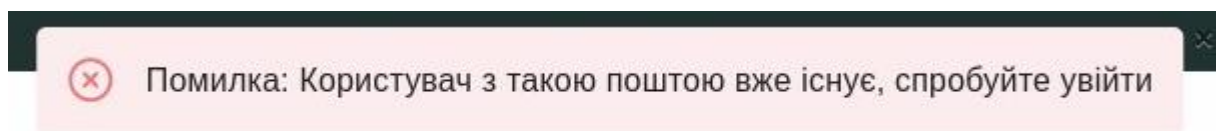


Рис. 3.21. Помилка про дублікат електронної пошти

Після реєстрації на вказану електронну пошту буде надіслано лист з посиланням на підтвердження електронної пошти. Без підтвердження користувач не зможе увійти у систему.

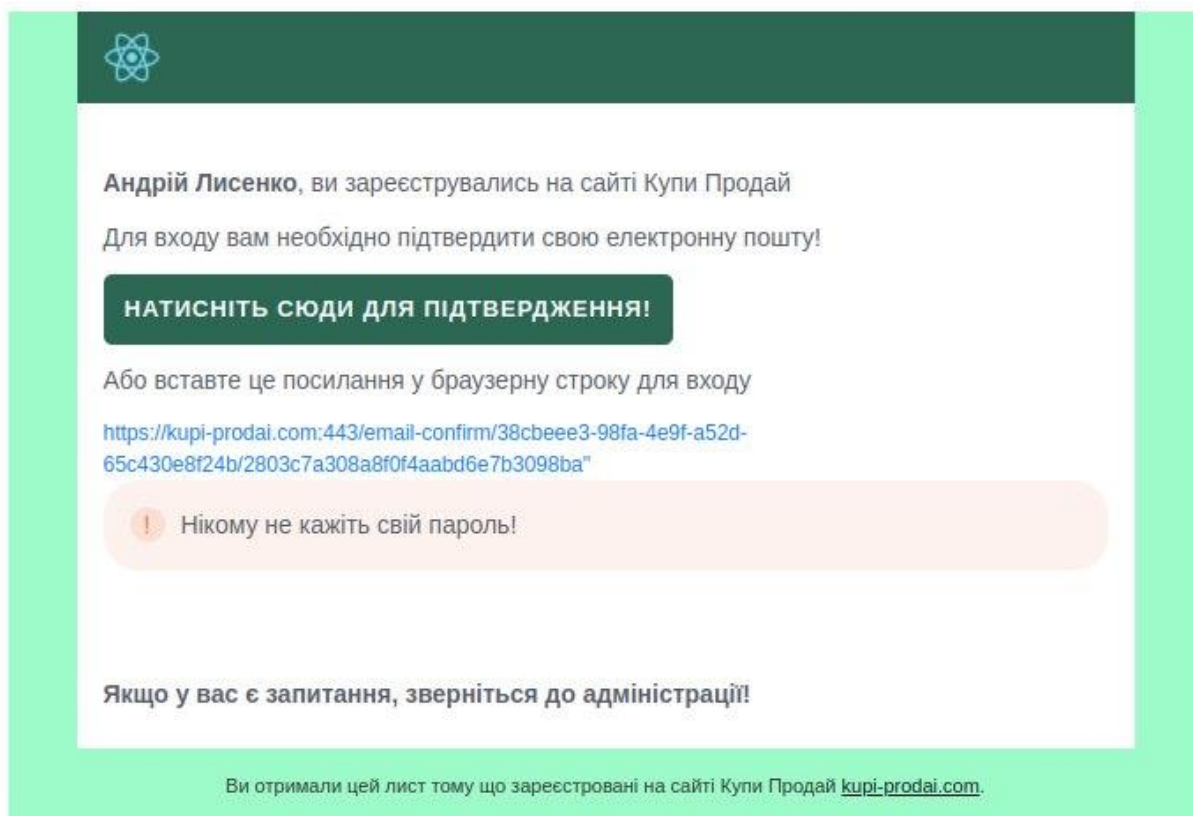


Рис. 3.22. Лист з підтвердженням реєстрації

Лист має посилання, що містить ідентифікатор клієнта та спеціальний токен, що було згенеровано при реєстрації. При переході по посиланню система перевіряє, чи співпадає токен в посиланні з токеном, що зберігається в базі даних для даного клієнта. Якщо перевірка успішна, користувачу відобразиться повідомлення та кнопка з переходом до логіну.

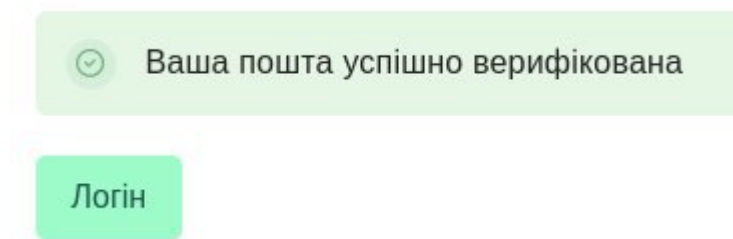


Рис. 3.23. Повідомлення про успішну верифікацію електронної пошти

Авторизований користувач має сторінку «Мої оголошення». Так як користувач щойно зареєструвався, він не має власний оголошень і бачить відповідне повідомлення.



Рис. 3.24. Повідомлення про відсутність оголошень

Кнопка «Створити оголошення» виконує перехід на форму створення оголошення. В ній потрібно вказати назву товару, опис, ціну, вибрати зі списку категорію та вибрати зображення. Вибір зображень не є обов'язковим, також можна вибрати одне чи декілька зображень.

 A screenshot of a form for creating an advertisement. At the top left is a green button labeled "Назад". The form contains several input fields: "Назва:" with the text "Мобільний телефон Samsung Galaxy S22 8/128 GB Phantom Black"; "Опис:" with a detailed description of the phone's specifications; "Ціна:" with the value "33599"; a category dropdown menu currently showing "Мобільні телефони"; and a section for "Зображення:" with a button "Вибрати файли" and the text "файлів: 2". At the bottom, there is a note "Оголошення буде закріплено за цією категорією" and a green button labeled "Зберегти".

Рис. 3.25. Форма створення оголошення

Після збереження оголошення відбувається перехід на сторінку «Мої оголошення» де відображається повідомлення про успішне додавання оголошення.

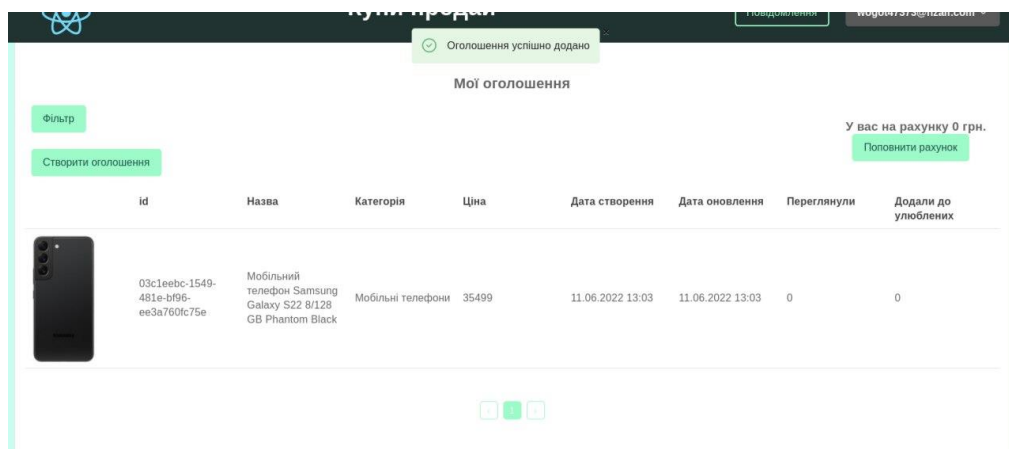


Рис. 3.26. Форма створення оголошення

Тепер на сторінці «Мої оголошення» відображається оголошення, що було додано. Також тут можна моніторити кількість переглядів та уподобань.

Після додавання оголошення його можна відредагувати. Можна змінити назву, опис, ціну, вибрати іншу картинку в якості головної, видалити чи додати нові.



Рис. 3.27. Зміна головної картинки

Інші користувачі бачать це оголошення у списку, та можуть написати повідомлення, якщо вони зацікавлені в ньому.

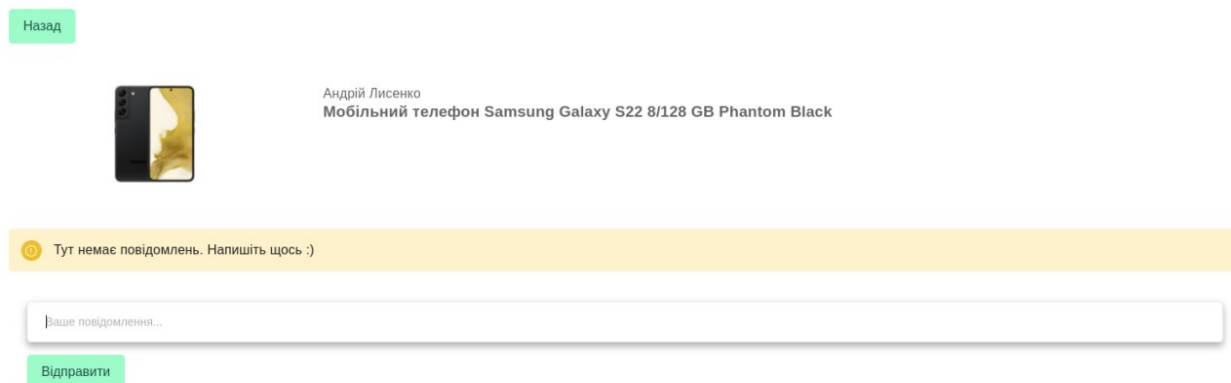


Рис. 3.28. Чат без повідомлень

При натисканні кнопки «Повідомлення» кімната не створюється, вона створюється лише після першого повідомлення. Це зроблено для того, щоб не засмічувати базу даних порожніми кімнатами.

Продавець може відповісти на повідомлення в той момент, як користувач знаходиться на сторінці чату. Нове повідомлення відобразиться йому в реальному часі, без перезавантаження сторінки.

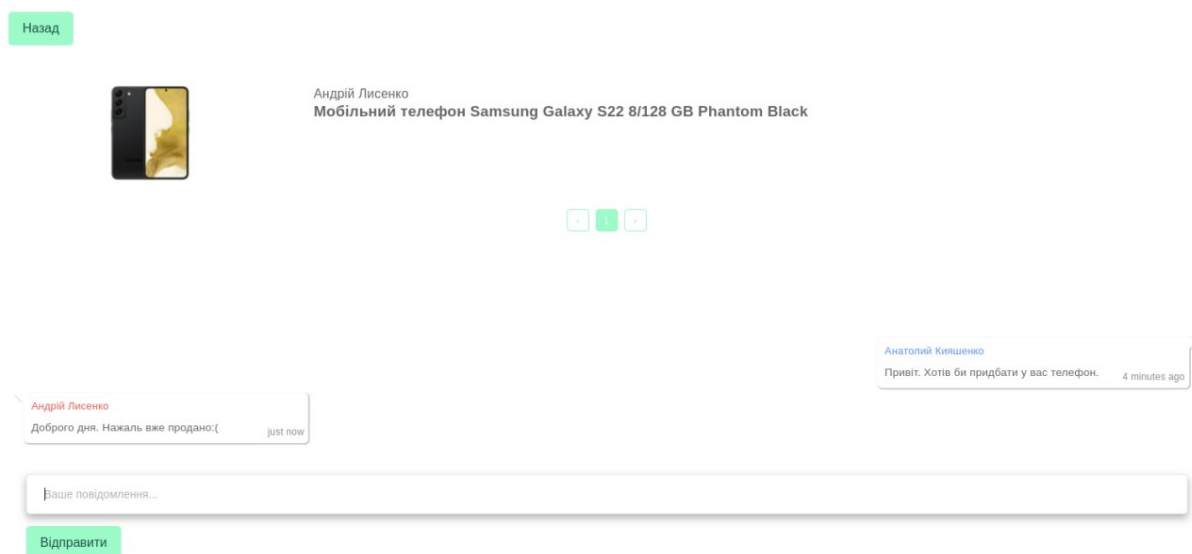


Рис. 3.29. Повідомлення та відповідь у чаті

Оголошення інших користувачів можна додавати в улюблені. В цьому випадку клієнт буде мати швидкий доступ до нього. Також це дає можливість

отримувати лист на електронну пошту, якщо ціна на це оголошення була знижена.

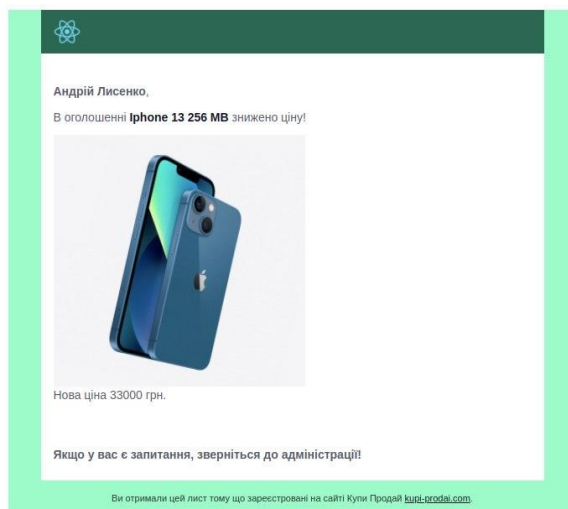


Рис. 3.30. Лист про зниження ціни в оголошенні

Адміністратор бачить всі оголошення, навіть видалені та список користувачів. Детальна сторінка користувача містить інформацію про користувача, його роль, та список оголошень. Також є посилання на повідомлення користувача. З цієї сторінки користувача можна заблокувати. В цьому випадку він не зможе увійти в акаунт, доки його не розблокують. Заблокувати можна не тільки клієнтів, але і адміністраторів, проте заблокувати самого себе не вийде.

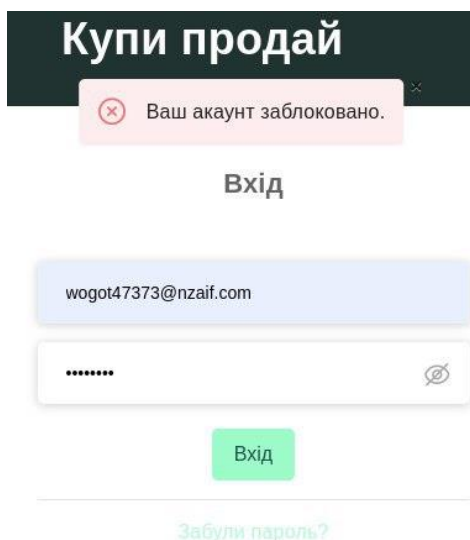


Рис. 3.31. Повідомлення про заблокований акаунт

При поповненні балансу, якщо не вказати платіжну систему буде відображено відповідне повідомлення.



Рис. 3.32. Повідомлення про відсутність платіжного методу

Також зроблена валідація для поля вводу на цифри, перевірка мінімумів та максимумів.

Оскільки веб-сайт працює як React веб-додаток і надсилає запити на API, особлива увага була приділена валідації на backend, щоб користувач не зміг підставити невалідні дані і надіслати їх наприклад через Postman, зламавши тим самим систему.

Логи помилок, системні логи, а також тіла запитів, що надходять від платіжних шлюзів зберігаються у базі даних MongoDB. Це зроблено для того щоб розвантажити основну базу даних MySQL. MongoDB ефективна у вставці нових даних, але повільніша ніж MySQL при виконанні запитів на відображення чи оновлення. Оскільки ці дані статичні, а переглядатися будуть тільки в разі необхідності розробниками чи адміністраторами, MongoDB показала себе чудовим вибором для цієї задачі, оскільки записи в ній це по факту текстові документи, але з можливостями фільтрації та сортування.

```

1 {
2   _id: ObjectId('657724119c15fel2010303d3'),
3   invoiceId: '57e4436d-3e94-4930-a286-b1c9374479f7',
4   data: {
5     data: {
6       returnUrl: 'https://kupi-prodai.ngrok.io/invoice/57e4436d-3e94-4930-a286-b1c9374479f7',
7       attributes: [
8         {
9           attribute: {
10            id: 'l0601',
11            name: 'isStabling',
12            type: 'hidden'
13          },
14          value: '1'
15        }
16      ],
17      flowData: [
18        {
19          name: 'address',
20          value: 'TGtJ7YNnySXKH69wG9aJpXAvFF2ZWkRLU5'
21        },
22        {
23          name: 'qr-code',
24          value: 'tron:TGtJ7YNnySXKH69wG9aJpXAvFF2ZWkRLU5?amount=15.00'
25        },
26        {
27          name: 'Link',
28          value: 'https://cp.exsoft.io/api/transaction-handler/fff9a20d-e684-440d-94c5-ddddcleed06c'
29        },
30        {
31          name: 'confirms',
32          value: '14'
33        },
34        {
35          name: 'expectedConfirms',
36          value: '20'
37        },
38        {
39          name: 'tx-hash',
40          value: 'f1b0baab641489f427ce28d75c585616444707df81875ce97f0be4493dab45a2'
41        },
42        {
43          name: 'amount',
44          value: '15'
45        },
46        {
47          name: 'blockchainUrl',
48          value: 'https://tronscan.org/#/transaction/f1b0baab641489f427ce28d75c585616444707df81875ce97f0be4493dab45a2'
49        },
50        {
51          name: 'BalanceBefore',
52          value: '5.710023324'
53        }
54      ]
55    }
56  }
57 }

```

Рис. 3.33. Приклад тіла запиту, збереженого в MongoDB

Висновки до розділу 3

Третій розділ описує порядок і налаштування системи. Оскільки система розміщена за допомогою Docker, було створена та описана схема його роботи, перераховані контейнери, що використовуються та представлена діаграма розгортання.

Далі наведено опис інтерфейсу платформи створення онлайн оголошень, показаний порядок її роботи. Продемонстровано процес поповнення балансу на платформі карткою та криптовалютами.

Виконано тестування реалізованих алгоритмів, що дозволило знайти і виправити недоліки, підтвердити точність та коректність роботи реалізованого програмного комплексу. Описано процес збереження статичних даних у MongoDB.

ВИСНОВКИ

Провівши дослідження за темою кваліфікаційної роботи магістра та можна зробити висновки, що для успішної реалізації даної задачі потрібно було провести досить кропітку працю над постановкою та аналізом задач, методів і засобів її рішення в першому розділі.

Проаналізувавши актуальність було зроблено висновок, що тема криптовалют дуже популярна. Не дивлячись на вже велику кількість наявних інтернет магазинів тема актуальності не втратила, оскільки це не звичайний інтернет магазин, а сайт з розміщення оголошень, в якому можна розміщувати свої оголошення практично будь-якої категорії. При цьому платний функціонал доступний для людей, яким важлива анонімність, чи відсутня можливість оплати гривнею завдяки прийому криптовалюти.

Перед початком розробки було проаналізовано існуючі платіжні сервіси та зроблені висновки про їх переваги та недоліки. Було визначено платіжний шлюз, що буде інтегровано в платформу створення оголошень. Також було вирішено якою має бути система, сформовані вимоги до неї. Відповідно до вимог необхідно було вибрати інструментальні засоби, за допомогою яких їх можна реалізувати. Оскільки програмний додаток має бути масштабованим, с можливістю подальшого розвитку архітектура MVC була вибрана в якості основи і відповідно до цього підбирались потрібні фреймворки.

На другому етапі були сформовані основні алгоритми роботи. Були створені діаграми активності, станів взаємодії компонентів. Створено діаграму послідовності поповнення балансу за допомогою криптовалюти, завдяки чому було визначено, які методи потрібно інтегрувати в систему. Також було розроблено об'єктно-орієнтовану модель системи з використанням патернів проектування. Після формувань алгоритмів було зроблено ядро системи, який потім дороблювався відповідно до заданих умов.

Третій етап розробки полягав в налаштуванні системи, заповнення її тестовими даними. Було розроблено дизайн веб-сайту. На останок було

проведено тестування компонентів системи, що дозволило знайти низку багів та дір.

В результаті чого було отримано програмний продукт, який відповідає всім поставленим вимогам та може стати непоганим стартапом. Розгортання проекту за допомогою Docker дозволяє продовжити розробку проекту практично на будь-якій Unix системі однією кнопкою.

У цілому, дана робота підкреслює важливість інтеграції криптовалютних платіжних систем у платформи онлайн оголошень. Вона виявляє, що інтеграція криптовалют може значно розширити ринкові можливості, залучаючи нових користувачів і пропонуючи інноваційні та безпечні способи оплати. Робота також визначає ключові технологічні інструменти та алгоритми, необхідні для ефективної реалізації таких платіжних систем, гарантуючи гнучкість та масштабованість платформи. Тестування розроблених алгоритмів підтверджує їх точність та коректність роботи, що забезпечує високий рівень безпеки та прозорості в транзакціях. Таким чином, дослідження відкриває нові можливості для бізнесу, пропонуючи інноваційні, безпечні та гнучкі рішення для сучасного цифрового ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jeff Nickoloff, Stephen Kuenzli, *Docker in Action*, Manning Publications, 2016, pp 120-126.
2. Ujan Mukhopadhyay, *A brief survey of Cryptocurrency systems*, IEEE, 2016.
3. Marcin Wątopek, *Multiscale characteristics of the emerging global cryptocurrency market*, Journal of Network and Computer Applications, vol 901, Elsevier, 2021.
4. Monia Milutinović, *CRYPTOCURRENCY*, Ekonomika - Journal of economic theory and practice and social issues, Društvo ekonomista "Ekonomika" Niš, 2018, pp. 105-122.
5. Jiajing Wu, *Analysis of cryptocurrency transactions from a network perspective: An overview*, Journal of Network and Computer Applications, vol 190, Elsevier, 2021.
6. Francois Zaninotto, Fabien Potencier, *The Definitive Guide to symfony*, Apress, 2017.
7. Majida Laaziri, *A comparative study of laravel and symfony PHP frameworks*, International Journal of Electrical and Computer Engineering (IJECE) Vol. 9, No. 1, 2019, pp. 704-712.
8. Bartosz Porebski, Karol Przystalski, Leszek Nowak, *Building PHP Applications with Symfony, CakePHP, and Zend Framework*, Willey Publishing inc., 2021.
9. Ерік Фрімен, Елізабет Робсон, *Паттерни проєктування*, Фабула, 2020.
10. L. Johnson and R. Miller, *Modern Web Development with React: A Practical Guide*, in Proc. of the International Conference on Web Engineering, 2022, pp. 125-132.
11. S. Thompson, *React for Beginners: An Introduction*, Journal of Web Development, vol. 15, no. 4, 2021, pp. 234-245.

12. A. Davis and T. Brown, *Building Scalable Web Applications with Symfony*, in Proc. of the Symposium on Advanced Web Technologies, 2021, pp. 310-318.
13. N. Patel, *Symfony versus Laravel: A Comparative Study*, Journal of Software Engineering, vol. 22, no. 3, 2020, pp. 456-465.
14. K. Zhang and M. Li, *Understanding the Blockchain Technology in Cryptocurrencies*, IEEE Transactions on Financial Technology, vol. 9, no. 2, 2022, pp. 120-130.
15. J. Turner and S. Gupta, *The Rise of Bitcoin: An Analysis*, in Proc. of the Conference on Digital Finance, 2023, pp. 542-549.
16. R. Sharma and L. Nguyen, *Digital Payment Systems: Evolution and Trends*, Journal of Payment Technologies, vol. 17, no. 1, 2021, pp. 75-84.
17. C. Wilson, *Integrating Blockchain in Payment Systems: Prospects and Challenges*, in Proc. of the International Conference on E-commerce and Blockchain, 2022, pp. 202-210.
18. Adam Boduch, *React and React Native*, Pact Publishing, 2017.
19. Вакалюк Т.А., Кияшенко А.С. *ПЕРСПЕКТИВИ ВИКОРИСТАННЯ NOSQL СИСТЕМ КЕРУВАННЯ БАЗАМИ ДАНИХ*, Матеріали III Всеукраїнської науково-практичної Інтернет-конференції «СУЧАСНІ КОМП'ЮТЕРНІ ТА ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ» 02-19 грудня 2022 р, 2022, С. 119.
20. Вакалюк Т.А., Кияшенко А.С. *ВИКОРИСТАННЯ ТЕХНОЛОГІЇ DOCKER В РОЗРОБЦІ ВЕБ-ОРІЄНТОВАНОГО ДОДАТКУ «ПЛАТФОРМА СТВОРЕННЯ ОНЛАЙН ОГолошень»*, Матеріали III Всеукраїнської науково-практичної Інтернет-конференції «СУЧАСНІ КОМП'ЮТЕРНІ ТА ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ» 02-19 грудня 2022 р. С. 256.
21. *React Doc*, [Online]. Available: <https://uk.legacy.reactjs.org>. [Accessed on: July 1, 2023].

22. *Api Platform Documentation*, [Online]. Available: <https://api-platform.com/docs>. [Accessed on: July 1, 2023].
23. *Symfony Documentation*, [Online]. Available: <https://symfony.com/doc/current/index.html>. [Accessed on: Oct. 1, 2023].
24. *Doctrine 2 ORM's documentation*, [Online]. Available: <https://www.doctrine-project.org/projects/doctrine-orm/en/2.10/index.html>. [Accessed on: Oct. 3, 2023].
25. *Mercure Documentation*, [Online]. Available: <https://mercure.rocks/docs>. [Accessed on: Oct. 7, 2023].
26. *Патерни/Шаблони проектування*, [Online]. Available: <https://refactoring.guru/uk/design-patterns>. [Accessed on: Oct. 13, 2023].
27. *Now Payments, About us*, [Online]. Available: <https://nowpayments.io/about>. [Accessed on: Nov. 25, 2023].
28. *Acceptcoin, Про нас*, [Online]. Available: <https://acceptcoin.io/ua/about-us>. [Accessed on: Nov. 25, 2023].
29. *Exsoft, About us*, [Online]. Available: <https://exsoft.io/about-company>. [Accessed on: Nov. 25, 2023].
30. *Plisio, Cryptocurrency Adoption Statistics Worldwide in 2023*, [Online]. Available: <https://plisio.net/blog/cryptocurrency-adoption-statistics-worldwide-in-2023>. [Accessed on: Nov. 28, 2023].
31. *Coinmap, Crypto ATMs & merchants of the world*, [Online]. Available: <https://coinmap.org/>. [Accessed on: Nov. 28, 2023].

ДОДАТКИ

Лістинг файлу «CreateInvoiceAction.php»

```

<?php

namespace App\Controller\Invoice;

use ApiClient\Api\ControlPanel\Response\Connection\Connection;
use ApiClient\Api\ControlPanel\Response\Transaction\Transaction;
use ApiClient\Api\ControlPanel\Response\Transaction\TransactionAttribute;
use ApiClient\ErrorResponse;
use App\Entity\Invoice\FlowData;
use App\Entity\Invoice\Invoice;
use App\Entity\Invoice\PaymentMethod;
use Doctrine\ORM\EntityManagerInterface;
use Exception;
use Symfony\Component\HttpFoundation\Request;
use App\Service\ControlPanel\ControlPanel;
use Symfony\Component\Routing\Generator\UrlGeneratorInterface;
use Symfony\Contracts\HttpClient\Exception\TransportExceptionInterface;

class CreateInvoiceAction
{
    /**
     * @var ControlPanel
     */
    private ControlPanel $controlPanel;

    /**
     * @var UrlGeneratorInterface
     */
    private UrlGeneratorInterface $generator;

    /**
     * @var EntityManagerInterface
     */
    private EntityManagerInterface $entityManager;

    /**
     * @param ControlPanel $controlPanel
     * @param UrlGeneratorInterface $generator
     * @param EntityManagerInterface $entityManager
     */
    public function __construct(
        ControlPanel $controlPanel,
        UrlGeneratorInterface $generator,
        EntityManagerInterface $entityManager
    )
    {
        $this->controlPanel = $controlPanel;
        $this->generator = $generator;
        $this->entityManager = $entityManager;
    }

    /**
     * @param Invoice $data
     * @param Request $request
     * @return Invoice
     * @throws TransportExceptionInterface
     * @throws Exception

```

```

    */
    public function __invoke(Invoice $data, Request $request): Invoice
    {
        /** @var PaymentMethod $paymentMethod */
        $paymentMethod = $this->entityManager->getRepository(PaymentMethod::class)->findOneBy([
            'currency' => $data->getCurrency()
        ]);

        if (!$paymentMethod) {
            throw new Exception('No payment method found');
        }

        $data->setPaymentMethod($paymentMethod->getPaymentMethod());
        $api = $this->controlPanel->getConnection();

        $connections = $api->connection()->getAll([
            'service.name' => $data->getPaymentMethod(),
            'currency.asset' => $data->getCurrency()
        ]);

        if (!$connections) {
            throw new Exception('No connections found');
        }

        /** @var Connection $connection */
        $connection = $connections->first();

        $data->addFlowData((new FlowData)->setName('UAH course')->setValue($this->controlPanel->getUAHCourse()->getSelling()));

        $transaction = $this->createPayment($data, $this->getInvoiceData($data, $paymentMethod, $connection));

        $data->setProcessingId($transaction->getId());

        foreach ($transaction->getFlowData() as $flowData) {
            if ($flowData['name'] === 'link') {
                $data->setPaymentUrl($flowData['value']);
            }
        }

        $data->addFlowData((new FlowData)->setName($flowData['name'])->setValue($flowData['value']));
    }

    $data->setStatus($transaction->getStatus()->getValue());

    return $data;
}

/**
 * @param Invoice $invoice
 * @param PaymentMethod $paymentMethod
 * @param Connection $connection
 * @return array
 */
    public function getInvoiceData(Invoice $invoice, PaymentMethod $paymentMethod, Connection $connection): array
    {
        $route = "payment_callback";

        $invoiceData = [
            'paymentSystem' => $paymentMethod->getPaymentSystem(),

```

```

        'amount'          => $invoice->getAmount(),
        'currency'        => $paymentMethod->getCurrency(),
        'referenceId'     => $invoice->getId()->toString(),
        'connection'      => $connection->getId(),
        'returnUrl'       => "{$_ENV['BASE_URL']}invoice/{$invoice->getId()}",
        'callBackUrl'     => $this->generator->generate($route, ["id" =>
$invoice->getId()], UrlGeneratorInterface::ABSOLUTE_URL),
        'note'            => "Kupi-prodai payment"
    ];

    if ($paymentMethod->getPaymentMethod() === PaymentMethod::CRYPTO) {
        $attr = $this->controlPanel->getConnection()-
>transactionAttribute()->getAll([
            'service.name'          => $paymentMethod->getPaymentMethod(),
            'paymentSystem.name' => $paymentMethod->getPaymentSystem(),
            'currency.asset'       => $paymentMethod->getCurrency(),
            'transactionType'      => 'payment'
        ]);

        /** @var TransactionAttribute $isStablingAttr */
        $isStablingAttr = $attr->first();

        $invoiceData['attributes'] = [
            [
                'attribute' => $isStablingAttr->getId(),
                'value'     => "1"
            ]
        ];
    }

    return $invoiceData;
}

/**
 * @param Invoice $invoice
 * @param array $invoiceData
 * @return Transaction
 * @throws TransportExceptionInterface
 */
private function createPayment(Invoice $invoice, array $invoiceData):
Transaction
{
    $requestContent = $this->controlPanel->getConnection()->payment()-
>create(
        $invoiceData['returnUrl'],
        $invoiceData['paymentSystem'],
        bcddiv($invoiceData['amount'], 1, 2),
        $invoiceData['currency'],
        $invoiceData['referenceId'],
        $invoiceData['callBackUrl'],
        $invoiceData['connection'],
        array_key_exists('attributes', $invoiceData) ?
$invoiceData['attributes'] : []
    );

    if ($requestContent instanceof ErrorResponse) {
        throw new \RuntimeException(json_encode($requestContent-
>getResponseContent()), 500);
    }

    /** @var Transaction $transaction */
    $transaction = $requestContent->first();

```

```
        return $transaction;
    }
}
```

Лістинг файлу «PaymentCallbackController.php»

```

<?php

namespace App\Controller\Invoice;

use ApiClient\Api\ControlPanel\Response\Transaction\Transaction;
use App\Entity\Invoice\FlowData;
use App\Entity\Invoice\Invoice;
use App\Service\ControlPanel\ControlPanel;
use App\Service\FlowData\FlowDataService;
use Doctrine\ORM\EntityManagerInterface;
use Exception;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Component\Serializer\Exception\ExceptionInterface;
use Symfony\Component\Serializer\Normalizer\DenormalizerInterface;
use MongoDB\Client;
use Throwable;

/**
 * @Route("/api")
 */
class PaymentCallbackController extends AbstractController
{
    /**
     * @var DenormalizerInterface
     */
    protected DenormalizerInterface $denormalizer;

    /**
     * @var EntityManagerInterface
     */
    protected EntityManagerInterface $entityManager;

    /**
     * @var Client
     */
    protected Client $client;

    /**
     * @var ControlPanel
     */
    private ControlPanel $controlPanel;

    /**
     * @param EntityManagerInterface $entityManager
     * @param DenormalizerInterface $denormalizer
     * @param ControlPanel $controlPanel
     */
    public function __construct(
        EntityManagerInterface $entityManager,
        DenormalizerInterface $denormalizer,
        ControlPanel $controlPanel
    )
    {
        $this->entityManager = $entityManager;
    }

```

```

        $this->denormalizer = $denormalizer;
        $this->client = new Client($_ENV['MONGODB_URL']);
        $this->controlPanel = $controlPanel;
    }

    /**
     * @param Request $request
     * @param string $id
     * @return Response
     * @Route("/payment-callback/{id}", name="payment_callback")
     */
    public function paymentCallback(Request $request, string $id): Response
    {
        /** @var Invoice $paymentInvoice */
        $paymentInvoice = $this->getDoctrine()->getRepository(Invoice::class)-
>find($id);

        if (!$paymentInvoice) {
            return new Response("UNSUBSCRIBE OF NOT EXIST INVOICE (PAYMENT)",
Response::HTTP_OK);
        }

        try {
            $requestPaymentContent = $this->checkInvoiceRequest($request,
$paymentInvoice, 'payment');
        } catch (Throwable $exception) {
            return new Response($exception->getMessage(), Response::HTTP_OK);
        }

        $paymentInvoice->setProcessedAmount($requestPaymentContent-
>getProcessedAmount());
        $paymentInvoice->setAmount($requestPaymentContent->getAmount());

        // INSERT FLOW DATA
        $paymentFlowContent = $requestPaymentContent->getFlowData();

        try {
            if ($paymentFlowContent) {
                FlowDataService::setFlowData(
                    $this->entityManager, $paymentInvoice, $paymentFlowContent
                );
            }
        } catch (\Exception $exception) {
            return new Response("INSERT FLOW DATA ERROR " . $exception-
>getMessage(), Response::HTTP_OK);
        }

        $paymentInvoice->setStatus($requestPaymentContent->getStatus()-
>getValue());

        if ($requestPaymentContent->getStatus()->getValue() !==
Invoice::STATUS_PROCESSED) {
            $this->getDoctrine()->getManager()->persist($paymentInvoice);
            $this->getDoctrine()->getManager()->flush();

            return new Response("END CALLBACK", Response::HTTP_OK);
        }

        $this->processUserBalance($paymentInvoice);

        $this->getDoctrine()->getManager()->persist($paymentInvoice);
        $this->getDoctrine()->getManager()->flush();
    }

```

```

        return new Response("END CALLBACK", Response::HTTP_OK);
    }

    /**
     * @param Request $request
     * @param Invoice $invoice
     * @param string $direction
     * @return Transaction
     * @throws Exception
     * @throws ExceptionInterface
     */
    public function checkInvoiceRequest(
        Request $request,
        Invoice $invoice,
        string $direction): Transaction
    {
        if ($invoice->getStatus() !== Invoice::STATUS_PENDING) {
            throw new Exception("UNSUBSCRIBE OF " . $invoice->getStatus() . "
STATUS ($direction CALLBACK)");
        }

        $this->client->monolog->CallbackLogs->insertOne(['invoiceId' =>
$invoice->getId()->toString(), 'data' => json_decode($request->getContent(),
true), 'createdAt' => time()]);

        $content = json_decode($request->getContent(), true)["data"];

        /** @var Transaction $requestContent */
        $requestContent = $this->denormalizer->denormalize($content,
Transaction::class, "array");

        return $requestContent;
    }

    /**
     * @param Invoice $invoice
     * @return void
     */
    private function processUserBalance(Invoice $invoice)
    {
        $balanceBefore = $invoice->getFlowDataByName('BalanceBefore');

        if ($balanceBefore) {
            $balanceBefore = $balanceBefore->getValue();
        } else {
            $balanceBefore = 0;
        }

        $balanceAfter = $invoice->getFlowDataByName('BalanceAfter');

        if ($balanceAfter) {
            $balanceAfter = $balanceAfter->getValue();
        } else {
            $balanceAfter = 0;
        }

        $uahCourse = $this->controlPanel->getUAHCourse();
        $uahCourseFlowData = $invoice->getFlowDataByName('UAH course');

        if (!$uahCourseFlowData) {
            $uahCourseFlowData = new FlowData;
            $uahCourseFlowData->setName('UAH course');
            $uahCourseFlowData->setInvoice($invoice);
        }
    }

```

```
    }  
  
    $uahCourseFlowData->setValue($uahCourse->getSelling());  
  
    $addedAmount = ($balanceAfter - $balanceBefore) * $uahCourse->getSelling();  
  
    $invoice->setProcessedAmount(round($addedAmount, 2));  
    $invoice->getUser()->setMoney(round($invoice->getUser()->getMoney() +  
$addedAmount, 2));  
    }  
}
```