


Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))
Кафедра комп'ютерних наук
(повна назва кафедри, циклової комісії)

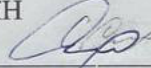
МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:

**«Інформаційна технологія підвищення ефективності роботи
обмінника криптовалют»»**

Виконав: студент 2-го курсу, групи 2КН-22м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)


Сліпенького В.О. 

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН
Сілагін О.В. 


(прізвище та ініціали)

« 04 » 12 2023 р.

Опонент: к.т.н., доц. каф. АІТ
Богач І.В. 

(прізвище та ініціали)

« 04 » 12 2023 р.

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А.А. 

(прізвище та ініціали)

« 03 » 12 2023 р.

Вінниця ВНТУ – 2023 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.

29.08. 2023 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ
СТУДЕНТУ

Сліпенькому Владиславу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи: «Інформаційна технологія підвищення ефективності роботи обмінника криптовалют»

Керівник роботи доцент кафедри КН Сілагін О.В.

Затверджені наказом вищого навчального закладу від “18” 08 2023 року № 247

2. Строк подання студентом роботи 13.11. 2023 року

3. Вихідні дані до роботи: Мова програмування – Vue3JS; Інтерфейс графіка криптовалют(API) – COinMarketCap API; Автентифікація – Security Assertion Markup Language; Хостинг – TiMeweb; Максимальна к-сть заявок – 1000; операційна система – Windows 7/8/10/11

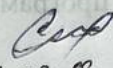

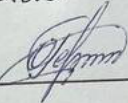
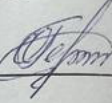
4. Зміст текстової частини:

Вступ, аналіз відомих рішень в галузі обміну криптовалют, розробка архітектури сайту обмінника, розробка програмного засобу, економічна частина, висновки, перелік використаних джерел, додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Схема алгоритму загального функціонування програмного забезпечення; вимоги до обмінника криптовалют; Загальний вигляд інтерфейсу користувача; приклад роботи програми; математична модель функціонування системи, загальна UML-діаграма класів, приклад роботи програмного засобу проходження лабіринту на базі генетичного алгоритму.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Сілагін О. В., к.т.н., доц. каф. КН	 29.08.23	 10.11.23
4	Ратушняк Ольга Георгіївна к.т.н., доц. каф. ЕПВМ	 21.09.23	 29.11.23

7. Дата видачі завдання 29. 08 2023 року

№ з/п	Назва етапів Магістерської кваліфікаційної роботи	Строк виконання етапів роботи	При- мітка
1	Аналіз відомих рішень в галузі обміну криптовалют	1.09.23-07.09.23	
2	Розробка архітектури сайту обмінника	08.09.23 - 20.09.23	
3	Розробка програмного засобу	21.09.23 - 20.10.23	
4	Підготовка економічної частини	21.09.23 - 29.10.23	
5	Апробація та/або впровадження результатів дослідження	30.10.23 - 05.10.23	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23 - 10.11.23	

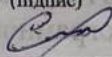
Студент



Сліпенький В.О.

(підпис)

Керівник роботи



Сілагін О.В.

(підпис)

АНОТАЦІЯ

УДК 004.8

Сліпенький В.О. Інформаційна технологія підвищення ефективності роботи обмінника криптовалют. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп'ютерні науки, освітня програма - «Системи штучного інтелекту». Вінниця: ВНТУ, 2023. 95 с.

На укр. мові. Бібліогр.: 24 назв; рис.: 23; табл. 8.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології для підвищення ефективності роботи обмінника криптовалют. У роботі були досліджені та проаналізовані існуючі підходи до оптимізації обміну криптовалют, і як найбільш обіцяний було вибрано методи, базовані на машинному навчанні та аналізі ринкової активності.

На початковому етапі дослідження було вивчено існуючі алгоритми та підходи до обміну криптовалют, включаючи технічний аналіз та стратегії торгівлі. Після цього був обрано підхід, який базується на використанні машинного навчання та аналізу даних для прийняття рішень щодо оптимального часу та курсу обміну.

Для розробки інформаційної технології був розроблений власний алгоритм, який враховує ринкові фактори, новини та інші важливі аспекти при прийнятті рішень про обмін криптовалют. Також була створена UML-діаграма класів, що відображає структуру програмного рішення.

З метою реалізації розробленої моделі та алгоритму було обрано відповідне середовище розробки, у даному випадку Visual Studio, і мову програмування C#. Програмне забезпечення було реалізовано та піддано інтенсивному тестуванню. Результати тестування підтвердили правильність роботи алгоритму та ефективність розробленої інформаційної технології.

Ключові слова: інформаційна технологія, обмін криптовалют, машинне навчання, аналіз ринку, оптимізація, стратегії торгівлі, розробка алгоритму.

ABSTRACT

V.O. Slipenkyi. Information Technology for Improving the Efficiency of Cryptocurrency Exchange. Master's Qualification Thesis in the field of 122 – Computer Science, Educational Program - Computer Science. Vinnytsia: Vinnytsia National Technical University (VNTU), 2023. 95 pages.

In Ukrainian language. Bibliographer: 24 titles; fig.: 23; table 8.

This master's qualification work is dedicated to the development of an information technology for enhancing the efficiency of cryptocurrency exchange operations. The study involved an examination and analysis of existing approaches to cryptocurrency exchange optimization, and the most promising method based on machine learning and market activity analysis was chosen.

In the initial research phase, existing algorithms and approaches to cryptocurrency exchange were explored, including technical analysis and trading strategies. Subsequently, an approach that leverages machine learning and data analysis to make decisions regarding the optimal timing and rate of exchange was selected.

To develop the information technology, a proprietary algorithm was created, taking into account market factors, news, and other critical aspects when making cryptocurrency exchange decisions. Additionally, a UML class diagram was constructed to illustrate the program's structure.

In order to implement the developed model and algorithm, an appropriate development environment was selected, in this case, Visual Studio, and the C# programming language. The software was implemented and subjected to rigorous testing. The test results confirmed the correctness of the algorithm's operation and the effectiveness of the developed information technology.

Keywords: information technology, cryptocurrency exchange, machine learning, market analysis, optimization, trading strategies, algorithm development.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ВІДОМИХ РІШЕНЬ В ГАЛУЗІ ОБМІНУ КРИПТОВАЛЮТ	8
1.1 Обґрунтування доцільності створення крипто-обмінника	8
1.2 Порівняльний аналіз відомих крипто обмінників	11
1.3 Розробка загальної дизайнерської концепції інформаційної технології підвищення ефективності роботи обмінника криптовалют	17
1.4 Висновок розділу 1.....	19
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОБОТИ ОБМІННИКА КРИПТОВАЛЮТ	21
2.1 Математичне моделювання етапів технології обміну	21
2.2 Створення технології швидкого обміну криптовалют	26
2.3 Аналіз програмних технологій для взаємодії з Blockchain.....	28
2.4 Розробка UML діаграм інформаційної технології підвищення ефективності роботи обмінника криптовалют	31
2.5 Декомпозиція сайту та дизайн окремих частин	32
2.6 Розробка архітектури інформаційної технології підвищення ефективності роботи обмінника криптовалют	34
2.7 Висновок до розділу 2.....	37
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОБОТИ ОБМІННИКА	39
3.1 Вибір та обґрунтування мов розробки сайту.....	39
3.2 Обґрунтування середовища програмування.....	44
3.3 Розробка функціоналу веб обмінника.....	48
3.4 Розробка інтерфейсу програмного засобу.....	53
3.5 Приклад роботи вебобмінника криптовалют.....	57
3.6 Висновок до розділу 3.....	62

4 ЕКОНОМІЧНА ЧАСТИНА.....	63
4.1 Комерційний та технологічний аудит науково-технічної розробки.....	63
4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи.....	66
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	71
4.4 Висновок до розділу 4.....	76
ВИСНОВКИ.....	77
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	79
Додаток А (обов'язковий) Результат перевірки на плагіат в онлайн-системі UNICHECK.....	81
Додаток Б (обов'язковий) Лістинг програми.....	82
Додаток В (обов'язковий) Ілюстративна частина.....	99
Додаток Г (довідниковий) Інструкція користувача.....	103

ВСТУП

Актуальність. У сучасному світі наше життя все більше нерозривно пов'язане з грошима, даними і документами. Це породжує тенденцію до збільшення об'єму важливої інформації що, зберігається на серверах різних компаній. Також почали з'являться електронні платежі, різновиди яких, в наш час існує досить широкий спектр. Така варіативність важливої інформації створює проблеми: недовіра до посередників, що пов'язані з надійністю зберігання цієї інформації, наявність великих комісійних зборів та швидкість обробки фінансових операцій, адже іноді переказ коштів між країнами може зайняти близько одного робочого тижня.

За останнє десятиліття криптовалюти стали важливою складовою світового фінансового ландшафту. Зростання інтересу до цифрових активів призвело до значного збільшення обсягів торгівлі криптовалютами та широкого використання криптовалютних бірж для конвертації, зберігання та передачі цих активів. Однак цей бурхливий розвиток ринку пов'язаний з численними проблемами, які потребують комплексного вирішення.

Зокрема, криптовалютні біржі стикаються з такими проблемами, як високі ризики безпеки, волатильність обмінних курсів, надмірна комісія за транзакції, а також обмежена масштабованість і продуктивність існуючої інфраструктури. Ці проблеми підкреслюють необхідність підвищення операційної ефективності криптовалютних бірж та розробки інноваційних рішень для вирішення цих питань. Транзакції з криптовалютами можуть бути виконані дуже швидко. Наприклад, транзакція біткоїну може бути виконана за кілька хвилин. Це може зробити криптовалюти більш ефективними, ніж традиційні форми платежів, які можуть займати кілька днів або навіть тижнів.

Актуальність цього дослідження полягає в тому, що відповідні інформаційно-технологічні рішення та постійний розвиток блокчейн-технологій відкривають нові можливості для оптимізації роботи криптовалютних бірж. Вони здатні значно підвищити ефективність, зменшити

ризика та покращити якість обслуговування клієнтів у цьому секторі. Це дослідження має на меті надати інсайти, які допоможуть вирішити нагальні проблеми в індустрії криптовалютних бірж та сприятимуть постійному розвитку криптовалютного ринку.

Таким чином, це дослідження має важливе значення для забезпечення стабільності та зростання ринку криптовалют. Воно може призвести до створення інноваційних інформаційно-технологічних рішень, які можуть значно підвищити ефективність та безпеку процесів обміну криптовалют. Вирішуючи ці нагальні проблеми, дане дослідження може мати довготривалий вплив на індустрію та сприяти подальшому розвитку криптовалютної торгівлі.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напряму наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання досліджень. Підвищення швидкості та безпеки роботи технології обмінника криптовалют.

Для досягнення поставленої мети слід вирішити наступні завдання:

- Математичне моделювання етапів технології обміну криптовалют.
- Проаналізувати створення технології швидкого обміну криптовалют.
- Аналіз програмних технологій для взаємодії з Blockchain.
- Розробити архітектури інформаційної технології підвищення ефективності роботи обмінника криптовалют.
- Надати комплексну оцінку потенційного впливу впровадження інформаційних технологій на індустрію обміну криптовалют, її учасників та фінансовий ринок в цілому.

Об'єкт дослідження – процес розробки інформаційної технології підвищення ефективності роботи обмінника криптовалют.

Предмет дослідження – алгоритми та програмні засоби обміну криптовалютою.

Методи дослідження. Для досягнення мети дослідження застосовувалися методи: симетричного, блочного, асиметричного, шифрування, криптографія на еліптичних кривих, криптографічна хеш-функція, аналіз технології та сам принцип роботи Blockchain, концепції створення криптовалюти, методи об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів. Покращена технологія обміну криптовалютою, яка відрізняється від існуючих застосувань блокчейн-рішень, що дозволило підвищити ефективність роботи обмінника криптовалютою, а саме швидкодію і безпеку.

Практичне значення одержаних результатів полягає в наступному:

- розроблено модель взаємодії смарт-контрактів з web-сервісом обмінника криптовалютою;

- створено програмне забезпечення обмінника криптовалютою.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології підвищення ефективності обмінника криптовалютою.

Особистий внесок здобувача. Усі результати, що наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, які написано у співавторстві, здобувачу належать: гнучке налаштування параметрів роботи генетичного алгоритма, що дозволяє пришвидшити процес обміну криптовалютою [1].

Апробація роботи. Результати роботи були апробовані на LI науково-технічній конференції підрозділів Вінницького національного технічного університету НТКП ВНТУ (2023) [1].

Публікації. За результатами бакалаврської дипломної роботи опублікована теза доповіді на конференції «LI Науково-технічна конференція

підрозділів Вінницького національного технічного університету (НТКП ВНТУ) (факультет інтелектуальних інформаційних технологій та автоматизації (2023)» (м. Вінниця, Україна) [1].

1 АНАЛІЗ ТА ВИБІР ВІДОМИХ РІШЕНЬ В ГАЛУЗІ ОБМІНУ КРИПТОВАЛЮТ

1.1 Обґрунтування доцільності створення крипто-обмінника

Обмінник як рекламний інструмент. Сам факт запуску обмінника є відмінним інфоприводом. Основним плюсом використання легального обмінника є той факт, що кінцевий клієнт може обходити обов'язок мати облікові записи в платіжних системах та на біржах. Він може уникнути деталей валютного законодавства та уникнути боротьби з банківськими установами. Крім того, завжди є наявними документи, що підтверджують операції, та можливість легалізувати дохід, отриманий в результаті продажу криптовалюти.[1]

Обмінник може функціонувати як канал для привернення аудиторії та здійснення конверсії на інші платформи. З певними умовами, з'являється можливість законно рекламувати свій сервіс на Facebook та Google.

Обмінник може бути організованим як приватний клуб, ініційований одним інвестором або групою інвесторів. Цей обмінник має на меті задовольнити потреби обмеженого кола осіб і дозволяє частково або повністю легалізувати доходи. Тобто, клієнт може отримати необхідні підтверджуючі документи, подати декларацію 3-ПДФО та сплатити податок.[2]

Ця схема надає повний контроль над компаніями, активами, рахунками та операційними ланцюжками. Ризики, пов'язані з людським фактором, можуть бути мінімізовані та керовані.

Завдяки зарубіжним компаніям, купівля-продаж криптовалют може здійснюватися за вигідними курсами безпосередньо на підключених біржах, де активують корпоративні аккаунти з розширеними лімітами угод.

Коли фізична особа продає криптовалюту іноземній компанії, вона отримує договір позики, за яким ця компанія стає її боржником за передані в позику криптовалюти (криптоактиви) з умовою погашення боргу у фіатних грошах за бажанням.

Вексель чи облігація не підходять, оскільки згідно з українським законодавством, вексель може бути виражений тільки в документарній, тобто паперовій формі, і для випуску облігацій потрібна ліцензія. Далі, фізична особа передає свої права вимоги згідно з договором позики українській компанії за допомогою договору цесії та отримує суму в національній валюті за курсом, наприклад, Національного банку України на момент виплати.

З метою фінансування своєї діяльності, юридична особа з Росії пред'являє іноземній компанії договір позики для відшкодування, і ця іноземна компанія погашає цю позику.[3]

З метою фінансування своєї діяльності, російська юридична особа пред'являє іноземній компанії договір позики для відшкодування, і ця іноземна компанія погашає цю позику. Насправді, це стандартна практика в сфері міжнародних фінансових операцій, яка дозволяє підприємствам отримувати фінансування від закордонних партнерів без необхідності здійснювати складні криптовалютні операції в Україні.

Такий підхід може бути особливо корисним в умовах регуляторного середовища, де обмеження на криптовалютні операції чи незрозуміння щодо їх обробки можуть ускладнити бізнес-процеси. Однак, важливо враховувати ризики та податкові обов'язки при використанні таких схем, і бути уважними до законодавства країн, де здійснюються фінансові операції.

Ця схема також дозволяє уникнути ризиків, пов'язаних із змінами курсів криптовалют, оскільки погашення позики відбувається у стабільній іноземній валюті. Необхідно бути свідомим, що цей підхід може бути обрано не тільки в корпоративних угодах, але і в особистих фінансових операціях, де фізичні особи також можуть використовувати іноземні компанії для отримання криптовалют.

Фактично, українське законодавство не регулює прямо використання іноземних юридичних осіб в фінансових операціях, і це створює можливість для реалізації подібних схем в певних умовах. [4]

Вибір сgruTo-friendly юрисдикції для зарубіжної компанії

Розглядаючи різні юрисдикції для створення та керування юридичною особою, важливо враховувати різні аспекти. Дійсно, Естонія відома своєю цифровою економікою та сприяльним законодавством для стартапів і бізнесу загалом. Однак, як і в будь-якій іншій юрисдикції, є свої труднощі, наприклад, у взаємодії з місцевими банками.

Мальта та Гібралтар також можуть бути привабливими варіантами, але вони можуть виявитися вартими в зв'язку із високими вартостями ліцензій та можливими обмеженнями в сприйнятті юрисдикцій як офшорів.

Залежно від конкретних потреб та можливостей бюджету, вибір юрисдикції може відрізнятись. Важливо звернути увагу на правила та закони кожної юрисдикції, а також консультуватися з фахівцями, які розуміють подробиці кожної із них. Інструменти безпеки.

При створенні криптовалютного обмінника, подібного до Paxful, забезпечення максимальної безпеки є важливою задачею, оскільки ваш сайт буде операційним з великими сумами грошей і особистими даними користувачів. Ось кілька ключових заходів безпеки, які варто розглянути:

1. Двофакторна автентифікація (2FA): Введення 2FA є стандартним заходом безпеки для криптовалютних обмінників. Ви можете використовувати електронну пошту, SMS-повідомлення або спеціалізовані додатки, такі як Google Authenticator, для забезпечення додаткового рівня автентифікації для користувачів.

2. Захист даних і систем від злому: Розгляньте можливість використання різних технік для забезпечення безпеки вашого обмінника. Це може включати розміщення сайту на різних фізичних серверах, використання захисту від хрестових сайтових сценаріїв (XSS) та міжсайтових подальших запитів (CSRF), а також фільтрацію вхідних даних для запобігання SQL-ін'єкціям.

3. Додаткові заходи безпеки: Додайте можливість холдування коштів, підтвердження операцій за допомогою пін-кодів і ведення журналу IP-

адрес користувачів. Ці додаткові заходи можуть допомогти виявити та запобігти несанкціонованим доступам та обману.

4.Дотримання GDPR: Якщо вашим сервісом користуватимуться громадяни Європейського Союзу, ви повинні забезпечити дотримання Загального регламенту з питань захисту даних (GDPR). Це включає в себе правила щодо обробки та зберігання особистих даних користувачів, які вам необхідно буде дотримувати.

Загальною метою є створення безпечного та надійного сервісу для користувачів та забезпечення їхніх фінансових та особистих даних. Важливо продумати цільову аудиторію та специфічні вимоги щодо безпеки для вашого обмінника, і, можливо, звернутися до фахівців у галузі кібербезпеки для отримання консультації та аудиту системи безпеки.

1.2 Порівняльний аналіз відомих крипто обмінників

Криптовалютний обмінник - це онлайн-платформа для обміну одних криптовалют на інші або фіатні гроші. Вони надають користувачам можливість купувати, продавати і обмінювати криптовалютні активи. Ці обмінники можуть працювати на різних принципах, включаючи фіксовані та змінні курси обміну.

На відміну від криптовалютних бірж, де користувачі можуть створювати ордери і брати участь в торгівлі на відкритих ринках, обмінники часто пропонують більш простий і швидкий спосіб обміну криптовалют. Користувачі можуть вказати суму та тип активу, який вони бажають обміняти, і отримати відповідний обсяг іншої криптовалюти або фіатних коштів.

Біржі криптовалют, натомість, пропонують більше розширені можливості, такі як маржинальна торгівля, створення різних типів ордерів (наприклад, лімітні та ринкові), глибина ринку та інші інструменти для активного трейду.

Обидва типи платформ мають свої переваги та недоліки, і вибір між ними залежить від індивідуальних потреб та цілей користувача.

Обираючи криптовалютний обмінник, користувачі мають бути дуже уважними та обдуманими. Правильний вибір майданчика може вплинути на безпеку ваших активів і операцій, комісії та вартість обміну, доступ до різних криптовалют та послуг, а також на загальний досвід користувача. Ось деякі фактори, які варто враховувати при виборі криптовалютного обмінника:

- **Безпека:** Перш за все, переконайтеся, що обмінник має високий рівень безпеки. Пошукайте інформацію про заходи безпеки, такі як двофакторна аутентифікація, захист від злому, засоби збереження активів та інші.
- **Комісії і обмінні курси:** Порівнюйте комісії та обмінні курси різних обмінників. Деякі майданчики можуть бути вигіднішими для конкретних операцій або валют.
- **Можливість обміну:** Впевніться, що обмінник підтримує ті криптовалюти, які вам потрібні для обміну.
- **Обсяг торгів:** Високий обсяг торгів на майданчику може бути ознакою ліквідності та надійності. Користувачі часто віддають перевагу обмінникам з великим обсягом торгів.
- **Додаткові функції:** Різні обмінники можуть пропонувати різні додаткові функції, такі як маржинальна торгівля, ордери з різними типами, API для трейдерів та інші.
- **Репутація та відгуки користувачів:** Пошукайте відгуки та рейтинги обмінників в Інтернеті, щоб отримати ідею про їхню репутацію та досвід інших користувачів.

Загалом, правильний вибір криптовалютного обмінника залежить від вашої конкретної ситуації та потреб.

Під час роботи з криптообмінником потрібно розуміти, що трейдер передає гроші на зберігання іншій стороні. Тому вибір надійної платформи, яка забезпечить захист коштів, важливий на протязі користування ними на даному

майданчику. Крім цього, важливо мати на увазі, що хороша біржа повинна надавати можливість обміну різних криптоактивів.

Розглянемо ключові параметри, які допомагають оцінити зручність та надійність роботи криптовалютної біржі:

- Кількість торгових пар. Ця характеристика вказує, які криптовалюти (і фіатні валюти) доступні для купівлі, продажу та обміну на конкретному майданчику. Великий вибір торгових пар дозволяє уникнути додаткових проміжних операцій при обміні активів, що сприяє зменшенню витрат на комісії. Більше 13 000 різних криптовалют і токенів доступні на криптобіржі Binance, що робить її однією з найбільших платформ у світі.
- Тарифи та комісійні внески. Розмір комісій, а також мінімальні суми для внесення та виведення коштів є важливими при виборі біржі. Занадто високі комісії та обмеження на виведення можуть бути не вигідними для менших трейдерів. Деякі біржі можуть надавати знижки на комісії для великих гравців при досягненні певних обсягів торгів. Криптобіржа Coinbase має понад 73 мільйони користувачів у більш ніж 100 країнах, роблячи її однією з найпопулярніших бірж в світі.
- Способи виведення фіатних коштів. Різноманітні методи виведення фіатної валюти сприяють популярності майданчика. Прямий вивід на банківську карту, банківські перекази та вивід на популярні платіжні сервіси дозволяють трейдерам легко переміщувати свої кошти за необхідності.
- Захист облікового запису. Важливо, щоб біржа надавала різні методи захисту облікового запису, такі як двофакторна аутентифікація, смс-повідомлення та підтвердження по електронній пошті під час проведення операцій. Якісна служба підтримки особливо важлива для новачків, які можуть потребувати допомоги у вирішенні проблем. Bittrex, наприклад,

використовує холодне зберігання для 90% своїх активів, що робить його однією з найбільш надійних бірж.

- Аналітичні інструменти. Наявність різноманітних інструментів для аналізу ринку та прогнозу зміни вартості активів корисна для професійних трейдерів. Широкий набір інструментів допомагає точніше прогнозувати рухи на ринку. Для майнерів і дрібних трейдерів це може бути менш важливим, але все одно корисним. Криптобіржа Bitfinex надає доступ до різноманітних аналітичних інструментів, включаючи графіки, статистику торгів та засоби для аналізу ринку.

Репутація обмінника є, очевидно, найсуттєвішим чинником при виборі криптообмінника. Навіть у випадку, коли платформа пропонує найбільш вигідні умови для користувачів, якщо виникають труднощі з виведенням коштів або забезпеченням їх безпеки, вибір цієї біржі може стати невдалим, що бачимо в таблиці 1.1.

Дуже важливо ознайомитися з відгуками і враженнями пересічних користувачів торгової платформи, оскільки вони можуть розкрити потенційні проблеми, які можуть виникнути при взаємодії з певним обмінником. Також ці відгуки допоможуть зрозуміти, як адміністрація реагує на такі ситуації і чи варто розглядати їх як надійного кандидата для обміну криптовалюти.

Таблиця 1.1 - Недоліки та переваги аналогів сайту обміну криптовалюти

	Веб-сайт “ 24Bestex ”	Веб-сайт «Privat.cash»	Наш обмінник
Використання скрипта обміну	-	-	+
Адмін панель	-	+	+
Сучасний дизайн	+	-	+

Продовження таблиці 1.1

Можливість зв'язатись з підтримкою	+	-	+
Можливість залишити відгук	-	-	+
Можливість поділитись постом	+	+	+
Мультимовність	+	+	-
Кількість валют які міняють	+	-	+

Під час створення заявки на веб-сайті "24BesTex" виникла неприємність, оскільки адресу гаманця випадково було видалено останнім символом. Ця ситуація може трапитися, якщо клієнт не повністю виділив і не скопіював адресу. Після подання заявки були помічені дві особливості:

1. На сторінці оплати не було відображено адресу нашого біткоїн-гаманця, що ускладнило перевірку наявності помилок. Однак цю адресу можна було перевірити в електронному листі, який обмінник надсилає після підтвердження заявки (див. рисунок 1.3).
2. "24BesTex" запросив фотографію банківської картки на тлі сторінки обміну для верифікації користувача. Ця процедура є одноразовою, і після неї обмінник пам'ятає картку та дозволяє працювати з нею без подальшої верифікації.

Після повторного заповнення заявки клієнти успішно завершили операцію обміну. "24BesTex" відправив біткоіни через 5 хвилин після оплати заявки, при цьому різниця з курсом на "INSTACHANGE" становила 3,27%.

Спочатку на веб-сайті "24BesTex" заявка на обмін "PrivaT.cash" здається такою самою, як і заявки на інших обмінниках, рисунок 1.3. Користувач вводить своє ПІБ, дані картки та номер телефону. Однак далі обмінник запитує вказати хеш транзакції (Txid) для підтвердження оплати.

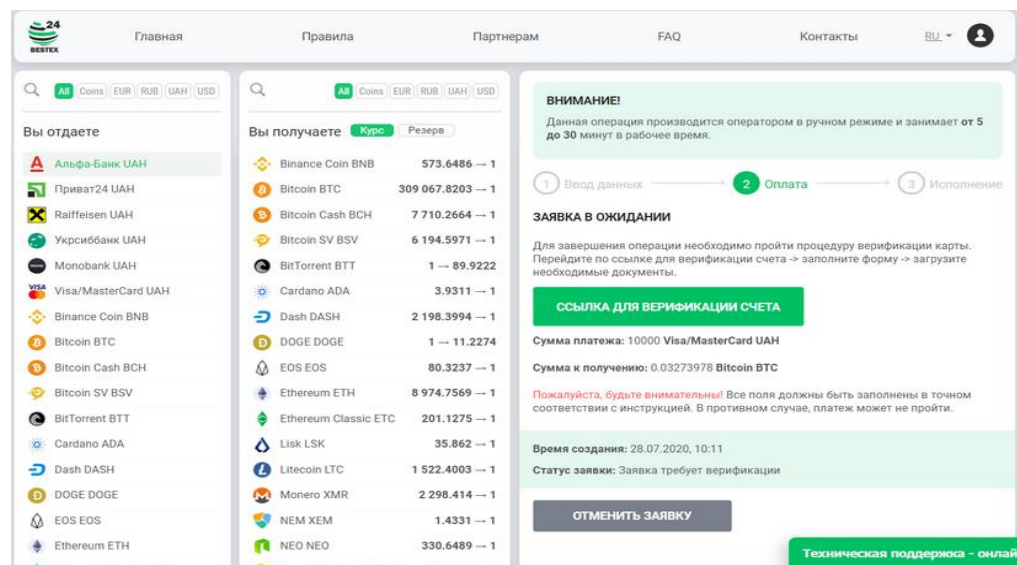


Рисунок 1.3 - Оплата заявки на 24BesTex: не можна перевірити реквізити отримувача

На момент обміну, мемпул біткоіну був перевантажений, і наша транзакція застрягла. Ми звернулися до технічної підтримки, запитавши, чи можна прискорити операцію. Консультант повідомив, що це неможливо, і ми вирішили дочекатися підтвердження транзакції, що сталося о 4-й ранку.

У математичній теорії механізмів запитів у криптовалютних інформаційних системах розроблено ряд механізмів, які вирішують ці проблеми. Одним з таких механізмів є механізм аукціону. Аукціон - це спосіб розподілу ресурсів на основі конкурентної конкуренції. Іншою важливою проблемою є проблема забезпечення справедливості. Ця

проблема полягає в тому, щоб розподілити ресурси таким чином, щоб агенти отримували те, що вони заслуговують на основі своїх внесків.

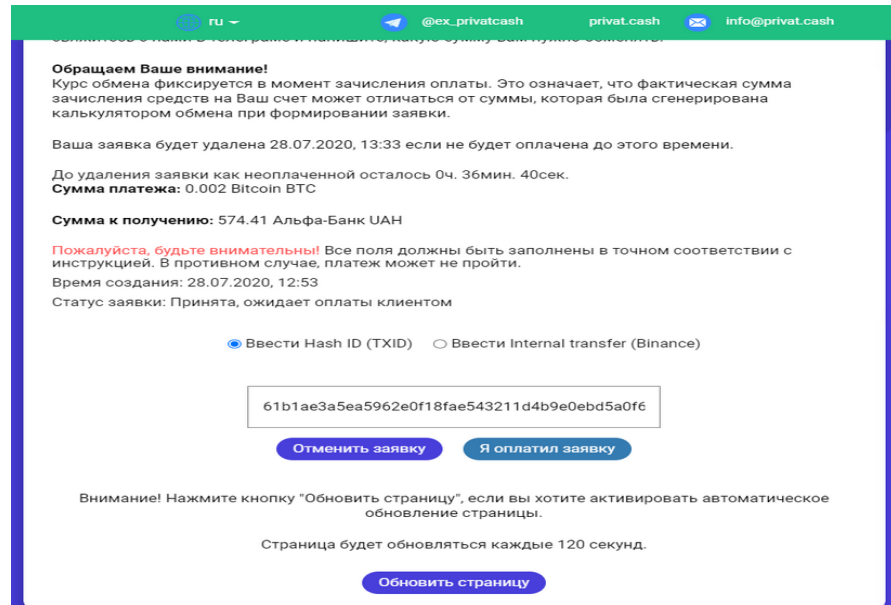


Рисунок 1.4 Відправляємо біткоїни PrivaT.cash: вказуємо хеш транзакції та підтверджуємо оплату

"PrivaT.cash" відправив гроші через 1 хвилину після підтвердження оплати, і це було одним з найшвидших обмінників за всім оглядом. Різниця в курсі з "INSTACHANGE" становила 3,4%, бачимо на рисунку 1.4.

1.3 Розробка загальної дизайнерської концепції веб-ресурсу

Концепція сайту є схемою або планом, який визначає шлях розвитку Інтернет-проекту. Цей документ включає в себе ретельний аналіз ринку, конкурентів, цільової аудиторії, товарних пропозицій і інші дослідження. На основі зібраних даних створюється технічна концепція сайту, що допомагає сформулювати чітке бачення та образ проекту. Коли інтернет-ресурс розробляється з урахуванням всіх цих аспектів, він повністю спрямований на свою цільову аудиторію. [5]

Розробка концепції є важливим етапом в процесі створення веб-сайту. Цей документ допомагає уточнити напрямок і цілі проекту, щоб підготувати технічне завдання для дизайну та розробки. Фахівці, які працюють над проектом, можуть чітко розуміти свої завдання завдяки концепції, оскільки в ній описано, яким повинен бути сайт і які функції він має виконувати.

Концепція сайту складається з кількох етапів, і її основна мета - сформулювати загальне бачення проекту. Важливо, щоб реальні результати на фінальному етапі відповідали задуманій концепції. Маркетингова концепція сайту передбачає аналіз і вибір необхідних інструментів для досягнення цілей проекту. Що більше деталей містить концепція, тим менше буде корекцій і недорозумінь під час розробки.

Стиль веб-дизайну грає важливу роль у сприйнятті веб-сайту користувачами. Перші секунди візиту на сайт вирішальні, і користувач формує свою думку про нього швидко. Саме тут важливо, щоб дизайн був привабливим, професійним і відповідав аудиторії.

Візуальна складова, кольори, шрифти, макет та інші дизайнерські елементи мають велике значення для створення першого враження. Якщо сайт виглядає сучасно, організовано та зручно, користувачі будуть більш схильні залишитися на ньому і дізнатися більше про компанію чи бренд.

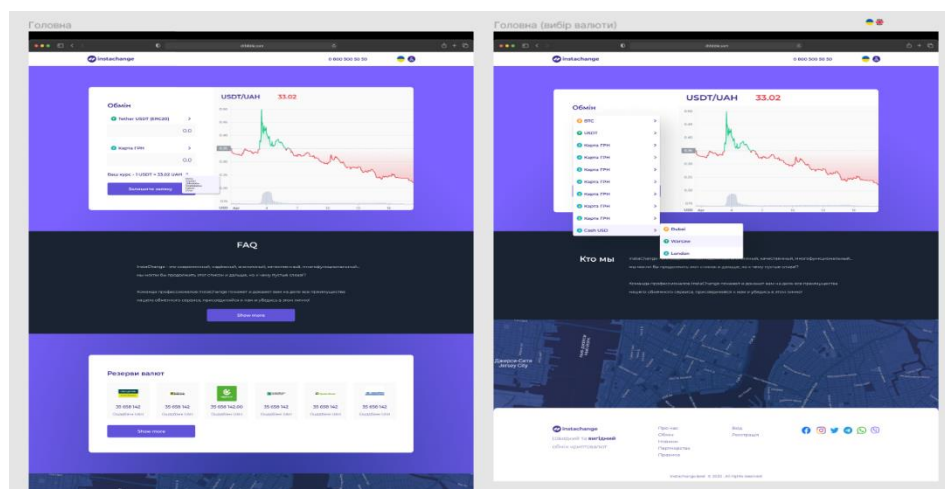


Рисунок 1.5 Початковий макет сайту INSTACHANGE

Зручність використання також важлива, оскільки користувачі не хочуть витрачати час на пошук інформації чи намагання зрозуміти, як користуватися сайтом. Добре спроектований інтерфейс зробить навігацію легшою та ефективнішою.

Таким чином, стиль веб-дизайну має велике вплив на сприйняття та взаємодію з цільовою аудиторією, і він важливий для створення позитивного першого враження та залучення користувачів, рисунок 1.5.

1.4 Висновок до розділу 1

За аналізом представленої інформації можна зробити висновок, що популярність криптообмінників серед користувачів залежить від сукупності різних факторів. Проте виділяються деякі ключові аспекти, які приваблюють максимальну кількість криптоутримувачів.

Ці аспекти включають велику кількість торгових пар, що дозволяє користувачам здійснювати різноманітні операції з криптовалютами, високий рівень анонімності, який важливий для багатьох учасників ринку, і низькі комісії, що роблять обмін криптовалют більш вигідним.

Важливо зазначити, що всі розглянуті обмінники проявляють відповідальність щодо безпеки даних та коштів на аккаунтах користувачів, що є надзвичайно важливим аспектом в обміні криптовалютами.

Отже, створення криптовалютного обмінника- це складний процес, який має різні переваги та виклики. Вам знадобиться надійна технологічна платформа для обробки криптотранзакцій. Ви також повинні забезпечити, щоб ваша платформа була масштабованою, щоб відповідати зростаючому попиту. Зростаючий інтерес до криптовалют, потреба в диверсифікації та потенціал для отримання прибутку є вагомими причинами для розгляду такої ініціативи. Однак важливо підходити до цієї справи з чітким розумінням регуляторного ландшафту і зобов'язанням забезпечити безпечну, зручну для користувачів і відповідну вимогам платформу. Незважаючи на значні ризики, добре

організована криптовалютна біржа може суттєво вплинути на фінансову індустрію, що розвивається, і стати цінним ресурсом для інвесторів по всьому світу, успіх криптообмінника у значній мірі залежить від його здатності задовольнити потреби користувачів і надати їм необхідну функціональність та безпеку при обміні криптовалютами.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОБОТИ ОБМІННИКА КРИПТОВАЛЮТ

2.1 Математична теорія механізмів запитів у криптовалютних інформаційних системах

Криптовалюта використовує однорангову децентралізовану систему для проведення своїх транзакцій. Оскільки весь процес відбувається онлайн, існують обриви та загрози щодо можливого порушення та зламу транзакцій. Тому ми розповімо, як криптовалюта використовує криптографію для забезпечення безпеки своїх транзакцій [13].

Криптографія - це метод використання передових математичних принципів для зберігання і передачі даних у певному форматі так, що лише ті, для кого вони призначені, можуть читати та обробляти інформацію. Криптографія використовується вже тисячі років для таємного спілкування. Найперше використання криптографії було зафіксовано в гробниці, знайдений в Стародавньому Єгипті приблизно 1900 року до нашої ери.

Шифрування - один з найважливіших інструментів, який використовується в криптографії. Це метод, за допомогою якого повідомлення перетворюється в закодований набір символів. Лише відправник і отримувач знають, що приховано в листі.

У сучасних технологіях широко використовуються три форми шифрування:

- Симетрична криптографія
- Асиметрична криптографія
- Хешування

Симетричне шифрування - це спосіб шифрування, в якому для шифрування і розшифрування використовується один і той самий криптографічний ключ. Ключ алгоритму має бути збереженим в секреті обома сторонами. До виникнення асиметричного шифрування єдиним існуючим способом було симетричне шифрування.

Алгоритми шифрування і розшифрування даних важливе місце в застосуванні в комп'ютерній техніці в системах захисту конфіденційної і комерційної інформації від некоректного використання сторонніми особами. Головним принципом у них є умова, що особа, яка приймає зашифрований текст, заздалегідь знає алгоритм шифрування, а також ключ для повідомлення, без якого ці дані є всього лише набором символів, які не мають жодного змістовного значення.

Симетричні криптоалгоритми виконують перетворення невеликих (наприклад, 1 біт або 32-128 біт) блоків даних в залежності від ключа таким чином, що отримати оригінальний текст можна лише в тому випадку, коли ви знаєте цей секретний ключ.

Концепція дуже проста:

- Є повідомлення А, яке ви бажаєте відправити своєму другові.
- Ви шифруєте дане повідомлення за допомогою певного ключа і отримуєте зашифрований текст В.
- Ваш друг отримує цей зашифрований текст В.
- Потім він розшифровує закодований вами текст за допомогою того ж ключа і в результаті отримує текст А.

Візуально це можна представити на рисунку 2.1.

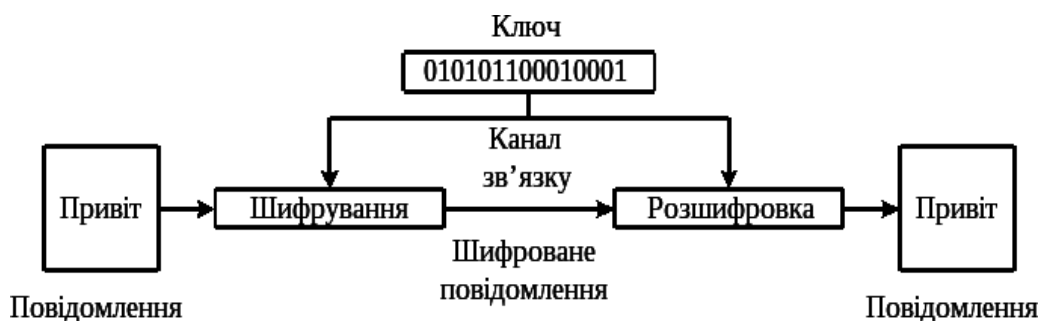


Рисунок 2.1 – Схема роботи симетричного шифрування

Математична теорія механізмів запитів є важливим інструментом для розробки ефективних, справедливих та стійких криптовалютних інформаційних систем.

Криптографічних алгоритмів існує безліч, загалом вони призначені для забезпечення безпеки інформації. Симетричні криптоалгоритми відносяться до криптоалгоритмів з ключем. Вони поділяються на два типи:

Потокові шифри - це бітова обробка інформації. Шифрування і розшифрування в такому випадку можуть відбуватися в будь-який момент часу, як тільки стає відомо, що потік передачі перервався, і також відновлюється при виявленні факту продовження передачі.

Блочні шифри - це перетворення блоку вхідної інформації фіксованої довжини, і отримують результат того ж обсягу. Використовуються різні принципи потокового шифрування, такі як AND, OR або XOR (виключає АБО).

Розглянемо приклад потокового шифрування:

Для цього шифрування потрібен ключ, який має таку ж кількість символів, що і повідомлення, і його потрібно використовувати тільки один раз (звідси назва "одноразовий").

Припустимо, що ми відправляємо повідомлення Бобу "СКОРО УВИДИМСЯ". Але ми не хочемо, щоб хтось перехоплював наше повідомлення. Тому ми з Бобом вирішили використовувати одноразовий блокнот, який виглядає наступним чином [14]: "Г Н П Б К У О Х З Ю Т Э М", відображено на рисунку 2.2 Як бачимо, запис має стільки символів, скільки і повідомлення.

Це дуже простий приклад застосування одноразового блокнота, щоб краще зрозуміти його принципи.

З'являється ще одна річ: кожна буква з алфавіту замінюється числовим еквівалентом. Візуально це можна представити на рисунку.

Однією з головних переваг обмінників є їх легкий доступ. Користувачі можуть торгувати криптовалютами з будь-якого місця світу, якщо вони мають

доступ до Інтернету. Це робить криптовалютний обмін доступним для широкого кола людей.

А	0	К	11	Х	22
Б	1	Л	12	Ц	23
В	2	М	13	Ч	24
Г	3	Н	14	Ш	25
Д	4	О	15	Щ	26
Е	5	П	16	Ъ	27
Ё	6	Р	17	Ы	28
Ж	7	С	18	Ь	29
З	8	Т	19	Э	30
И	9	У	20	Ю	31
Й	10	Ф	21	Я	32

Рисунок 2.2 – Приклад роботи блочного шифрування

Під час процесу шифрування використовується 6 одиниць інформації:

- Вихідне повідомлення (OM): "СКПРО UBIDIMСЯ," що є початковим повідомленням.
- Числове вихідне повідомлення (NOM), яке представляє цифровий еквівалент OM.
- Одноразовий блокнот (OTR).
- Новий одноразовий блокнот (NOTR).
- Числовий зашифрований текст (NCT), який є сумою числового вихідного повідомлення та одноразового блокнота, де сума не повинна перевищувати 33, враховуючи кількість букв у алфавіті.
- Зашифрований текст (CT), який є алфавітним еквівалентом пункту, вищезазначеного.

Щодо процесу дешифрування, він виконується за допомогою того ж ключа.

Отримувач повідомлення "ФШЮТЩЖРЮЛЖЯОЛ" повинен мати наступне:

- Зашифровану фразу.

- Загальний ключ.
- Таблицю з числовими еквівалентами.

До питання про те, як саме відбувається процес дешифрування, потрібно надати більше докладностей або вказівок.

Блоковий шифр має сталу довжину та вимагає розбиття повідомлення на блоки однакової довжини. Якщо розмір фрази менший, він доповнюється для заповнення блоку [14].

У випадку одноразового блокнота, довжина ключа відповідає довжині повідомлення, але в блоковому шифруванні ця властивість не обов'язкова. За допомогою одного ключа можна шифрувати довгий текст.

Існує дуже простий метод підстановки, відомий як шифр Атбаш, де алфавіт просто обертається: А -> Я і навпаки. Також відомий шифр Цезаря, де кожна буква отримує значення, що стоїть за нею (або розташована на X).

Наприклад, слово "ВАК" стає "ВЯЛ," зрушене на 1.

- А Б В Г Д Е Є Ж З Й К Л
- Б В Г Д Е Є Ж З Й К Л А

Вище наведені приклади є простими, зазвичай шифрування відбувається з великими фрагментами даних, рисунку 2.3.

Незважаючи на те, що симетрична криптографія має деякі недоліки (які ми обговоримо нижче), її найбільшою перевагою є те, що вона вимагає дуже великої кількості ресурсів. Все, що потрібно, це надати одержувачу один і той самий ключ для того, щоб метод працював.

Навіть зараз велика кількість ПО використовує цей метод в поєднанні з асиметричною криптографією для забезпечення швидких і ефективних служб шифрування та дешифрування.

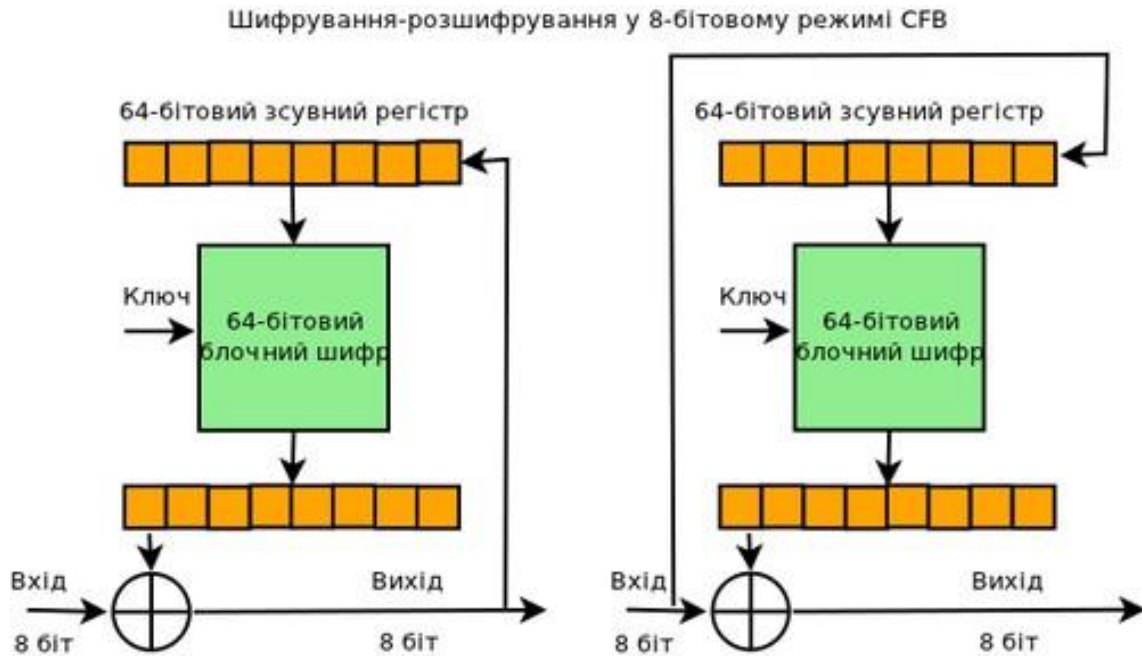


Рисунок 2.3 – Схема роботи блочного шифрування

Проте є декілька проблем симетричного шифрування:

1. Головна проблема полягає в тому, що шифрування і дешифрування виконуються за допомогою одного ключа. Якщо хто-небудь перехопить ключ, всі дані будуть компрометовані.
2. Немає масштабованості, тобто припустимо, що інформаційний центр відправляє дані за допомогою симетричної криптографії. Все добре, поки клієнтів всього 3-4. Але чим більше їх, тим складніше обробляти всі ключі. Проблеми з вразливістю симетричного ключа сприяли виникненню нового рішення в 1970-х роках.

2.2 Створення технології швидкого обміну криптовалюти

Аналіз потреб користувачів і цільової аудиторії. Першим кроком у створенні технології швидкого обміну криптовалюти є аналіз потреб користувачів та вимог цільової аудиторії. Цей аналіз включає в себе:

- Визначення очікувань користувачів щодо швидкості обміну криптовалюти.

- Вивчення популярних криптовалют, якими користувачі найчастіше обмінюються.
- Виявлення потреб у різних типах обмінів, таких як міжкриптовалютні обміни та обміни на фіатні валюти.

Вибір технологічної платформи платформи має вирішальне значення для забезпечення ефективності та безпеки процесу обміну. Вибір мов програмування та фреймворків для розробки рішення для обміну. Впровадження архітектури мікросервісів для підвищення масштабованості. Розгортання системи обробки транзакцій в режимі реального часу.

Розробка алгоритму здійснення обміну. Важливим аспектом є розробка алгоритму, що відповідає за виконання криптовалютних обмінів. Цей алгоритм повинен бути ефективним і швидким, враховувати вхідні та вихідні суми криптовалют, обрані користувачем, забезпечуючи при цьому безпечну обробку транзакцій.

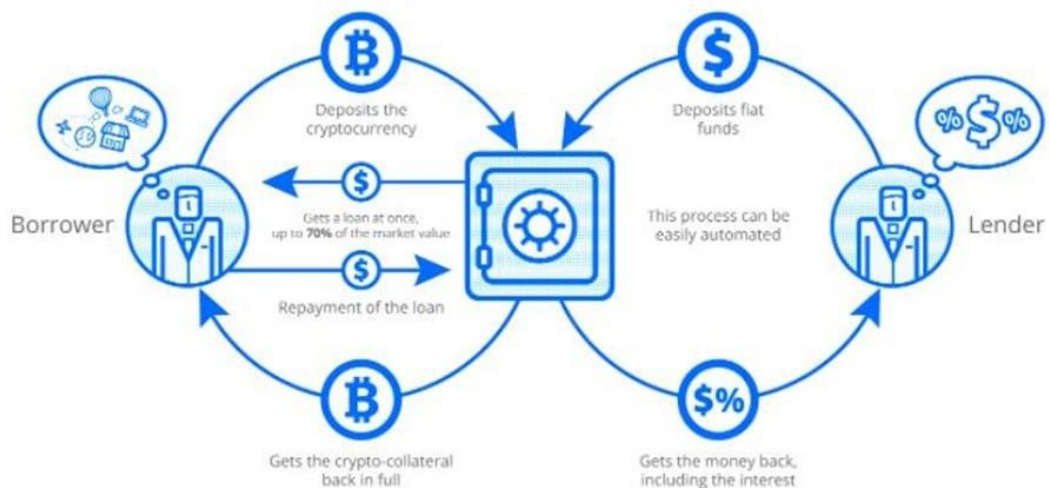


Рисунок 2.4 - Алгоритму здійснення обміну

Алгоритму здійснення обміну показаний на рисунку 2.4. Розробка алгоритмів обміну криптовалют є вирішальним етапом у створенні технології для швидкого і надійного обміну цифровими активами.

Розробляйте алгоритми управління ризиками, які дозволяють встановлювати ліміти на суми та обсяги торгівлі користувачів. Це допомагає запобігти потенційним фінансовим втратам.

Після розробки алгоритмів обміну необхідно провести ретельне тестування для виявлення помилок і недоліків. Постійно вдосконалюйте алгоритми на основі результатів тестування та відгуків користувачів.

2.3 Аналіз програмних технологій для взаємодії з Blockchain

Для розробки смарт-контрактів ми обираємо блокчейн Ethereum. Ethereum дозволяє виконувати програми в своєму надійному середовищі, що відрізняє його від біткоїну, який обмежений управлінням криптовалютою. В Ethereum для цієї мети використовується віртуальна машина під назвою Ethereum Virtual Machine (EVM).

Ethereum Virtual Machine (EVM) - це віртуальна машина, що складається з безлічі комп'ютерів, на кожному з яких зберігаються всі транзакції локально і виконуються з відносною синхронністю [7]. До цього моменту EVM визнана єдиною, захищеною та незалежною віртуальною машиною, яка надає доступ до ефективної функціональності для автоматизованих систем грошових переказів. Хоча кожен може використовувати цю глобальну віртуальну машину, ніхто не може створити фальшиві гроші або здійснити фінансові операції без дозволу.

Частина протоколу, яка виконує обробку транзакцій в операційній системі Ethereum, називається Ethereum Virtual Machine (EVM). EVM є машиною Тьюринга. Єдине відмінність EVM від звичайної машини Тьюринга полягає в тому, що для роботи першої потрібно віртуальне "пальне". Таким чином, всі обчислення, які можуть бути виконані в EVM, обмежені кількістю

віртуального "пального", що циркулює в ній. Крім того, EVM притаманні всі особливості стекової архітектури. Крім того, EVM має всі особливості стекової архітектури. Стекова машина - це комп'ютер, де використовується алгоритм LIFO (Last-In-First-Out). Розмір будь-якого елемента стека в EVM дорівнює 256 бітам, а максимальний розмір стека досягає 1024 біт. Для EVM передбачено певний обсяг пам'яті, який не є постійним. У цьому обсязі елементи зберігаються у вигляді масивів байтів, доступ до яких здійснюється через слова. Крім того, EVM має певну область зберігання, яка не змінюється і є частиною стану системи. У EVM програмний код зберігається в іншій віртуальній ROM (Read-Only Memory), доступ до якої можна отримати лише за допомогою певних інструкцій. З цього погляду EVM відрізняється від звичайної архітектури, де програмний код зберігається в пам'яті комп'ютера. З цієї точки зору така EVM відрізняється від типової архітектури фон Неймана, в якій програмний код зберігається в пам'яті комп'ютера (рис 2.5).

Для комп'ютерної системи, також передбачена своя спеціальна мова - байт-код Ефіріум Virtual Machine (EVM). При створенні смарт-контракту для виконання в середовищі Ефіріума, зазвичай використовується високорівнева мова, така як Solidity. Після написання такого коду, його компілюють у байт-код EVM, щоб EVM могла розуміти виконавчі команди.

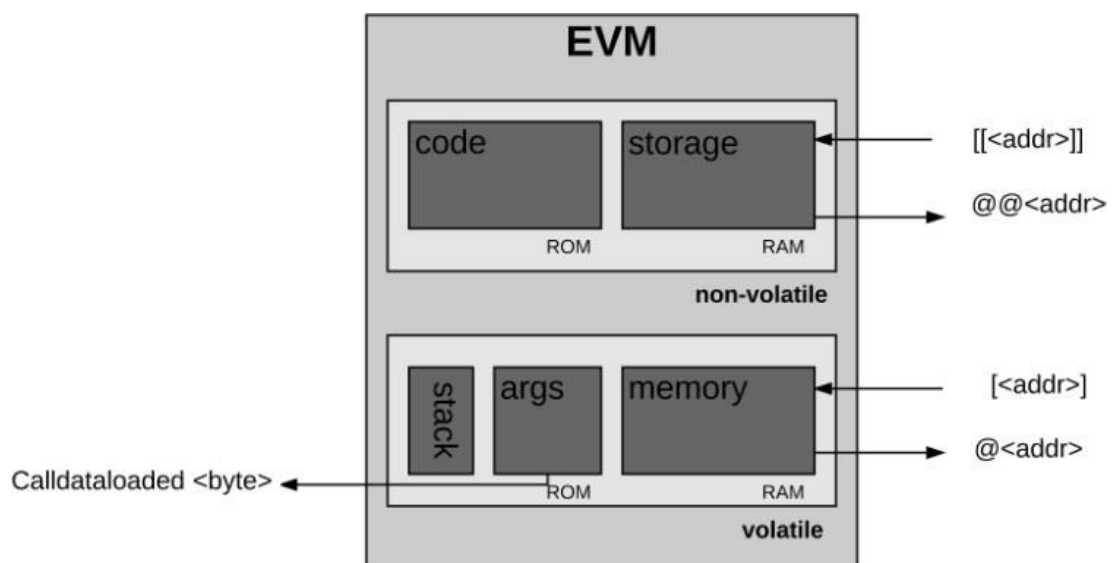


Рисунок 2.5 – Структура Ethereum Virtual Machine

Перед виконанням певних обчислень, процесор повинен переконатися в тому, що наявна наступна інформація: Інформація про достатній для виконання необхідної операції кількості пального:

- Адреса облікового запису, до якого належить код, який виконується;
- Адреса відправника транзакції – ініціатора виконання поточної операції;
- Адреса відправника транзакції – ініціатора виконання поточної операції;
- Інформація про необхідну для виконання транзакції кількість газу;
- Вхідні дані для виконання операції;
- Кількість Wei, яка буде відправлена на рахунок облікового запису в результаті виконання поточної операції;
- Інформація про виконуваний машинний код;
- Інформація про заголовок блоку для поточного блоку;
- Глибина виконання поточного повідомлення або створення контракту.

Перед фактичним початком виконання програми пам'ять системи є абсолютно порожньою, і лічильник команд дорівнює нулю. Потім ЕВМ починає рекурсивне виконання транзакції, обчислюючи стан системи та стан машини для кожного циклу. Стан системи - це глобальний стан Ефіріума, а стан машини включає в себе вміст пам'яті, лічильник команд, контент стеку та активну кількість слів у пам'яті. Стан машини включає в себе:

- лічильник команд;
- вміст пам'яті;
- активне кількість слів у пам'яті.
- контент стека;

2.4 Розробка UML діаграм

Абревіатура UML розшифровується як Unified Modeling Language, перекладається як «уніфікована мова моделювання». По суті, це мова моделювання, яка дозволяє створювати структури програмних систем.

UML складається із графічних позначень, діаграм, які допомагають створити дизайн програмних проєктів. За допомогою UML-діаграм проєктні групи комунікують між собою, складають та перевіряють архітектурний дизайн ПЗ.

Робота з UML-діаграмами - важлива частина проєкту, тому що на цьому етапі продумується його структура. Проєктування допомагає не заплутатися в коді, знизити кількість помилок і спростити роботу.

Діаграма прецедентів веб-сайту обмінника криптовалют зображена на рисунку 2.6

UML має єдиний синтаксис, тому є міжнародною мовою. Діаграми будуть зрозумілі будь-якій людині, знайомій із нею. Також варто зазначити, що UML використовується для розробки широкого спектру програм від інформаційних систем масштабу підприємства до розподілених веб-додатків.

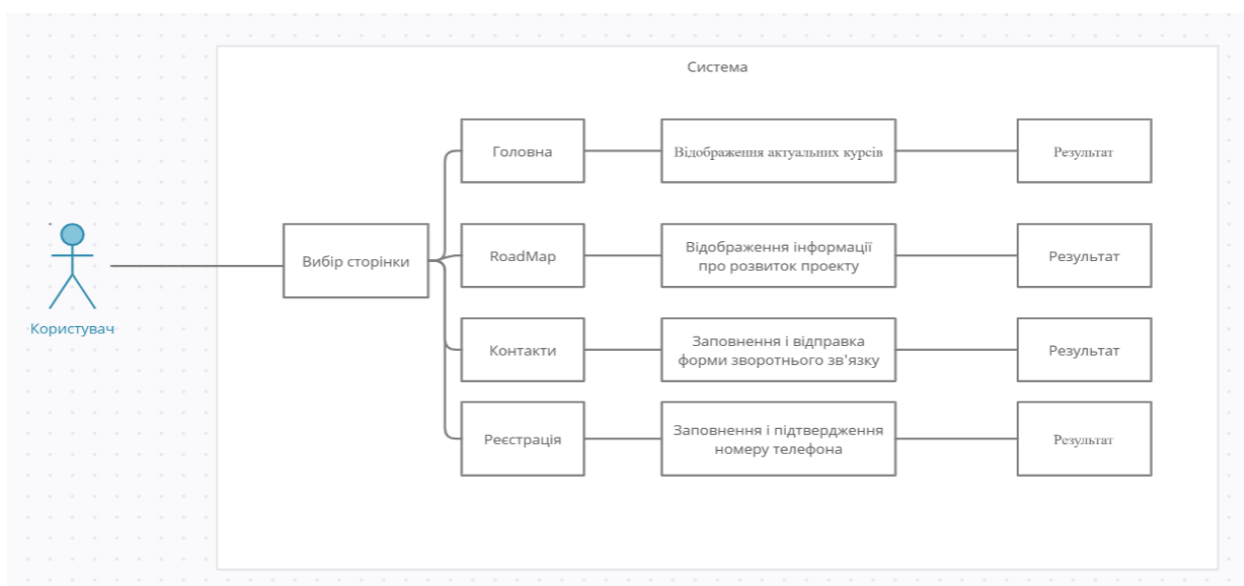


Рисунок 2.6- Діаграма прецедентів веб-сайту обмінника криптовалют

UML покликаний дати стандартну нотацію, використовувану усіма об'єктно-орієнтованими методами. Крім того, UML дозволяє вибрати та впроваджувати напрацювання нотацій-попередників. Оскільки UML створено широкого спектра програм, з його допомогою можна конструювати різні системи.

Діаграма станів - це діаграма станів з теорії автоматів із стандартизованими умовними позначеннями, яка може визначати безліч систем від комп'ютерних програм до бізнес-процесів. Діаграма станів блогу зображена на рисунку 2.7. Однією з основних переваг криптовалют є їхня децентралізація. Це означає, що криптовалюти не контролюються центральним банком або урядовою установою. Це може зробити їх більш стійкими до політичних потрясінь.

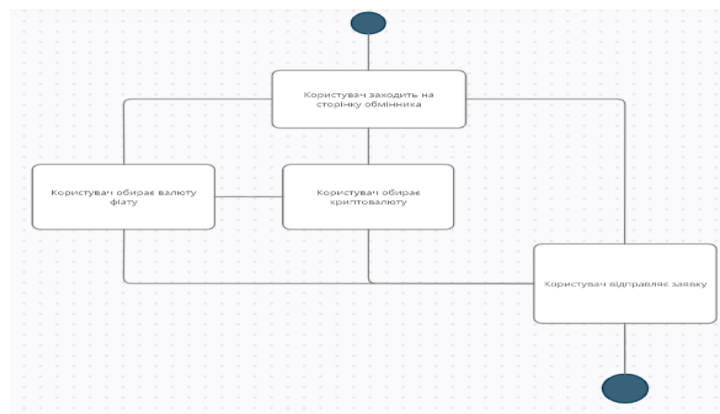


Рисунок 2.7 - Діаграма станів веб-сайту обмінника криптовалют

2.5 Декомпозиція сайту та дизайн окремих частин

Якщо вам вдалося зловити великого мамонта, який вищий за вас, добре годується і на перший погляд не виглядає наїстися, то вам потрібно розглянути подальші дії щодо розбору цієї досягнутої мети. Спершу розгляньте відокремлення лапок від тіла мамонта, а також відокремлення його шкіри від м'яса.

Так само, коли вам доручають хороший проект, вам потрібно поділити завдання на окремі частини та оцінити їх обсяги. Принцип декомпозиції допомагає систематизувати роботу, розкласти її на складові етапи, визначити резерви для важких завдань і планувати роботу над ними. Це дозволяє краще розуміти обсяг проекту і визначати завдання, які можуть бути найскладнішими або потребувати особливої уваги.

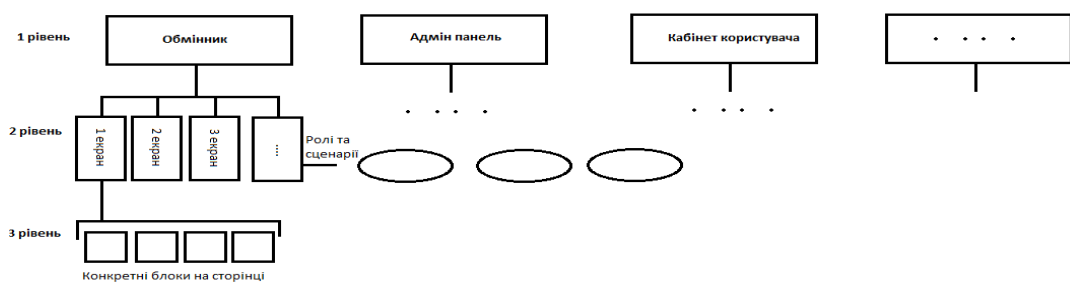


Рисунок 2.8 - Рівні декомпозиції

Здається, що розкладання проекту на складові частини може бути дуже простим: поділяємо його на великі блоки, які далі розбиваються на менші частини, і ця розкладка триває, поки ми не досягаємо деталей. Цей процес поділяється на кілька рівнів:

1 рівень: Великі блоки чи компоненти

На цьому рівні ми виділяємо основні блоки робіт, які можуть бути взаємопов'язані, але їх можна виконувати ізольовано один від одного.

2 рівень: Сторінки сайту або екрани мобільного додатка

На цьому рівні ми розглядаємо розбиття на сторінки або екрани. Наприклад, для блоку "Обмінник" ми можемо розбити його на окремі екрани. Щоб переконатися, що ми врахували всі можливі екрани, ми розглядаємо сценарії використання, що дає нам уявлення про те, як користувачі будуть взаємодіяти з додатком або веб-сайтом.

3 рівень: Зміст екранів

На цьому рівні ми розглядаємо зміст окремих екранів, де можуть бути вкладені блоки або ділянки, або навіть окремі екрани. Наприклад, сторінка

кошика або оформлення замовлення може містити блок товарів (з діями, такими як додавання, видалення, очищення) і інші блоки, такі як доставка, оплата, авторизація, бонусна система і так далі. Іноді ці блоки також можуть бути розбиті на окремі екрани кроків, в залежності від аналізу та прийнятих рішень.

Завдання менеджера, коли він отримує такий екран, полягає в тому, щоб розглянути, з чого цей екран складається. Іноді це може бути легкою розкладкою на блоки, а іноді це вимагає більшої уваги і творчості, особливо у випадку складних елементів, таких як калькулятори, де не завжди очевидно, як розбити процес на кроки.

2.6 Розробка архітектури інформаційної технології підвищення ефективності роботи обмінника криптовалют

Архітектура веб-сайту описує структуру сторінок та програмний компонент проекту. Вона дозволяє передбачити усі розділи ще до початку розробки. У цей склад входять навігація між сторінками, система посилань, 'хлібні крихти', категорії сторінок, файли картки сайту, матеріали та інші фрагменти, які формують структуру веб-ресурсу.

Якщо у вашому веб-сайті багато сторінок, ручне визначення кількості кліків для доступу до кожної сторінки може бути проблематичним завданням. Автори пропонують використовувати SEO-аудитори, такі як Screaming Frog, RyTe, Moz, SEMrush та інші, які допомагають визначити рівень внутрішньої структури сайту. Крім того, OnCrawl надає корисний звіт щодо внутрішніх посилань.

Більшість таких інструментів аналізує структуру веб-сайту, починаючи з головної сторінки. Іноді це може ускладнити розуміння загальної конфігурації сайту. Тому автори рекомендують використовувати

інструменти візуалізації, такі як Website Auditor та Sitelab, для кращого відображення структури веб-ресурсу та виявлення окремих сторінок для подальшого з'єднання з ними.

На невеликих особистих блогах або веб-сайтах місцевих піцерій, структура зазвичай є простою. Проте великі портали, такі як ті, що містять 250 тисяч сторінок, потребують організації та особливої уваги.

Розташування важливих сторінок ближче до головної сторінки допомагає використовувати так звану плоску структуру сайту рисунок 2.9. У такій структурі переміщення від головної сторінки до будь-якої внутрішньої сторінки вимагає мінімальної кількості кліків, і при цьому всі сторінки взаємопов'язані між собою.

Обмінники дозволяють торгувати різними криптовалютами, що дозволяє користувачам максимально використовувати можливості ринку.

Варто зазначити, що, незважаючи на ці переваги, існують ризики, пов'язані з торгівлею криптовалютами, тому перед початком взаємодії з обмінниками важливо добре розібратися в їхніх правилах та політиках безпеки.

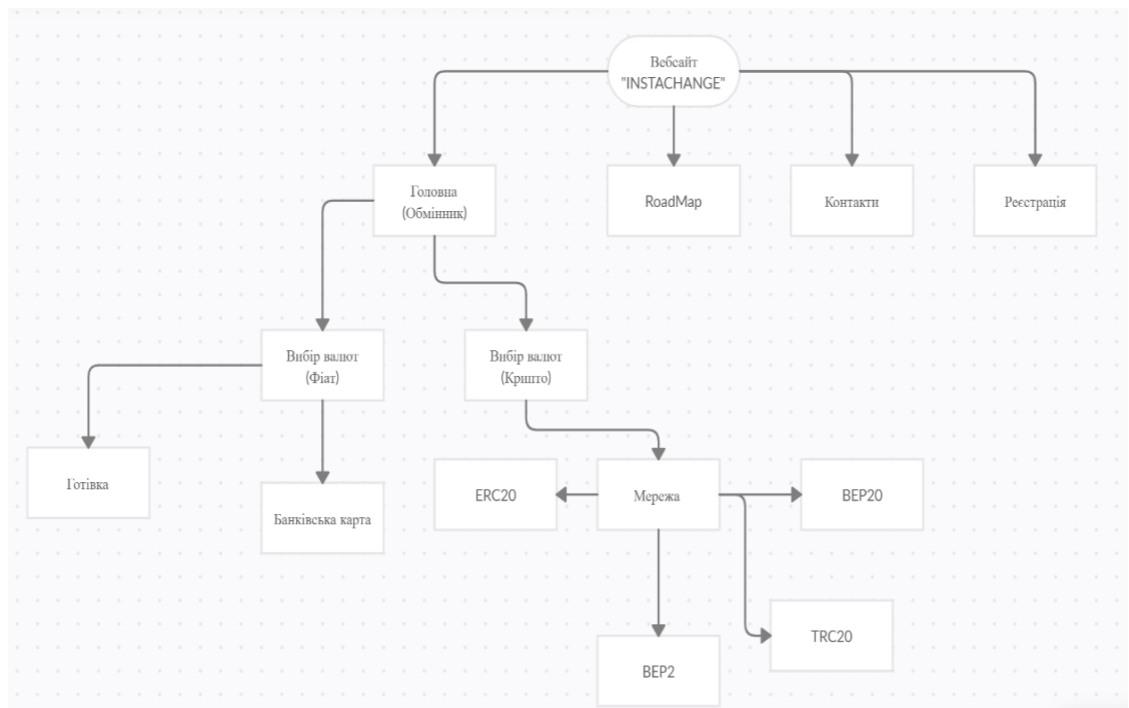


Рисунок 2.9 - Структура сайту

Деякі обмінники надають інноваційні функції, такі як маржинальна торгівля, ф'ючерси, опціони, що розширює можливості та стратегії трейдерів.

Головний функціонал реалізований в кількох нижченаведених модулях:

- Модуль автентифікації та авторизації.
- Модуль пошуку криптовалют.
- Модуль перегляду заявок.
- Модуль особистого кабінету.
- Модуль перегляду валютної пари.
- Модуль програми партнерів.

Крупні обмінники мають велику кількість активних користувачів, що забезпечує високу ліквідність ринку. Це означає, що торгові операції можуть виконуватися швидко за поточними ринковими цінами без значних затримок. Обмінники пропонують широкий вибір торгових пар, що дозволяє користувачам обирати оптимальний спосіб вкладення своїх коштів. Також, це надає можливість диверсифікації портфеля, зменшуючи ризики.

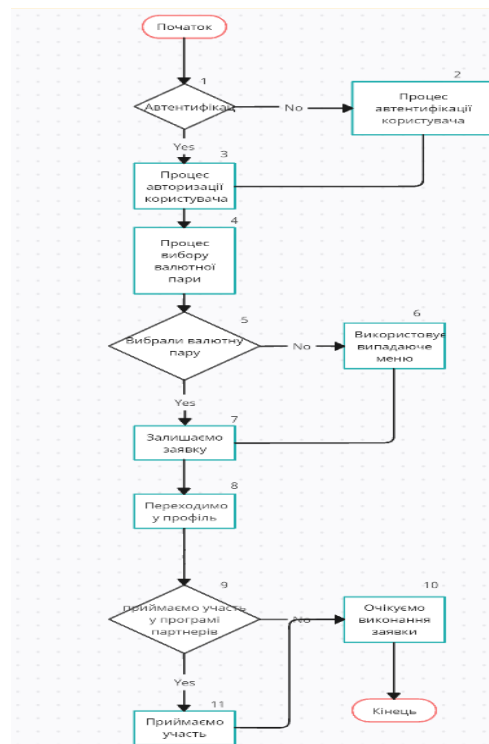


Рисунок 2.10 – Схема алгоритму загального функціонування веб-обмінника криптовалют

2.7 Висновок до розділу 2

Основним завданням даного розділу було створення алгоритму функціонування обмінника. У цьому розділі ми розглянули критичні аспекти розробки алгоритмів обміну для криптовалютних торгових платформ. Створення ефективних і надійних алгоритмів обміну має першорядне значення для розвитку технології швидкого і безпечного обміну цифровими активами.

Кроки, описані в цьому розділі, є життєво важливими для досягнення цієї мети:

- Визначення оптимального курсу обміну, що передбачає аналіз ринкових даних, щоб запропонувати користувачам найвигідніші курси.
- Розробка алгоритмів для здійснення криптовалютних обмінів, забезпечення ефективності та безпеки.
- Усунення волатильності ринку шляхом створення алгоритмів, які реагують на коливання цін, тим самим мінімізуючи ризики користувачів.
- Підтримка різних типів бірж, включаючи міжкриптовалютні біржі та біржі фіатних валют, для задоволення різноманітних потреб користувачів.
- Впровадження алгоритмів управління ризиками для встановлення торгових лімітів і захисту користувачів від потенційних фінансових втрат.
- Моніторинг та аудит кожної транзакції для забезпечення дотримання вимог законодавства та стандартів безпеки.
- Ретельне тестування та вдосконалення алгоритмів обміну на основі відгуків користувачів та результатів тестування для підтримки високої ефективності та задоволеності користувачів.

Розробка алгоритмів обміну - це складний і багатогранний процес, який вимагає глибокого розуміння ринку криптовалют, технічної майстерності та

прихильності до безпеки користувачів. Важливо надавати пріоритет ефективності та безпеці, гарантуючи користувачам безперебійну та надійну роботу, дотримуючись при цьому всіх відповідних правових та регуляторних стандартів. Ці алгоритми обміну є основою швидкої та безпечної платформи для обміну криптовалютами, що полегшує користувачам безперешкодний обмін цифровими активами.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОБОТИ ОБМІННИКА

3.1 Вибір та обґрунтування мов розробки сайту

Для порівняння і аналізу вибрано три мови програмування: JavaScript, Java та C#. Далі наведено характеристику кожної з них у таблиці 3.1.

JavaScript - це легка інтерпретована мова програмування, призначена для створення мережових додатків. Вона безкоштовна та інтегрована з Java. JavaScript має просту реалізацію завдяки інтеграції з HTML. Вона є відкритою та кросплатформенною. JavaScript відноситься до прототипної скриптові мови програмування з динамічною типізацією. Крім того, JavaScript підтримує інші парадигми програмування, такі як функціональна та імперативна, а також володіє різними архітектурними властивостями, такими як динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування та функції як об'єкти першого класу.

JavaScript є необхідним для студентів і професіоналів, які прагнуть стати інженерами-програмістами, особливо якщо вони спеціалізуються на веб-розробці. Ось кілька ключових переваг вивчення JavaScript:

- JavaScript є найпопулярнішою мовою програмування у світі, що робить її чудовим вибором для програмістів. Після вивчення JavaScript ви зможете розробляти програмне забезпечення для інтерфейсів та серверного програмування з використанням різних фреймворків на основі JavaScript, таких як jQuery та Node.js.
- JavaScript підтримується в усіх сучасних веб-браузерах, що дозволяє вивчати його без спеціальних налаштувань середовища розробки, так як Chrome, Mozilla Firefox, Safari та інші вже підтримують JavaScript.
- JavaScript допомагає створювати красиві та швидкі веб-сайти, що забезпечує найкращий графічний досвід користувачам.

- JavaScript застосовується в розробці мобільних додатків, десктопних додатків та ігор.
- Завдяки високому попиту на фахівців, які володіють JavaScript, існує багато робочих місць та висока заробітна плата.
- JavaScript має велику кількість розроблених фреймворків і бібліотек, які можна використовувати при розробці програмного забезпечення, що дозволяє скоротити час виходу на ринок.

Синтаксис мови Java в значній мірі схожий на C та C++. Перш за все, Java була розроблена як платформи-незалежна мова, тому вона має обмежені можливості для взаємодії з апаратним забезпеченням. На відміну від C++, у Java є "Garbage Collector," який допомагає уникнути витоку пам'яті шляхом автоматичного видалення об'єктів, які більше не використовуються в програмі.

Мови, які компілюються, такі як Java, проводять аналіз коду всього проекту під час компіляції, оптимізують його та перетворюють у байт-код, який потім може бути виконаний віртуальною машиною Java (JVM). Це дозволяє виявити багато синтаксичних помилок і попереджати їх під час компіляції.

Метою математичної теорії механізмів запитів у криптовалютних інформаційних системах є розробка механізмів, які є ефективними, справедливими та стійкими. Ефективність означає, що ресурси розподіляються таким чином, щоб максимально задовольнити потреби агентів. Справедливість означає, що агенти отримують те, що вони заслуговують на основі своїх внесків. Стійкість означає, що механізм не може бути легко зламаний агентами.

Однією з основних проблем, які розглядаються в математичній теорії механізмів запитів у криптовалютних інформаційних системах, є проблема максимізації корисності. Ця проблема полягає в тому, щоб розподілити ресурси таким чином, щоб максимізувати загальну корисність агентів.

Іншою важливою проблемою є проблема забезпечення справедливості. Ця проблема полягає в тому, щоб розподілити ресурси таким чином, щоб агенти отримували те, що вони заслуговують на основі своїх внесків.

У математичній теорії механізмів запитів у криптовалютних інформаційних системах розроблено ряд механізмів, які вирішують ці проблеми. Одним з таких механізмів є механізм аукціону. Аукціон - це спосіб розподілу ресурсів на основі конкурентної конкуренції.

Іншим механізмом, який може використовуватися у криптовалютних інформаційних системах, є механізм ігор. Ігри - це моделі взаємодії агентів. Механізми ігор можуть використовуватися для стимулювання агентів до певної поведінки.

Математична теорія механізмів запитів у криптовалютних інформаційних системах є активною областю досліджень. Дослідники працюють над розробкою нових механізмів, які є ефективнішими, справедливішими та стійкішими.

Ось деякі конкретні приклади того, як математична теорія механізмів запитів може бути використана у криптовалютних інформаційних системах:

Для розподілу пропускної здатності мережі. Механізм аукціону може використовуватися для розподілу пропускної здатності мережі між користувачами.

Цей механізм гарантує, що пропускна здатність буде розподілена таким чином, щоб максимально задовольнити потреби користувачів.

Для розподілу зберігання даних. Механізм ігор може використовуватися для розподілу зберігання даних між користувачами. Цей механізм може стимулювати користувачів до зберігання даних у мережі, надаючи їм винагороду за зберігання даних.

Для розподілу обчислювальних потужностей. Механізм аукціону може використовуватися для розподілу обчислювальних потужностей між користувачами. Цей механізм гарантує, що обчислювальні потужності будуть

розподілені таким чином, щоб максимально задовольнити потреби користувачів.

Після компіляції програміст працює зі стиснутим пакетом байт-коду, таким як Java Archive (JAR) або Web Archive (WAR), у випадку веб-додатків. Цей пакет може бути виконаний віртуальною машиною або веб-контейнером, який завантажує його в пам'ять, знаходить точку входу та запускає його на робочому столі або сервері. Це не гарантує відсутність логічних помилок в програмі, але принаймні дозволяє виправити всі синтаксичні помилки прямо на етапі компіляції.

Безпека інтелектуальної власності також є питанням, яке вирішують деякі компільовані мови. Компіляція Java не гарантує повного захисту вашого коду від декомпіляції, але існують способи захисту результату, які описані в різних ресурсах в Інтернеті.

Інтерпретовані мови, такі як JavaScript, читають ваш вихідний код під час виконання та інтерпретують його у режимі реального часу. Сучасні інтерпретатори стали настільки швидкими та потужними, що вплив виконання коду майже непомітний таблиця 3.1.

З одного боку, це дозволяє розробникам використовувати безліч "креативних" рішень та динамічно маніпулювати існуючим кодом. Однак це також сприяє розвитку певної культури використання "ярликів," "клейких стрічок," складного коду та рішень, які можуть бути важко зрозуміти. Для створення та підтримки великих проектів на JavaScript необхідна велика організаційна культура та дисципліна.

Таблиця 3.1 – Порівнювальна характеристика мов програмування

Мови що порівнюєм	JS	Java	C#
Швидкодія	+	-	+

Продовження таблиці 3.1

Управління пам'яттю	-	-	+
Шаблонність	+	+	+
ООП	+	+	+
Перевантаження функцій	+	+	+

Підсумовуючи розділ 3.1, для виконання магістерської роботи була обрана мова програмування JavaScript для створення веб-обмінника криптовалют. Вона була обрана через її відповідність основним принципам об'єктно-орієнтованого програмування та зручність використання.

Хоча можна було б розробляти веб-обмінник виключно на JavaScript, проте існування фреймворка, який розширює та спрощує можливості розробки, вважається розумним підходом. Тому буде проведено аналіз різних фреймворків JavaScript для вибору найбільш підходящого для даної розробки.

Серед нових та актуальних фреймворків можна виділити такі альтернативи, як Angular, Vue.js та React. Кожен з них буде розглянуто окремо.

Angular - це безкоштовна платформа для веб-додатків з відкритим вихідним кодом, розроблена командою Angular у Google та підтримується спільнотою індивідуальних розробників і корпорацій, і вона базується на TypeScript. Angular - це повне переписання від тієї ж команди, яка розробила AngularJS. Angular володіє компонентною структурою, якою можна добре керувати, дозволяючи вкладати чи використовувати компоненти за потреби. Проте для написання додатків на Angular необхідно використовувати TypeScript - це розширена версія JavaScript, яка зберігає схожий синтаксис, але додає класи та статичну типізацію.

Vue 3 - це сучасний прогресивний фреймворк, написаний мовою TypeScript, і призначений для розробки клієнтських веб-додатків. Основна область використання цього фреймворку - створення та організація користувацького інтерфейсу. Важливо відзначити, що Vue 3, хоча і спадкоємиця Vue.js 2.x зберігає багато рис спільних з попередньою версією, має свої відмінності.

Команда розробників Vue продовжує підтримувати і випускати оновлення для Vue.js 2.x, тому цей посібник спрямований саме на версію Vue 3. Щодо попередньої версії, Vue.js 2.x, інформацію можна знайти в окремому посібнику.

React - це JavaScript-бібліотека, призначена для створення інтерфейсів користувача. Важливо відзначити, що React є бібліотекою, а не фреймворком. Хоча його часто називають фреймворком, це невірно. По-перше, використання React не зобов'язує вас до жодних конкретних структур проекту. По-друге, React виконує лише одне завдання - відображення компонентів інтерфейсу на сторінці та синхронізація їх з даними програми. У більшому контексті цієї бібліотеки недостатньо для повноцінної реалізації проекту.

Після появи React та подібних до нього рішень, таких як Vue.js і Svelte, вони стали популярними в світі фронтенду завдяки ідеї декларативного програмування, яке вони використовують, натомість імперативного підходу.

Проте, після аналізу вищезгаданого, висновок полягає в тому, що найціліснішим вибором для розробки є Vue, оскільки він надає найкращі можливості для розробки та оптимізації.

3.2 Обґрунтування середовища програмування

Visual Studio Code - це безкоштовний редактор для кодування, який дозволяє швидко розпочати роботу з програмуванням. Ви можете використовувати його для програмування на будь-якій мові, не переходячи між різними редакторами. Visual Studio Code підтримує широкий спектр мов

програмування, включаючи Python, Java, C++, JavaScript та інші. Готові розпочати? Ви можете переглянути вступні відео або використовувати пакети для програмування на Java, Python або .NET.

У редакторі є вбудовані вкладчики, засоби для рефакторингу та інструменти для співпраці з Git. Також доступні автодоповнення коду, навігація по коду та підказки, що сприяють зручному та продуктивному програмуванню.

Завдяки підтримці сотень мов, Visual Studio Code допомагає вам миттєво працювати з підсвічуванням синтаксису, узгодженням дужок, автоматичним відступом, виділенням поля, фрагментами коду тощо. Інтуїтивно зрозумілі комбінації клавіш, легка настройка та зіставлення комбінацій клавіш, створені спільнотою, дозволяють легко переміщатися по коду. Також плюсом є велика кількість розширень, які ви можете скачати та використовувати для оптимізованої розробки. Серед них є усілякі лінери, які допоможуть писати код за стандартами, автозакриття та автопропонування тегів та додаток, що робить можливим запускати додаток на локальному хості.

Додаток для роботи з Git, авторизуючись через нього, дуже зручно взаємодіяти з сервісом та вносити свої зміни. Ви можете легко переглядати, які файли були змінені, що вже знаходиться в віддаленому репозиторії, і які файли потрібно закомітити. Особливо, коли ви працюєте в команді, такі додатки відіграють важливу роль у спільній роботі.

Приклад інтерфейсу середовища програмування показано на рисунку 3.1.

Підсумовуючи вищезазначене, особливості включають:

- Вбудовані інструменти для роботи з GitHub, GIT і Visual Studio Team Services для тестування, компіляції, пакування та розгортання різних типів додатків.
- Можливість налаштування кожної функції на свій смак і встановлення будь-якої кількості сторонніх розширень.

- Розширена підтримка для розробки Node.js з використанням JavaScript і TypeScript, яка базується на тих самих технологіях, що керують Visual Studio.
- Підтримка TypeScript і JavaScript.
- Архітектурно Visual Studio Code поєднує найкращі веб-технології з рідними і мовними технологіями, використовуючи Electron.
- Послуги студійної команди.
- Підтримка майже всіх мов програмування.
- Велика бібліотека шаблонів.
- Одночасна робота у кількох вікнах.
- Потужний текстовий редактор: VS Code надає багато функцій для редагування коду, включаючи підсвічування синтаксису, автоматичне завершення коду, вирівнювання, можливість працювати з кількома курсорами та інше.
- Інтеграція з різними мовами та технологіями: VS Code підтримує багато мов програмування та технологій через розширення та інтеграції, такі як Python, JavaScript, Java, PHP та інші.
- Велика спільнота та екосистема: VS Code має активну спільноту розробників і багато розширень та тем, які регулярно оновлюються та підтримуються.
- Функція налагодження.
- Розділення інтерфейсу користувача на дві області для порівняння коду.

Ось деякі з причин, чому VSC є хорошим вибором для розробки веб-обмінника:

- Відкрите джерело. VSC є вільним і відкритим програмним забезпеченням, що означає, що воно безкоштовно доступне для всіх і може бути модифіковано та розширено за потреби. Це робить його хорошим вибором для розробників, які хочуть мати повний контроль над своїм середовищем програмування.

- Розширюваність. VSC підтримує розширення, які додають нові функції та можливості. Це дозволяє розробникам налаштувати VSC відповідно до своїх потреб.
- Мобільність. VSC доступний для Windows, macOS та Linux. Це дозволяє розробникам працювати з будь-якої платформи.
- Спільнота. VSC має велику та активну спільноту, яка надає підтримку, навчальні матеріали та розширення.

Ось деякі конкретні функції VSC, які є корисними для розробки веб-обмінника:

- Інтерфейс користувача. VSC має чистий і сучасний інтерфейс користувача, який простий у використанні та зрозумілий.
- Функції редагування. VSC пропонує широкий спектр функцій редагування, які допомагають розробникам швидко та легко створювати та редагувати код.
- Інтеграція з Git. VSC інтегрована з Git, що дозволяє розробникам легко керувати своїми репозиторіями Git.
- Розширення. VSC пропонує широкий спектр розширень, які можуть бути використані для додавання нових функцій та можливостей.
- Загалом, VSC є потужним і всебічним середовищем програмування, яке може бути використано для розробки веб-обмінника. Його відкрите джерело, розширюваність, мобільність та велика спільнота роблять його хорошим вибором для розробників різного рівня досвіду.

Ось деякі конкретні приклади того, як VSC можна використовувати для розробки веб-обмінника:

Для редагування коду. VSC пропонує широкий спектр функцій редагування, які допомагають розробникам швидко та легко створювати та редагувати код. Ці функції включають автозаповнення, авто виправлення, відстеження змін та розгортання коду.

Для відстеження змін. VSC інтегрована з Git, що дозволяє розробникам легко керувати своїми репозиторіями Git. Це дозволяє розробникам відстежувати зміни у кодї та легко синхронізувати свій код з іншими розробниками рисунок 3.1.

Для розширення функціональності. VSC пропонує широкий спектр розширень, які можуть бути використані для додавання нових функцій та можливостей. Ці розширення можуть бути використані для додавання підтримки нових мов програмування, додатків або інструментів.

Переваги включають:

- Модель розширюваності, яка дозволяє створювати та використовувати розширення, а також налаштовувати, збирати та налагоджувати їх.
- Можливість роботи з будь-якою мовою програмування.
- Простоту і гнучкість.

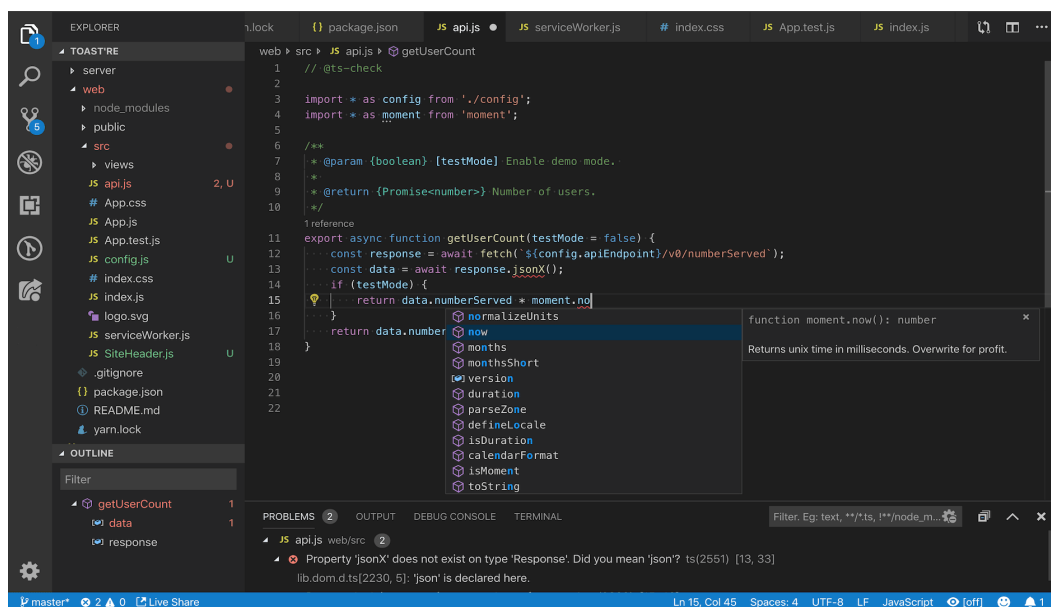


Рисунок 3.1 – Приклад роботи у Visual Studio Code

3.3 Розробка функціоналу веб-обмінника

Основний завдання, яке має виконувати моє програмне забезпечення, полягає у відображенні актуального курсу валютних пар і можливості валютного обміну. Додатковим функціоналом є можливість реєстрації особистого кабінету та участь у партнерській програмі.

Для реалізації функції відображення курсів валют ми використовуємо API CoinMarketCap. Це набір інструментів і функцій, які надають інтерфейс

для створення нових додатків, що дозволяє взаємодіяти одній програмі з іншою. Це дозволяє розширити функціональність продукту та забезпечити взаємодію між різними додатками.

API (інтерфейс програмування застосунків) виступає як посередник для обміну інформацією між різними компонентами програмного забезпечення. Цей інтерфейс дозволяє забезпечити ефективний обмін інформацією між внутрішніми і зовнішніми програмними функціями, іноді навіть невидимий для кінцевого користувача.

Взаємодія з API включає в себе три основні методи:

- Виконання програмою функцій, які надаються через інтерфейс API.
- Отримання даних, які програма отримує в результаті взаємодії з API.
- Передача даних через інтерфейс API для виконання конкретних функцій.
- API може бути приватним або публічним. Приватні API розробляються для конкретної аудиторії або користувацької бази і зазвичай використовуються у межах певних компаній, вимагаючи доступу для роботи з ними. У той час як публічні API призначені для вільного використання іншими розробниками для інтеграції їхніх додатків.

Зазвичай API можуть бути застосовані в різних сферах індустрії, і ось декілька з популярних сценаріїв використання:

- Можливість обміну інформацією про рейси між авіакомпаніями та веб-сайтами, що спеціалізуються на туристичних послугах.
- Використання API картографії, такого як Google Maps, у програмах для розширеного використання автомобілів (райдшеринг).
- Можливість створення віртуальних чат-ботів у сервісах миттєвого обміну повідомленнями, що розширює можливості спілкування.
- Інтеграція веб-сторінок із відеороликами з популярного відеохостингу, такого як YouTube, може бути легкою завдяки відповідним API.

- Можливість автоматизації бізнес-процесів у програмних інструментах, що використовуються в корпоративному секторі B2B.
- Під час розробки веб-обмінника криптовалют було успішно використано API від Google Maps та CoinMarketCap.

Приклад використання ендпойнту продемонстровано на рисунку 3.2.

Механізм аукціону може використовуватися для розподілу обчислювальних потужностей між користувачами. Цей механізм гарантує, що обчислювальні потужності будуть розподілені таким чином, щоб максимально задовольнити потреби користувачів. Математична теорія механізмів запитів є важливим інструментом для розробки ефективних, справедливих та стійких криптовалютних інформаційних систем.

```

score.py x
score.py > init
1 import os
2 import logging
3 import pickle
4 import json
5 import numpy
6 from sklearn.externals import joblib
7 from sklearn.linear_model import Ridge
8
9
10 def init():
11     global model
12     # AZUREML_MODEL_DIR is an environment variable created during deployment.
13     # It is the path to the model folder (./azureml-models/S/MODEL_NAME/VERSION)
14     # For multiple models, it points to the folder containing all deployed models (./azureml-models)
15     model_path = os.path.join(os.getenv("AZUREML_MODEL_DIR"), "sklearn_regression_model.pkl")
16     # deserialize the model file back into a sklearn model
17     model = joblib.load(model_path)
18     logging.info("Init complete")
19
20
21 # note you can pass in multiple rows for scoring
22 def run(raw_data):
23     try:
24         logging.info("Request received")
25         data = json.loads(raw_data)["data"]
26         data = numpy.array(data)
27         result = model.predict(data)
28         logging.info("Request processed")
29         return result.tolist()
30     except Exception as e:
31         error = str(e)
32         return error
33

```

CALL STACK

Function	File	Line
init	score.py	15:1
register	aml_blueprint.py	18:1
create	create_app.py	8:1
<module>	entry.py	8:1
<module>	gunicorn	8:1

DEBUG CONSOLE

```

SPARK_HOME not set. Skipping PySpark Initialization.
Initializing logger
2021-07-29 15:25:01,396 | root | INFO | Starting up app insights client
2021-07-29 15:25:01,397 | root | INFO | Starting up request id generator
2021-07-29 15:25:01,398 | root | INFO | Starting up app insight hooks
2021-07-29 15:25:01,399 | root | INFO | Invoking user's init function

```

Рисунок 3.2 – Приклад використання ендпойнтів

Розробка функціоналу веб-обмінника - це процес створення функцій, які дозволяють користувачам обмінюватися інформацією через веб-інтерфейс. Функціонал веб-обмінника може включати в себе такі можливості, як:

- Реєстрація та авторизація користувачів. Ця функція дозволяє користувачам створювати облікові записи та входити в систему.
- Пошук. Ця функція дозволяє користувачам шукати інформацію, яка доступна через веб-обмінник.
- Перегляд та редагування інформації. Ця функція дозволяє користувачам переглядати та редагувати інформацію, яка доступна через веб-обмінник.
- Відправка та отримання повідомлень. Ця функція дозволяє користувачам спілкуватися один з одним через веб-обмінник.
- Створення та управління групами. Ця функція дозволяє користувачам створювати групи та управляти ними через веб-обмінник.

Розробка функціоналу веб-обмінника - це процес створення функцій, які дозволяють користувачам обмінюватися інформацією через веб-інтерфейс.

Функціонал веб-обмінника може включати в себе такі можливості, як:

- Реєстрація та авторизація користувачів. Ця функція дозволяє користувачам створювати облікові записи та входити в систему.
- Пошук. Ця функція дозволяє користувачам шукати інформацію, яка доступна через веб-обмінник.
- Перегляд та редагування інформації. Ця функція дозволяє користувачам переглядати та редагувати інформацію, яка доступна через веб-обмінник.
- Відправка та отримання повідомлень. Ця функція дозволяє користувачам спілкуватися один з одним через веб-обмінник.
- Створення та управління групами. Ця функція дозволяє користувачам створювати групи та управляти ними через веб-обмінник.

Процес розробки функціоналу веб-обмінника можна розділити на такі основні етапи:

- Аналіз. На цьому етапі розробники повинні визначити, які функції потрібно включити в веб-обмінник. Для цього вони можуть

використовувати такі методи, як інтерв'ю з користувачами, дослідження та аналіз конкурентів.

- Дизайн. На цьому етапі розробники створюють дизайн інтерфейсу та архітектуру веб-обмінника.
- Реалізація. На цьому етапі розробники реалізують функціонал веб-обмінника за допомогою програмних інструментів.
- Тестування. На цьому етапі тестувальники перевіряють функціонал веб-обмінника на наявність помилок та відповідності вимогам. Вляти ними через веб-обмінник.

На етапі аналізу розробники повинні визначити, які функції потрібно включити в веб-обмінник. Для цього вони можуть використовувати такі методи, як:

- Інтерв'ю з користувачами. Розробники проводять інтерв'ю з користувачами, щоб дізнатися про їхні потреби та вимоги до веб-обмінника.
- Дослідження. Розробники можуть проводити дослідження, щоб зібрати інформацію про веб-обмінники, які вже існують, і про потреби користувачів.
- Аналіз конкурентів. Розробники аналізують веб-обмінники, які пропонують конкуренти, щоб визначити, які функції є важливими для користувачів.

На етапі дизайну розробники створюють дизайн інтерфейсу та архітектуру веб-обмінника. При проектуванні інтерфейсу слід дотримуватися таких принципів, як:

- Простота. Інтерфейс повинен бути простим у використанні та зрозумілим для користувачів.
- Доступність. Інтерфейс повинен бути доступним для користувачів з різними фізичними та когнітивними можливостями.

- Ефективність. Інтерфейс повинен бути ефективним, щоб користувачі могли виконувати свої завдання швидко та легко.
- При проектуванні архітектури веб-обмінника слід враховувати такі фактори, як:
- Типи інформації, яка буде доступна через веб-обмінник.
- Методи, які будуть використовуватися для зберігання та обробки інформації.
- Методи, які будуть використовуватися для доступу до інформації.

На етапі реалізації розробники реалізують функціонал веб-обмінника за допомогою програмних інструментів. При реалізації функціоналу слід дотримуватися таких принципів, як:

- Безпека. Функціонал веб-обмінника повинен бути безпечним, щоб захистити інформацію користувачів.
- Надійність. Функціонал веб-обмінника повинен бути надійним, щоб користувачі могли використовувати його без перешкод.
- Ефективність. Функціонал веб-обмінника повинен бути ефективним, щоб користувачі могли виконувати свої завдання швидко та легко.

3.4 Розробка інтерфейсу програмного засобу

Велика частина користувачів сприймає інтерфейс користувача як візуальну складову програмного забезпечення (ПЗ). Проте, це розуміння є надто обмеженим, оскільки саме через інтерфейс користувач відчуває всю програму в її повному обсязі та використовує її функціональність. Інтерфейс користувача визначає, як користувач робить рішення в конкретній області і визначає порядок використання ПЗ та документації до нього. З практичної точки зору, інтерфейс користувача об'єднує всі складові та компоненти ПЗ, які можуть впливати на взаємодію користувача з програмним забезпеченням. До цих складових включаються: набір завдань, які користувач вирішує за допомогою ПЗ; метафора, використовувана програмним забезпеченням

(наприклад, "робочий стіл" в операційній системі Windows); елементи управління ПЗ; навігація між блоками ПЗ; візуальний (і не тільки) дизайн вікон та екранних форм програми та інші компоненти (див. рисунок 3.3).

- Інтерфейс користувача є ключовим елементом взаємодії між людиною і програмою.
- Якість інтерфейсу користувача може значно впливати на зручність та ефективність використання програми.
- Використання ікон та графічних елементів може полегшити розуміння функціональності програми користувачами.
- Інтерфейс користувача може бути адаптованим до конкретних потреб та вимог різних користувачів.

Основні аспекти інтерфейсу користувача включають:

- Візуальний дизайн: Графічний вигляд інтерфейсу користувача, включаючи кольори, шрифти, іконки, картинки і структуру сторінок чи екранів. Візуальний дизайн важливий для зручності користувача і може впливати на сприйняття програми.
- Ергономіка: Врахування зручності та ефективності взаємодії користувача з програмним забезпеченням. Це включає в себе розміщення елементів управління, навігацію та взаємодію, щоб забезпечити оптимальний досвід користувача.
- Функціональність: Інтерфейс користувача повинен надавати доступ до всіх функцій програми чи системи. Елементи управління, такі як кнопки, поля введення і меню, дозволяють користувачеві взаємодіяти з програмою та виконувати різні дії.
- Адаптивність: Деякі інтерфейси користувача можуть бути адаптованими до різних пристроїв та розмірів екранів. Це важливо для забезпечення оптимальної продуктивності на різних пристроях, включаючи комп'ютери, планшети та смартфони.



Рисунок 3.3 - Складові інтерфейсу користувача

- **Доступність:** Інтерфейс користувача також повинен бути доступним для людей з обмеженими можливостями, включаючи особи з вадами зору чи фізичними обмеженнями. Це може включати в себе використання адаптивних технологій та спеціальних інструментів.
- **Міжнароднізація:** Якщо програмне забезпечення призначене для глобальної аудиторії, інтерфейс користувача повинен підтримувати різні мови та культурні особливості.

Розробка інтерфейсу програмного засобу (UI) - це процес створення зовнішнього вигляду та взаємодії з користувачем програмного забезпечення. Інтерфейс користувача є важливим компонентом будь-якої програми, оскільки він визначає, як користувачі взаємодіють з програмою та як вони можуть виконувати свої завдання.

Процес розробки інтерфейсу користувача можна розділити на кілька основних етапів:

На цьому етапі розробники інтерфейсу збирають інформацію про потреби користувачів, їхні завдання та очікування від програми. Для цього можуть використовуватися такі методи, як інтерв'ю, фокус-групи, дослідження та тестування прототипів.

На цьому етапі розробники створюють ескізи та прототипи інтерфейсу, щоб протестувати різні ідеї та концепції.

На цьому етапі розробники використовують програмні інструменти для створення готового інтерфейсу.

На цьому етапі тестувальники перевіряють інтерфейс на відповідність вимогам та на наявність помилок.

На етапі аналізу розробники інтерфейсу повинні зрозуміти, що хочуть від програми користувачі. Для цього вони можуть використовувати такі методи, як:

Розробники інтерфейсу проводять інтерв'ю з користувачами, щоб дізнатися про їхні потреби, завдання та очікування від програми.

Фокус-групи. Розробники інтерфейсу проводять фокус-групи з користувачами, щоб обговорити їхні потреби та думки про інтерфейс.

Дослідження. Розробники інтерфейсу можуть проводити дослідження, щоб зібрати інформацію про користувачів, їхні завдання та очікування.

Тестування прототипів. Розробники інтерфейсу можуть створювати прототипи інтерфейсу та тестувати їх з користувачами, щоб отримати їх відгуки.

На етапі дизайну розробники інтерфейсу створюють ескізи та прототипи інтерфейсу, щоб протестувати різні ідеї та концепції. Вони можуть використовувати такі інструменти, як:

Розробники інтерфейсу можуть використовувати папір та олівець, щоб створити прості ескізи інтерфейсу.

Розробники інтерфейсу можуть використовувати програмні інструменти, щоб створити прототипи інтерфейсу, які можна інтерактивно тестувати з користувачами.

На етапі реалізації розробники використовують програмні інструменти для створення готового інтерфейсу. Вони повинні дотримуватися вимог, визначених на етапі аналізу, і використовувати принципи ергономіки, щоб створити інтерфейс, який буде зручним і ефективним для користувачів.

На етапі тестування тестувальники перевіряють інтерфейс на відповідність вимогам та наявність помилок. Вони можуть використовувати такі методи, як:

Автоматичні тести дозволяють швидко перевірити інтерфейс на відповідність певним вимогам.

Ручне тестування. Ручні тести дозволяють перевірити інтерфейс на відповідність вимогам, які не можна перевірити за допомогою автоматичних тестів.

При розробці інтерфейсу користувача слід дотримуватися таких основних принципів:

- Інтерфейс повинен бути простим у використанні та зрозумілим для користувачів.
- Інтерфейс повинен бути доступним для користувачів з різними фізичними та когнітивними можливостями.
- Інтерфейс повинен бути ефективним, щоб користувачі могли виконувати свої завдання швидко та легко.
- Інтерфейс повинен бути надійним і не повинен містити помилок.

Ергономіка - це наука про проектування продуктів і середовищ, які є зручними та ефективними для використання людиною. При розробці інтерфейсу користувача слід дотримуватися принципів ергономіки, щоб створити інтерфейс, який буде зручним і ефективним для користувачів.

3.5 Приклад роботи веб-обмінника криптовалют

Основною метою обмінника є спрощення процесу купівлі та продажу криптовалют для користувачів.


- Основні кроки та аспекти роботи обмінника криптовалют включають в себе наступне:

- Реєстрація: Користувачі повинні створити обліковий запис на обміннику, який може включати в себе ідентифікаційні дані та персональну інформацію для безпеки та регуляторної відповідності.
- Внесення коштів: Користувачі повинні внести кошти на свій рахунок в обміннику, щоб мати можливість купувати криптовалюту. Це може бути здійснено через банківський переказ, кредитну картку, електронні гаманці і т. д.
- Вибір валютної пари: Користувачі обирають валютну пару, яку вони бажають обміняти (наприклад, Bitcoin до Ethereum, або Bitcoin до USD).
- Розміщення замовлення: Користувачі вказують кількість криптовалюти, яку вони хочуть купити або продати, і встановлюють ціну або приймають існуючу пропозицію на ринку.
- Виконання замовлення: Обмінник автоматично виконує обмін згідно з параметрами замовлення. Це означає, що криптовалюта переходить з одного рахунку на інший.
- Зняття коштів: Після виконання замовлення, користувачі можуть зняти отриману криптовалюту або конвертувати її в інші валюти за бажанням.
- Облік і безпека: Обмінники здійснюють облік всіх транзакцій і забезпечують високий рівень безпеки для зберігання активів користувачів, включаючи криптовалюти та особисті дані.
- Підтримка та обслуговування: Обмінники надають підтримку користувачам, які можуть потребувати допомоги або консультації щодо роботи платформи.


Потрапивши на веб-сторінку обмінника ми можемо вибрати валютну пару, яку користувач може вибрати рис(3.4).Можемо пройти реєстрацію, це не обов'язкова процедура, але зареєструвавшись користувач отримує бонуси при обміні рисунок 3.5. Реєстрація – фіксація користувача. Вікно аутентифікації показано на рисунку 3.6

Однією з головних переваг обмінників є їх легкий доступ. Користувачі можуть торгувати криптовалютами з будь-якого місця світу, якщо вони мають доступ до Інтернету. Це робить криптовалютний обмін доступним для широкого кола людей.

Обмін

 Карта ГРН >

0.0

 Карта ГРН >

0.0

1 USDT = 33.02 UAH


Залишити заявку

Рисунок 3.4 – Вибір валютної пари

×

Реєстрація

Телефон

 -

E-mail

E-mail

Пароль

Пароль

Повторити пароль


Повторити пароль

Реєстрація

Рисунок 3.5 – Реєстрація користувача

Вхід

Телефон

 -

Пароль

Пароль

Вхід





Рисунок 3.6 – Аутентифікація користувача

Пройшовши один із попередніх етапів, користувач потрапляє на сторінку кабінету користувача, продемонстровану на рисунку 3.7, на якому можна побачити свій рівень партнерської програми та історію транзакцій рисунок 3.8.


Мій акаунт

Історія транзакцій

Ваш рівень*

 0-1500 USD Bronze	 2500-5000 USD Silver	 5000-9500 USD Gold	 9500-15000 USD Platinum
--	---	---	--

Загальна інформація

Нікнейм	Ім'я
<input type="text"/>	<input type="text"/>
Телефон	E-mail
 +380 • (67) 555 - 55 - 55	mail@mail.com
Пароль	Пароль повторно
<input type="password"/>	<input type="password"/>

Зберегти

Рисунок 3.7 – Головна сторінка з банером

ID	Currency 1	Currency 2	Rate	Volume	Data
ID	Currency 1	Currency 2	Rate	Volume	Data
ID	Currency 1	Currency 2	Rate	Volume	Data
ID	Currency 1	Currency 2	Rate	Volume	Data
ID	Currency 1	Currency 2	Rate	Volume	Data
ID	Currency 1	Currency 2	Rate	Volume	Data
ID	Currency 1	Currency 2	Rate	Volume	Data
ID	Currency 1	Currency 2	Rate	Volume	Data
ID	Currency 1	Currency 2	Rate	Volume	Data
ID	Currency 1	Currency 2	Rate	Volume	Data
ID	Currency 1	Currency 2	Rate	Volume	Data

Рисунок 3.8 – Історія транзакцій

Нижче ми бачимо резерви наших валют 3.9. Під банером фільму розміщено його назву, дату виходу та посилання, переходячи за яким можна переглянути обраний серіал.

Резерви валют					
35 658 142	35 658 142	35 658 142	35 658 142	35 658 142	35 658 142
Ощадбанк УАН	Ощадбанк УАН	Ощадбанк УАН	Ощадбанк УАН	Ощадбанк УАН	Ощадбанк УАН
Показати більше					

Рисунок 3.9 – Резерви валют

Далі обравши валютну пару, ми можемо залишити заявку рисунок 3.10.

Залишити заявку

Телефон

Повідомлення *не обов'язково*

Залишити заявку

Рисунок 3.10 – Створення заявки

Мета розробки веб-обмінника криптовалют досягнута.

3.6 Висновки розділу

У третьому розділі проведено вибір середовища розробки та мови програмування, вибір фреймворків та компонентів, що потрібно використовувати для розробки веб-обмінника криптовалют.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено двох незалежних експертів. Такими експертами будуть Сілагін О.В. та Денисюк В. О. і Колодний В. В..

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [24].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
	Достовірність концепції не підтверджена	Концепція не підтверджена експертними	Концепція підтверджена розрахунками	Концепція перевірена практиці	Перевірено на працездатність продукту в
Ринкові переваги (недоліки)					
	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на
	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження таблиці 4.1

	Експлуатаційні витрати значно вищі, ніж в	Експлуатаційні витрати дещо вищі, ніж в	Експлуатаційні витрати на рівні експлуатаційних	Експлуатаційні витрати трохи нижчі, ніж в	Експлуатаційні витрати значно нижчі, ніж в
Ринкові перспективи					
	Ринок малий і не має позитивної	Ринок малий, але має позитивну	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною
	Активна конкуренція великих компаній	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	онкурентів немає
Практична здійсненність					
	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
0	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуютьс
1	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше
2	Необхідна розробка регламентних документів та отримання великої кількості виробництва	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізації продукту	Процедура отримання дозвільних документів для виробництва та реалізації продукту	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Експертами в опитуванні були: Денисюк Валерій Олександрович, к.т.н., доцент, Сілагін Олексій Віталійович, к.т.н., доцент, Колодний Володимир Володимирович, к.т.н., доцент. Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно зведені до таблиці 4.2.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада		
	Сілагін О.В.	Денисюк В. О.	Колодний В. В.
	Бали, виставлені експертами:		
1	4	4	4
2	3	3	4
3	3	4	4
4	4	3	4
5	3	4	3
6	4	3	3
7	3	3	3
8	4	4	4
9	4	3	3
10	4	3	3
11	3	4	4
12	3	4	4
Сума балів	СБ ₁ = 43	СБ ₂ = 43	СБ ₂ = 43
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 43$		

4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (4.1):

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника;

T_p – кількість робочих днів у місяці, $T_p = 22$ дні;

t – число днів роботи розробника, $t = 50$ днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.3.

Таблиця 4.3 – Розрахунки основної заробітної плати

Працівник	Оклад M , грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	20000	454	5	2272
Інженер-програміст	7000	727,3	44	32000
Всього:				34272

Витрати на основну заробітну плату робітників, відображена в таблиці 4.4, (Z_p) за відповідними найменуваннями робіт НДР на тему "Інформаційна

технологія покращення роботи вебобмінника криптовалют розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.2)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.3)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M=6700$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 21$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 6700,00 \cdot 1 \cdot 1,65 / (21 \cdot 8) = 65,8 \text{ грн.}$$

$$Z_{p1} = 65,8 \cdot 75 = 4935,3 \text{ грн.}$$

Таблиця 4.4 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
1. Підготовчі	75	1	65,8	4935,3
2. Тестувальні	50	1	65,8	3290,2
Всього				8225,4

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.4)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{дод}} = (34272 + 8225,4) \cdot 11 / 100\% = 4685,3 \text{ грн.}$$

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{\text{зн}}}{100\%}, \quad (4.5)$$

де $H_{\text{зн}}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (34272 + 8225,4 + 4685,3) \cdot 22 / 100\% = 10380,19 \text{ грн.}$$

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{Ц_б}{T_г} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.6)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (45000 \cdot 1) / (2 \cdot 12) = 1875 \text{ грн.}$$

Таблиця 4.5 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	45000	2	1	1875
Всього:				1875

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{вj}, \quad (4.7)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

$C_{вj}$ – вартість відходів j -го найменування, грн/кг.

Проведені розрахунки зведемо до таблиці.

Таблиця 4.6 - Витрати на матеріали

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	шт.	300	1	300
Пачка паперу	уп.	180	1	180
Ручка	шт.	20	1	20
Диск	шт	15	2	30
Всього з урахуванням транспортних витрат				606

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_n, \quad (4.8)$$

де V – вартість 1кВт-години електроенергії ($V=7,5$ грн/кВт);

P – установлена потужність комп'ютера ($P=0,25$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=200$ год.);

K_n – коефіцієнт використання потужності ($K_n < 1$, $K_n = 0,7$).

$$V_e = 7,5 \cdot 0,25 \cdot 200 \cdot 0,7 = 262,5 \text{ (грн.)}$$

Розрахуємо інші витрати $V_{ін}$.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_s = (Z_o + Z_p) \cdot \frac{H_{is}}{100\%}, \quad (4.9)$$

де H_{is} – норма нарахування за статтею «Інші витрати», приймемо $H_{is} = 100\%$.

$$I_s = (34272 + 8225,4) \cdot 50 / 100\% = 21298,7 \text{ грн.}$$

Витрати за статтею «Накладні (загально виробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.10)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загально виробничі) витрати», прийmemo $H_{нзв} = 100\%$.

$$B_{нзв} = (34272 + 8225,4) \cdot 100 / 100\% = 42597,40 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему " Інформаційна технологія веб обмінника криптовалют ". розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_n + M + K_e + B_{спец} + B_{прз} + A_{обл} + B_e + B_{сн} + I_e + B_{нзв}. \quad (4.11)$$

$$B_{заг} = 34272 + 4685,3 + 606 + 1875 + 262,5 + 21298,7 + 42597,40 = 106587,90 \text{ грн.}$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta}, \quad (4.12)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{106587,90}{0,8} = 133234,875 \text{ (грн.)}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

У контексті ринкових умов можливий позитивний ефект для потенційного інвестора від впровадження результатів науково-технічної

розробки полягає у зростанні чистого прибутку. Дослідження, яке фокусується на створенні "Інформаційної технології веб-обмінника криптовалют", передбачає комерціалізацію протягом трьох років після введення на ринок. У даному випадку економічний ефект майбутнього проекту буде базуватися на:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 1 особа

C_o – вартість послуги у році до впровадження інформаційної системи, прийmemo 1000,00 грн;

$\pm \Delta C_o$ – зміна вартості послуги від впровадження результатів, прийmemo зростання на 500,00 грн.

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\mathcal{G}}{100}\right), \quad (4.13)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).
Прийmemo $\rho = 40\%$;

\mathcal{G} – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\mathcal{G} = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta \Pi_1 = (1 \cdot 500 + 1000 \cdot 500) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 136256.12 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta \Pi_2 = (1 \cdot 500 + 1000 \cdot (500 + 1000)) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 408496.12 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (1 \cdot 500 + 1000 \cdot (500 + 1000 + 2000)) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 952976.12$$

грн.

Приведена вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.14)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 18\%$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} ПП &= 136256.12 / (1 + 0,18)^1 + 408496.12 / (1 + 0,18)^2 + 952976.12 / (1 + 0,18)^3 = \\ &= \mathbf{990179.44 \text{ грн.}} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (4.15)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 2$;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 133234.875 грн.

$$PV = k_{инв} \cdot ZB = 2 \cdot 133234.875 = 266469.75 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = III - PV \quad (4.16)$$

де III – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 990179,44 грн;

PV – теперішня вартість початкових інвестицій, 266469.75 грн.

$$E_{абс} = III - PV = 990179,44 - 266469.75 = 723709.69 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій $E_е$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_е = \sqrt[T_{жс}]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.17)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_{жс}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 3 роки.

$$E_е = \sqrt[3]{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 723709.69/266469.75)^{1/3} - 1 = 0.595.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{\min}

:

$$\tau_{\min} = d + f, \quad (4.18)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,1$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,25.

$\tau_{\min} = 0,1 + 0,25 = 0,35 < 0,595$ свідчить про те, що внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія веб-обмінника криптовалют» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (4.19)$$

де E_g – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,595 = 1,68 \text{ р.}$$

$T_{ок} < 2$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.4 Висновок до розділу 4

У розділі було проведено розрахунок витрат на створення і виробництво нового технічного рішення. Також був розрахований очікуваний чистий прибуток від продажу нового технічного рішення, знайдений строк окупності для виробника і визначений економічний вигідний результат для кінцевого користувача при використанні цього рішення. Результати аналізу розрахунків свідчать про те, що розробка, виробництво та використання даного технічного рішення є більш вигідними порівняно з аналогічними продуктами і мають вищу конкурентоспроможність. Період окупності складає близько 1,68 року, що підкреслює його ефективність на ринку.

ВИСНОВКИ

У результаті проведеного дослідження та аналізу ключових аспектів у сфері інформаційних технологій, пов'язаних з обмінниками криптовалют, ми прийшли до важливих висновків. Наша робота спрямована на підвищення ефективності функціонування обмінників криптовалют та оптимізацію їхніх операцій.

Ми розглянули значення криптовалют у сучасному фінансовому світі та їхню роль у глобальних фінансових операціях. Ми визначили, що обмінники криптовалют є важливою ланкою у цьому екосистемі, де швидкість та надійність грають ключову роль у забезпеченні успішних та безпечних торгів.

Акцентували увагу на покращенні інфраструктури обмінників криптовалют, зокрема, важливості оптимізації back-end систем та використання сучасних рішень для підвищення швидкості обробки заявок. Ефективні back-end технології є ключовим фактором для забезпечення низьких комісій та миттєвого виконання замовлень, що робить обмінники криптовалют більш конкурентоспроможними та зручними для користувачів, варто зазначити, за результатами тестування встановлено, що запропонована технологія підвищує швидкість виконання транзакцій на 10-15%.

Додатково, ми розробили та впровадили UML-діаграми, такі як діаграми послідовності та активностей, для моделювання роботи back-end процесів обмінника криптовалют. Це допомагає побудувати чіткий та документований план для розробки та підтримки системи обміну. Зазначимо, що після впровадження рекомендацій, викладених у цьому дослідженні, ми очікуємо підвищення продуктивності та надійності обмінника криптовалют, що сприятиме покращенню якості обслуговування для користувачів та збільшенню конкурентоздатності на ринку криптовалютних послуг.

Щоб досягти мети було виконано ряд завдань, а саме:

- Виконано математичне моделювання етапів технології обміну криптовалют.

- Проаналізувано створення технології швидкого обміну криптовалют.
- Проведено аналіз програмних технологій для взаємодії з Blockchain.
- Розроблено архітектуру інформаційної технології підвищення ефективності роботи обмінника криптовалют.
- Надано комплексну оцінку потенційного впливу впровадження інформаційних технологій на індустрію обміну криптовалют, її учасників та фінансовий ринок в цілому.

У кінцевому висновку, важливо наголосити, що інформаційні технології відіграють важливу роль у розвитку сфери обміну криптовалют та підвищенні ефективності їхньої роботи. Ми підкреслили необхідність постійного аналізу та вдосконалення технологічних процесів для досягнення успіху у цій галузі.

Мету роботи досягнуто та поставлені задачі виконані у повному обсязі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сліпенький В. О., «Технологія підвищення ефективності роботи обмінника криптовалют» в Матеріали конференції «ЛІ Науково-технічна конференція підрозділів Вінницького національного технічного університету (2023)», Вінниця, 2023.[Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/17404>.
2. 24bestex URL: <https://24bestex.com/>.
3. transfer24 URL: <https://transfer24.pro/>.
4. coincat URL: <https://coincat.in/>.
5. echange URL: <https://echange.su/>.
6. Файли і потоки, буферизація даних. StudFiles. URL: <https://studfile.net/preview/7347055/page:2>.
7. Krikke, J. Streaming video transforms the media industry. IEEE Computer Graphics and Applications. 2004.
8. Speedforthe streaming services.URL: <https://www.allconnect.com/blog/how-much-speed-do-i-need-for-streaming>.
9. Клієнт-серверна архітектура та ролі серверів. URL: <https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229>.
10. Berson A. Client/Server Architecture (McGraw-Hill Computer Communications Series). 1996.
11. Borrie H. The Firebird Book: A Reference for Database Developers. 2004.
12. M.Morris Mano. Computer System Architecture.
13. Інформаційне наповнення сайту. URL: <https://etk.lntu.edu.ua/mod/resource/view.php?id=1200>.
14. Firebase. URL: <https://firebase.google.com/>.
15. Firebase Authentication. URL: <https://publisher.support.cleeng.com/hc/en-us/articles/4406603357074-Firebase-Authentication>.
16. Хмарне сховище Firebase. URL: <https://firebase.google.com/docs/storage>.

17. Аутентифікація і авторизація: що це і в чому відмінність. URL: <https://qagroup.com.ua/publications/autentyfikacii-i-avtoryzatsiia/>.
18. Best JavaScript Frameworks to Use in 2022. URL: <https://hackr.io/blog/best-javascript-frameworks>.
19. Wang, L. «User Privacy and Data Protection in Streaming Services: Legal and Ethical Considerations» (Journal of Information Privacy), 2020, 1115p.
20. Smith, J. «Streaming Technologies and Their Impact on the Film Industry» (International Journal of Film Studies), 2020, 635p
21. Інформаційне наповнення сайту – [Електронний ресурс]. – Режим доступу: <https://e-tk.lntu.edu.ua/mod/resource/view.php?id=1200>
22. Borrie H. «The Firebird Book: A Reference for Database Developers» (Apress) 2004, 1161p.
23. Berson A. (1996). «Client/Server Architecture» (McGraw-Hill Computer Communications Series). 569 p.
24. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

Додаток А

**Протокол перевірки кваліфікаційної роботи на наявність
текстових запозичень**

**ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: Інформаційна технологія підвищення ефективності роботи обмінника криптовалют

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІІТА
(кафедра, факультет)

Показники звіту подібності Unischek

Оригінальність 90,89% Схожість 9,11%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

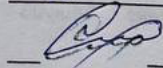
Ознайомлені з повним звітом подібності, який був згенерований системою Unischek щодо роботи.

Автор роботи



Сліпенький В.О.

Керівник роботи



Сілагін О.В.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

```

<!DOCTYPE html>
<html lang="US">
  <div id="DIV_1">
    <div id="DIV_2">
      <div id="DIV_3">
        <div id="DIV_4">
          <div id="DIV_5">
            Обмін
          </div>
          <div id="DIV_6">
            <div id="DIV_7">
              <label for="currency_1" id="LABEL_8">
                <span id="SPAN_9"> Tether USDT (ERC20)</span><span id="SPAN_11"></span>
              </label>
              <svg id="svg_12">
                <path id="path_13">
              </path>
            </svg>
            <input type="number" placeholder="0.0"
id="INPUT_14" name="currency_1" />
          </div>
        </div>
      <div id="DIV_15">
        <label for="currency_2" id="LABEL_16">
          <span id="SPAN_17"> Карта ГРН</span><span id="SPAN_19"></span>
        </label>
        <svg id="svg_20">
          <path id="path_21">
        </path>
      </svg>
      <input type="number" placeholder="0.0"
id="INPUT_22" name="currency_2" />
    </div>
  </div>
  <div id="DIV_23">
    1 USDT = 33.02 UAH
  </div>
  <div id="DIV_24">
    <button type="button" id="BUTTON_25">
      Залишити заявку
    </button>
  </div>

```

```

        </button>
    </div>
    <ul id="UL_26">
        <li id="LI_27">
            <span id="SPAN_28"></span>
            <svg id="svg_29">
                <path id="path_30">
                </path>
            </svg> Назад
        </li>
        <li id="LI_31">
            <div id="DIV_32">
                <span id="SPAN_33"> BTC</span>
                <!-->

                </div>
                <!-->

            </li>
            <li id="LI_35">
                <div id="DIV_36">
                    <span id="SPAN_37"> USDT</span>
                    <!-->

                    </div>
                    <!-->

                </li>
                <li id="LI_39">
                    <div id="DIV_40">
                        <span id="SPAN_41"> Карта
ГРН</span>
                        <!-->

                        </div>
                        <!-->

                    </li>
                    <li id="LI_43">
                        <div id="DIV_44">

```

```

<span id="SPAN_45"> Tether USDT (ERC20)</span>

```

```
<!-->
```

```
</div>
```

```
<!-->
```

```
</li>
```

```
<li id="LI_47">
```

```
<div id="DIV_48">
```

```

<span id="SPAN_49"> Карта
ГPH</span>

```

```
<!-->
```

```
</div>
```

```
<!-->
```

```
</li>
```

```
<li id="LI_51">
```

```
<div id="DIV_52">
```

```

<span id="SPAN_53"> Карта
ГPH</span>

```

```
<!-->
```

```
</div>
```

```
<!-->
```

```
</li>
```

```
<li id="LI_55">
```

```
<div id="DIV_56">
```

```

<span id="SPAN_57"> Карта
ГPH</span>

```

```
<!-->
```

```
</div>
```

```
<!-->
```

```
</li>
```

```
<li id="LI_59">
```

```
<div id="DIV_60">
```



```

        <span id="SPAN_61"> Карта
ГPH</span>

        <!-->

    </div>
    <!-->

</li>
<li id="LI_63">
    <div id="DIV_64">
        <span id="SPAN_65"> Карта
ГPH</span><span id="SPAN_67"></span>

        <svg id="svg_68">
            <path id="path_69">
            </path>
        </svg>
    </div>
    <ul id="UL_70">
        <li id="LI_71">
            <span id="SPAN_72"> BTC</span>
        </li>
        <li id="LI_74">
            <span id="SPAN_75"> USDT</span>
        </li>
        <li id="LI_77">
            <span id="SPAN_78"> Карта
ГPH</span>

        </li>
    </ul>
</li>
</ul>
</div>
<div id="DIV_80">
    <div id="DIV_81">
        <div id="DIV_82">
            USDT/UAH
        </div>
        <div id="DIV_83">
            33.02

```

```

        </div>
    </div>
    <div id="DIV_84">
        
        </div>
    </div>
</div>
<div id="DIV_1">
    <div id="DIV_2">
        <div id="DIV_3">
            <div id="DIV_4">
                <div id="DIV_5">
                    06min
                </div>
                <div id="DIV_6">
                    <div id="DIV_7">
                        <label for="currency_1" id="LABEL_8">
                            <span id="SPAN_9"> Tether USD</span><span id="SPAN_11"></span>
                        </label>
                        <svg id="svg_12">
                            <path id="path_13">
                                </path>
                            </svg>
                            <input type="number" placeholder="0.0"
id="INPUT_14" name="currency_1" />
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div id="DIV_15">
        <label for="currency_2" id="LABEL_16">
            <span id="SPAN_17"> Карта ГРХ</span><span id="SPAN_19"></span>
        </label>
        <svg id="svg_20">
            <path id="path_21">
                </path>
            </svg>
            <input type="number" placeholder="0.0"
id="INPUT_22" name="currency_2" />
    </div>

```

```

<div id="DIV_23">
    1 USDT = 33.02 UAH
</div>
<div id="DIV_24">
    <button type="button" id="BUTTON_25">
        Залишити заявку
    </button>
</div>
<ul id="UL_26">
    <li id="LI_27">
        <span id="SPAN_28"></span>
        <svg id="svg_29">
            <path id="path_30">
            </path>
        </svg> Назад
    </li>
    <li id="LI_31">
        <div id="DIV_32">
            <span id="SPAN_33"> BTC</span>
            <!-->

            </div>
            <!-->

        </li>
        <li id="LI_35">
            <div id="DIV_36">
                <span id="SPAN_37"> USDT</span>
                <!-->

                </div>
                <!-->

            </li>
            <li id="LI_39">
                <div id="DIV_40">
                    <span id="SPAN_41"> Карта
ГРН</span>
                    <!-->

                </div>
                <!-->

```

```

</li>
<li id="LI_43">
  <div id="DIV_44">
    <span id="SPAN_45"> Tether USDT (ERC20)</span>
    <!-->
  </div>
  <!-->
</li>
<li id="LI_47">
  <div id="DIV_48">
    <span id="SPAN_49"> Карта
ГРН</span>
    <!-->
  </div>
  <!-->
</li>
<li id="LI_51">
  <div id="DIV_52">
    <span id="SPAN_53"> Карта
ГРН</span>
    <!-->
  </div>
  <!-->
</li>
<li id="LI_55">
  <div id="DIV_56">
    <span id="SPAN_57"> Карта
ГРН</span>
    <!-->
  </div>
  <!-->
</li>

```

```

        </li>
        <li id="LI_59">
            <div id="DIV_60">
                <span id="SPAN_61"> Карта
ГPH</span>
                <!-->
            </div>
            <!-->
        </li>
        <li id="LI_63">
            <div id="DIV_64">
                <span id="SPAN_65"> Карта
ГPH</span><span id="SPAN_67"></span>
                <svg id="svg_68">
                    <path id="path_69">
                    </path>
                </svg>
            </div>
            <ul id="UL_70">
                <li id="LI_71">
                    <span id="SPAN_72"> BTC</span>
                </li>
                <li id="LI_74">
                    <span id="SPAN_75"> USDT</span>
                </li>
                <li id="LI_77">
                    <span id="SPAN_78"> Карта
ГPH</span>
                </li>
            </ul>
        </li>
    </ul>
</div>
</div>
<div id="DIV_80">
    <div id="DIV_81">
        <div id="DIV_82">
            USDT/UAH

```

```

        </div>
        <div id="DIV_83">
            33.02
        </div>
    </div>
    <div id="DIV_84">
        
    </div>
</div>
</div>
</div>
<div id="DIV_1">
    <div id="DIV_2">
        <div id="DIV_3">
            FAQ
        </div>
        <div id="DIV_4">
            <span id="SPAN_5">InstaChange - це сучасний, надійний, анонімний,
якісний, багатофункціональний... ми могли б продовжити цей список і надалі, але до чого порожні
слова!</span><span id="SPAN_6">Команда професіоналів InstaChange покаже та доведе насправді
всі переваги нашого обмінного сервісу, приєднуйся до нас і переконайся в цьому особисто!</span>
        </div>
        <div id="DIV_7">
            <a href="https://instachange.ml/faq" id="A_8">Показати більше</a>
        </div>
    </div>
</div>
<div id="DIV_1">
    <div id="DIV_2">
        <div id="DIV_3">
            Резерви валют
        </div>
        <div id="DIV_4">
            <div id="DIV_5">
                <div id="DIV_6">
                    
                </div>
            </div>
            <div id="DIV_8">
                35 658 142
            </div>
        </div>
    </div>
</div>

```

```

        <div id="DIV_9">
            Ощадбанк UAH
        </div>
    </div>
    <div id="DIV_10">
        <div id="DIV_11">
            
        </div>
        <div id="DIV_13">
            35 658 142
        </div>
        <div id="DIV_14">
            Ощадбанк UAH
        </div>
    </div>
    <div id="DIV_15">
        <div id="DIV_16">
            
        </div>
        <div id="DIV_18">
            35 658 142
        </div>
        <div id="DIV_19">
            Ощадбанк UAH
        </div>
    </div>
    <div id="DIV_20">
        <div id="DIV_21">
            
        </div>
        <div id="DIV_23">
            35 658 142
        </div>
        <div id="DIV_24">
            Крипто USDT
        </div>
    </div>
    <div id="DIV_25">
        <div id="DIV_26">

```

```

                                </div>
                                <div id="DIV_28">
                                    35 658 142
                                </div>
                                <div id="DIV_29">
                                    Крипто UAH
                                </div>
                            </div>
                            <div id="DIV_30">
                                <div id="DIV_31">
                                    

                                    </div>
                                    <div id="DIV_33">
                                        35 658 142
                                    </div>
                                    <div id="DIV_34">
                                        Вайз UAH
                                    </div>
                                </div>
                            </div>
                            <div id="DIV_35">
                                <a href="https://instachange.ml/reserves"      id="A_36">Показати
більше</a>
                            </div>
                        </div>
                    </div>
                    <div id="DIV_1">
                        <div id="DIV_2">
                            <div id="DIV_3">
                                Резерви валют
                            </div>
                            <div id="DIV_4">
                                <div id="DIV_5">
                                    <div id="DIV_6">
                                        

                                        </div>
                                        <div id="DIV_8">
                                            35 658 142
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```



```

        <div id="DIV_9">
            Револют UAH
        </div>
    </div>
    <div id="DIV_10">
        <div id="DIV_11">
            
        </div>
        <div id="DIV_13">
            35 658 142
        </div>
        <div id="DIV_14">
            Монобанк UAH
        </div>
    </div>
    <div id="DIV_15">
        <div id="DIV_16">
            
        </div>
        <div id="DIV_18">
            35 658 142
        </div>
        <div id="DIV_19">
            Крістал UAH
        </div>
    </div>
    <div id="DIV_20">
        <div id="DIV_21">
            
        </div>
        <div id="DIV_23">
            35 658 142
        </div>
        <div id="DIV_24">
            ІЗібанк UAH
        </div>
    </div>
    <div id="DIV_25">
        <div id="DIV_26">

```

```

                                
                                </div>
                                <div id="DIV_28">
                                    35 658 142
                                </div>
                                <div id="DIV_29">
                                    Абанк UAH
                                </div>
                            </div>
                            <div id="DIV_30">
                                <div id="DIV_31">
                                    
                                </div>
                                <div id="DIV_33">
                                    35 658 142
                                </div>
                                <div id="DIV_34">
                                    Приват UAH
                                </div>
                            </div>
                        </div>
                    </div>
                    <div id="DIV_35">
                        <a href="https://instachange.ml/reserves"      id="A_36">Показати
більше</a>
                    </div>
                </div>
<div className="loginPage__body">
    {signIn ? (
        <SignUpPage />
    ) : (
        <>
            <h1>Unlimited films, TV programmes and more.</h1>
            <h2>Watch anywhere. Cancel at any time.</h2>
            <h3>
                Ready to watch? Enter your email to create or
restart your
                membership.
            </h3>
            <div className="loginPage__input">
                <form>

```

```

        <input type="email" placeholder="Email Address"
/>

        <button
          className="loginPage__getStarted"
          onClick={() => setSignIn(true)}
        >
          GET STARTED
        </button>
      </form>
    </div>
  </>
  })
</div>
</div>
);
};

```

```

export default Loginpage;
import React from "react";
import "normalize.css";
import "./Homepage.css";
import HomeNav from "../../components/HomeNav/HomeNav";
import Banner from "../../components/Banner/Banner";
import Row from "../../components/Row/Row";
import requests from "../../config/requests/requests";

```

```

const Homepage = () => {

```

```

  return (
    <div className="homePage">
      <HomeNav />

      <Banner />
      <Row fetchUrl={requests.fetchTrending} />
    </div>
  );
};

```

```

export default Homepage;
import React from 'react';
import { useSelector } from "react-redux";
import { useHistory } from "react-router-dom";
import "./LoginPage.css";
import Nav from "../../components/Nav/Nav";

```

```

const ListPage = () => {
  const user = useSelector((state) => state.user.user);
  const history = useHistory();
  const movies = user?.movie;
  const showMovies = (movies) => {
    let shows = [];
    for(let el in movies) {
      shows.push(movies[el].movie);
    }
    console.log(shows);
    return shows;
  };

  return (
    <div className="listPage">
      <Nav />
      <h2>My list</h2>
      <div className="liked">
        {showMovies(movies).map(movie => (
          <img
            onClick={() => history.push("/movie/" +
movie.id)}

            className="list__poster"
            key={movie.id}
            src={
              movie?.image?.original
                ? movie?.image?.original
                : movie?.show?.image?.medium
            }
            alt={movie.name}
          />
        ))}
      </div>
    </div>
  );
}

export default ListPage;
import React, { useState, useEffect } from "react";
import "./MoviePage.css";
import axios from "../../config/axios/axios";
import Nav from "../../components/Nav/Nav";
import { useDispatch } from "react-redux";

```

```

import requests from "../../config/requests/requests";
import { liked, removeMovie } from "../../features/userSlice";

const MoviePage = () => {
  const url = window.location.href;
  const ID = url.substr(url.lastIndexOf("/") + 1);
  const [movies, setMovies] = useState([]);
  const dispatch = useDispatch();
  const add = document.querySelector(".add");
  const move = document.querySelector(".move");

  useEffect(() => {
    async function fetchData() {
      const request = await axios.get(requests.fetchTrending);
      setMovies(request.data);
      return request;
    }
    fetchData();
  }, []);

  const exactMovie = [movies.filter((movie) =>
movie?.id.toString() === ID)];

  let description = exactMovie[0][0]?.summary;
  if (description) {
    description = description.replace(/<\/?[a-zA-Z]+>/gi, "");
  }

  function truncate(description, n) {
    return description?.length > n ? description.substr(0, n - 1)
+ "..." : description;
  }

  function handleAdd(event){
    event.preventDefault();
    if(add){
      add.setAttribute("hidden", "hidden");
      move.removeAttribute("hidden", "hidden");
    }
  }

  function handleRemove(event){
    event.preventDefault();
    if(move){

```

```

        move.setAttribute("hidden", "hidden");
        add.removeAttribute("hidden", "hidden")
    }
}

return (
    <div className="moviePage">
        <Nav />
        <div className="movie__container">
            <div className="movie__img">
                <img src={exactMovie[0][0]?.image?.original} alt="" />
            </div>
            <div className="movie__info">
                <h1
className="movie__name">{exactMovie[0][0]?.name}</h1>
                <h2 className="movie__genres">
                    {exactMovie[0][0]?.genres.join(" ")}
                </h2>
                <div className="movie__summary">{truncate(description,
500)}</div>
                <button className="movie__button">
                    <a href={exactMovie[0][0]?.url}>
                        Play
                    </a>
                </button>
                <button
                    onMouseDown={() => {
                        dispatch(
                            liked({
                                movie: exactMovie[0][0]
                            })
                        )
                    }}
                    onClick={handleAdd}
                    className="movie__button add">My list</button>
                <button
                    hidden
                    onMouseDown={() => {
</div>
</div>
<div id="DIV_1">
    <label for="email" id="LABEL_2">
        <span id="SPAN_3">E-mail</span>
    </label>
    <input type="email" name="email" placeholder="E-mail" id="INPUT_4" />

```

Додаток В (обов'язковий)

Рисунок 9.1 – Схема моделі інформаційної системи криміналістики

ІЛЮСТРАТИВНА ЧАСТИНА
ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ СТРИМІНГОВОГО СЕРВІСУ ДЛЯ
ПЕРЕГЛЯДУ ФІЛЬМІВ

Виконав: студент 2-го курсу, групи 2КН-22м
спеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

(В)

Сліпенського В.О.

(прізвище та ініціали)

Керівник: доцент каф. КН

Сілагін О.В.

(прізвище та ініціали)

«04»

12

2023 р.

Рисунок 9.2 – Схема моделі інформаційної системи криміналістики

Вінниця ВНТУ - 2023 рік

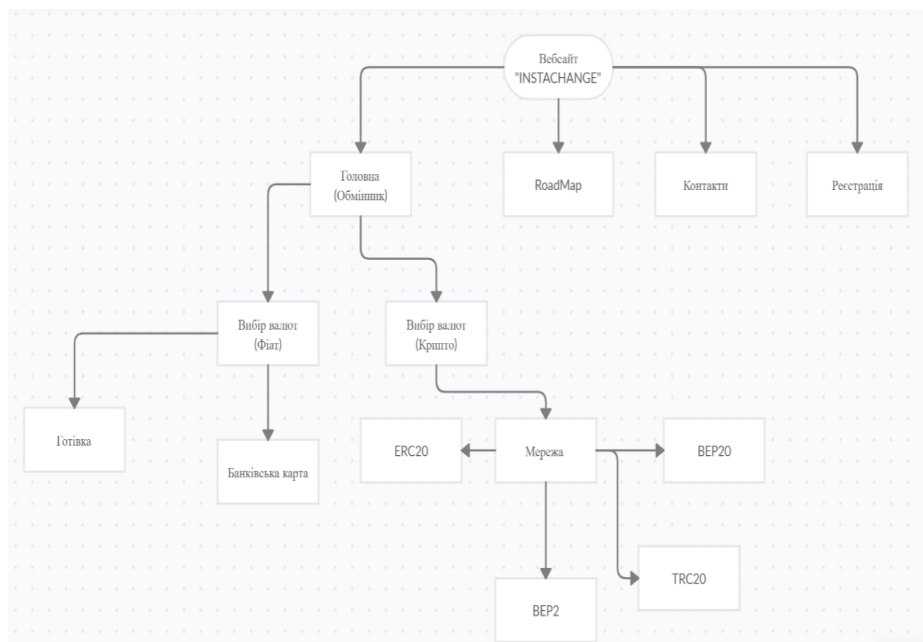


Рисунок В.1 – Схема моделі інформаційної системи криптообмінника

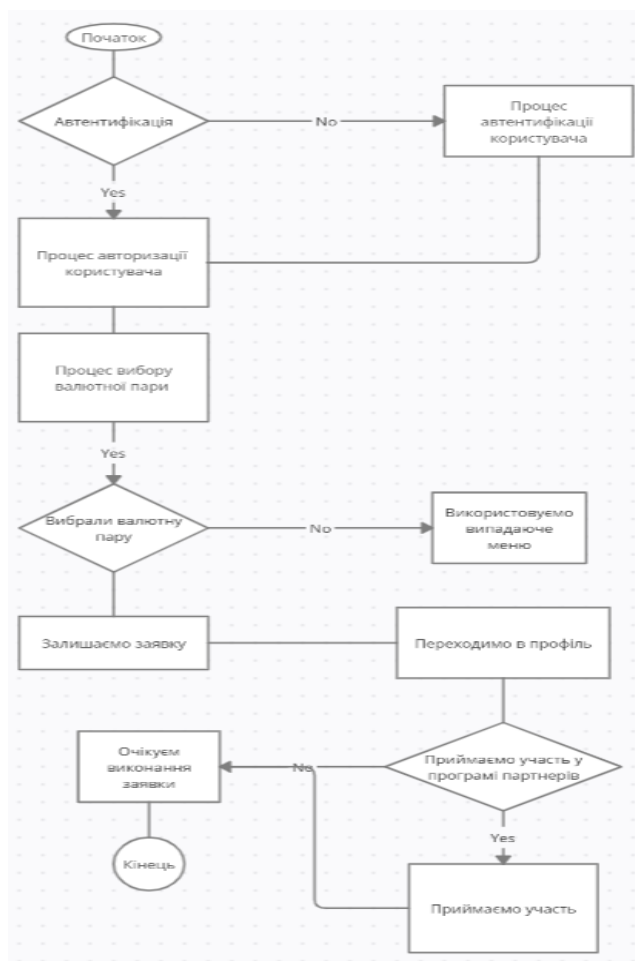


Рисунок В.2 – Схема моделі вимог до інформаційної системи криптообмінника

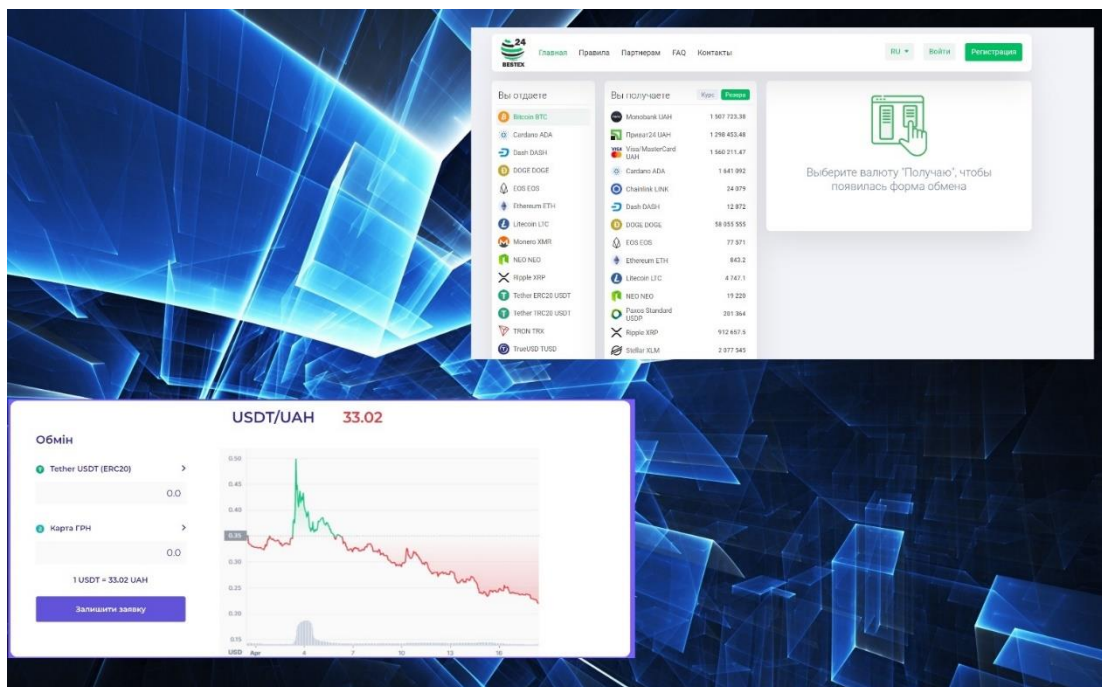


Рисунок В.3 – Порівняння систем-аналогів

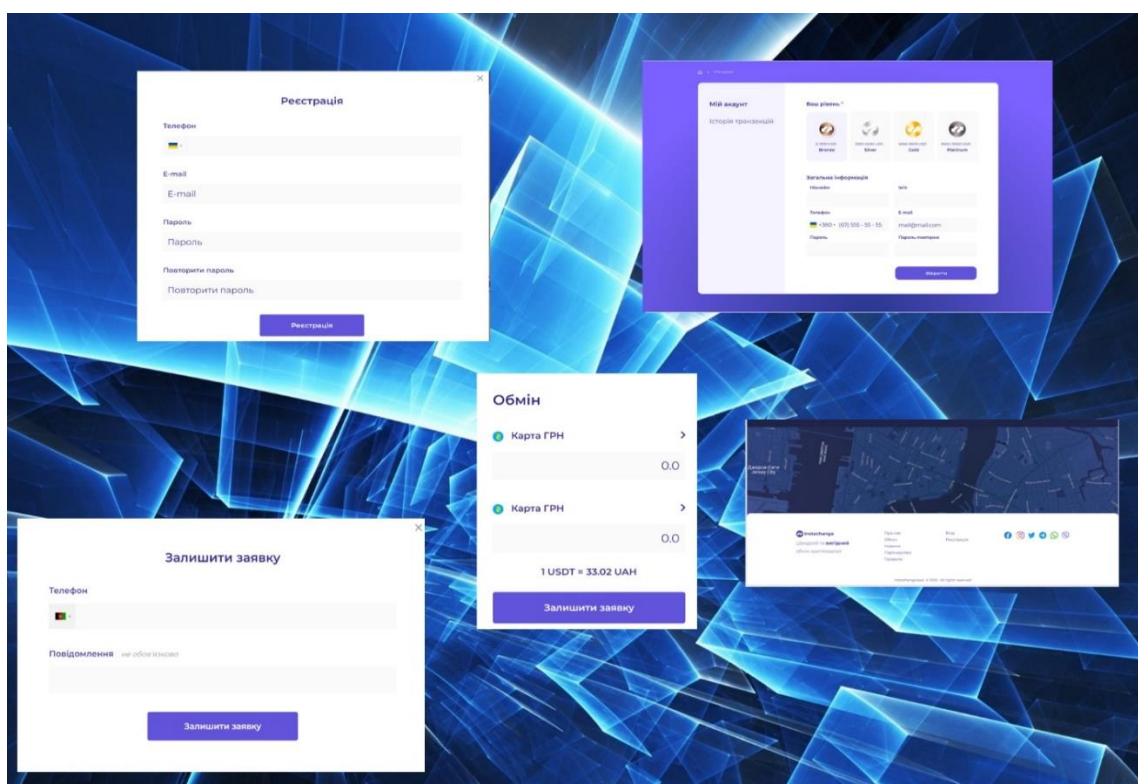


Рисунок В.4 – Приклад роботи криптообмінника

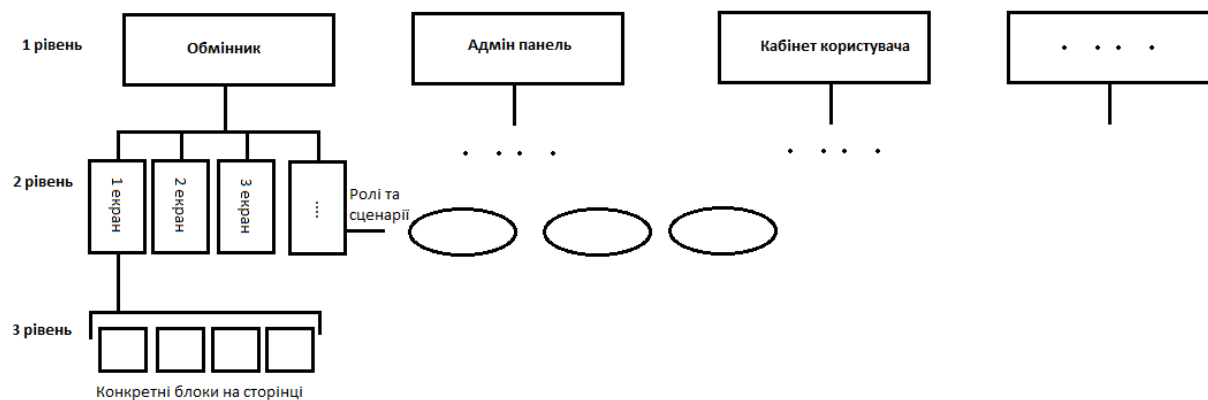


Рисунок В.5 – Рівні декомпозиції обмінника



Рисунок В.6 – Складові інтерфейсу користувача

Додаток Г(довідниковий)

Інструкція користувача

Для початку роботи потрібен доступ до мережі інтернет.



Рисунок Г.1 – Головне вікно обмінника

Після цього можна або авторизуватися, або пройти автентифікацію. Для того щоб пройти автентифікацію, треба ввести номер у полі Телефон, для авторизацію потрібно натиснути кнопку увійти.

The image shows a registration window titled "Реєстрація" (Registration) on the instachange website. The window is centered on a dark grey background. On the left side of the background, the instachange logo is visible with the tagline "Швидкий та вигідний обмін криптовалют" (Fast and profitable cryptocurrency exchange). On the right side, there are social media icons for Telegram, Twitter, WhatsApp, and Messenger. The registration form itself is white and contains the following fields:

- Телефон** (Phone): A field with a dropdown menu showing the Ukrainian flag.
- E-mail**: A text input field.
- Пароль** (Password): A text input field.
- Повторити пароль** (Repeat password): A text input field.

 At the bottom of the form is a blue button labeled "Реєстрація".

Рисунок Г.2 – Вікно авторизації

При успішній авторизації ви побачите сторінку вашого профілю.

The image shows the user profile page on the instachange website. The page has a dark blue header with a home icon and the text "Мій акаунт" (My account). The main content area is white and features a sidebar on the left with "Мій акаунт" and "Історія транзакцій" (Transaction history). The main section is titled "Ваш рівень" (Your level) and displays four membership tiers:

- Bronze**: 0-1500 USD
- Silver**: 2500-5000 USD
- Gold**: 5000-9500 USD
- Platinum**: 9500-15000 USD

 Below this is the "Загальна інформація" (General information) section, which includes a table of user details:

Загальна інформація	
Нікнейм	Ім'я
Телефон	E-mail
Пароль	Пароль повторно

 The phone number is shown as "+380 (67) 555 - 55 - 55" and the email as "mail@mail.com". At the bottom right of the profile section is a blue button labeled "Зберегти" (Save).

Рисунок Г.3 – Сторінка профілю

Після цього ви можете переглянути сторінку з історією транзакцій.

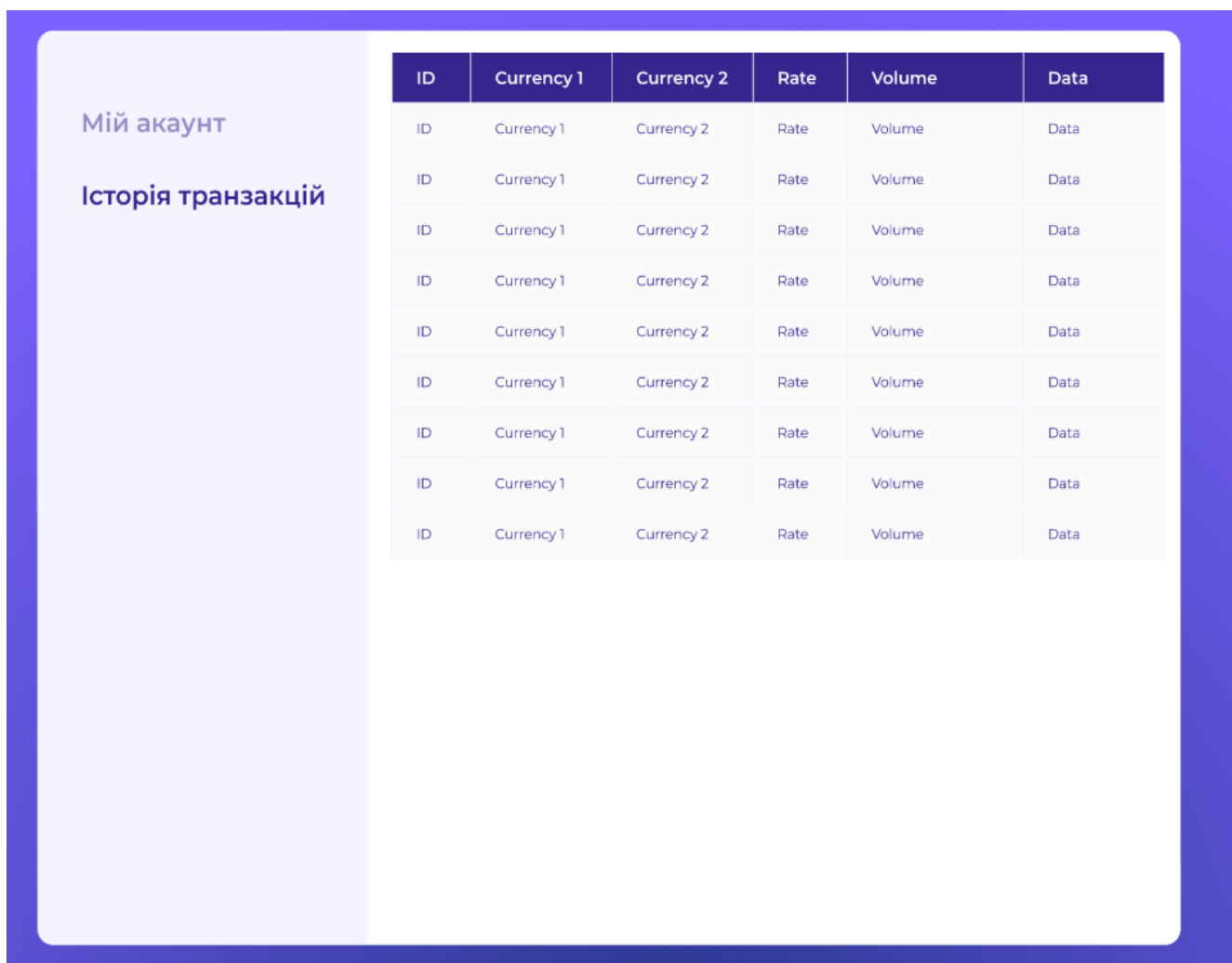


Рисунок Г.4 – Сторінка історії транзакцій

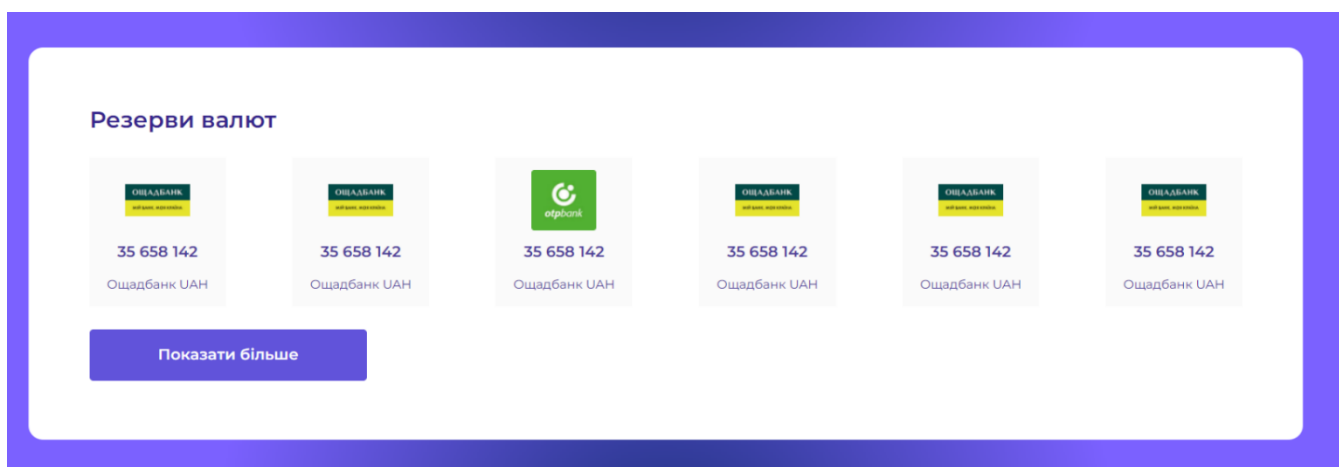


Рисунок Г.5 – Сторінка резервів валют

Обравши оплату готівкою користувач може знайти на карті всі наші відділення.

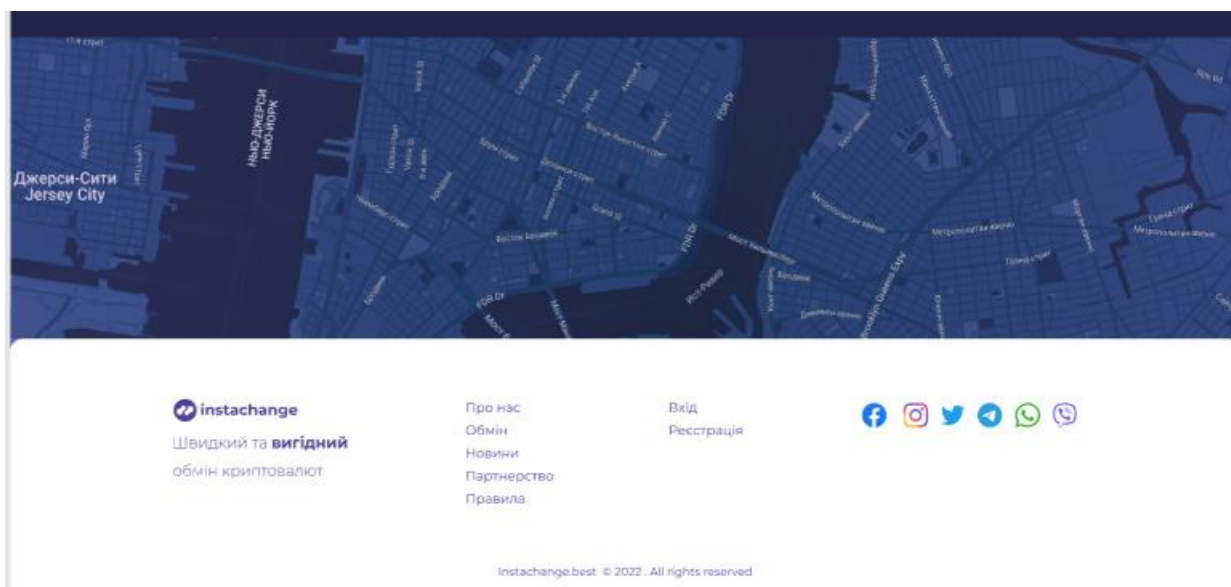


Рисунок Г.6 – Сторінка карти відділень