

Міністерство освіти і науки
України Університет митної справи
та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного забезпечення

Кваліфікаційна робота бакалавра

на тему Розробка телеграм боту для обміну криптовалют

Виконав: студент групи ІПЗ20-1

Спеціальність 121 «Інженерія Програмного
Забезпечення»

Ізварін Михайло Ігорович

(прізвище та ініціали)

Керівник кфмн. Фірсов О.Д.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент

(місце роботи)

(посада)

(науковий ступінь, вчене звання, прізвище та ініціали)

Дніпро – 2024

АНОТАЦІЯ

Ізварін М.І. Розробка телеграм боту для обміну криптовалют

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за

спеціальністю 121 «Інженерія Програмного Забезпечення» – Університет митної справи та фінансів, Дніпро, 2024.

Дана кваліфікаційна робота присвячена створенню телеграм боту для автоматизації процесу обміну криптовалют. У роботі розглянуті методи та технічні рішення, необхідні для розробки системи, сформульовані вимоги до програмного забезпечення. Було проведено проектування та розробку боту, а також його тестування. Отримані результати мають практичне значення, оскільки сприяють зручності та швидкості обміну криптовалют, забезпечуючи безпеку операцій. Розроблений бот є надійним та ефективним, використовуючи передові технології, такі як алгоритми обробки даних та захисту інформації, та є важливим кроком у покращенні процесів обміну криптовалют.

Розроблений бот має клієнтську частину, яка дозволяє користувачам здійснювати обмін криптовалют, та серверну частину, яка обробляє дані та надає відповіді. Система отримує дані про курси криптовалют від клієнтів та аналізує їх на сервері. Згідно з цими даними бот виконує операції обміну. Додаток має сучасний та зручний інтерфейс для користувачів.

Крім того, бот забезпечує високу масштабованість, що дозволяє легко додавати нові функції та підтримку інших криптовалют у майбутньому. Це робить систему гнучкою і готовою до розвитку разом з потребами ринку. Бот також пропонує інтеграцію з зовнішніми сервісами для отримання аналітики та прогнозів, що допомагає користувачам приймати більш обґрунтовані рішення щодо обміну. Така функціональність значно підвищує цінність бота, роблячи його корисним інструментом для широкого кола користувачів у світі криптовалют.

Ключові слова: телеграм бот, обмін криптовалют, автоматизація, безпека, технології, швидкість, ефективність, програмне забезпечення, тестування, клієнтська та серверна частини, аналіз даних.

ABSTRACT

Izvarin M.I. Development of a telegram bot for cryptocurrency exchange
Qualification work for obtaining a bachelor's degree in
specialty 121 “Software Engineering” - University of Customs and Finance, Dnipro,
2024.

This qualification work is devoted to the creation of a telegram bot to automate the process of cryptocurrency exchange. The work considers the methods and technical solutions necessary for the development of the system, formulates the requirements for the software. The bot was designed and developed, as well as tested. The results obtained are of practical importance, as they contribute to the convenience and speed of cryptocurrency exchange, ensuring the security of operations. The developed bot is reliable and efficient, using advanced technologies such as data processing and information security algorithms, and is an important step in improving cryptocurrency exchange processes.

The developed bot has a client part that allows users to exchange cryptocurrencies and a server part that processes data and provides answers. The system receives data on cryptocurrency rates from clients and analyzes it on the server. Based on this data, the bot performs exchange operations. The application has a modern and user-friendly interface.

In addition, the bot provides high scalability, which makes it easy to add new features and support for other cryptocurrencies in the future. This makes the system flexible and ready to evolve along with market needs. The bot also offers integration with external services to obtain analytics and forecasts, which helps users make more informed decisions about exchanges. This functionality significantly increases the value of the bot, making it a useful tool for a wide range of users in the cryptocurrency world.

Keywords: telegram bot, cryptocurrency exchange, automation, security, technology, speed, efficiency, software, testing, client and server parts, data analysis.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1	6
ТЕХНОЛОГІЯ БЛОКЧЕЙН ТА ТЕОРЕТИЧНІ АСПЕКТИ КРИПТОВАЛЮТИ	6
1.1 Теоретичні аспекти криптовалюти.....	6
1.2 Криптовалюта та її потенційні можливості на світовому ринку	8
1.3 Основи безпеки та вразливості криптовалют.....	10
1.4 Огляд і аналіз існуючих бірж для криптовалют	14
1.5 Огляд існуючих причин та способів обміну криптовалюти.....	18
1.6 Постановка задачі.....	21
1.7 Висновки до розділу 1	21
РОЗДІЛ 2	23
ТЕХНОЛОГІЇ СТВОРЕННЯ ТЕЛЕГРАМ БОТІВ	23
2.1. Обмін криптовалюти в Telegram боті	24
2.2 Використання Python у створенні Telegram бота	26
2.3 Огляд API використаного у програмі	28
2.3.1. Приклади використання API TronGrid	29
2.4 Бібліотеки TronPy та Aiogram.....	31
2.4.1. Інтеграція та взаємодія	32
2.5. Висновки до розділу 2	32
РОЗДІЛ 3	35
ПРОГРАМНИЙ ДОДАТОК	35
3.1. Бізнес-планування	35
3.2 Аналіз технічних вимог	36
3.3. Проектування та реалізація програмного забезпечення	37
3.3.1. API TronGrid	45
3.4 Експлуатація та супроводження. Інструкція користувача.....	48
3.5 Висновки до розділу 3.	55
ВИСНОВОК.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58

ВСТУП

Актуальність теми дослідження. У сучасному світі криптовалюти набули значної популярності та впливу на фінансові ринки. З кожним днем все більше людей, проектів, організацій та країн долучаються до використання цифрових активів для зберігання, платежів та інвестицій. У зв'язку з цим, зростає попит на ефективний, швидкий та безпечний спосіб обміну криптовалюти, що забезпечить користувачам різних масштабів, зручність і надійність у проведенні операцій. Такий проект має бути інтуїтивно зрозумілим, з простим інтерфейсом, щоб навіть новий користувач, який вчора дізнався що таке криптовалюта, міг за допомогою бота розібратися, що де і куди. Люди розуміють, що криптовалюта це найшвидший спосіб входу до сфери діяльності інвесторів, бо купуючи валюту ви вже інвестуєте в проект, інші просто бачать у цьому спосіб заробітку, вкладаючи гроші у певну монету та чекають поки вона дасть ріст від ціни покупки, є ще ті хто просто зберігає свої кошти на крипто гаманці, бо немає довіри до стандартної системи банкінгу, а може комусь потрібно замести сліди прибутку грошей і тому вони переводять їх у криптовалюту. Все це причини актуальності, які об'єднують інтереси людей і підбадьорюють інших входити до цієї сфери діяльності.

Мета та завдання дослідження. Метою даного дослідження є розробка автоматизованого обмінника криптовалют на платформі Telegram, що забезпечить безпечний, швидкий та зручний обмін цифрових активів для користувачів. Цей обмінник повинен мати високу ступінь автоматизації. Для досягнення поставленої мети потрібно поставити і виконати низку завдань:

- 1) дослідити особливості роботи криптовалют для переказів між гаманцями;
- 2) дати аналіз безпечності використання криптовалют та блокчейн технологій;
- 3) визначити параметри що впливають на необхідність криптовалют;

- 4) дослідити вже існуючі методи купівлі і обміну криптовалют на популярних біржах;
- 5) розібрати інструментарій який вже існує для створення автоматизованого обмінника криптовалют;
- 6) методом порівняння проєктів конкурентів, визначити переваги та недоліки;
- 7) вибрати мову програмування та платформу які краще підійдуть для виконання завдання;
- 8) продумати функціонал на реалізувати його в коді програми;
- 9) визначити напрямки покращення створеної моделі та майбутні напрямки використання.

Об'єкт та предмет дослідження. Об'єктом дослідження даної кваліфікаційної роботи виступає розробка автоматизованого обмінника криптовалют, а предметом - особливості створення та використання даної системи в реальних умовах.

Методи дослідження. При виконанні завдання використовувалися методи аналізу теоретичних аспектів та існуючих рішень, методи проектування, програмування та розробки, методи впровадження та експлуатації.

Практичне значення отриманих результатів. Практичне значення отриманих результатів полягають в розробці бота який автоматизує роботу обмінника для обміну криптовалют, що допоможе підвищити ефективність даної сфери діяльності.

Структура роботи. Структуру роботи складають вступ, три розділи, висновок та список використаних джерел для інформації.

РОЗДІЛ 1

ТЕХНОЛОГІЯ БЛОКЧЕЙН ТА ТЕОРЕТИЧНІ АСПЕКТИ КРИПТОВАЛЮТИ

1.1 Теоретичні аспекти криптовалюти

У сучасному світі криптовалюта відіграє ключову роль у фінансовій сфері, забезпечуючи альтернативний механізм обміну вартості та засіб збереження капіталу. Теоретичний аналіз криптовалют включає вивчення їх природи, принципів функціонування та потенційних переваг та ризиків[1].

Криптовалюта – це віртуальний актив, що використовує криптографічні техніки для забезпечення безпечних транзакцій, для контролю над створенням нових або подібних одиниць та для підтверджень власності. Однією з головних характеристик криптовалют є їх децентралізована природа, що означає відсутність централізованого органу контролю та регулювання, як у випадку традиційних фіатних валют.

Принципи функціонування криптовалют засновані на технологіях блокчейну, що розуміє під собою розподілену базу даних, в якій інформація про транзакції зберігається у блоках, які зв'язані між собою за допомогою криптографічних хешів. Саме така технологія забезпечує максимальну безпеку та надійність транзакцій, а також унікальність та невідмінність історії операцій.

Потенційні переваги криптовалют включають швидкість та ефективність переказу коштів, низькі комісійні витрати, анонімність та конфіденційність операцій, а також можливість здійснення міжнародних та незвичайних транзакцій без необхідності посередництва банків чи фінансових установ.

Проте разом з перевагами криптовалют приходять і ризики, серед яких варто виділити високу волатильність цін, можливість використання для нелегальних цілей, ризик кібератак та втрати доступу до коштів. Тому перед використанням криптовалют необхідно детально вивчити їхню природу та ризики[1].

До важливих теоретичних аспектів криптовалюти можна віднести розгляд самої екосистеми криптовалюти, вона складається з таких понять як стейблкоїни та альткоїни.

Стейблкоїни та альткоїни - це дві різні категорії криптовалют, які відіграють важливу роль у цифровій екосистемі, але відрізняються за призначенням, технологіями та властивостями.

Давайте розглянемо що означають поняття стейбл та альткоїни:

- Стейблкоїни - це криптовалюти, які створені з метою мінімізації волатильності їхньої вартості шляхом прив'язки до стабільних активів, таких як фіатні валюти (наприклад, долар США, євро) або товари (наприклад, золото). Це робить їх більш придатними для використання у щоденних фінансових операціях, таких як платежі та зберігання вартості. Деякі з найвідоміших стейблкоїнів включають Tether (USDT), USD Coin (USDC), TrueUSD (TUSD) та Dai (DAI).

- Альткоїни (альтернативні криптовалюти) - це всі криптовалюти, що не є біткоїном. Вони представляють собою широкий спектр цифрових активів, які були створені після біткоїна з різними технологічними рішеннями, цілями та властивостями. Наприклад, Ефір (ETH) є другим за розміром криптовалютним активом за ринковою капіталізацією та є основою для розробки різних децентралізованих додатків та смарт-контрактів на блокчейні Ethereum. Інші альткоїни включають Ripple (XRP), Litecoin (LTC), Cardano (ADA) та багато інших.

Біткоїн є першою та найбільш відомою криптовалютою, що була створена в 2008 році під псевдонімом Сатоші Накамото[4]. Ця криптовалюта базується на технології блокчейн і має ряд унікальних властивостей, які роблять її важливим активом на фінансовому ринку.

Роль і значення біткоїну для ринку можна розглянути з декількох аспектів:

1. Піонерство: Біткоїн виступає піонером у світі криптовалют та блокчейн технологій. Він встановив перший прецедент для створення та використання цифрових активів, що відкрило двері для розвитку сотень інших криптовалют та блокчейн проектів.

2. Децентралізація: Біткоїн є децентралізованим активом, що означає, що немає централізованого органу контролю або влади над ним. Це надає йому рівень безпеки та незалежності, що приваблює користувачів, які шукають альтернативу традиційним фіатним валютам.

3. Зберігання вартості: Біткоїн вважається цифровим засобом зберігання вартості, подібним до золота. Його обмежена випускність (лише 21 мільйон біткоїнів може бути видобуто) та природна волатильність роблять його привабливим для інвесторів як інструмент для диверсифікації портфеля та захисту від інфляції.

4. Технологічний розвиток: Розвиток Біткоїну стимулює інновації в галузі фінансів та технологій. Багато компаній та установ починають дослідження та розвивають продукти та послуги на основі блокчейн технології, що може привести до змін у фінансовому секторі та суспільстві в цілому.

В цілому, біткоїн відіграє ключову роль у розвитку криптовалют та блокчейн технологій, і його вплив на ринок має колосальні масштаби.

1.2 Криптовалюта та її потенційні можливості на світовому ринку

Стан криптовалют на світовому ринку визначається різноманітними факторами, які включають технічні, економічні, політичні та соціальні аспекти. Ось деякі з найважливіших факторів, які мають вплив на стан:

1. Попит та пропозиція: Велика різниця в попиті та пропозиції на ринку можуть значно впливати на ціну криптовалют. Великий попит може призвести до зростання ціни, тоді як велика пропозиція може спричинити зниження ціни.

2. Регуляторна діяльність: Рішення урядових органів щодо регулювання криптовалют можуть впливати на їхню ціну та стабільність. Наприклад, нові законодавчі ініціативи або заборони можуть викликати велику волатильність на ринку.

3. Технічні фактори: Технічні аспекти, такі як оновлення програмного забезпечення, проблеми масштабованості, кібератаки та інші технічні проблеми, можуть впливати на стан та ціну криптовалют.

4. Глобальні події: Геополітичні події, економічні кризи, великі події в сфері технологій та інші фактори змінюють настрої ринку та можуть впливати на криптовалютні ціни.

5. Сприйняття громадськістю: Усвідомлення та відношення громадськості до криптовалют може впливати на їхню популярність та ціну. Новини, події та соціальні медіа можуть впливати на загальний настрій ринку.

Ці фактори є лише деякими з багатьох, які впливають на стан криптовалют на світовому ринку. Важливо розуміти, що цей ринок дуже динамічний та піддається впливу багатьох різних чинників[5].

Регіональні фактори також дуже важливі, бо в різних країнах різний рівень дозволеності криптовалютної діяльності на території країни. Деяка вона зовсім заборонена, а деяка дозволено все що тільки можливо, для таких прикладів візьмемо до розгляду дві країни такі як Сполучені Штати Америки та Китай. Наприклад в США мають розвинену криптовалютну індустрію і вважаються одним з кращих місць для розвитку проектів у цій галузі. Хоча в країні існують регуляторні вимоги, які регулюють діяльність криптовалютних компаній, криптовалюти в цілому легальні і прийняті у більшості сфер життя. А ось наприклад в Китаї прийняли досить суворий підхід до криптовалют, забороняючи торгівлю криптовалютами та ICO (Initial Coin Offerings) у вересні 2017 року. Однак, використання блокчейн технології у різних галузях залишається допустимим.

Японія, наприклад, ще одна країна де прийняли більш прогресивний підхід до такого явища як криптовалюта, у країні криптовалюти визнаються законним засобом платежу, а також регулюються законодавством, що встановлює стандарти для криптовалютних бірж та фінансових установ, що працюють з криптовалютами. Іншим прикладом стане Індія, яка ще не визначила криптовалюту чіткою регуляцією, уряд висловив занепокоєння стосовно ризиків, пов'язаних з криптовалютами, але також розглядає можливості розвитку блокчейн технології. Наразі у країні ведуться дискусії щодо визнання криптовалют та розробки відповідного законодавства[2].

Ці приклади демонструють різноманітність підходів до регулювання криптовалют у різних країнах та важливість розуміння їхнього впливу на розвиток цього сектору.

Стан ринку криптовалют є динамічним і піддається значним змінам. За останні роки спостерігається зростання інтересу до криптовалют, збільшення кількості проєктів та інвестицій, а також впровадження блокчейн технологій у різні сфери бізнесу та громадського життя криптовалютний ринок має величезний потенціал для прогресу та винахідливості. Він може сприяти технологічному прогресу та сприяти фінансовій інклюзії. Незважаючи на це, великі проблеми, такі як регуляторність, волатильність і стійкість до кібератак, супроводжують потенціал.

Криптовалютний ринок це дуже складне та динамічне середовище, воно тільки зі сторони здається легким і зрозумілим, багато новачків вважають що це легкий ринок для інвестицій і зберігання, але не розуміння і велика волатильність ринку криптовалют дають зрозуміти, що без ефективного управління та грамотного розподілу фінансів і можливостей та і в цілому розуміння складових, ринок криптовалют стає ямою в якій люди втрачають свої фінанси. Розвиток ринку буде залежати від взаємодії між технологічними інноваціями, регуляторними органами влади та усвідомленим інвестиційним кліматом.

1.3 Основи безпеки та вразливості криптовалют

Блокчейн[8] - це технологія, що дозволяє зберігати та передавати дані у вигляді послідовності зв'язаних блоків. Кожен блок містить інформацію та посилання на попередній - разом вони формують ланцюжок. Таким чином, дані в блокчейні захищені від змін та фальсифікації. Простими словами це унікальні данні які неможливо фальсифікувати, вони завжди будуть мати один і лише оригінальний вигляд. У мережі блокчейн багато учасників, які співпрацюють між собою для обробки та підтвердження операцій. Кожен вузол активно бере участь у перевірці та додаванні нових блоків з цими операціями.

Блокчейн можна назвати базою даних, але не традиційною. Основним відмінністю між блокчейном і звичайною базою даних є їх структура та доступність.

Найважливіше, що вам потрібно знати про блокчейн, це те, що це технологія, яка використовується для передачі та зберігання даних через Інтернет. Таким чином, блокчейн є базою даних, розташованою по всьому світу, оскільки всі записи знаходяться на комп'ютерах кожного користувача. Тобто інформація зберігається в кількох місцях, а не в одному.

Деякі криптовалюти використовуються в своїх власних блокчейн-мережах, тоді як інші використовують уже існуючі блокчейни. Основна відмінність між ними:

- Монета — це термін, який використовується для опису криптовалюти, чия основна валюта — блокчейн. такі як Solana, Bitcoin і Ethereum.
- Токен — це слово, яке використовується для опису криптовалюти, яка розташована в існуючому блокчейні. Такі як, WhiteBIT Token (WBT), Maker і Uniswap[7].

Блокчейн не викликає сумнівів щодо своєї надійності у зберіганні даних. Однак не буде великим здивуванням що велика більшість компаній які працюють з криптовалютою, прагнуть розробити свої власні технології. Саме це свідчить про надійність та масштабність технологій блокчейну.

Однією з таких компаній є одна з найбільших європейських криптобірж з українськими коріннями - WhiteBIT. Вони запустили тестову мережу власного блокчейну. Тепер WB Network Testnet дозволяє випробувати їхню блокчейн-технологію в реальному часі, хоча без використання реальних цифрових активів[13].

Блокчейни захищені різноманітними механізмами, включаючи передові криптографічні методи та математичні моделі поведінки та прийняття рішень. Ця технологія є основою більшості криптовалютних систем і запобігає подвоєнню або знищенню цифрових активів.

Блокчейн створює мережу комп'ютерів, які працюють разом, щоб зберігати та підтверджувати транзакції. Кожен блокчейн складається з блоків, які містять дані про всі транзакції, які відбувалися в мережі.

Хеш-код попереднього блоку міститься в кожному новому блоку. Усі комп'ютери в мережі перевіряють і підтверджують новий блок, гарантує унікальність. Блокчейн створюється завдяки спільній роботі децентралізованих комп'ютерів, які знаходяться у мережі, де вони пов'язані.

Опис формування блокчейну можна побачити наступним чином (рис. 1.1.):

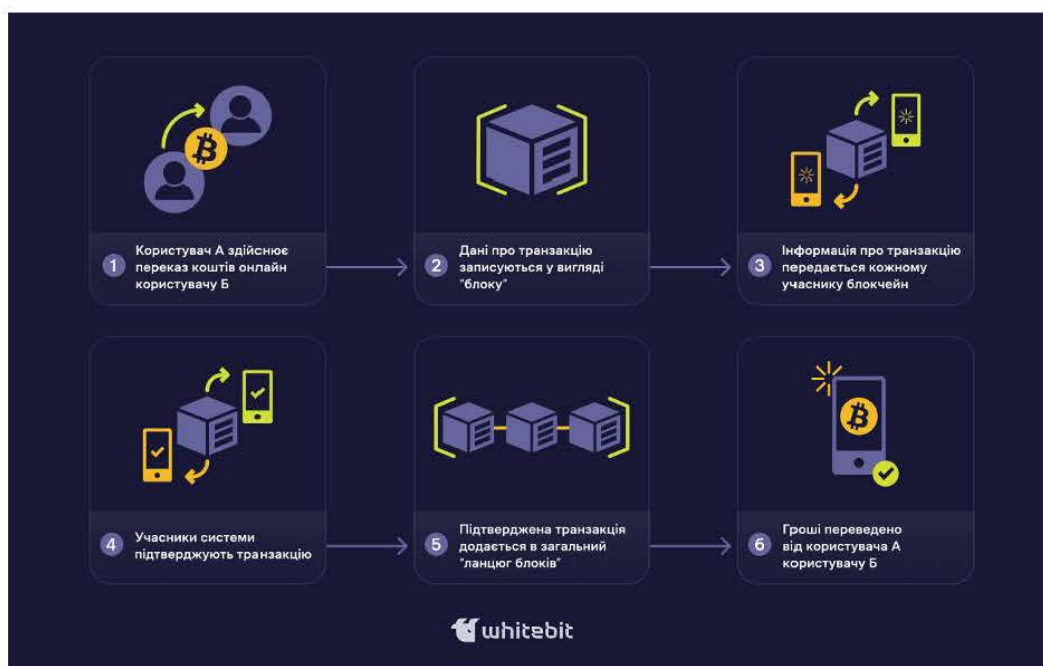


Рис. 1.1. Формування блокчейну (з сайту whitebit.com)

Розрізняються приватний, публічний і гібридний блокчейни.

- Приватний блокчейн зазвичай контролює одна компанія чи корпорація, тому він мало відрізняється від закритої корпоративної бази даних.
- Публічний блокчейн доступний для всіх бажаючих без обмежень у будь-який час.
- Прикладами груп, які можуть використовувати гібридний формат блокчейну, є приватні нотаріуси та агенції з продажу нерухомості, які прагнуть постійно мати доступ до даних, пов'язаних із певним сектором, але не хочуть ділитися цими даними зі сторонніми учасниками. Такий тип організації називається «блокчейн-консорціум»[8].

Розглянемо 4 основних видів захисту блокчейну від шахрайських махінацій та кібератак:

- 1) Запобігання шахрайству: Блокчейн перед додаванням транзакцій ретельно їх перевіряє через підтвердження з обох сторін обміну.
- 2) Захист особистих даних: Блокчейн шифрує данні, що допомагає їх захистити від кібератак та маніпуляцій.
- 3) Забезпечення прозорості: Блокчейн записує усі данні про транзакцію, роблячи цю інформацію прозорою, кожна зі сторін обміну може переглянути одну і ту саму інформацію.
- 4) Підвищення безпеки у ланцюгах постачання: Блокчейн підвищує безпеку ланцюгів постачання, надаючи прозорий та безпечний запис всіх операцій у ланцюгу постачання. Це забезпечує, що всі сторони, що беруть участь у ланцюгу постачання, мають доступ до однієї і тієї ж інформації, забезпечуючи прозорість та відповідальність.

Але є основна проблема того що ви як звичайна людина яка таки вирішила увійти у світ криптовалют, спочатку реєструється на біржі і вже на ній купуєте або обмінюєте кошти на криптовалюту, ці кошти будуть фіксуватися на біржі яка працює на технології блокчейну, але безпека блокчейну ніяк не впливає на безпеку самої біржі і якщо буде здійснений злом самої біржі, то ваші данні та фінанси будуть скоріш за все втрачені. Тобто це дві сторони медалі, це потрібно також розуміти[3].

Усі ці принципи безпеки блокчейну та його вразливості це все звісно добре, але за рахунок того що історію транзакцій в блокчейні неможна відслідкувати, у гру вступають люди які через криптовалюту відмивають чорні гроші, тобто у криптовалюту можуть поступати великі гроші через так само біржу, через обмінники, тим самим утворюючи нові блоки в блокчейні і все, ці гроші блокчейн запам'ятовує як ваші, і до них не має ніяких питань.

Насамперед саме такий фактор як не можливість відслідкувати транзакції і не можливість виникнення питань, як наприклад в банку «Звідки у вас з'явилися ці гроші» і надали криптовалюту таку популярність.

Далі іде небезпека зі сторони самого користувача, криптовалюта яку він вже купив, роздивився та розібрався, вистроїв план її росту наприклад, тобто коротше говорячи проінвестував в якусь монету і збирається її тримати якийсь період часу, та через деякий час він може забути свої логіни або паролі, або електронний код свого гаманця, тощо. Це вже більш людський фактор, але сам факт що якщо він забуває такі данні, то повернути їх уже буде неможливо, це потрібно також враховувати у ризики.

Також шахраї, у світі криптовалют це в основному великі команди людей, які працюють за наказом, або за для своєї вигоди, вони можуть навантажувати мережу на якій працює блокчейн, це може надати шахраям 50% від інформації усього блокчейну, але і у цей відділ інформації можуть потрапити коди наприклад ваших гаманців. З іншого боку, навантаження мережі може затримати переведення транзакцій, до збільшення часу, або навіть їх перехоплення[1].

Отже, значення безпеки блокчейну в сучасному світі та в майбутньому є беззаперечним. Проте, забезпечення безпеки блокчейну - це непроста задача, яка потребує рішення для забезпечення надійності та захисту блокчейн-проектів.

1.4 Огляд і аналіз існуючих бірж для криптовалют

Популярні біржі криптовалют у світі та Україні мають велике значення для трейдерів та інвесторів, надаючи їм можливість купувати, продавати та обмінювати різні цифрові активи. Давайτε спочатку зрозуміємо що бірж існує дуже велика кількість, але як люди вибирають ту або іншу біржу і чому, зараз розглянемо головні аспекти вибору біржі:

- Вартість торгів і біржові комісії: Важливо розглянути кількість комісій, які вимагаються за торгові операції на кожній біржі, і загальні витрати на торгівлю. Умови на деяких біржах можуть бути більш вигідними для певних типів операцій або обсягів торгів.
- Рівень безпеки та регулювання: Кожна біржа використовує заходи безпеки, такі як двофакторна аутентифікація, зберігання грошей у холодних гаманцях

тощо. Крім того, слід взяти до уваги ступінь регулювання, який впливає на безпеку інвесторів.

- Клієнтська підтримка та репутація: Важливо, щоб клієнтська підтримка була якісною та доступною на кожній біржі. Крім того, досвідність і репутація біржі є важливими, оскільки вони вказують на надійність біржі та її відповідальність перед користувачами.
- Функціональні можливості та інтерфейс: деякі біржі можуть мати функціональні можливості, такі як ордери зупинки, маржинальна торгівля, ф'ючерси тощо. Оцініть також зручність використання інтерфейсу користувача та доступність мобільного додатку для торгівлі в дорозі.
- Регіональні обмеження та доступність мов: деякі біржі можуть встановлювати обмеження щодо реєстрації користувачів з певних країн або обмеження щодо використання певних мов. Важливо переконатися, що біржа доступна для вашого району та пропонує мовну підтримку.

Такі аспекти допоможуть як новим так і старим користувачам вибрати для себе найвигідніші біржі для торгів своїми електронними активами.

Розглянемо популярні світові біржі, які вже зробили собі ім'я на світовому ринку криптовалют, та мають найбільші обсяги користувачів:

- Binance:
 - Світовий рівень: Binance - одна з найбільших та найпопулярніших криптобірж у світі. Вона пропонує широкий вибір торгових пар та продуктів, включаючи ф'ючерси та опціони.
 - Україна: Хоча Binance не має прямої підтримки для гривні, українські користувачі можуть використовувати послуги цієї біржі через криптокошельки або платіжні системи.
- Coinbase:
 - Світовий рівень: Coinbase - одна з найбільших та найбільш відомих криптобірж, особливо серед новачків.

- Україна: Українські користувачі можуть використовувати послуги Coinbase для купівлі, продажу та зберігання криптовалют через карткові операції або банківські перекази.

- Kraken:

- Світовий рівень: Kraken відома своєю великою кількістю доступних криптовалютних пар та продуктами для інституційних інвесторів.

- Україна: Українські користувачі можуть торгувати на Kraken, використовуючи різні методи оплати, такі як банківські перекази або криптовалютні депозити.

- Bybit:

- Світовий рівень: прямий конкурент біржі Binance, бо може надати одні й ті самі функції, але трішки їх примножити, біржа яка точно не буде у вас нічого запитувати, за все ви відповідаєте самі.

- Україна: повна підтримка усіх існуючих банків в Україні, для оплати або купівлі активів.

Повний список бірж за рейтингом наведено у (<https://meta.ua/uk/crypto/>).

Особливу увагу необхідно приділити такій Українській біржі як WhiteBit. WhiteBIT - це криптовалютна біржа, яка була заснована в 2018 році та базується в Києві. Вона набула значної популярності серед трейдерів та інвесторів завдяки своїм широким можливостям торгівлі, швидкому внесенню та виведенню коштів, а також високому рівню безпеки.

Основні риси та характеристики WhiteBIT:

1. Широкий вибір криптовалют: WhiteBIT пропонує широкий вибір криптовалютних пар для торгівлі, включаючи популярні та менш відомі цифрові активи.

2. Функціональність: Платформа WhiteBIT має ряд функціональних можливостей, таких як ордери зупинки, лімітні та ринкові ордери, а також API для автоматизованої торгівлі.

3. Безпека: WhiteBIT забезпечує високий рівень безпеки, використовуючи різноманітні методи, такі як двофакторна аутентифікація (2FA), захист від DDoS-атак та захист кошельків.

4. Регуляторність: WhiteBIT дотримується вимог до регуляторних стандартів і зобов'язань у сфері валютних операцій та фінансових послуг.

5. Підтримка клієнтів: Біржа надає швидку та ефективну підтримку клієнтів через різні канали зв'язку, такі як електронна пошта та чат підтримки.

6. Співпраця з українськими проектами: WhiteBIT активно співпрацює з українськими криптовалютними та блокчейн проектами, що сприяє розвитку місцевої криптоспільноти та інноваціям в цьому секторі.

Існування саме такої біржі на території України, це дуже важлива річ, бо це допомагає продвинути Україну у фінансовому світі криптовалюти, допомагає розвивати саме такий напрямок як криптовалюта у самій країні, люди можуть легше обмінювати саме гривню на крипто активи, це дає змогу людям швидше адаптуватися до інновацій, дає змогу швидко переводити гроші, та зберігати їх.

Отже, біржа WhiteBIT відіграє важливу роль у розвитку криптовалютного ринку в Україні, надаючи українським користувачам доступ до широкого спектру можливостей та фінансових інструментів.

1.5 Огляд існуючих причин та способів обміну криптовалюти

Кожен пов'язує своє життя з криптовалютою по-різному, хтось навіть не знає що вона існує, але все ж таки вона серед нас і вже присутня. Таким чином було встановлено що криптовалюта це:

- Альтернативний спосіб зберігання та переказу коштів, зберігання яке може працювати як побічний заробіток і перекази які не залежать від традиційних банківських систем та обмежень;

- Також використовується для інвестицій та торгівлі. Деякі люди використовують як легкий вхід до світу інвестицій та побічного доходу, вкладаючи свої кошти у певну монету і сподіваються на її ріст або вони дійсно провели технічний аналіз і впевнені у проекті і купують його токен. Також є люди які називаються трейдерами, які активно торгують криптовалютами, спекулюють на їх постійній зміні ціни на ринку для отримання прибутку;

- Міжнародні перекази коштів, без усіляких обмежень в сумах, часу або банківських заборон, криптовалюта дозволяє швидко та з низькими комісіями здійснювати міжнародні перекази коштів, що може бути корисним для осіб, які ведуть бізнес за кордоном;

- Платежі за товари та послуги, деякі люди навіть в Україні, використовують криптовалюту для прийому та оплати платежів. Продаючи майно людина може приймати оплату у криптовалюті, тобто люба сфера може бути оплачена віртуальною валютою.

Обмін криптовалют може бути дуже необхідний для збереження своїх фінансів у певній допустимій зоні втрат або прибутку, тому що як вже розібралися ринок криптовалют має велику волатильність, тобто ціна на куплену монету або токен може постійно змінюватись то вгору то вниз, ще криптовалюти дуже тендітні в плані світових подій, якщо певна ваша крипта починає падати люди докуповують або обмінюють на іншу яка дає у даний момент приріст до своєї стартової або постійної ціни. Деяким людям потрібно відправити гроші закордон, закордоном не приймають

гривню, потрібно обмінювати і відправляти в доларах, тут допоможе швидкий обмін гривні на валюту долар.

Також є обмеження центрального банку на відправлення певної суми грошей за кордон або купівлю іноземної валюти. Тут також допомагає криптовалюта та її обмін, наприклад в боті, ви просто переводите вашу валюту на вказаний гаманець, бот відправляє вам потрібну валюту, в потрібних вам обсягах і в потрібне вам місце, ви ніяк не обмежені в сумах та переказах. Наприклад, ще є люди які через обмін криптовалют зберігають свою анонімність та приватність в своїх фінансових операціях, оскільки як вже розібралися транзакції в криптовалютній мережі можуть бути менш прозорими, ніж у традиційних банках[2].

Тому обмін крипти може стати важливим, майже для всіх, чи ви просто тримаєте там свої активи, чи відправляєте платежі, чи спекулюєте і заробляєте на цьому. Ось чому ця тема так важлива у даний період часу.

Розглянемо існуючі обміни криптовалют[9]:

1. Криптовалютні біржі: вже з цим розібралися, що це місце де створюємо власні гаманці або профілі, на яких перший раз купляємо або заводимо криптовалюту. Біржі це ще платформи, керовані централізованими організаціями, які надають послуги з обміну криптовалют. Перевагами буде велика кількість торгових пар та безпека ваших фінансів, до недоліків можна віднести повне проходження верифікацій.

2. Обмінники криптовалют: сервіси які пропонують прямий і швидкий обмін однієї монети на іншу, без реєстрацій та якогось складного інтерфейсу, недоліком може бути вища комісія за транзакції.

3. Банкова платіжна система: деякі банки в Україні, вже дозволяють купляти криптовалюту на пряму зі своєї банківської карти, самостійно конвертуючи наприклад гривні у якусь криптовалюту, так це швидко, зручно але головним недоліком можуть стати комісії які ви сплачуєте за свою швидкість та комфорт.

Вже є багато існуючих обмінників які так само працюють з криптовалютами, але у них простий обмін віртуальних активів наприклад долара (USDT) на електронну

гривню(UAH). Мета дипломної роботи створити обмінник розрахований на роботу між криптовалютними активами, тобто токен на токен, монету на монету.

Розглянемо конкретний приклад на образі київського обмінника «Garant» (Рис. 1.3).



Рис. 1.3 Реальний обмінник

Якщо порівнювати обмінник і вже реально існуючий можна зрозуміти, що схожі тільки на 30%, бо @TezChanger_bot працює повністю автоматично, через прикладання великих зусиль для написання бота, який буде замість консультанта сидіти і обробляти транзакції і приймати запити, а в прикладі існуючого, йде робота клієнта з консультантом, щоб забронювати або обміняти валюту, вам потрібно зв'язатися з ним, відправити свої електронні кошти, а потім піти до відділення та забрати виміняну валюту.

Якщо так дивитися у них також є бот, але він у них робить тільки розсилку, щодо зміни курсів та спеціальних акційних пропозицій, тобто зараз порівнюємо створений бот зі сторони автоматизації, не будемо сперечатись у моментах із кількістю валют або клієнтів, якщо так подивитися, то ті самі функції які є в

реальному обміннику можна реалізувати і в @TezChanger_bot, так це займе багато часу але тоді будемо точно поза конкуренцією.

Обмінник дає змогу спробувати повну автоматизацію та економію часу, нові технології та їх впровадження у старі системи. Тому порівняння таке розпливчате, бо аналогів які працюють конкретно в такому самому напрямку не багато. З іншої сторони питання про довіру, звісно людина буде більше довіряти системі яку вона зможе побачити вживу, ані ж система яка робить все сама, а вам просто потрібно кинути кудись свої гроші.

Для таких інновацій людям потрібен час, а обміннику за цей час набити собі ім'я.

1.6 Постановка задачі

Метою стала розробка Telegram бота на основі розібраних даних. Для цього потрібно правильно сформулювати завдання для повного виконання даного проекту.

Потрібно вирішити такі завдання:

1. Вибрати платформу на якій буде існувати та працювати бот.
2. Вибрати мову програмування, яка підходить по своєму функціоналу для написання потрібних функцій.
3. Вивчити бібліотеки які підключаються для виконання функцій.
4. Вибір API котрі будуть використані для реалізації проекту.

Цей бот буде створений на платформі месенджера, то таких питань, як наприклад вибір інтерфейсу, виникати не буде, так як будемо працювати в рамках дозволеності функцій платформи, а саме Telegram.

1.7 Висновки до розділу 1

У першому розділі були розглянуті теоретичні аспекти криптовалюти як об'єкта дослідження. Було дано визначення поняттю «Криптовалюта», розглянуто їх різновиди, переваги та ризики криптовалют як інтернет-активів. Окремо було

описано Біткоїн та його вплив на ринок криптовалют, а також зазначено, що криптовалюта вже є невід'ємною складовою сучасного життя.

Також було проаналізовано фактори впливу на стан криптовалют та їх різноманітність. Було проведено порівняння країн, де криптовалюта повністю дозволена, та тих, де її ще не легалізовано. На основі цього зроблено висновок, що ринок криптовалют є дуже складним і динамічним середовищем.

Було також розглянуто основи безпеки та вразливості криптовалют. Описано систему блокчейн, проаналізовано різницю між монетами та токенами, додано ілюстрацію покрокового пояснення формування блокчейна, а також проведено аналіз ризиків при купівлі та зберіганні криптовалют.

Детально оглянуто та проведено огляд і аналіз існуючих бірж для криптовалют. Вивчено фактори, за якими люди обирають біржі для особистого користування, наведено приклади популярних бірж у світі та порівняно їх функціонал як для світового ринку, так і для ринку України. Особливу увагу приділено крипто-біржі WhiteBit, підкреслено її важливість для українського ринку криптовалют та внесок у підвищення обізнаності суспільства.

Було проведено огляд існуючих причин та способів обміну криптовалют. Розглянуто конкретні ситуації, у яких і для кого може знадобитися функція обміну віртуальної валюти, а також виявлено, які привілеї надає обмін та як криптовалюта впливає на життя у сучасних умовах. Було встановлено, що необхідність обміну віртуальної валюти є надзвичайно важливою.

Нарешті, у завершальній частині розділу було сформульовано завдання для реалізації проекту.

РОЗДІЛ 2

ТЕХНОЛОГІЇ СТВОРЕННЯ ТЕЛЕГРАМ БОТІВ

На базі поставленої задачі з попереднього розділу, можна сказати, що для реалізації обміну криптовалюти існує декілька різних місць, наприклад біржа, банк або обмінник. Відрізняються вони за своїми недоліками та перевагами які вже розібрали, якщо розглядати обмінник який дійсно буде працювати та приносити дохід, був кращим варіантом особистий спеціальний обмінник, в якому зможемо вказувати комісію за перекази самостійно, щоб могли вводити якісь свої спеціальні функції, та в цілому такі обмінники ні від кого не залежать. Тепер слід розібратися в питаннях:

1. Чому бот буде існувати на платформі месенджера Telegram?
2. Яка мова програмування підходить для розробки?
3. Які бібліотеки використовуються ?
4. API?
5. Які валюти будуть використовуватися в боті?

До переваг вибору саме бота у цій сфері послуг, стає автоматизація та швидкість зв'язку між клієнтом та самим обміном валют, бот зможе приймати запити користувачів, відправляти їм інформацію щодо відправки коштів та відправляти інформацію обміннику що до створеної операції самостійно. А працівнику лише потрібно буде підтверджувати транзакцію з гарантією на біржі. Нажаль принцип роботи безпеки блокчейну про які вже говорили не дозволить запровадити робота який буде ще й самостійно підтверджувати та переводити віртуальну валюту.

2.1. Обмін криптовалюти в Telegram боті

Проектування системи обміну криптовалюти за допомогою Telegram бота, передбачає під собою вирішення базових питань, наприклад функціонал, вміст, зручність. Перш ніж говорити про сам обмін, потрібно розібратися що таке бот і як він стає основною частиною проекту. Бот це можна сказати штучний інтелект, який програмує людина для виконання послідовних дій, але виконує він саме ті дії та саме за тим порядком, який в ньому був закладений під час програмування, в цілому програмування бота, це і є занесення певних дій, виконання цих дій, в вся ця система вже і називається «БОТ».

З кожним роком функціональність ботів росте, на самому початку їх використовували тільки для якоїсь розсилки повідомлень у певний час і для певних користувачів, з часом люди навчилися закладати в них більш складні процеси, наприклад спілкування з новими користувачами, бот може провести екскурс що до продукту або системи, розказати базову закладену інформацію коли це потрібно.

Саме потрібен бот, який зможе прийняти замовлення клієнта що до обміну валюти, опрацювати це замовлення, взяти потрібну інформацію із бази даних, розподілити її, відправити клієнту інформацію яка потрібна йому, наприклад код електронного гаманця, на який клієнт має відправити свої кошти які він хоче обміняти, а самому обміннику відправити форму, що наприклад з'явився новий запит на обмін, клієнт з ніком @borishuma, хоче обміняти певну валюту на іншу, і вже людина доопрацьовує прийом та підтвердження на відправку віртуальної валюти.

Вирішення задач з розробки бота можна поділити на такі основні етапи:

- етап підготовки текстових даних;
- етап створення блоків з даними;
- створення бота та реєстрація його в системі;
- підключення функцій та баз даних;
- тестування та оцінка працездатності;
- оцінка спроможності бота вилучати інформацію з баз даних[10].

Продумати такого бота та зареєструвати в системі, зараз у популярних месенджерах не є проблемою, для цього самі месенджери та їх розробники додають такий функціонал.

Telegram месенджер на сьогоднішній день, є найпопулярнішим місцем зв'язку як у текстовому форматі так і у форматі дзвінків так голосових розмов, велика кількість функцій, які дозволяють створювати особисті канали сповіщення та розповсюдження інформації, підтримка самого месенджера та м'які правила що до ботів, тобто кожен користувач може зробити свого бота, який буде виконувати дії як для особистого користування так і публічного, людина може підписатися на вашого бота та так само отримати його функціонал.

Telegram це місце де зібралися різні вікові категорії у величезних кількостях людей, тобто знайти тут свою аудиторію не буде проблемою. Кожен користувач зможе натиснути дві кнопки та почати користуватися вашим ботом.

Простий інтерфейс самого месенджеру прибирає питання що до користування будь чим у межах месенджера. Саме тому що ця супер-проста система яка є майже в кожного на телефоні і стало причиною вибору платформи Telegram для «домівки» для бота. Telegram лояльно ставиться до створення ботів, якщо ті не порушують правила системи, тобто таким чином, можемо запровадити бота робити все що завгодно, обмежені тільки функціоналом та правилами. Правил та функціоналу повністю вистачає для реалізації проекту.

Тож можемо перейти до реєстрації особистого ключа для бота в месенджері.

Для цього потрібен лише інший системний бот який вже існує в системі і відповідає за реєстрацію нових, це бот від самого месенджера Telegram (t.me/botfathermeb). Цей бот дозволить створити ключ свого особистого бота та надасть функції для менеджменту над ним.

2.2 Використання Python у створенні Telegram бота

Сам додаток був розроблений на мові програмування Python тому що дивлячись на умови завдання, він здатний дати найбільшу кількість переваг порівняно з іншими мовами. Використання мови Python надало такі переваги як:

1) Легкий синтаксис: Python відомий своєю простотою та читабельністю коду. Це значно полегшує розробку та підтримку коду. Синтаксис Python дозволяє швидко зрозуміти логіку роботи програми, що особливо важливо при розробці ботів, де система працює за принципом подія = відповідь.

2) Велика база готових бібліотек: Python має величезну кількість готових бібліотек, що охоплюють майже всі можливі завдання. Наприклад, для створення телеграм-ботів існують бібліотеки, такі як `'python-telegram-bot'`, `'Telepot'`, `'pyTelegramBotAPI'` та інші. Це дозволяє швидко і легко реалізувати необхідний функціонал, не витрачаючи час на написання базового коду з нуля.

3) Можливість підключення додаткових бібліотек: Python має гнучку систему підключення зовнішніх бібліотек. Це означає, що, крім великої кількості стандартних бібліотек, розробники можуть легко знайти та підключити будь-які додаткові модулі, необхідні для специфічних завдань.

4) Кросплатформеність: Python працює на всіх основних операційних системах, таких як Windows, macOS, Linux. Це дозволяє розробляти та тестувати боти на будь-якій платформі, що значно підвищує зручність і гнучкість роботи.

5) Універсальність: Python є універсальною мовою програмування. Це означає, що за потреби можна змінити частково або повністю тип застосунку або, як у даному випадку, рід діяльності бота. Наприклад, бот, створений для Telegram, може бути адаптований для інших месенджерів, таких як Viber або WhatsApp.

6) Підтримка асинхронного програмування: Python підтримує асинхронне програмування за допомогою бібліотек, як `'asyncio'`. Це дозволяє створювати високопродуктивних ботів, які можуть обробляти багато запитів одночасно, що є важливим для забезпечення швидкої реакції та ефективної роботи бота.

7) Широка підтримка спільноти: Python має велику і активну спільноту розробників. Це означає, що знайти допомогу або рішення для конкретної проблеми дуже легко. Спільнота регулярно ділиться знаннями, прикладами коду, бібліотеками і фреймворками.

8) Легкість розгортання: Існує багато платформ для хостингу Python-ботів, таких як Heroku, AWS Lambda, Google Cloud Functions та інші, що робить процес розгортання простим і зручним. Це дозволяє швидко запускати бота в роботу і забезпечувати його стабільну роботу.

9) Інтеграція з іншими системами та сервісами: Python має потужні можливості для інтеграції з різноманітними базами даних, веб-сервісами та іншими інструментами (наприклад, з AWS, Google Cloud). Це дозволяє створювати складні боти з розширеною функціональністю[6].

Вся робота над програмним кодом здійснювалась в PyCharm Community Edition версії 2023.3.4. Це легкий та дуже потужний редактор вихідного коду, який доступний на більшості операційних систем і є абсолютно безкоштовним та доступним для кожного. PyCharm має велику кількість інструментів для зручної роботи з Python, включаючи вбудовану підтримку роботи з віртуальними середовищами, можливість налагодження коду, інтеграцію з системами контролю версій та багато іншого. Це дозволило значно прискорити процес розробки та забезпечити високу якість коду[12].

Підсумовуючи, використання Python для розробки телеграм-бота дозволило швидко і ефективно реалізувати поставлені завдання, використовуючи всі переваги цієї мови програмування. Простота синтаксису, велика кількість готових бібліотек, підтримка асинхронного програмування, кросплатформеність та інші переваги роблять Python відмінним вибором для розробки ботів та інших програмних рішень.

2.3 Огляд API використаного у програмі

API TronGrid є потужним інструментом для розробників, які працюють з блокчейн-платформою TRON. TronGrid надає доступ до всієї інфраструктури блокчейну TRON, що дозволяє легко взаємодіяти з мережею, здійснювати транзакції, отримувати інформацію про стан рахунків, аналізувати блоки та багато іншого.

У цьому тексті розглянемо основні переваги використання API TronGrid, його можливості та приклади використання.

1. Доступність та простота використання: TronGrid API створений з урахуванням потреб розробників, тому він простий у використанні та добре документований. Це дозволяє швидко розпочати роботу з API навіть новачкам у розробці блокчейн-додатків.

2. Швидка обробка запитів: Завдяки високопродуктивній інфраструктурі TronGrid, обробка запитів здійснюється швидко та ефективно. Це особливо важливо для додатків, які вимагають високої швидкості реакції, таких як фінансові сервіси або ігрові платформи.

3. Масштабованість: TronGrid API розроблений з урахуванням масштабованості. Це означає, що ви можете створювати додатки, які будуть працювати з великим обсягом даних та обробляти велику кількість запитів без втрати продуктивності.

4. Безпека: TronGrid використовує сучасні методи шифрування та аутентифікації для забезпечення безпеки даних та транзакцій. Це гарантує, що ваші додатки будуть захищені від несанкціонованого доступу та атак.

5. Інтеграція з іншими сервісами: TronGrid API легко інтегрується з іншими сервісами та платформами, що дозволяє створювати комплексні рішення, які поєднують можливості блокчейну TRON з іншими технологіями.

API TronGrid надає широкий спектр можливостей для розробників, включаючи:

1. Отримання інформації про блоки та транзакції: Ви можете отримувати детальну інформацію про блоки, транзакції, стан рахунків та інші дані, що зберігаються в блокчейні TRON. Це дозволяє аналізувати історію транзакцій, перевіряти стан рахунків та багато іншого.

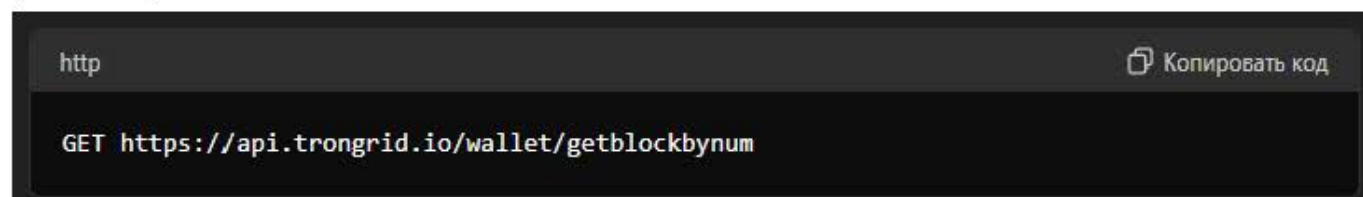
2. Створення та відправка транзакцій: API TronGrid дозволяє створювати та відправляти транзакції в мережу TRON. Це може бути корисним для розробки додатків, які вимагають виконання фінансових операцій, таких як перекази коштів, купівля-продаж токенів тощо.

3. Взаємодія зі смарт-контрактами: TronGrid API надає інструменти для взаємодії зі смарт-контрактами на платформі TRON. Це дозволяє створювати додатки з використанням децентралізованих обчислень та автоматизації процесів.

4. Моніторинг подій: Ви можете налаштувати моніторинг подій у блокчейні TRON за допомогою TronGrid API. Це дозволяє отримувати сповіщення про певні події, такі як надходження транзакцій, зміни стану рахунків та інші події в реальному часі[16].

2.3.1. Приклади використання API TronGrid

Щоб отримати інформацію про блоки, можна використовувати такий запит (Рис. 2.1):



```
http
GET https://api.trongrid.io/wallet/getblockbynum
```

Копировать код

Рис. 2.1 – Приклад запиту, щоб отримати інформацію про блоки

Цей запит повертає детальну інформацію про блок за його номером. Це корисно для аналізу історії блокчейну та перевірки конкретних блоків.

Для створення та відправки транзакції використовується наступний запит (Рис. 2.2):

```
http Копировать код
POST https://api.trongrid.io/wallet/createtransaction
```

Рис. 2.2 - Приклад запиту створення та відправки транзакції

У тілі запиту необхідно вказати дані про транзакцію, такі як відправник, одержувач та сума переказу. Після цього транзакція буде створена та відправлена до мережі TRON.

Для взаємодії зі смарт-контрактами використовується запит (Рис. 2.3):

```
http Копировать код
POST https://api.trongrid.io/wallet/triggersmartcontract
```

Рис. 2.3 – Приклад запиту, щоб викликати функції смарт-контракту

Цей запит дозволяє викликати функції смарт-контракту та отримувати результати їх виконання. Це відкриває можливості для створення децентралізованих додатків з використанням смарт-контрактів.

Для налаштування моніторингу подій можна використовувати WebSocket-з'єднання (Рис 2.4).

```
js Копировать код
const socket = new WebSocket('wss://api.trongrid.io');
socket.onmessage = function(event) {
  console.log('Event:', event.data);
};
```

Рис 2.4 - WebSocket-з'єднання для отримання сповіщень

Цей приклад показує, як можна налаштувати WebSocket-з'єднання для отримання сповіщень про події в реальному часі.

API TronGrid є потужним інструментом для розробників, які працюють з блокчейн-платформою TRON. Його використання надає безліч переваг, включаючи простоту використання, швидку обробку запитів, масштабованість, безпеку та

можливість інтеграції з іншими сервісами. Завдяки широким можливостям API TronGrid, розробники можуть створювати потужні та ефективні блокчейн-додатки, що відповідають сучасним вимогам та стандартам.

2.4 Бібліотеки TronPy та Aiogram

Для повноцінної реалізації проекту телеграм-бота, необхідно було додатково використовувати різноманітні бібліотеки та допоміжні функції, які підключалися до мови програмування Python. Оскільки стандартний набір функцій Python був недостатнім для досягнення бажаного функціоналу, ми залучили такі бібліотеки, як ``tronpy``, ``aiogram``, ``requests`` та ``aiohttp``. Кожна з цих бібліотек відіграла важливу роль у функціонуванні створеного бота, дозволяючи реалізувати різноманітні функції та забезпечити його ефективну роботу.

1. Tronpy: Бібліотека ``tronpy`` використовується для взаємодії з блокчейном мережі TRON. Оскільки в створеному проекті передбачено обмін криптовалюти, яка базується на мережі TRON, ця бібліотека стала невід'ємною складовою програмного забезпечення. Вона відповідає за автоматизацію платежів, підписування та відправлення транзакцій. Бібліотека ``tronpy`` дозволяє ефективно управляти коштами та взаємодіяти з гаманцями користувачів безпосередньо з програмного коду бота.

2. Aiogram: Фреймворк ``aiogram`` є сучасним і потужним інструментом для розробки Telegram-ботів, написаним на мові програмування Python. Він забезпечує можливість легко створювати функціонал бота, включаючи обробку повідомлень, створення кнопок, обробку дій користувачів та розсилку повідомлень. Завдяки ``aiogram`` ми змогли ефективно взаємодіяти з користувачами та забезпечити їм зручний інтерфейс для використання створеного бота.

3. Requests: Бібліотека ``requests`` є допоміжним інструментом для спрощення взаємодії з HTTP-запитами. Вона дозволяє здійснювати звернення до веб-сервісів з легкістю та ефективністю. В створеному проекті ``requests`` використовується для взаємодії з зовнішніми сервісами та отримання необхідних даних для функціонування бота.

4. Aiohttp: Бібліотека `'aiohttp'` встановлена для обробки великої кількості запитів одночасно. Оскільки створений бот має обробляти багато запитів в реальному часі, включаючи переказ коштів, отримання платежів та відправлення повідомлень, потрібна була ефективна та швидка бібліотека для обробки цих запитів. `'Aiohttp'` дозволяє забезпечити асинхронну обробку запитів, що підвищує продуктивність бота та забезпечує його безперебійну роботу навіть при великому навантаженні[6].

2.4.1. Інтеграція та взаємодія

За допомогою бібліотеки `"Aiogram"` було забезпечено прямий зв'язок з ботом, запущеним на комп'ютері через Python, та самим телеграм-ботом. Це надає зручний і простий спосіб взаємодії з ботом та моніторингу його роботи в реальному часі.

Крім того, інтеграція бібліотеки `"Tronpy"` дозволяє безпечно та ефективно обробляти фінансові транзакції, забезпечуючи безпеку та надійність платежів у мережі TRON.

Завдяки використанню різноманітних бібліотек та допоміжних функцій Python, вдалось ефективно реалізувати проект телеграм-бота. Кожна з використаних бібліотек має свою важливу роль у функціонуванні бота та забезпеченні його необхідного функціоналу.

Бібліотека `"Tronpy"` забезпечує безпечну та ефективну взаємодію з блокчейном мережі TRON, `"Aiogram"` дозволяє легко та зручно створювати функціональність бота та інтегрувати його з Telegram, а `"Requests"` та `"Aiohttp"` забезпечують ефективну обробку та взаємодію з зовнішніми сервісами.

В цілому, використання різноманітних бібліотек та допоміжних функцій Python значно полегшило розробку та реалізацію проекту телеграм-бота, забезпечивши необхідний функціонал та продуктивність для успішної роботи бота у реальному часі.

2.5. Висновки до розділу 2

Обмін криптовалюти за допомогою Telegram бота є сучасним та зручним рішенням для користувачів, які прагнуть швидко та безпечно здійснювати фінансові операції. Telegram як платформа надає численні переваги завдяки своїй відкритій API, що дозволяє легко інтегрувати бот із зовнішніми сервісами та забезпечує високу продуктивність. Користувачі отримують доступ до функціоналу обміну криптовалюти безпосередньо у своєму месенджері, що підвищує зручність та швидкість операцій. Автоматизація процесів обміну, включаючи прийом запитів, відправку повідомлень про статус транзакцій та обробку платежів, знижує навантаження на обслуговуючий персонал і підвищує загальну ефективність системи.

Python став основною мовою програмування для створення Telegram бота завдяки своїй простоті, гнучкості та широкому спектру бібліотек. Легкий у використанні синтаксис Python дозволяє швидко розробляти та тестувати нові функції, що значно скорочує час розробки. Підтримка асинхронного програмування за допомогою бібліотек, таких як `asyncio`, забезпечує можливість обробки великої кількості запитів одночасно, що є критично важливим для забезпечення високої продуктивності бота. Використання Python також спрощує інтеграцію з різноманітними API та зовнішніми сервісами, що підвищує загальну функціональність та гнучкість бота.

Одним із ключових аспектів роботи створеного Telegram бота є використання API для взаємодії з блокчейном TRON. API TronGrid дозволяє отримувати інформацію про стан акаунтів, транзакції та блоки у реальному часі. Це забезпечує необхідну функціональність для проведення криптовалютних операцій безпосередньо у боті. Завдяки API TronGrid ми можемо створювати, підписувати та відправляти транзакції, що робить процес обміну криптовалюти швидким і безпечним. Інтеграція з API дозволяє автоматизувати багато процесів, що знижує ризики людської помилки та підвищує загальну надійність системи.

Використання спеціалізованих бібліотек є невід'ємною частиною створеного проекту. Бібліотека TronPy забезпечує взаємодію з блокчейном TRON, дозволяючи автоматизувати процеси створення, підписання та відправки транзакцій. Це особливо важливо для забезпечення безпеки та надійності криптовалютних операцій.

Бібліотека Aiogram, у свою чергу, є потужним фреймворком для розробки Telegram ботів. Вона забезпечує зручні інструменти для обробки повідомлень, створення кнопок та реакцій на дії користувачів. Aiogram дозволяє швидко та ефективно розробляти функціональність бота, забезпечуючи високу продуктивність та зручність використання.

Розробка Telegram бота для обміну криптовалюти є складним та багатокомпонентним процесом, який вимагає ретельного підходу та використання сучасних технологій. Використання платформи Telegram дозволяє забезпечити високу зручність та швидкість обміну для користувачів, надаючи їм доступ до необхідних фінансових операцій безпосередньо у месенджері. Вибір Python як основної мови програмування обумовлений його простотою, гнучкістю та широким спектром бібліотек, що дозволяє швидко реалізовувати та тестувати нові функції.

Інтеграція з API TronGrid забезпечує необхідну функціональність для взаємодії з блокчейном TRON, що дозволяє автоматизувати процеси створення, підписання та відправки транзакцій. Використання бібліотек TronPy та Aiogram забезпечує високу продуктивність та надійність системи, дозволяючи швидко та ефективно обробляти запити користувачів та проводити криптовалютні операції.

Загалом, створення Telegram бота для обміну криптовалюти є ефективним рішенням, яке забезпечує високу швидкість, надійність та зручність для користувачів. Використання сучасних технологій та бібліотек дозволяє створити надійний та продуктивний інструмент, який відповідає всім вимогам сучасного ринку криптовалютних послуг. Автоматизація процесів, інтеграція з блокчейном та використання потужних інструментів для розробки Telegram ботів робить цей проект успішним та перспективним рішенням для обміну криптовалюти.

РОЗДІЛ 3

ПРОГРАМНИЙ ДОДАТОК

3.1. Бізнес-планування

Розроблена програма представляє собою повністю автоматичний обмінник криптовалют в месенджері Telegram, автоматизація здійснюється через самостійно написаного бота.

Цей крипто обмінник дозволяє на даний момент часу обмінювати власні кошти такі як USDT та BTC на монету TRX у мережі TRON.

Весь обмін відбувається у месенджері Telegram встановленому на ~~вашому~~ телефоні або комп'ютері. Ви просто заходите в бота и через прості кнопки починаєте працювати та обмінювати свою криптовалюту.

Від вас запрошують тільки провести оплату у вибраній монеті, а далі просто чекати 30 секунд, як бот обробить вашу угоду, побачить ваші кошти на рахунку обмінника і відпустить вам вже обміняну криптовалюту.

Реєстрація та авторизація в боті відбувається автоматично, бот запам'ятовує ваше ім'я вказане в Telegram при реєстрації месенджера, бот також самостійно запише вас в базу даних, запам'ятає вашу угоду та транзакцію.

Бот через опцію «Інформація» повідомить вас актуальний курс обмінника, перед початком транзакції бот сам розрахує кінцеві суми і повідомить вам їх разом із кодом гаманця на який вам потрібно буде відправити свої віртуальні активи для обміну.

Інтерфейс бот має той самий що і ваші особисті повідомлення у месенджері, тобто як звичайний чат, але до строки для написання повідомлення додається ще й кнопки, нічого складного у використанні.

Сповіднення. Протягом всієї транзакції, бот буде сповіщати вас про її статус, тобто про початок угоди та її завершення або якщо щось пішло не так то бот також вас повідомить.

Безпека та довіра, бот налаштований так щоб ви точно вказали гаманець який підійде вам для прийняття валюти, тобто якщо у боті міняємо USDT на TRX, то на потрібен гаманець мережі Tron, сюди не підійде наприклад гаманець Matic, бот вам про це скаже і попросить ввести дійсний гаманець Tron.

Всі данні про транзакції будуть автоматично записані на біржі якими ви користуєтесь та в базу бота, дані записуються тільки для звітності, щоб адміністратор системи по закінченню дня наприклад, міг зробити звіт, та підрахувати прибуток обмінника.

3.2 Аналіз технічних вимог

Забезпечення проекту здійснює комп'ютер, на якому був написаний код програми. Для працездатності та запуску проекту, потрібен комп'ютер з операційною системою яка підтримує встановлення мови програмування Python, запуск може бути здійснений як за допомогою командної строки Python, так і напряду із застосунку PyCharm, різниці майже немає. Проект працює рівно той час який він запущений на комп'ютері, в даній ситуації комп'ютер виступає як сервер, який забезпечує працездатність бота та надає розрахункові потужності.

Розібрано, що для постійної роботи бота потрібен або комп'ютер, навіть не самий сильний підійде, потрібно тільки запустити бота і мати постійне підключення до мережі інтернет або помістити програму на сервер, який працює в безперебійному режимі та дозволяє боту працювати постійно, але сервер знадобиться тільки в тих випадках, коли проект вже затверджений і працює на реальну аудиторію.

Для опису характеристик системи візьмемо мінімальні обов'язкові комплектуючі ПК які пропонує сайт розробника мови Python[11], а саме:

- 64 або 32 розрядний процесор на архітектурах Intel або AMD;
- 2ГБ оперативної пам'яті;
- 5ГБ вільного місця на жорсткому диску комп'ютера;
- Версія операційної системи не нижче Windows 7, macOS X 10.11, Red

Nat Enterprise Linux 6/7.

Конкретно на момент розробки та тестування, проект був створений, протестований та продемонстрований на комп'ютері з такими характеристиками:

- 64 розрядний процесор на базі Intel (Intel Core i9-9400F);
- 16ГБ оперативної пам'яті;
- Зі 700ГБ вільно місця на SSD диску;
- Операційна система Windows 10 PRO;
- Високошвидкісний інтернет 100мб/с.

Специфікації проекту були розглянуті тільки на операційній системі Windows 10, без розгляду запуску програми на macOS та Linux.

Програма запускається з APK файлу, відкриється спеціальне вікно терміналу Python, в якому ви будете повідомленні що програма з назвою бота почала роботу, та в цілому що не виникало ніяких помилок.

3.3. Проектування та реалізація програмного забезпечення

Для написання наповнення бота, була обрана мова програмування Python. Python доволі проста при вивченні мова, яка ж до того підтримує купу важливих бібліотек для роботи з Telegram месенджером. Його в цілому найчастіше використовують для написання ботів для Telegram, завдяки простоті інтегрування у месенджер.

Послідовними кроками після створення тіла та отримання спеціального ключа для бота, потрібно наповнити його, це найважливіша частина проекту.

Для початку розпишемо який функціонал має бути у боті та будемо рухатися за планом:

1. Бот буде працювати за хендлерами, тобто при натисканні кнопок створюється подія, завдяки якій бот буде опрацьовувати команди.

2. Додається панель адміністратора, в якій буде керуватися повністю бот. В ній адміністратор може змінювати курси, робити розсилку по клієнтам.

3. Обмін, створення угоди між клієнтом та обмінником.

4. Автоматична відправка коштів або віртуальної валюти з гаманця обмінника.

5. Створення бази даних та історії обміну.

6. Простий і зрозумілий інтерфейс[6].

Для початку до бота підключається електронний гаманець. Зроблено так щоб бот отримав повний доступ до гаманця, разом із ключем доступу, завдяки якому, бот самостійно зможе опрацьовувати угоди з обміну криптовалюти (Рис. 3.1).

```

1  [[Bot]
2  token = 7183809101:AA6RdlqwQExdU0GsDt57uCfVKETjV30UFDg
3  channel = -1002008677051
4  admin_id = 497802251
5
6  [Send]
7  private_key = fc52dbfffd21bbaddde9f6b8fa43daab6d0c529492f2a436b1bd1f0c353d2d95
8  from_address = TKWqdunv82RmyakJN2c4hnEkPTbgugnuXV
9  api_key = 0f72fe80-1eb7-45c2-9067-09973382d30c
10
11 [Wallet]
12 wallet_BTC = 1Ma6XxukmsJkumZEh6cyhTQpRiH5esVTJ
13 wallet_USDT = TS4SXmSdAhjM96f3ax6g8aRtYjCvNBC3hk
14
15 [Misc]
16 name_exchange = CryptoChanger
17 helper = https://www.python.org/downloads/release/python-3105/
18 minimal_trx = 100
19 maximal_trx = 1000

```

Рис. 3.1 Ключі доступу

На рисунку 3.1 можемо побачити:

- 2-4: Інформація про бота, його особливий токен який отримували, код каналу та ID адміністратора бота.
- 7-9: Інформація яка необхідна боту для автоматичної відправки та обміну криптовалюти, тобто приватний ключ доступу до гаманця, його адреса, та ключ API.
- 12-13: Інформація що до гаманців для різних методів оплати, тобто за хендлером бот буде вибирати один із цих гаманців, згідно з угодою клієнта.

- 16-19: Інформація про назву бота, та про мінімальні і максимальні кількості монети для обміну.

Наступним кроком додається головну сторінку, опрацьовується інтерфейс і прості, але головні кнопки які будуть створювати подію для опрацювання інформації ботом. Так як в цілому у месенджері немає стандартного вигляду головного меню, воно буде відображено через 3 початкових кнопки (Рис. 3.2).

```

1  import loader as load
2  from aiogram import types
3  from aiogram.types import ReplyKeyboardMarkup, KeyboardButton, InlineKeyboardMarkup, InlineKeyboardButton, ReplyKeyboardRemove
4
5  main_menu = types.ReplyKeyboardMarkup(resize_keyboard=True)
6  items_1 = ['🏠 Головне меню']
7  main_menu.add(*items_1)
8
9  menu = types.InlineKeyboardMarkup()
10 callback_1 = types.InlineKeyboardButton(text="Купити TRX", callback_data="buy")
11 callback_2 = types.InlineKeyboardButton(text="Інформація", callback_data="info")
12 callback_3 = types.InlineKeyboardButton(text="Правила", callback_data="rules")
13 menu.add(callback_1)
14 menu.add(*args: callback_2, callback_3)
15
16 way = types.InlineKeyboardMarkup()
17 callback_1 = types.InlineKeyboardButton(text="Купити за USDT", callback_data="way-USDT")
18 callback_2 = types.InlineKeyboardButton(text="Купити за BTC", callback_data="way-BTC")
19 callback_3 = types.InlineKeyboardButton(text="« Назад", callback_data="back")
20 way.add(*args: callback_1, callback_2)
21 way.add(callback_3)
22
23 stop = types.InlineKeyboardMarkup()
24 callback_1 = types.InlineKeyboardButton(text="« Відміна", callback_data="stop")
25 stop.add(callback_1)
26
27 helper = types.InlineKeyboardMarkup()
28 callback_1 = types.InlineKeyboardButton(text="🆘 Підтримка", url=load.helper)
29 callback_2 = types.InlineKeyboardButton(text="« Головне меню", callback_data="back")
30 helper.add(callback_1)
31 helper.add(callback_3)
32
33 back = types.InlineKeyboardMarkup()
34 callback_1 = types.InlineKeyboardButton(text="« Назад", callback_data="back")
35 back.add(callback_1)
36
37 usage
38 def start(order_id):
39     markup = types.InlineKeyboardMarkup()
40     callback_1 = types.InlineKeyboardButton(text="📄 Почати обмін", callback_data=f"start-{order_id}")
41     callback_2 = types.InlineKeyboardButton(text="« Головне меню", callback_data="back")
42     markup.add(callback_1)

```

Рис. 3.2 Головне меню

На даному етапі було розроблено 3 основні кнопки які створюють подію для подальших дій бота, такі як:

- купити TRX;
- інформація;
- правила;
- допоміжні кнопки підфункцій, такі як «назад», «відміна», «підтримка», «головне меню», «почати обмін».

Далі пишеться блок для адмін-панелі, до якої може увійти користувач, ID якого внесено у початкову таблицю дозволу бота (Рис. 2.1), в панелі адміністратора, можна повністю керувати ботом (Рис. 3.3).

```

25 class AdminState(StatesGroup):
26     markup_set = State()
27
28 @dp.message_handler(commands=['admin'])
29 async def admin(message: types.Message):
30     if message.chat.id == int(admin_id):
31         await message.answer('text: <b>👑 Ви авторизовані в адмін-панелі!</b>', parse_mode='HTML', reply_markup=kb.admin)
32
33 @dp.message_handler(text='📧 Рассылка', state='*')
34 async def mail(message: types.Message, state: FSMContext):
35     if message.chat.id == int(admin_id):
36         await state.reset_state(with_data=False)
37         await message.answer('text: <b>📧 Перешліть мені або надішліть повідомлення, яке потрібно розіслати:</b>', reply_markup=kb.stop_admin, parse_mode='HTML')
38         await MailingState.forwarded.set()
39
40 @dp.message_handler(text='📊 Статистика', state='*')
41 async def static(message: types.Message, state: FSMContext):
42     if message.chat.id == int(admin_id):
43         await state.reset_state(with_data=False)
44         daily_orders, daily_amount, weekly_orders, weekly_amount, monthly_orders, monthly_amount, total_orders, total_amount = db.static_order()
45         count_users = db.count_users()
46         count_order = db.count_order()
47
48         text = f"<b>📊 Статистика обмінника:</b> \n\n" \
49             f"<b>📈 Успішних угод:</b> \n" \
50             f"<b>📅 За день {daily_orders or 0} угод \n" \
51             f"<b>📅 За тиждень {weekly_orders or 0} угод \n" \
52             f"<b>📅 За місяць {monthly_orders or 0} угод \n" \
53             f"<b>📅 За весь час {total_orders or 0} угод \n\n" \
54             f"<b>📈 Оборот обмінника:</b> \n" \
55             f"<b>📅 За день {daily_amount or 0} TRX \n" \
56             f"<b>📅 За тиждень {weekly_amount or 0} TRX \n" \
57             f"<b>📅 За місяць {monthly_amount or 0} TRX \n" \
58             f"<b>📅 За весь час {total_amount or 0} TRX \n\n" \
59             f"<b>👤 Користувачів в боті: {count_users or 0} люд.</b> \n" \
60             f"<b>📄 Відкрито угод {count_order} шт.</b>"
61
62         await message.answer(text, parse_mode='HTML')
63
64 @dp.message_handler(text='🛒 Змінити курс', state='*')
65 async def set_markup(message: types.Message, state: FSMContext):
66     if message.chat.id == int(admin_id):
67         await set_markup()

```

Рис. 3.3 Адміністративна панель

Користувач із правами панелі адміністратора у боті, сам власник, може змінювати курс для купівлю або обміну, розпочати розсилку наприклад про зміну курсу, користувачам які вже користувались ботом, та були занесенні у базу даних бота, про базу трішки згодом, побачити статистику обмінника, за день, за місяць і за весь час, побачити повну інформацію щодо діяльності бота.

Далі було розроблено меню покупки та обміну бота, де користувач обирає метод оплати, кількість і бачить дозволені суми від мінімальної до максимальної (Рис. 3.4).

```

26 @dp.callback_query_handler(text='buy')
27 async def buy(callback_query: CallbackQuery, state: FSMContext):
28     await callback_query.message.edit_text(text='<b>👉 Виберіть метод оплати:</b>', parse_mode='HTML', reply_markup=kb.way)
29
30 #Просим ввести сколько TRX клиент хочет приобрести.
31 @dp.callback_query_handler(text_startswith='way-')
32 async def way(callback_query: CallbackQuery, state: FSMContext):
33     currency = callback_query.data.split("-")[1]
34     user_id = callback_query.message.chat.id
35
36     course = await func.Convert_TRX(TRX=1, currency)
37     await callback_query.message.edit_text(text=f'<b>👉 Введіть скільки TRX ви хочете придбати: \n\nКурс: 1 TRX - <code>{course:.7f} {currency}</code> \n\n👉 Введіть',
38
39     information[user_id] = {}
40     information[user_id]['currency'] = currency
41     await UserState.amount.set()
42
43 #Просим клиента ввести свой TRX адрес.
44 @dp.message_handler(state=UserState.amount)
45 async def amount(message: types.Message, state: FSMContext):
46     try:
47         amount = float(message.text)
48         user_id = message.chat.id
49
50         if amount < float(minimal_trx) or amount > float(maximal_trx):
51             await message.answer(text=f'<b>👉 Введіть суму від <code>{minimal_trx}</code> до <code>{maximal_trx}</code> TRX</b>', reply_markup=kb.stop, parse_mode='HTML')
52             return
53
54         await message.answer(text='<b>👉 Введіть свій адресу TRX:</b>', parse_mode='HTML', reply_markup=kb.stop)
55         await UserState.wallet.set()
56         information[user_id]['amount'] = amount
57
58     except ValueError:
59         await message.answer(text='<b>👉 Суму необхідно ввести в числовому форматі:</b>', parse_mode='HTML', reply_markup=kb.stop)
60
61 #Проверяем что введен именно TRX-адрес, если все верно генерируем сделку и ожидаем подтверждения.
62 @dp.message_handler(state=UserState.wallet)
63 async def wallet(message: types.Message, state: FSMContext):
64     wallet = message.text
65
66 async wallet() except KeyError

```

Рис. 3.4 Початок обміну

Після покрокового натискання вище сказаних кнопок, бот просить ввести клієнта вже свою адресу гаманця, куди він відправить криптовалюту, після проходження та підтвердження угоди. Бот попередить про кінцеві суми та остаточні платежі, які мають відбутися зі сторони клієнта та сторони обмінника.

Бот самостійно також може перевірити правильність вашої адреси, бо для кожної монети вона має свої особливі позначки, тобто у вас не буде такого, що ви випадково ввели не ту адресу і кошти не прийшли від бота, він вас попросить повторно ввести справжню адресу. Якщо ж оплата від клієнта не відбувається певний час, то бот вам відпише що час вашої сесії скінчився і вам потрібно буде заповнити форму спочатку.

Далі був створений блок який відповідає за створення бази даних, завдяки якому бот буде самостійно реєструвати нових користувачів і додавати їх в базу. До речі розсилка з адмін-панелі працює за цією базою, тобто бере користувачів з історії і розсилає їм написане повідомлення (Рис. 3.5).

```

1  Import ...
2
3  1 usage
4
5  def user_info(user_id):
6      cursor.execute('SELECT * FROM users WHERE id=?', (user_id,))
7      user = cursor.fetchone()
8      return user
9
10 2 usage
11
12 def add_users(user_id, username):
13     current_datetime = datetime.now()
14     cursor.execute('INSERT INTO users (id, username, date) VALUES (?, ?, ?)', (user_id, username, current_datetime))
15     conn.commit()
16
17 2 usages
18
19 def markup():
20     cursor.execute('SELECT markup FROM admin')
21     markup_order = cursor.fetchone()[0]
22     return markup_order
23
24 1 usage
25
26 def create_order(order_id, amount_trx, amount_coin, wallet, coin):
27     current_datetime = datetime.now()
28     cursor.execute('INSERT INTO deal (id, amount_trx, amount_coin, wallet, coin, date) VALUES (?, ?, ?, ?, ?, ?)', (order_id, amount_trx, amount_coin, wallet, coin,
29     conn.commit()
30
31 1 usage
32
33 def data_info(order_id):
34     cursor.execute('SELECT * FROM deal WHERE id=?', (int(order_id),))
35     data = cursor.fetchone()
36     return data
37
38 2 usages
39
40 def succesfull_order(user_id, amount, amount_coin, name_coin, wallet):
41     current_datetime = datetime.now()
42     cursor.execute('INSERT INTO history (id, amount, amount_coin, name_coin, wallet, date) VALUES (?, ?, ?, ?, ?, ?)', (user_id, amount, amount_coin, name_coin, wal
43     conn.commit()
44
45 1 usage
46
47 def select_users_mail():
48     cursor.execute('SELECT id FROM users')
49     user_id = cursor.fetchall()

```

Рис. 3.5 Створення бази

Пояснення до рисунку 3.5:

- 7-10: Створення та збір інформації про користувача, за його ID у Telegram.
- 12-15: Додавання нового користувача до бази.
- 17-20: Позначка, яку може зробити адміністратор на користувачі, для довіри.
- 22-25: Створення нової угоди, бот це одразу документує і відправляє адміну.
- 27-30: Інформація щодо усіх угод.
- Далі йдуть допоміжні функції запису, наприклад імейл користувачів, кількість купленої валюти на кожного користувача.

Бот вже має велику кількість команд для виконання і обробки, тому настав момент, що можуть виникати різні проблеми, наприклад користувач був різко

від'єднаний від мережі інтернет, або понатискав на кнопки в боті і просто вийшов, для цього був розроблений блок з опрацюванням помилок (Рис. 3.6).

```

19  ***
20
21  if isinstance(exception, CantDemoteChatCreator):
22      logging.exception('Can't demote chat creator')
23      return True
24
25  if isinstance(exception, MessageNotModified):
26      pass
27      return True
28  if isinstance(exception, MessageCantBeDeleted):
29      logging.exception('Message cant be deleted')
30      return True
31
32  if isinstance(exception, MessageToDeleteNotFound):
33      logging.exception('Message to delete not found')
34      return True
35
36  if isinstance(exception, MessageTextIsEmpty):
37      logging.exception('MessageTextIsEmpty')
38      return True
39
40  if isinstance(exception, Unauthorized):
41      logging.exception(f'Unauthorized: {exception}')
42      return True
43
44  if isinstance(exception, InvalidQueryID):
45      logging.exception(f'InvalidQueryID: {exception} \nUpdate: {update}')
46      return True
47
48  if isinstance(exception, TelegramAPIError):
49      logging.exception(f'TelegramAPIError: {exception} \nUpdate: {update}')
50      return True
51  if isinstance(exception, RetryAfter):
52      logging.exception(f'RetryAfter: {exception} \nUpdate: {update}')
53      return True
54  if isinstance(exception, CantParseEntities):
55      logging.exception(f'CantParseEntities: {exception} \nUpdate: {update}')
56      return True
57
58  logging.exception(f'Update: {update} \n{exception}')
59

```

Рис. 3.6 Робота з помилками

У рядках коду з 21 по 58 розписані різні помилки які можуть статися в момент роботи з ботом, бот зможе опрацювати кожену із них, та видати користувачу відповідний до помилки текст із поясненням[15].

Тепер подивимось базу даних яку створює бот (Рис. 3.7).

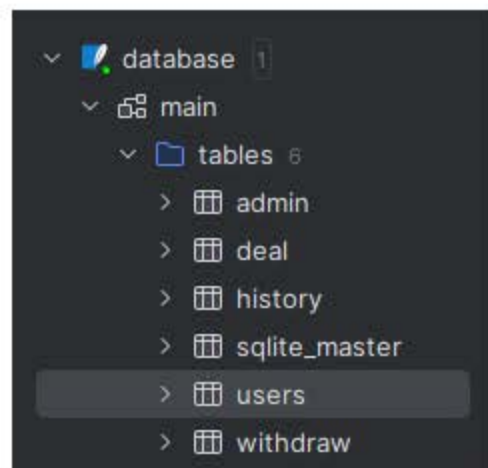


Рис. 3.7 База даних

Тут можемо переглянути адміністраторів, які мають доступ і які входили, історію угод, самі угоди та користувачів (Рис. 3.8), (Рис. 3.9).

WHERE		ORDER BY					
id	amount_tx	amount_coin	wallet	coin	date		
1	50023785	10	1.05074	18EhmyUpuvq3rEJCfrh6jrdxLssJ8KxSSz	USD	2023-11-24 18:45:55.627447	
2	18324694	110	11.52754	1KX3VtmH0qF8Wct587K98ank5oX0Zawij	USD	2023-11-24 20:09:46.159215	
3	68427085	30	5.485374830303031	11XJ8AKH6H1YMYXapPunhriLsbq9abBgaN	USD	2023-11-28 18:25:53.184757	
4	40745475	190	18.66602	17Bwz2dW2yV8KRI1P5K2yc8KngxKunt8M	USD	2023-11-29 18:37:38.369892	
5	29917381	190	18.64142	186duqZfYt29i6kvWc5tKa5J6M5XZBnoBM	USD	2023-11-29 19:19:39.598879	
6	75401549	30	5.567776	18EhmyUpuvq3rEJCfrh6jrdxLssJ8KxSSz	USD	2023-12-02 08:11:46.029459	
7	14294654	200	37.2312	11XJ8AKH6H1YMYXapPunhriLsbq9abBgaN	USD	2023-12-02 08:55:40.886498	
8	79154819	150	27.9254	11XJ8AKH6H1YMYXapPunhriLsbq9abBgaN	USD	2023-12-02 08:56:11.591922	
9	35561568	150	27.854283630303035	11XJ8AKH6H1YMYXapPunhriLsbq9abBgaN	USD	2023-12-02 18:13:30.574689	
10	89487693	50	9.26752	18DpNvAqVqFqH8H8iE58LnYyx0vz1Jeddr	USD	2023-12-02 13:16:51.499887	
11	94288759	190	18.65918	18L5F8p5645SkwWduquc5zAS6kySkmtH	USD	2023-12-04 08:33:28.017807	
12	35481554	200	37.236723630303034	11XJ8AKH6H1YMYXapPunhriLsbq9abBgaN	USD	2023-12-04 12:26:39.687998	
13	52915793	250	66.5006	11XJ8AKH6H1YMYXapPunhriLsbq9abBgaN	USD	2023-12-04 12:27:28.215238	
14	46885190	190	18.67582	11w15fNqD0E5y7Yq8g8HdKlpaYbEgkY6F	USD	2023-12-05 12:06:09.655622	
15	75895552	190	18.93584	18L5F8p5645SkwWduquc5zAS6kySkmtH	USD	2023-12-08 08:17:36.821256	
16	85181527	200	38.765	11jHfHwvzHjH0cZgq171dCts2FlaHlmm	USD	2023-12-09 18:13:52.922186	
17	1208660	60	7.761578	18EhmyUpuvq3rEJCfrh6jrdxLssJ8KxSSz	USD	2023-12-09 10:44:51.1821368	
18	56351533	190	18.785	18K8oelH8e7AC1J8a7AiQ2Xx0HqJ9pRtH	USD	2023-12-13 23:48:08.715059	
19	7729000	250	67.628	182NqYcprcdQs8H1Pelle1J2giYygh1H	USD	2023-12-16 18:18:05.1861905	
20	67855656	50	9.53956	18L5F8p5645SkwWduquc5zAS6kySkmtH	USD	2023-12-16 23:13:25.676088	
21	20220680	500	92.7641	1Q2K5H1x8oeVJedrAKBJK0HecJAmQYSrk	USD	2023-12-16 11:18:24.507360	
22	56556786	200	55.62452	11XJ8AKH6H1YMYXapPunhriLsbq9abBgaN	USD	2023-12-18 11:15:57.586655	
23	37666679	150	28.79307	18L5F8p5645SkwWduquc5zAS6kySkmtH	USD	2023-12-23 22:03:22.768313	
24	51012263	190	57.5615	11XJ8AKH6H1YMYXapPunhriLsbq9abBgaN	USD	2023-12-26 09:36:39.639893	
25	56802694	190	19.107965959999997	18S8XxjH85FiwY195fW67Jz7DdXGgq1Z	USD	2023-12-25 00:08:51.948798	

Рис. 3.8 Угоди

На цьому скріншоті представлений перелік всіх угод, які були укладені на даному обміннику. Цей список відображає угоди, які були створені та виконані в системі. Кожна угода включає в себе деталі та параметри транзакції, що здійснювались на обміннику.

	WHERE	ORDER BY					
	id	username	date	language	balance	invited	purchase_amount
19	6782251283	porshclubb	2024-01-09 21:28:00.553448	ru	0	0	0
20	5479550241	myltik66	2024-01-10 06:47:49.870296	ru	0	0	0
21	906638172	JordanBelfort_va	2024-01-10 07:53:20.997536	ru	0	0	0
22	634232316	hw12013	2024-01-10 08:11:50.034885	ru	0	0	0
23	5699670236	potapov000	2024-01-10 14:23:55.245673	ru	0	0	0
24	5028564878	counnntry	2024-01-17 19:09:13.759970	ru	0	0	0
25	790784201	oldheffner	2024-01-20 11:54:23.322407	ru	0	0	0
26	6167842245	tierce_bella	2024-01-23 16:53:02.599033	ru	0	0	0
27	1288591897	geocesh	2024-02-01 09:54:51.019340	ru	0	0	0
28	439646916	TDI1312	2024-02-04 12:33:10.647911	ru	0	0	0
29	1975466832	WhiteNikolasWild	2024-02-08 07:53:09.306104	ru	0	0	0
30	6699493378	Ostap13129	2024-02-08 21:27:41.237770	ru	0	0	0
31	6703836181	sayhihis	2024-02-10 11:54:37.933124	ru	0	0	0
32	5751955006	raziebxolodka	2024-02-18 21:30:06.142670	ru	0	0	0
33	5817903536	damirjan888	2024-02-19 08:27:35.457456	ru	0	0	0
34	5635650244	uragan_999	2024-02-22 13:43:35.425105	ru	0	0	0
35	5864126999	alohatyt	2024-02-26 16:44:06.709376	ru	0	0	0
36	6799497129	JonSnow_Market	2024-02-26 22:17:01.064426	ru	0	0	0
37	5869407008	raymondforester	2024-02-26 22:20:07.778526	ru	0	0	0
38	475031509	Chikatill0	2024-02-26 22:22:18.454582	ru	0	0	0
39	5930945562	aUp22x82	2024-02-26 22:43:34.373867	ru	0	0	0
40	5211074556	joseffin77	2024-02-26 23:03:39.246967	ru	0	0	0
41	712050388	Extazzz_zayakin	2024-02-26 23:19:22.875615	ru	0	0	0
42	1006657959	Evgrobo	2024-02-27 07:11:00.687596	ru	0	0	0
43	5908047610	xsportdrugsairmax	2024-02-27 07:55:31.291191	ru	0	0	0
44	6175582747	AlfCashinovich	2024-02-27 07:57:33.825130	ru	0	0	0
45	478779213	<null>	2024-02-28 08:25:52.347679	ru	0	0	0
46	5908268730	kkkarat777	2024-02-28 08:56:47.171130	ru	0	0	0
47	6376335215	Graham_Benjamin_Support	2024-03-01 19:29:16.820430	ru	0	0	0
48	355373634	TimeSnatcher	2024-03-04 17:32:37.201665	ru	0	0	0
49	5528811415	TezChanger	2024-03-04 17:33:31.824006	ru	0	0	0
50	497802251	ownfam3	2024-05-20 16:37:00.798559	en	0	355373634	0

Рис. 3.9 Користувачі

Бот ще може підтримувати безліч різних команд та усіляких фішок але навіть із цим набором функцій він може коректно працювати та приносити прибуток[14].

3.3.1. API TronGrid

В проєкті використовується API TronGrid (trongrid.io). Це сервіси та інструменти або їх набір, які розроблені спеціально для розробників, які працюють або використовують блокчейн мережі Tron. Він надає можливість програмування додатків (API), в даному випадку дозволяє перевіряти чи прийшла оплата на гаманець в мережі (Рис 3.10).

```

45  """
46  Надсилаємо TRX через бібліотеку tronpy
47  Угоду проводимо за допомогою API trongrid.io
48  """
49  2.usages
50  async def send_TRX(wallet, amount_trx):
51      send = f'{int(amount_trx)}_000_000'
52      amount = int(send)
53      to_address = wallet
54
55      try:
56          transaction = client.trx.transfer(from_address, to_address, amount).build()
57          signed_transaction = transaction.sign(priv_key)
58          response = signed_transaction.broadcast().wait()
59          tx_id = response['id']
60          return tx_id
61
62      except Exception as ex:
63          await bot.send_message(channel, f'<b>⚠ Помилка під час надсилання TRON: {ex}</b>', parse_mode='HTML')
64          return False

```

Рис. 3.10 Tron grid

Так у створеного бота є схожа бібліотека Tronpy, але вони не виконують одне й те саме, вони працюють в гармонії один з одним, тобто через TronGrid API бот перевіряє чи пройшла оплата на гаманець, після чого дає команду на відправку валюти клієнту, цю команду на відправку вже виконує Tronpy (Рис. 3.11).

```

"""
Функція автоперевірка оплати USD!
Перевірка 60 разів кожні 60 сек - 60 хвилин навів перевірка
Функція перевірки у файлі vutils/function/check.py
"""
4.usages
5.async def check_USDT(amount_check, user_id, order_id, amount_trx, wallet):
6.    max_attempts = 60
7.    attempts = 0
8.
9.    while attempts < max_attempts:
10.        transaction = await func.check_USDT(amount_check)
11.        if transaction:
12.            await bot.send_message(user_id, f'<b>✅ Успішно пополнено USDT!</b> Відправлено {amount_trx} TRX на адресу {code}</code>{wallet}</code>...', parse_mode='HTML')
13.            send = await send_TRX(wallet, amount_trx)
14.
15.            if send:
16.                await bot.send_message(user_id, f'<b>✅ {amount_trx} TRX успішно відправлено на ваші реквізити!</b>', parse_mode='HTML')
17.                await bot.send_message(channel, f'<b>✅ {user_id} успішно купив {amount_trx} TRX. Замовлення - {order_id}</b>', parse_mode='HTML')
18.                db.successfull_order(user_id=user_id, amount=amount_trx, amount_send=amount_check, name_coin='USDOT', wallet=wallet)
19.                if True:
20.                    await bot.send_message(user_id, f'<b>⚠ Помилка під час надсилання {amount_trx} TRX! Відсутність підписки!</b>', parse_mode='HTML')
21.
22.                break
23.
24.            attempts += 1
25.            await asyncio.sleep(60)
26.
27.        if attempts == max_attempts:
28.            await bot.send_message(user_id, f'<b>⚠ Час на оплату закінчив {order_id}! Припиніть! Платіж не було знайдено.</b>', parse_mode='HTML')
29.            break

```

Рис. 3.11 Перевірка оплати

Приклад коду з використанням Tron Grid для відправки на блокчейн:

```

import requests
import json
from tronapi import Tron
from tronapi import HttpProvider

# Ініціалізація Tron з'єднання
full_node = HttpProvider('https://api.trongrid.io')
solidity_node = HttpProvider('https://api.trongrid.io')
event_server = HttpProvider('https://api.trongrid.io')
tron = Tron(full_node=full_node, solidity_node=solidity_node,
event_server=event_server)

# Ваша приватна та публічна адреса
private_key = 'your_private_key'
from_address = tron.address.from_private_key(private_key)
to_address = 'recipient_tron_address'

# Створення транзакції
transaction = tron.trx.send_transaction(to_address, 1000000) # 1 TRX = 1000000

Sun

# Підписання транзакції
signed_tx = tron.trx.sign(transaction, private_key)

# Відправка транзакції
response = tron.trx.broadcast(signed_tx)

print(response)

```

Взагалі TronGrid API надає розробнику широкий вибір кінцевих точок, які дозволяють отримувати таку наприклад інформацію як: вище сказана інформація про транзакції, інформацію про блоки, дані про облікові записи, події угод та статус мережі. Надає швидкий доступ до блокчейну, відзначається високою продуктивністю, що особливо підходить додаткам таким, якому потрібно опрацьовувати багато запитів або транзакцій одночасно[16].

3.4 Експлуатація та супроводження. Інструкція користувача.

Обов'язково для запуску програми потрібно встановити на ваш комп'ютер інтерпретатор мови програмування Python, версії 3.10.X - наприклад 3.10.5. У наступному кроці перейти до папки з програмою та заповнити данні в файлі settings.ini, там підписані пункти які вам потрібно буде замінити. Після чого відкриваємо консоль мови Python, та прописуємо такі команди одна за одною:

- pip install tronpy;
- pip install aiogram==2.23.1;
- pip install requests;
- pip install aiohttp.

Ці команди допоможуть встановити та підключити необхідні для проекту бібліотеки Python.

Якщо при запуску вам буде видавати помилки по типу *ModuleNotFoundError: no module named «Назва бібліотеки яка не пройшла підключення»*, тоді повторюємо процедуру встановлення бібліотеки через pip install як у прикладі і так до того моменту поки бібліотека не запрацює, така процедура робиться лише один раз, і тут вже впливають проблеми з самим бібліотеками та Python і чому він не може їх одразу знайти, зазвичай якщо і з'являється така проблема то вона одразу вирішується повторним приписанням команди, тобто з другої спроби установки.

Якщо все пройшло успішно, то ви можете запускати програму через файл з назвою start.bat (Рис. 3.12).

Имя	Дата изменения	Тип	Размер
.idea	24.05.2024 13:46	Папка с файлами	
__pycache__	10.03.2024 22:42	Папка с файлами	
database	20.05.2024 19:31	Папка с файлами	
handlers	10.03.2024 22:42	Папка с файлами	
keyboard	20.05.2024 16:56	Папка с файлами	
state	10.03.2024 22:42	Папка с файлами	
utils	10.03.2024 22:42	Папка с файлами	
app.py	04.03.2024 12:07	Python File	1 КБ
loader.py	04.03.2024 13:31	Python File	1 КБ
logs.log	24.05.2024 16:02	Текстовый докум...	400 КБ
settings.ini	04.03.2024 17:30	Параметры конф...	1 КБ
start.bat	24.02.2023 20:04	Пакетный файл ...	1 КБ

Рис. 3.12 Запуск програми

Після запуску ви бачите нове вікно терміналу з написом, що бот працює (Рис. 3.13).

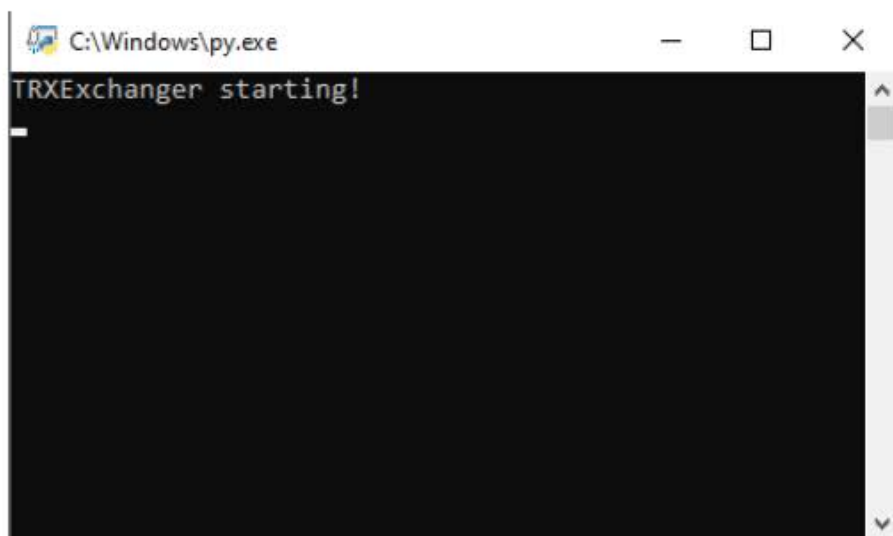


Рис. 3.13 Вдалиий запуск

Тепер ваш комп'ютер став сервером для бота і бот готовий до роботи.

Після запуску самого проекту, можемо переходити в Telegram месенджер, в пошукову строку писати назву бота, у створеного бота вона така : @TezChanger_Bot (https://t.me/TezChanger_Bot) (Рис. 3.14).

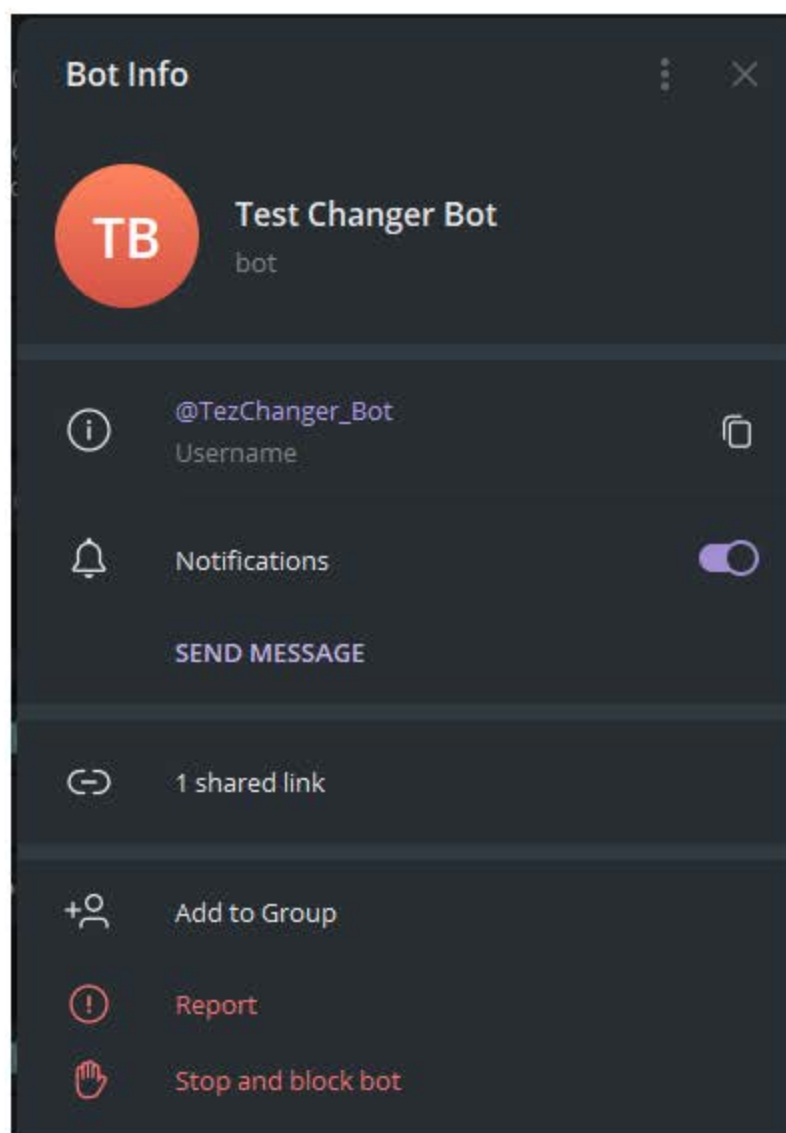


Рис. 3.14 Бот

Натискаємо кнопку «Написати», після чого в строку для повідомлення пишемо /start, і бачимо головне меню бота (Рис. 3.15).

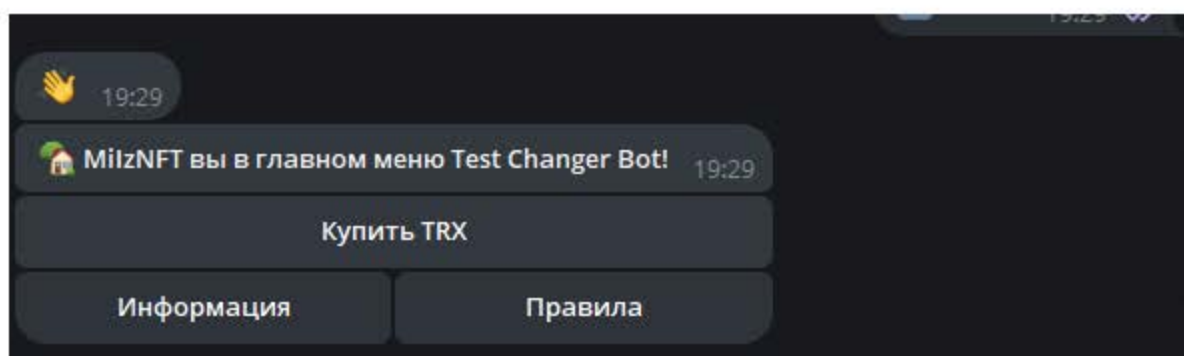


Рис. 3.15 Головне меню бота

Далі при натисканні кнопок «Інформація або Правила» побачимо ось таке повідомлення (Рис. 3.16), воно відобразить певний заготовлений текст.

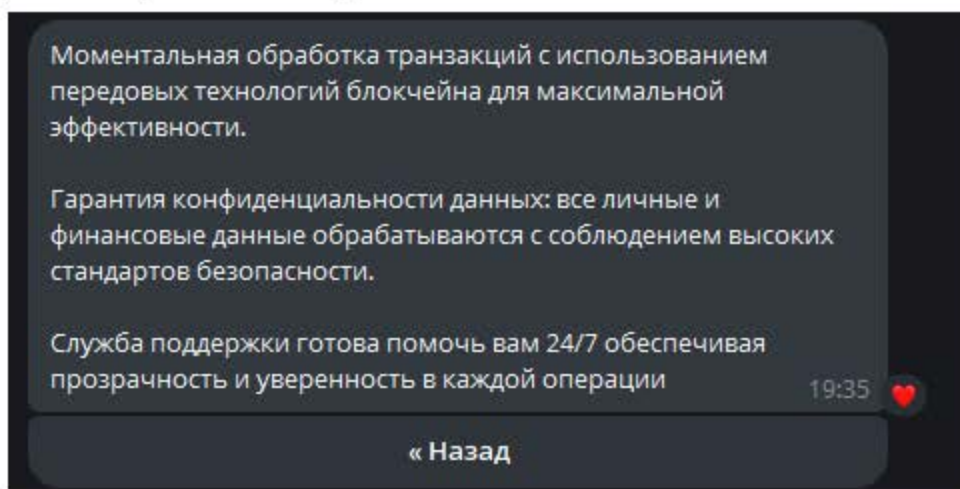


Рис. 3.16 Допоміжні кнопки

Після натискання кнопки «Купити TRX» ви поступово будете переходити до оплати, та здійснення обміну валюти. (пошагові дії зображені на Рис. 3.17-3.22).

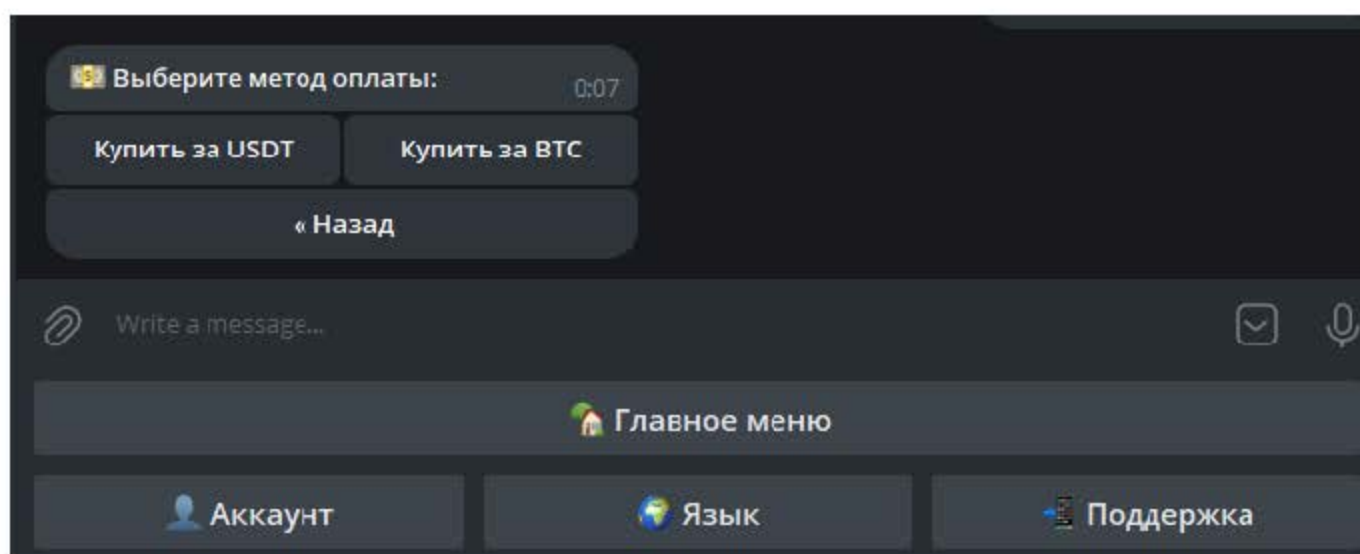


Рис. 3.17 Вибір валюти яку міняємо

На цьому етапі користувач може вибрати, яку саме криптовалюту обмінювати на TRX: USDT або BTC. Це надає гнучкість у виборі валюти для обміну. Користувач самостійно вирішує, яку з цих монет використовувати для транзакції.

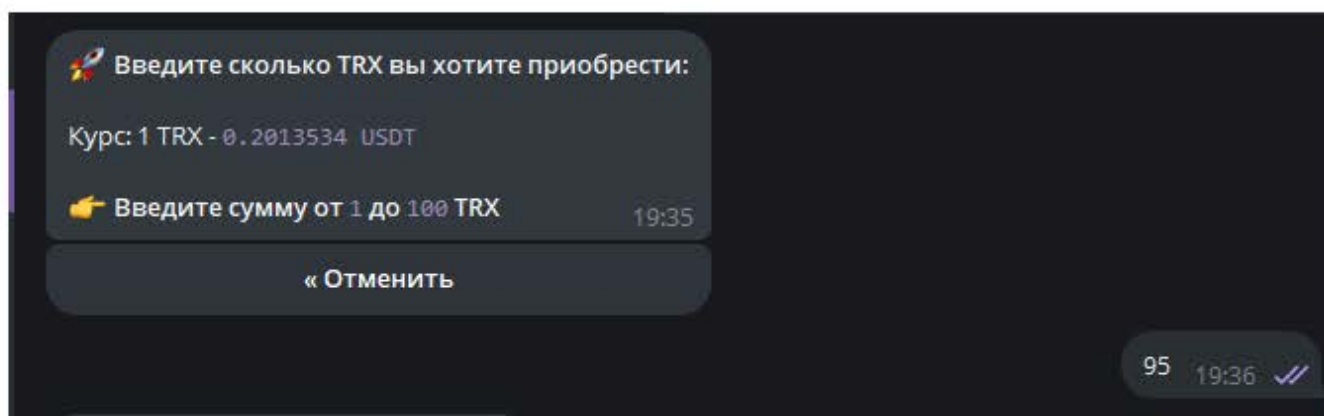


Рис. 3.18 Прохання ввести кількість монет для обміну

На цьому етапі користувач може вибрати, яку кількість монет він бажає отримати. Це дозволяє налаштувати суму відповідно до своїх потреб. Користувач самостійно визначає обсяг криптовалюти для обміну.

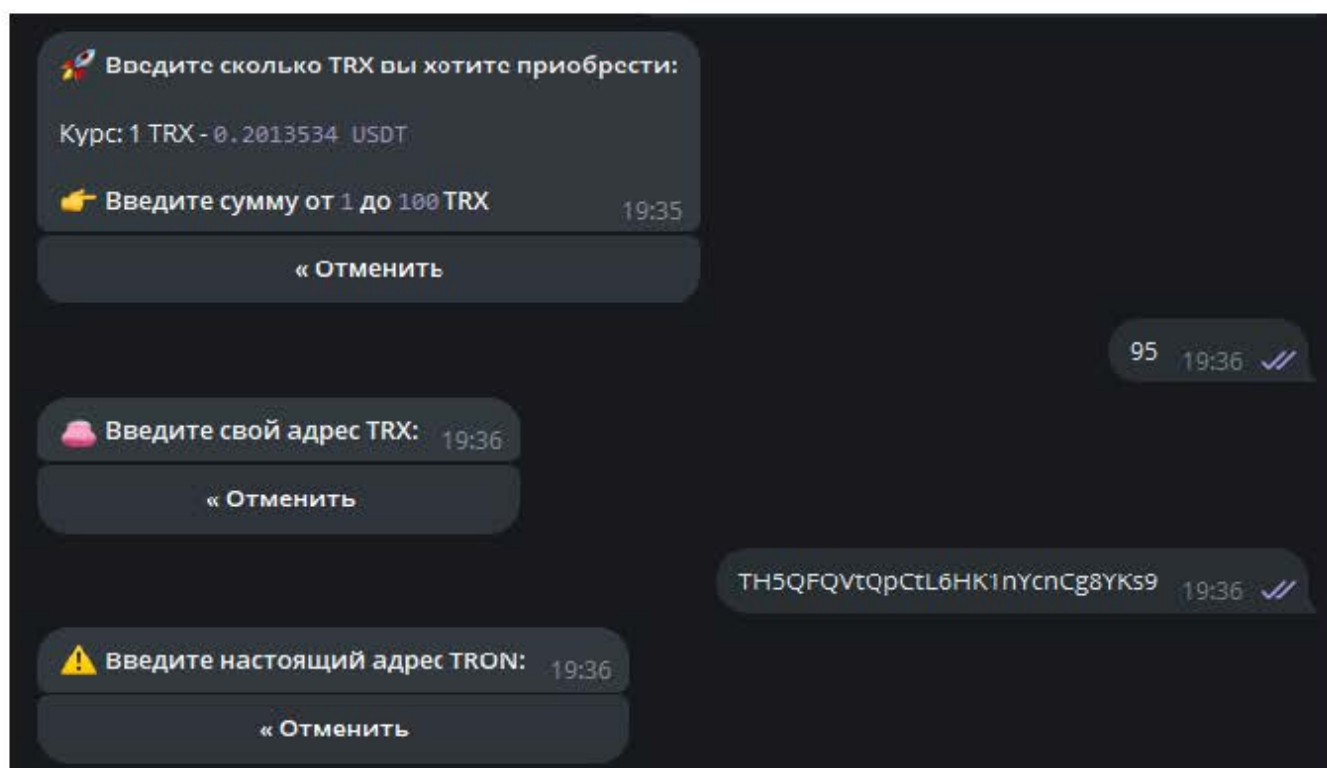


Рис. 3.19 Приклад неправильного вводу гаманця

Цей етап введений для захисту користувача від помилок при введенні свого гаманця або даних гаманця іншої мережі. Він допомагає запобігти випадковому введенню неправильних даних. Таким чином, користувач може переконатися в правильності введенної інформації.

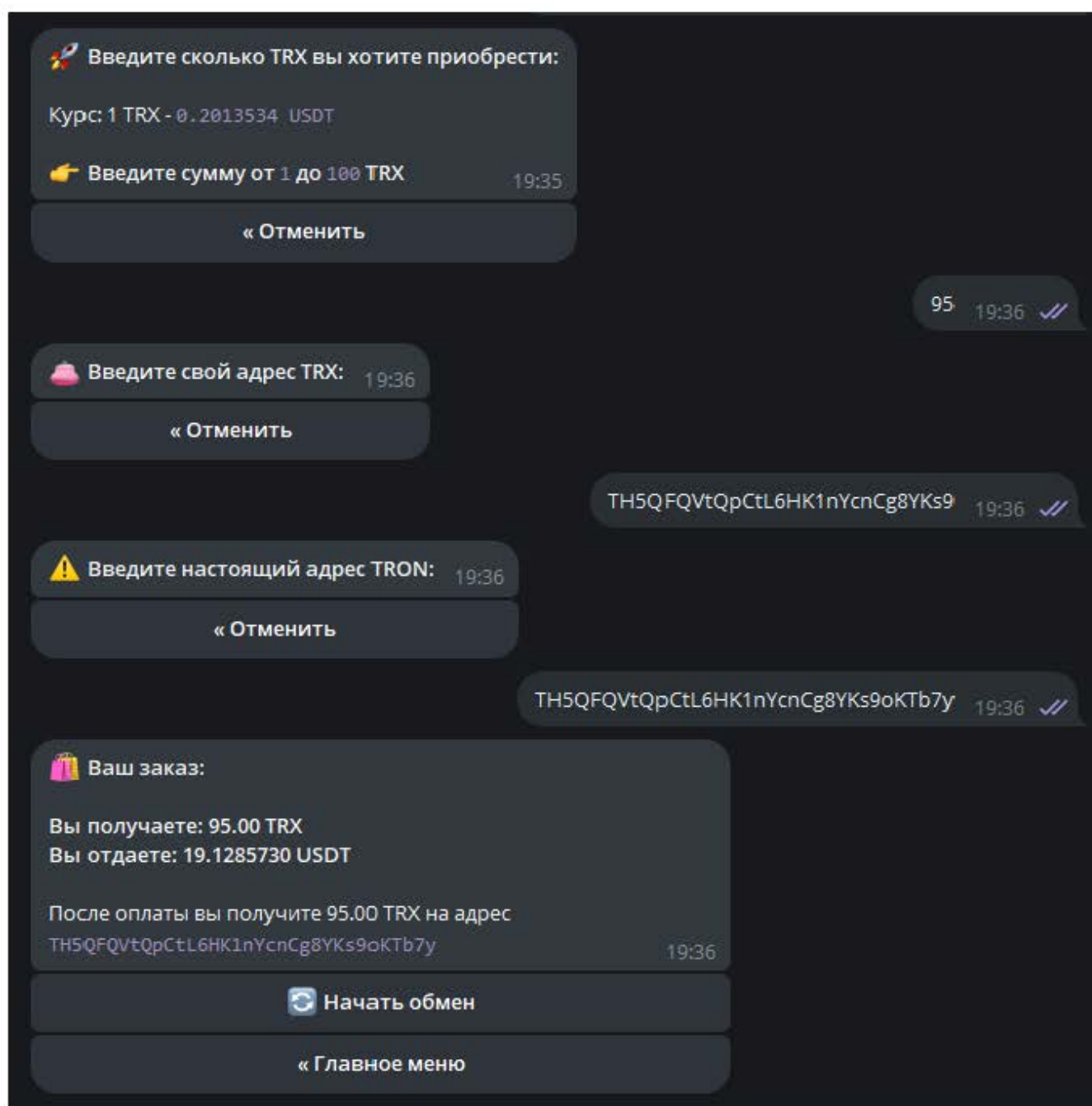


Рис. 3.20 - Приклад правильного ввода гаманця, розрахунок та початок обміну

Користувач отримує повідомлення з необхідною інформацією про угоду. У ньому вказується, яку кількість і яку саме монету потрібно відправити на відповідний гаманець. Якщо всі дані вірні, користувач підтверджує угоду.

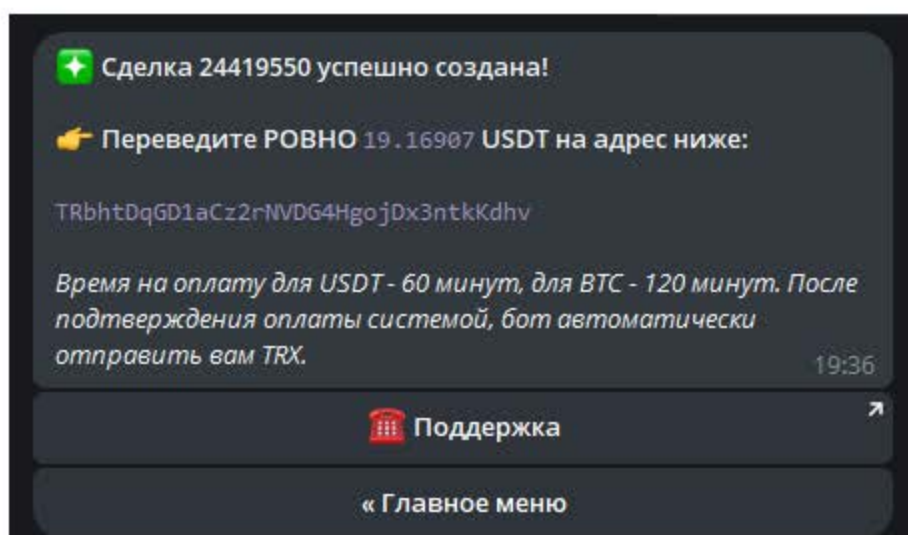


Рис. 3.21 Створена угода

Після створення угоди користувач отримує повідомлення з необхідною інформацією. У повідомленні вказується, яку кількість і яку саме монету потрібно відправити. Також надається адреса відповідного гаманця для транзакції.

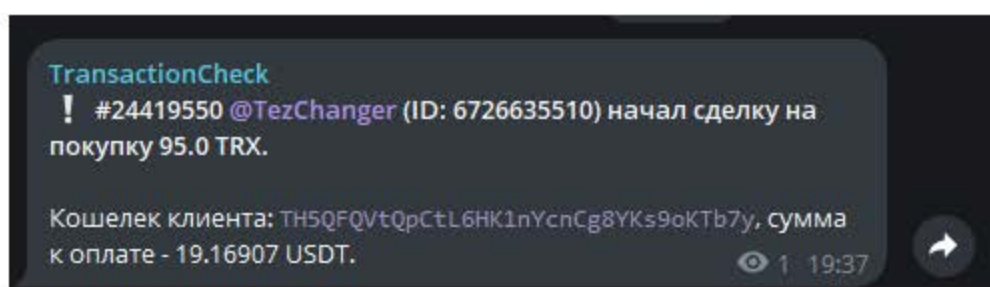


Рис. 3.22 Повідомлення для адміністратора

Після підтвердження угоди користувачем, система надсилає повідомлення у спеціальну групу. Це повідомлення містить інформацію про створений запит на обмін та угоди, які були укладені. Таким чином, інші учасники групи (адміні, модератори) будуть проінформовані про створений запит на обмін та угоди.

На поданих рисунках вибираємо валюту, яку хочемо поміняти на монету TRX. У створеного бота доступно для обміну USDT (долар) або BTC(біткоїн), після чого вводимо кількість монет яку ми хочемо отримати при обміні (Рис. 3.7), після чого бот просить ввести ваш гаманець, для отримання оплати, адреса гаманця TRX на яку бот зможе відправити валюту.

Тут як раз спрацьовує режим безпеки, тобто на рисунку 3.8, ввели адресу зовсім іншої монети, API Tron Grid зафіксувало те що це не адреса мережі Tron і повідомило клієнта про це, далі вводимо справжній адресу (Рис. 3.9), бот одразу розраховує ваше замовлення та створює угоду, представляючи вам чек із точними сумами. Після підтвердження обміну, бот підписує угоду, та дає вам 60 хвилин на те щоб скинути оплату у вигляді 37,23542 USDT і гаманець на який потрібно скинути. Бот вже автоматично заніс вас у свою базу даних, вас і вашу угоду з особливим номером.

В цей же час, бот інформує адміністратора, про нову угоду, відправляючи повідомлення з записаною транзакцією з бази даних (Рис. 3.11).

3.5 Висновки до розділу 3.

У даному розділі була поставлена задача показати реалізації моделі яку представили та розробили у розділі 2.

Отже ось і весь виконаний проєкт згідно поставленому завданню. Були показані зовсім нові технології автоматизації бота для обміну криптовалюти.

Звісно бот потребує професійного доопрацювання, наприклад додавання більшої кількості валют для обміну, але навіть на прикладі трьох валют, можемо побачити всю суть проєкту.

Після виконання представлених дій у підрозділі 3.3, користувач отримує бажану валюту, витративши на всю роботу 2 хвилини, це є неймовірний приклад запровадження технологій у сферу діяльності по обміну валют та криптовалют в цілому.

ВИСНОВОК

Під час процесу виконання кваліфікаційної роботи було розроблено автоматизованого бота для обміну віртуальної криптовалюти, використовуючи широкий спектр сучасних технологій та сучасних рішень. Основною мовою для програмування став Python, що з використанням додаткових бібліотек дозволило створити стабільний і масштабний проект, який працює за простими функціями.

Розглянуто поняття «криптовалюта», її різновиди, переваги та недоліки криптовалют як інтернет активів, приведено приклади використання криптовалют у сучасних сферах життєдіяльності. Проведено аналіз факторів які впливають на стан криптовалюти та приведено приклади різних висновків різних країн щодо дозволеності криптовалют.

Розглянуто основну систему роботи блокчейну, та бірж які вже існують для купівлі і обміну криптовалют, доведено важливість її у нашому житті.

Представлено розроблений скелет проекту, з доступними роз'ясненнями щодо способів функціонування бота, та використаних додаткових джерел для роботи, таких як бібліотеки та додаткові API.

Проведено порівняння між обмінником та обмінниками які вже існують, чим довели конкурентоспроможність проекту і чим він зможе зацікавити користувачів.

Показано реалізацію проекту згідно поставленому завданню, підготовлено інструкцію для користувачів.

Таким чином, розроблений проект зміг об'єднати в собі використання сучасних методів та технологій розробки та створення віртуальних автоматизованих систем. Цей проект показав реальний рух технологій, які можуть бути інтегровані і у інші сфери життя.

Було спроектовано та розроблено програмне забезпечення з використанням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

Реалізовано багаторівневу обчислювальну модель на основі архітектури клієнт-сервер.

Були використанні інструментальні засоби розробки клієнт-серверних застосунків.

Були використані мови системного програмування та методи розробки програм, які взаємодіють з компонентами комп'ютерних систем, а також інтеграція з API TronGrid.

Було розроблено програмні моделі предметних середовищ, обрано парадигму програмування з урахуванням зручності та якості застосування для впровадження методів та алгоритмів в галузі комп'ютерних наук.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Когут Ю.І. Технології блокчейн та криптовалюта: ризики та кібербезпека / Ю.І. Когут. – Київ: Дакор, 2024. - 316 с
2. Дрозд О.Ю. Цифрова валюта у світі та в Україні: правове регулювання, судова практика, цифрові валюти, е-гривня, віртуальні активи криптовалюти / О.Ю. Дрозд. – Київ, Центр навчальної літератури, 2022. – 298 с
3. Рафал Лапаць. Електронна книга Як безпечно і без поспіху увійти у світ криптовалют / Рафал Лапаць. – Київ: ОЛЕАН, 2023. – 237 с
4. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Електронний ресурс: <https://bitcoin.org/bitcoin.pdf>
5. Antonopoulos, A. M. Mastering Bitcoin: Unlocking Digital Cryptocurrencies. O'Reilly Media.
6. Kolluri, V. (2021). Python Programming with Blockchain. Packt Publishing.
7. What is Bitcoin (BTC). Електронний ресурс: <https://blog.whitebit.com/uk/btc/>
8. Grounded and secure: blockchain is the new data protector. Електронний ресурс: https://24tv.ua/economy/ru/chto-takoe-blokchejn-kak-zashhitit-svoi-dannye-ot-moshennichestva-jekonomika-24_n2316166
9. CCXT. (2023). CCXT: Cryptocurrency Exchange Trading Library. Електронний ресурс: <https://github.com/ccxt/ccxt>
10. Dugin, Y. Building Telegram Bots: Develop Bots in 15 Minutes. Independently Published.
11. Python Software Foundation. (2023). Python Documentation. Електронний ресурс: <https://docs.python.org/3/>
12. Kolluri, V. (2023). Python Programming with Blockchain. Packt Publishing.
13. WhiteBIT. (2023). WhiteBIT Documentation and API Guide. . Електронний ресурс: <https://whitebit.com/>
14. Material: Python project “Currency Converter”. Електронний ресурс: <https://vseosvita.ua/library/proekt-python-konverter-valiut-695808.html>

15. Blockchain technology .python cryptocommunicator development. Электронный ресурс: <https://avada-media.ua/ua/services/razrabotka-kriptoobmennika-na-python/>
16. TronGrid is a high-performance API interface for developers. Электронный ресурс: <https://www.trongrid.io>