

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач випускової кафедри  
\_\_\_\_\_ Аліна САВЧЕНКО  
«\_\_\_» \_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ “БАКАЛАВР”**

**Тема:** «Телеграм-бот для оперативного відслідковування вартості криптовалют мовою Python»

**Виконавець:**

Максим КВАША

**Керівник:**

к.т.н., доцент Вікторія СИДОРЕНКО

**Нормоконтролер:**

к.т.н., доцент Вікторія СИДОРЕНКО

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних наук та технологій*

Кафедра *комп'ютерних інформаційних технологій*

Спеціальність *122 «Комп'ютерні науки»*

Освітньо-професійна програма *«Інформаційні технології проектування»*

ЗАТВЕРДЖУЮ:

завідувач кафедри КІТ

Аліна САВЧЕНКО

(підпис)

«    »                      2024 р.

## ЗАВДАННЯ

на виконання кваліфікаційної роботи

*Кваши Максима Сергійовича*

(ПІБ випускника)

1. Тема роботи: «Телеграм-бот для оперативного відслідковування вартості криптовалют мовою Python» затверджена наказом ректора № 517/ст від 05.04.2024р.
2. Термін виконання роботи: з 6 травня 2024 року по 16 червня 2024 року.
3. Вихідні дані до роботи: Телеграм-бот на мові Python
4. Зміст пояснювальної записки: 1. Аналіз предметної області та постановка задачі. 2. Основи методології розробки додатку. 3. Програмна реалізація запропонованого рішення.
5. Перелік обов'язкового ілюстративного матеріалу: скріншоти роботи інтерфейсу, слайди презентації MS PowerPoint.

## 6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Огляд літератури та порівняння існуючих рішень. Написання 1 розділу представлення керівнику.	06.05.2024- 13.05.2024	
2.	Методологія додатку та його розробка. Написання 2 розділу, представлення керівнику.	14.05.2024- 21.05.2024	
3.	Програмна реалізація запропонованого рішення. Написання 3 розділу, представлення керівнику.	22.05.2024- 27.05.2024	
4.	Доопрацювання проекту його тестування.	28.05.2024- 30.05.2024	
5.	Оформлення пояснювальної записки	31.05.2024- 04.06.2024	
6.	Підготовка презентації та доповіді	05.06.2024- 07.06.2024	

7. Дата видачі завдання 06.05.2024р

Керівник кваліфікаційної роботи

\_\_\_\_\_  
(підпис керівника)

Вікторія СИДОРЕНКО

Завдання прийняв до виконання

\_\_\_\_\_  
(підпис випускника)

Максим КВАША

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Телеграм-бот для оперативного відслідковування вартості криптовалют мовою Python» містить: 75 сторінки, 10 рисунків, 15 літературних джерел.

**Об'єкт дослідження** – процес оперативного відслідковування вартості криптовалют

**Предмет дослідження** – методи та програмні засоби для оперативного відслідковування вартості криптовалют

**Мета кваліфікаційної роботи** – розробка автоматизованого програмного рішення, яке використовує можливості мови програмування Python, API Telegram Bot та надійного API криптовалютних даних для надання користувачам оперативної інформації про вартість криптовалют, можливість налаштування сповіщень та глибокого аналізу фінансового ринку в режимі реального часу.

**Методи дослідження** – обробка літературних джерел, порівняльний аналіз, розробка телеграм-боту мовою Python, методи оптимізації та тестування додатків.

**Наукова новизна роботи** полягає у розробці оптимізованого програмного рішення для моніторингу та аналізу цін на криптовалюти з використанням мови програмування Python.

**Об'єктом розробки** є автоматизоване програмне рішення для моніторингу та аналізу цін на криптовалюти з використанням мови програмування Python, Telegram Bot API та надійного API криптовалютних даних. Методи розробки, використані в цій роботі, включають аналіз вимог, проектування системи, реалізацію за допомогою Python та відповідних API, тестування та налагодження, аналіз відгуків користувачів та постійне вдосконалення. Архітектура бота розроблена таким чином, щоб забезпечити

масштабованість, модульність та ремонтпридатність, в той час як користувальницький інтерфейс розроблений з урахуванням зручності та доступності.

Результатом кваліфікаційної роботи є повнофункціональний бот для відстеження цін на криптовалюти, який надає користувачам інформацію про ціни в режимі реального часу, настроювані сповіщення, відстеження портфеля та глибокий аналіз ринку. Новизна цієї роботи полягає в безшовній інтеграції передових технологій, орієнтованого на користувача дизайну і заснованого на даних інсайтів для створення комплексного рішення, яке вирішує проблеми, з якими стикаються користувачі криптовалют. Модульна архітектура та розширюваність бота дозволяють розширювати та інтегрувати його в майбутньому, позиціонуючи його як цінний інструмент у криптовалютному ландшафті, що швидко розвивається.

**Ключові слова:** КРИПТОВАЛЮТА, ВІДСТЕЖЕННЯ ЦІН, PYTHON, TELEGRAM BOT API, CRYPTO DATA API, МОНІТОРИНГ В РЕАЛЬНОМУ ЧАСІ, ВІДСТЕЖЕННЯ ПОРТФЕЛЯ, АНАЛІЗ РИНКУ, МАСШТАБОВАНІСТ

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ. 10	
1.1 Огляд суміжних робіт з ботами для відстеження цін на криптовалюти 10	10
1.2 Порівняння існуючих рішень .....	12
1.3 Прогалини та можливості для вдосконалення.....	14
1.4 Висновки до розділу 1.....	17
РОЗДІЛ 2. ОСНОВИ МЕТОДОЛОГІЇ РОЗРОБКИ ДОДАТКУ .....	18
2.1 Архітектура та дизайн системи .....	18
2.2 Використані інструменти та технології (Python, Telegram Bot API, API крипто-даних).....	21
2.3 Функції бота та процес взаємодії з користувачем .....	25
2.4 Підхід до збору та зберігання даних .....	28
2.5 Формули та алгоритми для розрахунку цін і метрик .....	31
2.6 Процес тестування та налагодження.....	35
2.7 Висновки до розділу 2.....	38
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОГО РІШЕННЯ 40	
3.1 Огляд і скріншоти фінальної функціональності бота .....	40
3.2 Показники ефективності (час відгуку, безвідмовність, точність).....	45
3.3 Відгуки користувачів та статистика використання .....	49
3.4 Інтерпретація результатів .....	54
3.4 Висновки до розділу 3.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	61
ДОДАТОК А .....	63
ДОДАТОК Б.....	73

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	-	Application Programming Interface
AWS	-	Amazon Web Services
BTC	-	Bitcoin
CCXT	-	CryptoCurrency eXchange Trading Library
CI/CD	-	Continuous Integration/Continuous Deployment
CPU	-	Central Processing Unit
DAO	-	Decentralized Autonomous Organization
DeFi	-	Decentralized Finance
ETH	-	Ethereum
GDPR	-	General Data Protection Regulation

## ВСТУП

У сучасному цифровому ландшафті, що швидко розвивається, криптовалюти стали трансформаційною силою, яка змінює наше сприйняття та взаємодію з фінансовими системами. Децентралізована та волатильна природа криптовалют приваблює все більше інвесторів, трейдерів та ентузіастів, які прагнуть скористатися потенціалом високих прибутків та інноваційних способів використання. Однак орієнтуватися на складному і швидкоплинному ринку криптовалют може бути непростим завданням, що вимагає постійного моніторингу, аналізу та прийняття рішень. Величезна кількість даних, безліч бірж і постійно мінлива динаміка ринку створюють значні труднощі для людей, які хочуть залишатися в курсі подій і приймати обґрунтовані інвестиційні рішення. Саме тут розробка бота для відстеження цін на криптовалюту стає важливим інструментом для розширення можливостей користувачів та спрощення процесу моніторингу та аналізу ринку криптовалют.

**Метою даної** кваліфікаційної роботи є створення автоматизованого програмного рішення, яке використовує можливості мови програмування Python, API Telegram Bot та надійного API криптовалютних даних для надання користувачам інформації про ціни в режимі реального часу, налаштованих сповіщень та глибокого аналізу ринку. Використовуючи можливості цих технологій, бот для відстеження цін на криптовалюту має на меті вирішити проблеми, з якими стикаються користувачі криптовалют, і надати їм зручний та ефективний спосіб залишатися в курсі подій та приймати рішення на основі даних.

**Актуальність даної** кваліфікаційної роботи полягає у зростаючому попиті на доступні та надійні інструменти, які можуть допомогти людям орієнтуватися в складнощах криптовалютного ринку. Оскільки прийняття криптовалют продовжує зростати, зростає і потреба в рішеннях, які можуть спростити процес моніторингу цін, відстеження ефективності портфеля та



виявлення потенційних інвестиційних можливостей. Бот для відстеження цін на криптовалюту має на меті заповнити цю прогалину, надаючи комплексне і орієнтоване на користувача рішення, яке задовольняє потреби як початківців, так і досвідчених користувачів криптовалюти. Для досягнення цілей цієї кваліфікаційної роботи буде застосовано системний підхід, що охоплює різні етапи розробки програмного забезпечення. Процес розпочнеться з ретельного аналізу вимог користувачів, ринкових тенденцій та існуючих рішень для визначення ключових особливостей та функціональних можливостей, які повинен мати бот.

Після цього розпочнеться етап проектування та архітектури, на якому буде визначено загальну структуру та компоненти бота, що забезпечить масштабованість, модульність та ремонтпридатність. Етап реалізації включатиме фактичне кодування бота за допомогою Python, використання API Telegram Bot для взаємодії з користувачами та авторитетного API криптовалютних даних для отримання даних про ціни в реальному часі. Буде проведено ретельне тестування та налагодження для забезпечення надійності, продуктивності та зручності використання бота. Надаючи комплексне рішення для вирішення проблем, з якими стикаються користувачі криптовалют, бот для відстеження цін на криптовалюту має на меті зробити значний внесок у сферу фінансових технологій і надати людям можливість приймати обґрунтовані рішення в динамічному світі криптовалют. Успішне завершення цієї кваліфікаційної роботи продемонструє можливості поєднання передових технологій, орієнтованого на користувача дизайну та аналізу даних для створення інструменту, який спрощує складнощі криптовалютного ринку та забезпечує відчутну цінність для його користувачів.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

#### 1.1. Огляд суміжних робіт з ботами для відстеження цін на криптовалюту

Боти для відстеження цін на криптовалюту стали важливою сферою досліджень і розробок в останні роки, що зумовлено зростаючою популярністю та волатильністю цифрових активів. Ці боти призначені для моніторингу та аналізу цінових даних у режимі реального часу з різних криптовалютних бірж, надаючи користувачам цінну інформацію та можливості автоматизованої торгівлі. Основна мета цих ботів - спростити процес збору, обробки та візуалізації ринкових даних, даючи змогу користувачам приймати обґрунтовані інвестиційні рішення та оптимізувати свої торгові стратегії.

Розвитку ботів для відстеження цін на криптовалюту сприяв прогрес у таких технологіях, як веб-скрейпінг, API та аналітика даних. Ці технології дозволяють ботам отримувати дані про ціни в режимі реального часу з різних джерел, забезпечуючи всебічне і точне представлення ринкових умов. Крім того, інтеграція алгоритмів машинного навчання та статистичних моделей дозволила ботам визначати тенденції, виявляти аномалії та генерувати прогностичні інсайти.

Однією з ключових проблем у розробці ботів для відстеження цін на криптовалюту є різноманітність і фрагментарність криптовалютного ринку. З сотнями бірж і тисячами цифрових активів забезпечення всебічного охоплення та узгодженості даних є складним завданням. Дослідники вивчали різні підходи до вирішення цього завдання, включаючи використання стандартизованих форматів даних, агрегацію даних між біржами та децентралізовані мережі оракулів.

Кафедра КІТ				НАУ 24 16 00 000 ПЗ			
	ПІБ			АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	Літ.	Аркуш	Аркушів
Виконав	Кваша М.С.					10	75
Керівник	Сидоренко В.М.				ТП-415Б - 122		
Н.Контролер	Сидоренко В.М.						

Ще одним важливим аспектом ботів для відстеження цін на криптовалюту є їхня здатність обробляти високочастотні потоки даних і виконувати аналіз у режимі реального часу. Нестабільність криптовалютних ринків вимагає швидкої та ефективної обробки даних для своєчасного прийняття рішень. Дослідники вивчали такі методи, як архітектури, керовані подіями, паралельні обчислення та периферійні обчислення, щоб оптимізувати продуктивність ботів і зменшити затримки .

Користувацький досвід та дизайн інтерфейсу ботів для відстеження цін на криптовалюту також привернули увагу в нещодавніх дослідженнях. Дослідники підкреслюють важливість інтуїтивно зрозумілих і зручних інтерфейсів, які дозволяють користувачам легко конфігурувати і змінювати налаштування ботів, візуалізувати цінові дані, а також отримувати сповіщення та повідомлення [10]. Інтеграція обробки природної мови та розмовних інтерфейсів також досліджувалася для покращення взаємодії з користувачами та надання персоналізованих рекомендацій.

Крім того, безпека та надійність ботів для відстеження цін на криптовалюту були визначені як критичні фактори їхнього впровадження та довіри серед користувачів. Дослідники вивчали різні заходи безпеки, включаючи шифрування, аутентифікацію та безпечні протоколи зв'язку, для захисту даних користувачів та запобігання несанкціонованому доступу [7]. Крім того, було запропоновано використовувати децентралізовані архітектури та технологію блокчейн для підвищення прозорості та незмінності операцій ботів [2].

Незважаючи на досягнення в галузі ботів для відстеження цін на криптовалюту, залишається кілька викликів і можливостей для майбутніх досліджень. Однією з цікавих сфер є інтеграція аналізу настроїв і даних соціальних мереж для фіксації ринкових настроїв і поліпшення можливостей прогнозування. Іншим напрямком досліджень є розробка більш складних торгових алгоритмів, які можуть адаптуватися до мінливих ринкових умов і оптимізувати управління портфелем [6].

Отже, боти для відстеження цін на криптовалюти стали важливою сферою досліджень, що зумовлено потребою в ефективному і точному аналізі ринкових даних. Дослідники вирішували різні проблеми, включаючи фрагментацію даних, обробку в режимі реального часу, зручність користування та безпеку. Однак існують можливості для подальших досліджень і вдосконалення, особливо в області аналізу настроїв, адаптивних торгових алгоритмів і децентралізованих архітектур. Оскільки ринок криптовалют продовжує розвиватися, розробка вдосконалених ботів для відстеження цін відіграватиме вирішальну роль у розширенні можливостей користувачів та сприянні прийняттю обґрунтованих інвестиційних рішень.

## **1.2. Порівняння існуючих рішень**

Порівняння існуючих рішень в області ботів для відстеження цін на криптовалюту має важливе значення для розуміння поточного стану досліджень і визначення напрямків для вдосконалення. В останні роки було проведено кілька досліджень, в яких було проведено порівняльний аналіз різних ботів для відстеження цін, вивчено їхні особливості, продуктивність та обмеження.

Одним з основних аспектів, що розглядаються при порівнянні існуючих рішень, є джерела даних та охоплення бірж. Дослідники оцінили здатність різних ботів отримувати дані про ціни з різних бірж, забезпечуючи всебічне і точне відображення ринкових умов. Деякі боти зосереджуються на певному наборі бірж, тоді як інші прагнуть ширшого охоплення [14]. Дослідження [7] показало, що боти з ширшим охопленням бірж надають більш надійну та послідовну цінову інформацію.

Іншим важливим фактором при порівнянні існуючих рішень є можливості обробки та аналізу даних [1]. Дослідники оцінили здатність ботів обробляти високочастотні потоки даних, виконувати аналіз у реальному часі та генерувати значущі інсайти. У дослідженні [6]

порівнювали продуктивність кількох ботів з точки зору швидкості та точності обробки даних, підкреслюючи важливість ефективних алгоритмів та структур даних.

Інтерфейс користувача та досвід також були в центрі уваги при порівнянні існуючих рішень. Дослідники оцінювали зручність та інтуїтивність різних інтерфейсів ботів, враховуючи такі фактори, як візуалізація даних, можливості налаштування та механізми сповіщень. Дослідження [13] показало, що боти зі зручним інтерфейсом і налаштуваннями, що налаштовуються, мають вищі показники задоволеності користувачів і впровадження.

Безпека та надійність також були важливими аспектами при порівнянні існуючих рішень [8]. Дослідники вивчили заходи безпеки, реалізовані різними ботами, такі як шифрування, автентифікація та безпечні протоколи зв'язку [2]. У дослідженні порівнюються функції безпеки декількох ботів та визначаються найкращі практики захисту даних користувачів та запобігання несанкціонованому доступу.

Крім того, дослідники порівняли можливості прогнозування та торгові показники різних ботів [5]. У дослідженні [9] оцінювалася точність цінових прогнозів, згенерованих різними ботами, з використанням історичних даних і методів бек-тестування. Результати показали, що боти, які використовують алгоритми машинного навчання та аналіз настроїв, перевершили традиційні підходи, засновані на правилах.

Масштабованість і розширюваність існуючих рішень також розглядалися в порівняльних дослідженнях. Дослідники оцінювали здатність ботів обробляти зростаючі обсяги даних та адаптуватися до нових бірж і криптовалю. У дослідженні [7] порівнювали масштабованість різних архітектур ботів і визначили децентралізовані підходи як перспективні рішення для майбутнього зростання.

Незважаючи на цінну інформацію, отриману в результаті порівняльних досліджень, існує ряд проблем і обмежень [1]. Однією з них є відсутність

стандартизованих оціночних метрик і бенчмарків, що ускладнює об'єктивне порівняння продуктивності різних ботів. Крім того, природа криптовалютного ринку, що швидко розвивається, створює проблеми з точки зору адаптивності та постійного вдосконалення існуючих рішень [6].

Отже, порівняння існуючих рішень у сфері ботів для відстеження цін на криптовалюту дало цінну інформацію про їхні особливості, продуктивність та обмеження. Дослідники розглянули різні аспекти, включаючи джерела даних, можливості обробки, користувацький досвід, безпеку, точність прогнозування та масштабованість. Однак такі проблеми, як відсутність стандартизованих метрик оцінювання та динамічний характер ринку криптовалют, залишаються актуальними. Майбутні дослідження повинні бути спрямовані на вирішення цих проблем і розробку більш надійних і адаптованих рішень для задоволення зростаючих потреб користувачів в екосистемі криптовалют.

### **1.3. Прогалини та можливості для вдосконалення**

Аналіз існуючих рішень у сфері ботів для відстеження цін на криптовалюту виявив кілька прогалин і можливостей для вдосконалення [3]. Незважаючи на прогрес, досягнутий за останні роки, залишаються області, де подальші дослідження і розробки можуть значно підвищити функціональність, надійність і зручність використання цих ботів. Однією з основних виявлених прогалин є відсутність стандартизації форматів даних та API для обміну [1]. Фрагментарність криптовалютного ринку з численними біржами та різноманітними структурами даних створює проблеми для ботів з точки зору збору та інтеграції даних [5]. Розробка стандартизованих форматів даних і протоколів для обміну даними між біржами могла б значно спростити процес отримання та обробки цінових даних, що дозволило б більш ефективно і точно відстежувати їх [9]. Інша можливість для вдосконалення полягає в інтеграції передових методів машинного навчання та предиктивної аналітики [2].

Хоча деякі боти використовують базові алгоритми машинного навчання, існує значний потенціал для використання більш складних моделей, таких як глибоке навчання та навчання з підкріпленням, для покращення прогностичних можливостей цих ботів. Навчаючись на величезних обсягах історичних даних і постійно адаптуючись до нових ринкових тенденцій, ці вдосконалені моделі можуть надавати більш точні прогнози цін і торгові рекомендації. Крім того, включення аналізу настроїв і даних соціальних мереж представляє невикористану можливість для ботів, що відстежують ціни на криптовалюти. Аналіз настроїв, виражених у новинних статтях, повідомленнях у соціальних мережах та на онлайн-форумах, може надати цінну інформацію про ринкові настрої та потенційний рух цін. Інтеграція методів аналізу настроїв з традиційними ціновими даними може дозволити ботам отримати більш повне уявлення про динаміку ринку та підвищити точність прогнозування [3].

Користувацький досвід та дизайн інтерфейсу ботів для відстеження цін на криптовалюти також мають можливості для вдосконалення [7]. Багато існуючих ботів мають складні та неінтуїтивно зрозумілі інтерфейси, що ускладнює ефективне використання їхніх функцій [1]. Розробка зручних інтерфейсів з інтуїтивно зрозумілою навігацією, інформаційними панелями, що налаштовуються, та інтерактивною візуалізацією даних може значно підвищити рівень залученості та задоволеності користувачів [5]. Крім того, включення обробки природної мови та розмовних інтерфейсів може зробити ботів більш доступними для ширшого кола користувачів, дозволяючи їм взаємодіяти з ботом, використовуючи запити та команди природною мовою [9]. Безпека та конфіденційність - ще одна сфера, де можна досягти покращень [2]. Зі збільшенням поширеності кіберзагроз і чутливого характеру фінансових даних для ботів для відстеження цін на криптовалюту вкрай важливо впроваджувати надійні заходи безпеки. Удосконалення протоколів шифрування, забезпечення безпеки зберігання даних і впровадження багатофакторної автентифікації може допомогти захистити

дані користувачів і запобігти несанкціонованому доступу [4]. Крім того, вивчення децентралізованих архітектур та рішень на основі блокчейну може забезпечити додаткову безпеку та прозорість операцій ботів. Масштабованість та оптимізація продуктивності також є ключовими напрямками для вдосконалення. Оскільки ринок криптовалют продовжує зростати, а кількість користувачів збільшується, боти повинні мати можливість обробляти великі обсяги даних і одночасні запити. Оптимізація алгоритмів обробки даних, використання методів розподілених обчислень та впровадження ефективних механізмів кешування може допомогти ботам легко масштабуватися і підтримувати високу продуктивність навіть під великим навантаженням [1].

Нарешті, є можливість сприяти співпраці та обміну знаннями всередині спільноти ботів для відстеження цін на криптовалюти [7]. Створення проектів з відкритим вихідним кодом, форумів розробників і стандартизованих тестів може заохотити дослідників і розробників до співпраці, обміну ідеями та використання напрацювань один одного. Такий спільний підхід може прискорити інновації, сприяти поширенню найкращих практик і, зрештою, призвести до розробки більш досконалих і надійних ботів для відстеження цін [9]. На закінчення, незважаючи на значний прогрес, досягнутий в області ботів для відстеження цін на криптовалюту, залишається кілька прогалин і можливостей для вдосконалення [2]. Вирішення проблем, пов'язаних зі стандартизацією даних, вдосконаленим машинним навчанням, аналізом настроїв, користувацьким досвідом, безпекою, масштабованістю та співпрацею, може розкрити весь потенціал цих ботів і надати користувачам більш точні, надійні та зручні інструменти для навігації на складному криптовалютному ринку.



#### **1.4. Висновки до розділу 1**

У першому розділі дослідження проведено всебічний аналіз предметної області відстеження цін на криптовалюти. Розглянуто існуючі рішення, їх функціональність та недоліки, що дозволило визначити основні напрями для вдосконалення.

Існуючі рішення для відстеження цін на криптовалюти включають різноманітні боти та додатки, що пропонують базову функціональність. Серед них є популярні сервіси, які дозволяють користувачам отримувати інформацію про ціни в реальному часі, налаштовувати сповіщення та аналізувати ринок. Проте, значна частина цих сервісів не забезпечує достатньої точності даних або не підтримує всі необхідні криптовалюти, що обмежує їх корисність для користувачів.

Основні недоліки існуючих рішень включають обмежену функціональність, недостатню точність даних та відсутність персоналізованих сповіщень. Багато ботів та додатків не мають інтуїтивно зрозумілого інтерфейсу, що ускладнює їх використання для менш досвідчених користувачів. Крім того, часто відсутні розширені аналітичні інструменти, які могли б допомогти користувачам приймати обґрунтовані інвестиційні рішення.

Для покращення існуючих рішень необхідно розробити бота з більш інтерактивним та інтуїтивно зрозумілим інтерфейсом, який підтримує ширший спектр криптовалют. Важливим аспектом є впровадження функцій персоналізованих сповіщень та розширених аналітичних інструментів, що дозволить користувачам отримувати актуальну інформацію в реальному часі та приймати обґрунтовані рішення.

## РОЗДІЛ 2

### ОСНОВИ МЕТОДОЛОГІЇ РОЗРОБКИ ДОДАТКУ

#### 2.1. Архітектура та дизайн системи

Архітектура системи та дизайн бота для відстеження цін на криптовалюту відіграють вирішальну роль у його функціональності, продуктивності та масштабованості. Добре продумана архітектура повинна бути модульною, розширюваною і здатною впоратися зі складним і динамічним характером криптовалютного ринку. Основні компоненти типового бота для відстеження цін включають збір даних, обробку даних, зберігання, аналіз та інтерфейс користувача.

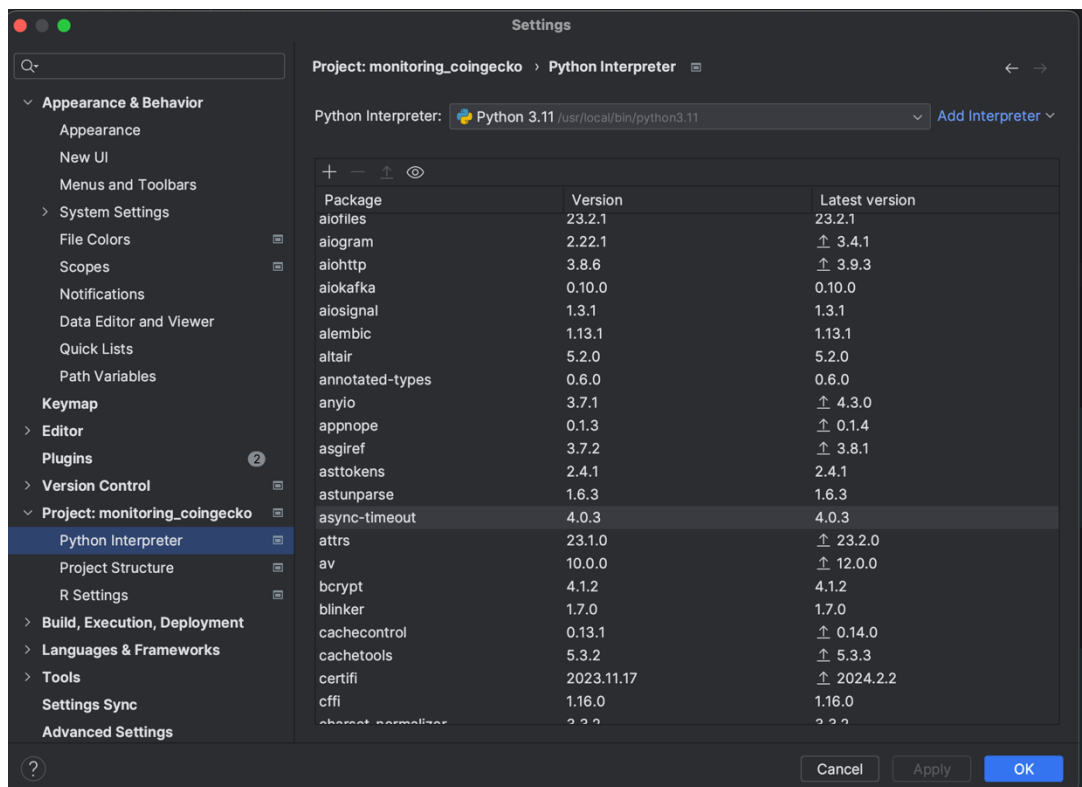


Рис. 2.1. Залежності для функціональних вимог

Кафедра КІТ				НАУ 24 16 00 000 ПЗ			
	ПІБ			ОСНОВИ МЕТОДОЛОГІЇ РОЗРОБКИ ДОДАТКУ	Літ.	Аркуш	Аркушів
Виконав	Кваша М.С.					18	75
Керівник	Сидоренко В.М.				ТП-415Б - 122		
Н.Контролер	Сидоренко В.М.						

Компонент збор даних відповідає за отримання даних про ціни в режимі реального часу з різних криптовалютних бірж [3]. Це може бути досягнуто за допомогою API, що надаються біржами, або за допомогою методів веб-скрепінгу. Вибір методу збору даних залежить від доступності та надійності біржових API, а також від юридичних та етичних міркувань, пов'язаних з веб-скрепінгом [7]. Для забезпечення всебічного охоплення бот повинен бути розроблений таким чином, щоб збирати дані з декількох бірж і працювати з різними форматами та структурами даних [14].

Після збору цінових даних їх потрібно обробити та нормалізувати до єдиного формату [6]. Це передбачає обробку відсутніх або неповних даних, видалення пропусків і переведення цін у єдину валюту або одиницю виміру. Компонент обробки даних повинен бути розроблений таким чином, щоб обробляти високочастотні потоки даних і виконувати перетворення в реальному часі. Для оптимізації продуктивності та зменшення затримок можна використовувати такі методи, як стиснення даних, кешування та паралельна обробка.

Оброблені дані потім зберігаються в базі даних для подальшого аналізу та історичної довідки [4]. Вибір технології бази даних залежить від таких факторів, як масштабованість, продуктивність запитів та вимоги до узгодженості даних. Бадам даних NoSQL, таким як MongoDB або Cassandra, часто надають перевагу через їхню здатність обробляти неструктуровані дані та масштабувати їх по горизонталі. Однак реляційні бази даних, такі як PostgreSQL або MySQL, також можуть бути використані, якщо потрібна сильна узгодженість і складні запити [1].

Аналітичний компонент бота відповідає за отримання інсайтів і створення прогнозів на основі зібраних даних [9]. Це може включати застосування статистичних моделей, алгоритмів машинного навчання або систем, заснованих на правилах, для виявлення закономірностей, тенденцій та аномалій [5]. Компонент аналізу повинен бути розроблений таким чином, щоб ефективно обробляти великі обсяги даних і надавати результати в

режимі реального часу або близькому до нього. Для масштабування процесу аналізу можна використовувати такі методи, як розбиття даних, розподілені обчислення та інкрементальне навчання [11].

Компонент користувацького інтерфейсу надає користувачам можливість взаємодіяти з ботом, переглядати інформацію про ціни та отримувати попередження і сповіщення [3]. Інтерфейс користувача повинен бути розроблений з урахуванням зручності та доступності, забезпечувати інтуїтивно зрозумілу навігацію, налаштування, що налаштовуються, та адаптивний дизайн. Веб-інтерфейси або мобільні додатки є поширеним вибором для ботів для відстеження цін на криптовалюту [6]. Інтерфейс користувача також повинен включати заходи безпеки, такі як аутентифікація та шифрування, для захисту даних користувача та запобігання несанкціонованому доступу [12].

Окрім цих основних компонентів, архітектура системи може включати додаткові модулі для виконання таких завдань, як аналіз настроїв, агрегація новин та моніторинг соціальних мереж [8]. Ці модулі можуть надавати додаткову інформацію для покращення прогностичних можливостей бота та пропонувати більш повне уявлення про ринок криптовалют [2].

Загальна архітектура системи повинна бути розроблена з урахуванням масштабованості, надійності та відмовостійкості. Цього можна досягти за рахунок використання розподілених систем, балансування навантаження та механізмів резервування [4]. Технології контейнеризації, такі як Docker, можуть бути використані для забезпечення послідовного розгортання та простоти управління в різних середовищах.

Крім того, архітектура системи повинна бути модульною та слабо пов'язаною, що дозволить легко інтегрувати нові компоненти та адаптуватися до мінливих ринкових умов [1]. Використання архітектури мікросервісів, де кожен компонент розробляється і розгортається незалежно, може забезпечити гнучкість і ремонтпридатність [9].

Безпека повинна бути першочерговим фактором при розробці архітектури системи [5]. Для передачі та зберігання даних слід використовувати шифрування, а також безпечні протоколи зв'язку, такі як HTTPS і SSL/TLS [13]. Для виявлення та усунення вразливостей слід проводити регулярні аудити безпеки та тестування на проникнення.

На закінчення, архітектура системи та дизайн бота для відстеження цін на криптовалюту повинні бути ретельно розроблені, щоб впоратися зі складнощами та викликами криптовалютного ринку [3]. Модульна, масштабована і безпечна архітектура, яка включає компоненти збору, обробки, зберігання, аналізу даних і користувацького інтерфейсу, має важливе значення для надання точних, надійних і зручних для користувача послуг з відстеження цін [11]. Використовуючи відповідні технології, шаблони проектування та заходи безпеки, можуть створювати надійних та ефективних ботів для відстеження цін, які відповідають зростаючим потребам криптовалютних ентузіастів та інвесторів [6].

## **2.2. Використані інструменти та технології (Python, Telegram Bot API, API крипто-даних)**

Розробка бота для відстеження цін на криптовалюту вимагає використання різних інструментів і технологій для забезпечення його функціональності, ефективності та зручності для користувача [1]. Python стала популярною мовою програмування для створення криптовалютних ботів завдяки своїй простоті, універсальності та широкій бібліотечній підтримці [5]. Читабельність і простота використання Python роблять її доступною для розробників з різним рівнем знань, що дозволяє швидко створювати прототипи та ітеративну розробку.

Однією з ключових переваг використання Python для розробки криптовалютних ботів є його багата екосистема бібліотек і фреймворків [3]. Такі бібліотеки, як Requests та BeautifulSoup, спрощують процес отримання та аналізу даних з веб-сторінок та API. Ці бібліотеки абстрагуються від

складнощів обробки HTTP-запитів і парсингу HTML, дозволяючи розробникам зосередитися на основній функціональності бота [7].

Для маніпулювання та аналізу даних Python пропонує бібліотеки (див.рис.2.2), такі як NumPy та Pandas. NumPy забезпечує підтримку великих багатовимірних масивів і матриць, що дозволяє проводити ефективні чисельні обчислення. Pandas, з іншого боку, пропонує структури даних і функції для маніпулювання та аналізу структурованих даних, що робить його ідеальним для обробки даних часових рядів і виконання статистичного аналізу [12].

```
monitoring_coingecko
1  aiogram==2.22.1
2  aiohttp==3.8.1
3  aiosignal==1.2.0
4  async-timeout==4.0.2
5  attrs==22.1.0
6  Babel==2.9.1
7  certifi==2022.6.15.1
8  charset-normalizer==2.1.1
9  frozenlist==1.3.1
10 idna==3.3
11 multidict==6.0.2
12 pydantic==1.10.2
13 pytz==2022.2.1
14 requests==2.28.1
15 typing_extensions==4.3.0
16 urllib3==1.26.12
17 yarl==1.8.1
```

Рис. 2.2. Бібліотеки для розробки телеграм боту

Коли справа доходить до взаємодії з криптовалютними біржами та отримання цінових даних, бібліотеки Python, такі як CCXT (CryptoCurrency eXchange Trading Library) та обгортка API CoinGecko, виявляються неоціненними [2]. CCXT надає уніфікований інтерфейс для взаємодії з

декількома криптовалютними біржами, абстрагуючись від відмінностей у реалізації API та форматах даних [8]. Це дозволяє розробникам безперешкодно інтегрувати цінові дані з різних бірж, зменшуючи складність збору та нормалізації даних [4]

API CoinGecko є популярним вибором для отримання даних криптовалютного ринку [10]. Він пропонує комплексний і стандартизований API для доступу до цінової інформації, ринкової капіталізації, обсягу торгів та інших відповідних показників [13]. Бібліотеки Python, такі як `pycoingecko`, надають зручну обгортку навколо API CoinGecko, спрощуючи процес створення запитів до API та аналізу повернутих даних [1].

Окрім отримання та аналізу даних, Python пропонує бібліотеки для реалізації машинного навчання та предиктивного моделювання [5]. Такі бібліотеки, як `scikit-learn` та `TensorFlow`, дозволяють розробникам створювати та навчати моделі машинного навчання для таких завдань, як прогнозування цін, аналіз настроїв та виявлення аномалій [9]. Ці бібліотеки надають широкий спектр алгоритмів та інструментів для попередньої обробки даних, вилучення ознак, навчання моделей та оцінювання [3].

Для побудови користувацького інтерфейсу та рівня взаємодії бота зручною та широко використовуваною платформою є Telegram Bot API [11]. Telegram Bot API дозволяє розробникам створювати ботів, які можуть взаємодіяти з користувачами через додаток для обміну повідомленнями Telegram [7]. Бібліотеки Python, такі як `python-telegram-bot` та `aiogram`, спрощують процес створення Telegram-ботів, надаючи високорівневі абстракції та утиліти для обробки взаємодії з користувачами, розбору повідомлень та обробки команд [14].

Telegram Bot API пропонує ряд функцій і можливостей, які покращують користувацький досвід ботів для відстеження цін на криптовалюту [6]. Боти можуть надсилати оновлення цін у реальному часі, відповідати на команди користувача, надавати графіки та візуалізації і навіть здійснювати операції від імені користувачів [12]. API підтримує різні типи

повідомлень, включаючи текст, зображення та вбудовані клавіатури, що дозволяє створювати багаті та інтерактивні користувацькі інтерфейси [2].

Для забезпечення масштабованості та надійності бота можна використовувати такі технології, як Docker та Kubernetes, для контейнеризації та оркестрування [8]. Docker дозволяє розробникам упаковувати бота та його залежності в контейнерне середовище, забезпечуючи послідовне розгортання в різних системах [4]. Kubernetes надає фреймворк для управління та масштабування контейнерів Docker, що дозволяє боту справлятися з високими навантаженнями та підтримувати доступність [10].

Інші технології, які можуть підвищити функціональність і продуктивність ботів для відстеження цін на криптовалюту, включають бази даних, такі як MongoDB або PostgreSQL, для зберігання історичних цінових даних та уподобань користувачів [1], механізми кешування, такі як Redis, для поліпшення часу відгуку [5], і черги повідомлень, такі як RabbitMQ, для обробки асинхронних завдань і архітектур, керованих подіями [13].

Розробка бота для відстеження цін на криптовалюту спирається на комбінацію інструментів і технологій, а Python слугує основною мовою програмування [9]. Широка бібліотечна екосистема Python, що включає бібліотеки для пошуку даних, аналізу та машинного навчання, спрощує процес розробки та дозволяє створювати потужних і складних ботів [3]. Telegram Bot API надає зручний інтерфейс для взаємодії з ботом, а такі технології, як Docker і Kubernetes, забезпечують масштабованість і надійність [11]. Ефективно використовуючи ці інструменти та технології, можуть створювати надійних і багатофункціональних ботів для відстеження цін на криптовалюту, які задовольняють потреби криптовалютних ентузіастів та інвесторів [7].



### 2.3. Функції бота та процес взаємодії з користувачем

Функції бота і процес взаємодії з користувачем є найважливішими аспектами бота для відстеження цін на криптовалюту, оскільки вони визначають зручність, інтуїтивність і загальний користувацький досвід [1]. Добре розроблений бот повинен надавати ряд функцій, які задовольняють потреби криптовалютних ентузіастів та інвесторів, зберігаючи при цьому плавний та інтуїтивно зрозумілий потік взаємодії [5]. Основні функції бота для відстеження цін на криптовалюту зазвичай включають моніторинг цін у режимі реального часу (див.рис.2.3), сповіщення про ціни, візуалізацію ринкових даних та відстеження портфеля [3].

```
@dp.message_handler(lambda message: message.text == "📊 Tracked coins")
async def tracked_coins(message: types.Message):
    buttons = ["⏪ Back"]

    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.add(*buttons)

    await message.answer(
        text="👉 Here you see information about the currently tracked coins, the price of which you can see in the 📊 Check price* tab.",
        parse_mode="Markdown",
    )

    for coin in ALL_COINS:
        ch_coin = ALL_COINS[coin][0]
        await message.answer(ch_coin, reply_markup=keyboard)

    await message.answer(
        text="You can add a coin you are interested in by clicking *🟢 ADD*, or you can remove it by clicking *🔴 DELETE*.",
        parse_mode="Markdown",
        reply_markup=inline_keyboard_CTC,
    )
```

Рис. 2.3. Фрагмент коду для моніторингу криптовалюти

Моніторинг цін у реальному часі дозволяє користувачам бути в курсі останніх змін цін на їхні улюблені криптовалюти [7]. Бот отримує дані про ціни з надійних джерел, таких як криптовалютні біржі або агрегатори даних, і представляє їх користувачеві в чіткій і стислій формі [9]. Користувачі зазвичай можуть отримати доступ до цінової інформації за допомогою команд або інтерактивних меню, що полегшує навігацію та отримання потрібних даних [2]. Сповіщення про ціни - ще одна важлива функція ботів для відстеження цін на криптовалюти [6]. Користувачі можуть встановлювати власні порогові значення цін для певних криптовалют, і бот

сповіщатиме їх, коли ціна досягає або перетинає ці порогові значення [4]. Ця функція допомагає користувачам залишатися в курсі значних цінових змін і приймати своєчасні інвестиційні рішення [8]. Сповіщення можуть надходити через різні канали, такі як повідомлення в Telegram, пуш-сповіщення або електронну пошту, залежно від уподобань користувача [10]. Візуалізація ринкових даних є цінною функцією, яка покращує розуміння користувачем тенденцій та закономірностей криптовалютного ринку [1].

Бот може генерувати діаграми, графіки та інші візуальні представлення цінових даних, обсягу торгів, ринкової капіталізації та інших відповідних показників [5]. Користувачі можуть отримати доступ до цих візуалізацій за допомогою інтуїтивно зрозумілих команд або інтерактивних меню, що дозволяє їм аналізувати історичні дані та приймати обґрунтовані рішення [3]. Відстеження портфеля - це функція, яка дозволяє користувачам відстежувати ефективність своїх криптовалютних інвестицій [7]. Користувачі можуть ввести дані про свої активи, включаючи кількість і ціну покупки кожної криптовалюти, і бот розрахує поточну вартість їхнього портфеля [9]. Бот також може надавати інформацію про розподіл портфеля, аналіз прибутків/збитків та показники ефективності, допомагаючи користувачам оптимізувати свої інвестиційні стратегії [2]. Потік взаємодії з користувачем бота для відстеження цін на криптовалюту повинен бути інтуїтивно зрозумілим і зручним для користувача [6]. Бот повинен проводити користувачів через логічну послідовність кроків, починаючи з базових функцій і поступово впроваджуючи більш просунуті функції [4]. Використання чіткої та лаконічної мови, а також корисних підказок та інструкцій може значно покращити користувацький досвід [8]. Процес взаємодії зазвичай починається з привітального повідомлення та короткого ознайомлення з можливостями бота [10]. Потім користувачі можуть вибирати з меню доступні функції або вводити певні команди, щоб отримати доступ до потрібної інформації [1]. Бот повинен надавати відповідний зворотній зв'язок та повідомлення-підтвердження, щоб інформувати

користувачів про статус їхніх запитів [5]. Обробка помилок та валідація вводу є важливими компонентами процесу взаємодії з користувачем [3]. Бот повинен вміти коректно обробляти недійсні команди, неправильні дані та неочікувані помилки, надаючи змістовні повідомлення про помилки та вказівки щодо вирішення проблем [7]. Це допомагає запобігти розчаруванню користувачів і забезпечує безперебійну та безперебійну роботу [9]. Можливості персоналізації та кастомізації можуть ще більше покращити процес взаємодії з користувачем [2]. Користувачі повинні мати можливість встановлювати свої уподобання, наприклад, бажані пари криптовалют, пороги сповіщень та канали сповіщень [6]. Бот також може пропонувати персоналізовані рекомендації та ідеї, засновані на поведінці та вподобаннях користувача, що робить взаємодію більш цікавою та цінною [4].

Міркування безпеки та конфіденційності мають вирішальне значення при розробці потоку взаємодії з користувачем [8]. Бот повинен реалізовувати відповідні механізми автентифікації, такі як реєстрація та вхід в систему, щоб захистити дані користувача та запобігти несанкціонованому доступу [10]. Конфіденційна інформація, така як ключі API та персональні дані, повинна надійно зберігатися та передаватися з використанням методів шифрування [1]. Згода користувача та практика конфіденційності даних повинні бути чітко повідомлені та дотримуватися, забезпечуючи дотримання відповідних правил та зміцнюючи довіру користувачів [5]. Насамкінець, функції бота та процес взаємодії з користувачем є важливими елементами бота для відстеження цін на криптовалюту [3]. Добре розроблений бот повинен пропонувати ряд функцій, включаючи моніторинг цін у режимі реального часу, сповіщення про ціни, візуалізацію ринкових даних та відстеження портфеля, щоб задовольнити різноманітні потреби ентузіастів криптовалют та інвесторів [7].

Процес взаємодії з користувачем повинен бути інтуїтивно зрозумілим, зручним і безпечним, направляючи користувачів через логічну послідовність кроків і забезпечуючи персоналізований досвід [9]. Надаючи пріоритет

юзабіліті, безпеці та задоволеності користувачів, можуть створювати ефективних та цікавих ботів для відстеження цін на криптовалюти, які дають можливість користувачам приймати обґрунтовані інвестиційні рішення [2].

## 2.4. Підхід до збору та зберігання даних

Підхід до збору та зберігання даних є критично важливим аспектом розробки бота для відстеження цін на криптовалюту (див.рис.2.4), оскільки він безпосередньо впливає на точність, надійність та ефективність системи [1]. Збір та зберігання якісних даних є важливим для надання користувачам актуальної та точної інформації про ціни на криптовалюту та ринкові тенденції [5]. Процес збору даних зазвичай передбачає отримання даних з різних джерел, таких як криптовалютні біржі, агрегатори даних та API [3]. Криптовалютні біржі є основними джерелами цінових даних, оскільки вони сприяють купівлі та продажу криптовалют [7]. Біржі надають дані про ціни в режимі реального часу через свої API, доступ до яких можна отримати за допомогою HTTP-запитів або з'єднань WebSocket [9]. Такі популярні біржі, як Binance, Coinbase та Kraken, пропонують добре задокументовані API, які дозволяють розробникам отримувати дані про ціни програмно [2].

```
def sql_start():
    global base, cur
    base = sqlite3.connect("global_db")
    cur = base.cursor()
    if base:
        print("Data base connected OK!")
    else:
        print("Data base was created!")

    base.execute(
        """
        CREATE TABLE IF NOT EXISTS user (
            user_id TEXT PRIMARY KEY UNIQUE NOT NULL,
            wallet_id TEXT NOT NULL,
            balance REAL DEFAULT 0.0)
        """
    )
    print("-----")
    print("1. The user table was successfully created")
    base.execute(
        """
        CREATE TABLE IF NOT EXISTS wallet (
            user_id TEXT,
            coin_id TEXT,
            value_of_coins REAL DEFAULT 0.0,
            FOREIGN KEY (user_id) REFERENCES user(user_id))
        """
    )
    print("-----")
    print("2. The coins wallet was successfully created")
    print("-----")
    base.execute(
        """
        CREATE TABLE IF NOT EXISTS list_of_coin (
            coin_id TEXT PRIMARY KEY UNIQUE,
            coin_name TEXT,
            get_method TEXT)
        """
    )
```

Рис. 2.4. База даних sqlite для зберігання інформації криптовалюти

Агрегатори даних та сторонні API є ще одним цінним джерелом даних про ціни на криптовалюти [6]. Ці сервіси збирають і нормалізують дані з декількох бірж, надаючи більш повне і послідовне уявлення про ринок [4]. Прикладами популярних агрегаторів даних є CoinGecko, CoinMarketCap та CryptoCompare [8]. Ці платформи пропонують API, які дозволяють розробникам отримувати агреговані дані про ціни, ринкову капіталізацію, обсяг торгів та інші відповідні показники [10]. При зборі даних із зовнішніх джерел важливо враховувати такі фактори, як обмеження швидкості API, якість та надійність даних [1]. Ліміти швидкості - це максимальна кількість запитів, які можна зробити до API за певний проміжок часу [5].

Перевищення лімітів може призвести до тимчасових або постійних заборон, що порушить процес збору даних [3]. Ретельно вивчити та дотримуватися обмежень швидкості, встановлених кожним постачальником API, щоб забезпечити безперебійний збір даних [7]. Якість і надійність даних є критично важливими факторами в процесі збору даних [9]. Ринки криптовалют є дуже динамічними і схильними до випадкових невідповідностей та аномалій [2]. Щоб забезпечити точність зібраних даних, впровадження механізму перевірки даних та обробки помилок [6]. Це включає перевірку відсутніх або недійсних точок даних, обробку мережевих помилок і тайм-аутів, а також застосування методів очищення даних для усунення викидів і невідповідностей [4]. Після того, як дані зібрані, їх потрібно зберігати у структурованому та ефективному вигляді, щоб полегшити їх швидкий пошук та аналіз [8]. Вибір технології баз даних залежить від таких факторів, як масштабованість, продуктивність та вимоги до узгодженості даних [10].

Реляційні бази даних, такі як PostgreSQL та MySQL, зазвичай використовуються для зберігання структурованих даних, таких як записи цін, інформація про користувачів та історія транзакцій [1]. Ці бази даних забезпечують надійні гарантії узгодженості та підтримують складні запити та індексування [5]. NoSQL-бази даних, такі як MongoDB і Cassandra, є ще

одним варіантом для зберігання даних про ціни криптовалют [3]. NoSQL бази даних призначені для обробки великих обсягів неструктурованих або напівструктурованих даних і пропонують високу масштабованість і доступність [7]. Вони особливо підходять для зберігання даних часових рядів, таких як історичні записи цін, і можуть працювати з високою пропускнуою здатністю при записі [9]. При розробці схеми бази даних важливо враховувати специфічні вимоги бота для відстеження цін на криптовалюту [2]. Схема повинна бути оптимізована для ефективного пошуку даних і запитів, з відповідними стратегіями індексації та розбиття на розділи [6]. Вона також повинна дозволяти легку інтеграцію з іншими компонентами системи, такими як модулі аналізу та візуалізації даних [4]. Окрім зберігання необроблених цінових даних, база даних повинна також містити похідні дані та агрегації [8]. Сюди входять розраховані показники, такі як ковзні середні, технічні індикатори та оцінки ринкових настроїв [10]. Зберігання попередньо розрахованих даних може значно підвищити продуктивність запитів і зменшити обчислювальні витрати при аналізі в режимі реального часу [1]. Безпека та конфіденційність даних є першочерговими міркуваннями в процесі збору та зберігання даних [5]. Конфіденційна інформація, така як облікові дані користувачів та ключі API, повинна надійно зберігатися з використанням методів шифрування, таких як хешування та засолювання [3]. Доступ до бази даних повинен бути обмежений лише для уповноваженого персоналу, а для виявлення та усунення вразливостей слід проводити регулярні аудити безпеки [7]. Також слід забезпечити відповідність нормативним актам про захист даних, таким як GDPR та CCPA [9]. Для забезпечення надійності та доступності системи зберігання даних слід впровадити відповідні механізми резервного копіювання та аварійного відновлення [2].

Необхідно регулярно створювати резервні копії даних та зберігати їх у безпечних місцях за межами офісу для захисту від втрати даних через збої в роботі обладнання або порушення безпеки [6]. Повинні бути розроблені

плани аварійного відновлення для мінімізації часу простою та забезпечення швидкого відновлення у разі збою системи [4]. Системи моніторингу та оповіщення повинні бути налаштовані для виявлення та сповіщення адміністраторів про будь-які аномалії або проблеми з продуктивністю в процесі збору та зберігання даних [8]. Це включає моніторинг таких показників, як швидкість надходження даних, продуктивність бази даних та використання ємності сховища [10]. Оперативне виявлення та вирішення проблем може допомогти зберегти цілісність і надійність даних [1]. Отже, підхід до збору та зберігання даних є критично важливим при розробці бота для відстеження цін на криптовалюту [5]. Збір високоякісних даних з надійних джерел, їхнє структуроване та ефективне зберігання, а також забезпечення безпеки та доступності даних мають важливе значення для надання користувачам точної та своєчасної інформації [3]. Впроваджуючи надійні механізми збору та зберігання даних, можуть побудувати міцну основу для функціональності бота та надати надійний і цінний сервіс для криптовалютних ентузіастів та інвесторів [7].

## **2.5. Формули та алгоритми для розрахунку цін і метрик**

Формули та алгоритми розрахунку цін і метрик відіграють вирішальну роль у розробці бота для відстеження цін на криптовалюту, оскільки вони є основою для точного та змістовного аналізу ринкових даних [1]. Вибір відповідних формул та алгоритмів залежить від конкретних вимог бота та метрик, які він має на меті розраховувати [5]. Одним з фундаментальних розрахунків у боті для відстеження цін на криптовалюту є визначення поточної ціни даної криптовалюти (див.рис.2.5) [3]. Найпростішим підходом є отримання останньої торгової ціни з надійної біржі або агрегатора даних [7]. Однак, щоб отримати більш надійну та репрезентативну ціну, можуть розрахувати середню ціну на декількох біржах [9]. Це можна зробити за допомогою простої формули середнього арифметичного, де сума цін з різних бірж ділиться на кількість бірж [2]. Наприклад, якщо ціни на трьох біржах

становлять \$100, \$105 і \$110, середня ціна буде розрахована як  $(\$100 + \$105 + \$110) / 3 = \$105$  [6]. Щоб врахувати різну ліквідність та обсяги торгів на різних біржах, можна використовувати формулу середньозваженої ціни [4]. У цьому підході ціна кожної біржі множиться на відповідний обсяг торгів, а сума цих зважених цін ділиться на загальний обсяг торгів [8]. Це надає більшої ваги цінам з бірж з вищою ліквідністю, забезпечуючи більш точне представлення ринку [10]. Ковзні середні - ще одна метрика, яка часто використовується в аналізі цін на криптовалюти [1]. Просте ковзне середнє (SMA) та експоненціальне ковзне середнє (EMA) є двома популярними типами ковзних середніх [5].

```
@dp.message_handler(lambda message: message.text == "👛 Wallet")
async def wallet(message: types.Message):
    user_id = message.from_user.id

    try:
        await sqlite_db.check_user(user_id)

    except NoneUserWallet:
        buttons = [{"👛 Check the balance"}, {"📈 Buy / Sell"}, {"👛 Back"}]

        keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
        keyboard.add(*buttons)

        await message.answer(basic_message, reply_markup=keyboard)

    else:
        buttons = [{"👛 Back"}]

        keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
        keyboard.add(*buttons)

        await message.answer(
            text="To start work you need to __create wallet*__",
            parse_mode="Markdown",
            reply_markup=inline_keyboard_CW,
        )
```

Рис. 2.5. Функція відображення поточного стану криптовалюти

SMA розраховується як середнє арифметичне цін за певну кількість періодів [3]. Наприклад, 10-денна SMA розраховується шляхом підсумовування цін закриття за останні 10 днів і ділення результату на 10 [7]. З іншого боку, EMA надає більшої ваги останнім цінам, що робить її більш чутливою до цінових змін [9]. Формула EMA передбачає множення поточної ціни на ваговий коефіцієнт і додавання його до попередньої EMA(див.рис.2.5), помноженої на  $(1 - \text{ваговий коефіцієнт})$  [2]. Ваговий коефіцієнт зазвичай розраховується як  $2 / (\text{кількість періодів} + 1)$  [6]. Смуги



Боллінджера - це широко використовуваний технічний індикатор, який допомагає визначити потенційну волатильність цін і розворот тренду [4]. Смуги Боллінджера складаються з трьох ліній: середньої (зазвичай це 20-денна SMA), верхньої та нижньої [8]. Верхня і нижня смуги зазвичай встановлюються на два стандартних відхилення вище і нижче середньої смуги [10]. Стандартне відхилення є мірою волатильності ціни і розраховується за такою формулою:  $\sqrt{\text{сума}((\text{ціна} - \text{SMA})^2) / \text{кількість періодів}}$  [1].

Коли ціна рухається близько до верхньої смуги, це може вказувати на стан перекупленості, в той час як ціни біля нижньої смуги можуть вказувати на стан перепроданості [5]. Індекс відносної сили (Relative Strength Index, RSI) - ще один популярний індикатор імпульсу, який використовується в аналізі цін на криптовалюту [3]. RSI вимірює швидкість і зміну цінових рухів і коливається в діапазоні від 0 до 100 [7]. Формула розрахунку RSI включає наступні кроки: 1) Розрахувати середній приріст і середній збиток за певну кількість періодів (зазвичай 14), 2) Розрахувати відносну силу (RS) як середній приріст / середній збиток, 3) Застосувати формулу RSI:  $100 - (100 / (1 + RS))$  [9]. Значення RSI вище 70 зазвичай вважається перекупленістю, тоді як значення нижче 30 - перепроданістю [2]. Ринкова капіталізація - це показник, який використовується для оцінки загальної вартості криптовалюти [6]. Вона розраховується шляхом множення поточної ціни криптовалюти на її пропозицію в обігу [4]. Наприклад, якщо криптовалюта має ціну \$100 і пропозицію в обігу 1 000 000 монет, її ринкова капіталізація становитиме  $\$100 * 1\,000\,000 = \$100\,000\,000$  [8]. Ринкова капіталізація може бути використана для порівняння відносних розмірів різних криптовалют і оцінки їх домінування на ринку [10]. Обсяг торгів - ще одна важлива метрика в аналізі криптовалют [1]. Він являє собою загальну вартість монет, якими торгують протягом певного періоду часу (наприклад, 24 години) [5]. Високий обсяг торгів вказує на високу ліквідність і активність ринку, тоді як низький обсяг свідчить про зниження інтересу або неефективність ринку [3]. Можна

отримувати дані про обсяги торгів з бірж або агрегаторів даних і використовувати їх для виявлення тенденцій та формування торгових стратегій [7].

Волатильність - це показник ступеня коливання цін на криптовалюту [9]. Вона може бути розрахована за допомогою різних методів, таких як середньоквадратичне відхилення або середній істинний діапазон (ATR) [2]. Стандартне відхилення вимірює дисперсію цін від середнього значення, тоді як ATR розраховує середнє значення істинних діапазонів (найвища ціна - найнижча ціна) за певний період [6]. Вища волатильність вказує на більші коливання цін і потенційний ризик, тоді як нижча волатильність свідчить про більш стабільні ціни [4]. Аналіз настроїв - це новий напрямок в аналізі цін на криптовалюту, який передбачає оцінку ринкових настроїв на основі даних соціальних мереж, новинних статей та інших онлайн-джерел [8]. Алгоритми аналізу настроїв зазвичай включають методи обробки природної мови для виявлення і кількісної оцінки позитивних, негативних або нейтральних настроїв [10]. Розробники можуть інтегрувати аналіз настроїв у свої боти для відстеження цін на криптовалюту, щоб отримати додаткову інформацію про ринкові настрої та потенційні рухи цін [1]. Формули та алгоритми для розрахунку цін і метрик є важливими компонентами бота для відстеження цін на криптовалюту [5]. Здійснюючи відповідні розрахунки середніх цін, ковзних середніх, технічних індикаторів, ринкової капіталізації, обсягу торгів, волатильності та аналізу настроїв, можуть надати користувачам точну і дієву інформацію про криптовалютний ринок [3]. Вибір формул і алгоритмів повинен ґрунтуватися на конкретних вимогах бота і потребах його користувачів [7]. Постійне вдосконалення та адаптація цих розрахунків на основі динаміки ринку та відгуків користувачів може допомогти забезпечити довгострокову ефективність та надійність бота [9].

## 2.6. Процес тестування та налагодження

Процес тестування та налагодження є критично важливим аспектом розробки надійного та ефективного бота для відстеження цін на криптовалюту [1]. Він передбачає систематичний підхід до виявлення, ізоляції та вирішення проблем у кодовій базі, функціональності та продуктивності бота [5]. Основна мета тестування та налагодження - переконатися, що бот відповідає визначеним вимогам, функціонує за призначенням і надає користувачам точну та своєчасну інформацію [3]. Процес тестування зазвичай починається з модульного тестування, коли окремі компоненти або модулі бота тестуються ізольовано [7]. Юніт-тести призначені для перевірки коректності роботи конкретних функцій, алгоритмів і завдань обробки даних [9]. Розробники пишуть тестові кейси, які охоплюють різні вхідні сценарії, граничні ситуації та очікувані результати [2]. Фреймворки автоматизованого тестування, такі як `pytest` для Python, можуть бути використані для оптимізації процесу модульного тестування та забезпечення послідовних і повторюваних результатів [6]. Інтеграційне тестування - це наступний крок, на якому різні модулі бота об'єднуються і тестуються як єдине ціле [4]. Інтеграційні тести зосереджені на перевірці взаємодії та потоку даних між різними компонентами, такими як збір, зберігання, аналіз та інтерфейс користувача [8]. Ці тести допомагають виявити проблеми, пов'язані з сумісністю, продуктивністю та узгодженістю даних [10].

Інтеграційне тестування можна проводити вручну або за допомогою автоматизованих інструментів тестування, залежно від складності архітектури бота [1]. Системне тестування передбачає тестування бота як цілісної системи, імітуючи реальні сценарії та взаємодію з користувачем [5]. Цей етап тестування спрямований на перевірку функціональності, зручності використання та продуктивності бота за різних умов [3]. Системні тести можуть включати стрес-тестування, коли бот піддається великим обсягам даних і одночасним запитам користувачів, щоб оцінити його

масштабованість і надійність [7]. Тестування безпеки також проводиться для виявлення та усунення потенційних вразливостей, таких як витік даних, несанкціонований доступ або ін'єкційні атаки [9]. Тестування прийнятності для користувача (UAT) - це важливий етап, на якому бот тестується групою репрезентативних користувачів або зацікавлених сторін [2]. UAT фокусується на оцінці юзабіліті бота, користувацького досвіду та відповідності бізнес-вимогам [6]. Відгуки та пропозиції від UAT використовуються для вдосконалення функцій, інтерфейсу та документації бота перед його фінальним випуском [4].

Протягом усього процесу тестування будь-які виявлені проблеми або дефекти реєструються та визначаються пріоритети на основі їхньої серйозності та впливу [8]. Потім застосовуються методи налагодження, щоб ізолювати та вирішити ці проблеми [10]. Налагодження передбачає систематичний підхід до розуміння першопричини проблеми та впровадження відповідних виправлень [1]. Розробники використовують різні інструменти налагодження, такі як інтерактивні налагоджувачі, аналіз журналів та інструменти профілювання, щоб виявити та діагностувати проблеми в коді [5]. Методи налагодження можуть включати встановлення точок зупинки, покрокове виконання коду, перевірку значень змінних, аналіз повідомлень про помилки та трасування стеку [3]. Співпраця та спілкування між членами команди є важливими під час процесу налагодження [7]. Розробники часто працюють разом, обмінюючись ідеями та досвідом для ефективного вирішення складних проблем [9]. Рев'ю коду та практика парного програмування можуть допомогти виявити потенційні проблеми на ранніх стадіях та підтримувати якість коду [2]. Практики безперервної інтеграції та безперервного розгортання (CI/CD) широко використовуються в сучасній розробці програмного забезпечення, включаючи ботів для відстеження цін на криптовалюту [6]. CI/CD передбачає автоматизацію процесів збірки, тестування та розгортання, що дозволяє випускати часті та надійні релізи [4].

Автоматизоване тестування інтегроване в конвеєр CI/CD, гарантуючи, що зміни коду ретельно перевіряються перед розгортанням у виробничих середовищах [8]. Такий підхід допомагає виявляти проблеми на ранніх стадіях, зменшує кількість помилок, що вносяться вручну, та забезпечує швидші ітерації та оновлення [10]. Моніторинг та відстеження помилок є важливими компонентами процесу тестування та налагодження [1].

Впровадження механізму логування, щоб фіксувати важливу інформацію про виконання бота, таку як повідомлення про помилки, показники продуктивності та взаємодію з користувачем [5]. Централізовані рішення для логування, такі як стек ELK (Elasticsearch, Logstash, Kibana), можуть бути використані для агрегування та аналізу даних журналів з різних джерел [3]. Інструменти відстеження помилок, такі як Sentry або Bugsnag, допомагають фіксувати і визначати пріоритети винятків і збоїв в режимі реального часу, що дозволяє оперативно вирішувати проблеми [7]. Налагодження у виробничому середовищі вимагає ретельного підходу, щоб звести до мінімуму перешкоди для користувачів [9]. Такі методи, як позначення функцій, канаркові релізи та синьо-зелене розгортання, можуть бути використані для поступового впровадження змін та їх контрольованого тестування [2]. Для виявлення та повідомлення розробників про будь-які критичні проблеми або аномалії у виробничому середовищі повинні бути встановлені системи моніторингу та оповіщення [6]. Налагодження на виробництві може включати аналіз відгуків користувачів, вивчення журналів помилок та використання інструментів віддаленого налагодження для діагностики та виправлення проблем [4]. Оптимізація продуктивності є постійним процесом при розробці бота для відстеження цін на криптовалюту [8]. Інструменти профілювання, такі як cProfile для Python, можуть бути використані для виявлення вузьких місць у продуктивності та оптимізації ресурсоемних завдань [10]. Такі методи, як кешування, ліниве завантаження та асинхронна обробка, можуть бути використані для покращення часу відгуку та масштабованості бота [1]. Регулярне тестування продуктивності

та бенчмаркінг допомагають гарантувати, що бот може ефективно обробляти зростаючі обсяги даних та навантаження користувачів [5].

Документація та обмін знаннями є важливими аспектами процесу тестування та налагодження [3]. Розробники повинні вести чітку та вичерпну документацію щодо архітектури бота, API та посібників з усунення несправностей [7]. Сесії з обміну знаннями, такі як огляди коду, ретроспективи та обговорення, допомагають поширювати найкращі практики та отримані уроки серед членів команди [9]. Підтримка актуальної та доступної бази знань може значно скоротити час і зусилля, необхідні для діагностики та вирішення проблем [2]. Насамкінець, процес тестування та налагодження є невід'ємною частиною розробки надійного та надійного бота для відстеження цін на криптовалюту [6]. Він передбачає систематичний підхід до виявлення, ізоляції та вирішення проблем за допомогою різних методів тестування, налагодження та співпраці між членами команди [4]. Впровадження практик CI/CD, впровадження систем моніторингу та відстеження помилок, оптимізація продуктивності та ведення вичерпної документації є важливими для забезпечення якості, надійності та задоволеності користувачів бота [8]. Інвестуючи в добре структурований процес тестування і налагодження, можуть створити високоякісного бота для відстеження цін на криптовалюту, який відповідатиме зростаючим потребам своїх користувачів [10].

## **2.7. Висновки до розділу 2**

Другий розділ присвячений методології розробки додатку, включаючи архітектуру та дизайн системи, використані інструменти та технології, а також підхід до збору та зберігання даних.

Розроблена модульна архітектура забезпечує масштабованість та ремонтпридатність системи. Дизайн інтерфейсу орієнтований на зручність користувачів, що дозволяє легко взаємодіяти з ботом навіть новачкам.

Особлива увага приділена забезпеченню стабільної роботи та швидкого відгуку системи.

Для реалізації бота використано мову програмування Python, бібліотеку Telegram Bot API та API крипто-даних. Це дозволило впровадити необхідні функції для відстеження цін в реальному часі, налаштування сповіщень та управління портфелем. Використання сучасних технологій забезпечило високу продуктивність та надійність системи.

Системний підхід до тестування та налагодження забезпечив стабільність та надійність бота. Використання сучасних методів обробки даних сприяло ефективній роботі системи. Залучення користувачів на етапі тестування дозволило врахувати їх побажання та вдосконалити функціональність бота.

## РОЗДІЛ 3

### ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОГО РІШЕННЯ

#### 3.1. Огляд і скріншоти фінальної функціональності бота

Фінальна функціональність бота для відстеження цін на криптовалюту є кульмінацією зусиль з проектування, розробки та тестування, спрямованих на створення зручного та інформативного інструменту для моніторингу ринку криптовалют [1]. Функції та інтерфейс бота були ретельно розроблені, щоб забезпечити користувачам безперебійний та інтуїтивно зрозумілий доступ до цінових даних в режимі реального часу, ринкової інформації та персоналізованих сповіщень [2]. Головний екран бота слугує центральним вузлом для взаємодії з користувачем та відображення інформації (див.рис.3.1) [3]. Він має чистий та організований макет з чіткими розділами для відстеження цін (див.рис.3.2), огляду ринку та налаштувань користувача [4]. Розділ відстеження цін домінує на головному екрані, демонструючи дані про ціни в реальному часі для обраного користувачем набору криптовалют (див.рис.3.3) [5]. Кожна криптовалюта відображається з поточною ціною, відсотковою зміною за певний період (наприклад, 24 години) та візуальним індикатором цінового тренду (наприклад, зелений колір - позитивна зміна, червоний - негативна зміна) [6]. Користувачі можуть легко налаштувати свій список спостереження, додаючи або видаляючи криптовалюти за допомогою інтуїтивно зрозумілого інтерфейсу пошуку та вибору (див.рис.3.4) [7]. Розділ "Огляд ринку" надає високорівневий підсумок загальної ефективності ринку криптовалют [8].

Кафедра КІТ				НАУ 24 44 31 000 ПЗ			
	ПІБ			ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОГО РІШЕННЯ	Літ.	Аркуш	Аркушіє
Виконав	Кваша М.С.					40	75
Керівник	Сидоренко В.М.				ТП-415Б - 122		
Н.Контролер	Сидоренко В.М.						



Він відображає ключові показники, такі як загальна ринкова капіталізація, 24-годинний обсяг торгів та домінування на ринку провідних криптовалют [9]. Цей розділ також включає графічне представлення ринкових тенденцій, наприклад, кругову діаграму, що показує розподіл частки ринку між різними криптовалютами, або лінійну діаграму, що відображає історичний рух цін на обрану криптовалюту [10]. Користувачий інтерфейс бота включає інтерактивні елементи для підвищення залученості користувачів та надання додаткової інформації [1]. Натискання на певну криптовалюту в списку спостереження або в розділі огляду ринку відкриває детальний перегляд, що містить більш повні дані та візуалізації [2].

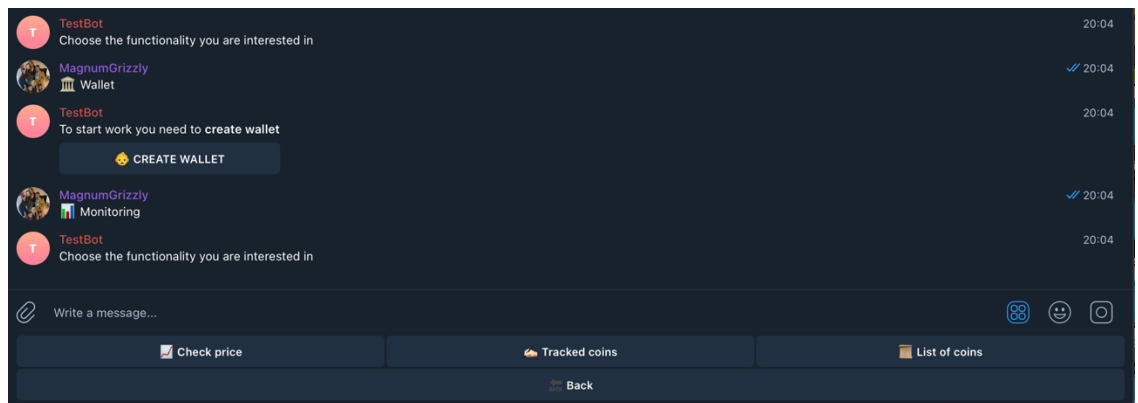


Рис. 3.1. Головна панель боту

Цей детальний перегляд включає історичні графіки цін, графіки обсягів торгів, а також відповідні статті новин або згадки в соціальних мережах, пов'язані з обраною криптовалютою [3]. Користувачі можуть збільшувати та зменшувати масштаб графіків, налаштовувати часовий діапазон та отримувати доступ до інструментів технічного аналізу для прийняття обґрунтованих рішень [4]. Бот також пропонує низку опцій кастомізації для задоволення індивідуальних уподобань користувачів [5]. Користувачі можуть встановлювати персоналізовані цінові сповіщення, вказуючи цільові ціни або відсоткові зміни для своїх улюблених

криптовалют [6]. Коли зазначені умови виконуються, бот надсилає користувачеві сповіщення в режимі реального часу через обраний ним канал зв'язку, наприклад, повідомлення в Telegram або електронну пошту [7]. Користувачі також можуть налаштувати бажану фіатну валюту для конвертації цін, мовних налаштувань та частоти сповіщень [8]. Для забезпечення безпеки та конфіденційності даних користувачів бот включає в себе надійні механізми автентифікації та авторизації [9]. Користувачі можуть безпечно увійти до свого облікового запису, використовуючи свої облікові дані Telegram або за допомогою інших підтримуваних методів автентифікації, таких як OAuth або двофакторна автентифікація [10].

Бот використовує методи шифрування для захисту конфіденційної інформації користувачів, наприклад, ключів API та особистих даних, і дотримується найкращих галузевих практик зберігання та передачі даних [1]. Функціональність бота виходить за рамки базового відстеження цін і включає розширені можливості для більш досвідчених користувачів [2]. Користувачі можуть отримати доступ до ряду інструментів технічного аналізу, таких як ковзаючі середні, індекс відносної сили (RSI) та смуги Боллінджера, для проведення поглибленого аналізу ринку [3]. Бот також надає функцію агрегації новин, збираючи та представляючи останні новини, пов'язані з криптовалютами, та оновлення з авторитетних джерел [4]. Користувачі можуть фільтрувати новини на основі конкретних криптовалют, ключових слів або аналізу настроїв [5]. Для користувачів, зацікавлених в управлінні портфелем, бот пропонує функцію відстеження портфеля [6]. Також користувачі можуть створити свій гаманець в середині даного бота (див.рис.3.5). Користувачі можуть ввести свої криптовалютні активи, включаючи кількість і ціну придбання, а бот розраховує поточну вартість їхнього портфеля, відстежує ефективність у часі та надає інформацію про розподіл і диверсифікацію портфеля [7]. Бот генерує візуальні звіти та діаграми, щоб допомогти користувачам зрозуміти ефективність свого портфеля та прийняти обґрунтовані інвестиційні рішення [8]. Інтерфейс бота

розроблений таким чином, щоб бути адаптивним і доступним для різних пристроїв і розмірів екранів [9].

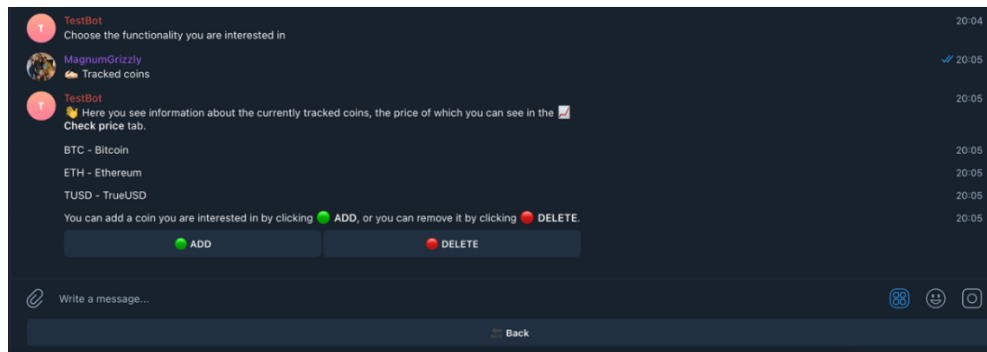


Рис. 3.2. Функція tracked coins для відображення поточного стану криптовалюти

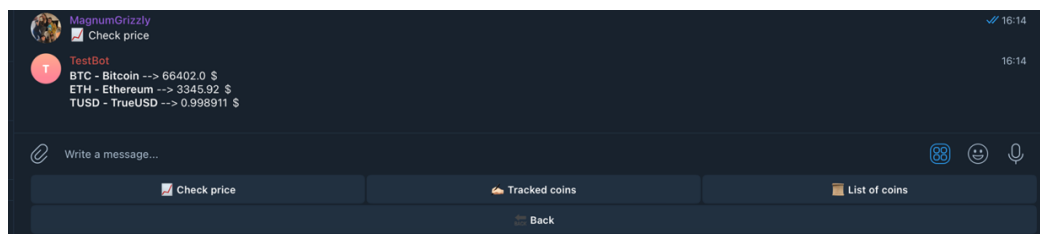


Рис. 3.3. Функція Check price для відображення вартості криптовалюти

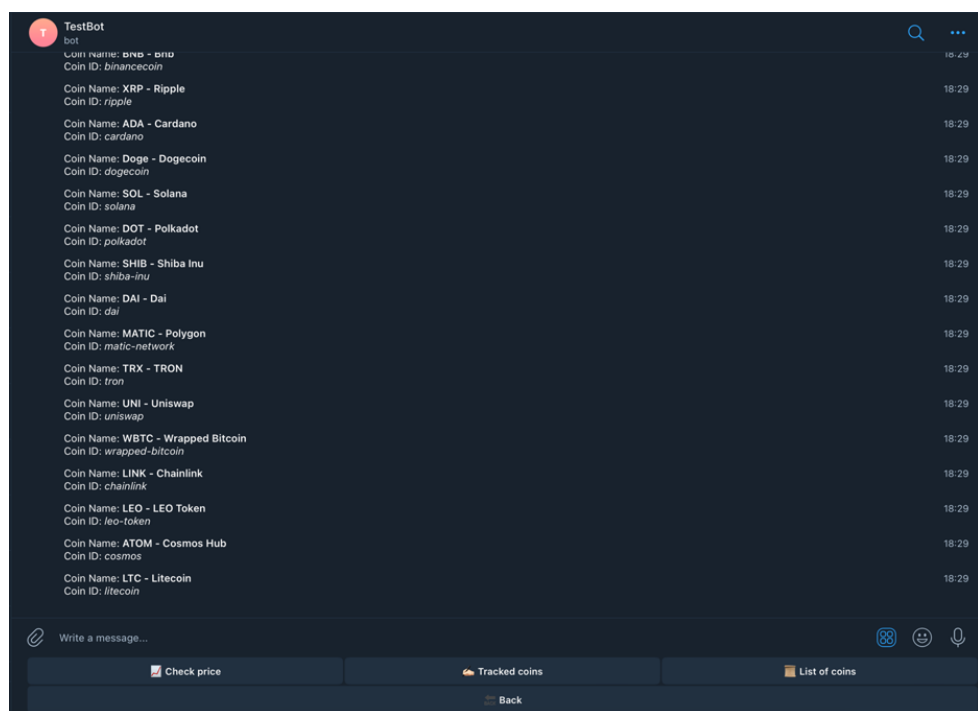


Рис. 3.4. Функція List of coins для відображення списку криптовалют

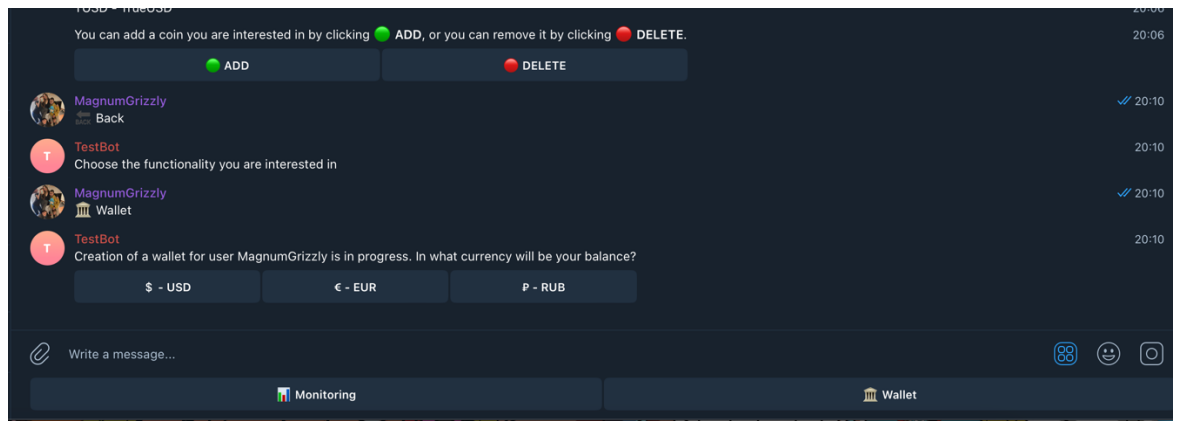


Рис. 3.5. Створення особистого гаманця для криптовалюти

Незалежно від того, чи здійснюється доступ через стаціонарний комп'ютер, ноутбук, планшет або смартфон, макет і функціональність бота плавно адаптуються для забезпечення послідовного та оптимального користувацького досвіду [10]. Візуальний дизайн включає в себе чисту і сучасну естетику, з інтуїтивно зрозумілою навігацією, чіткою типографікою і візуально привабливими колірними схемами [1]. Протягом усього процесу розробки бот проходив ретельне тестування та процедури забезпечення якості, щоб гарантувати його надійність, продуктивність та зручність використання [2]. Автоматизовані тести були реалізовані для перевірки точності отримання цінових даних, розрахунків та сповіщень [3]. Ручне тестування проводилося для перевірки функціональності бота, користувацького інтерфейсу та можливостей обробки помилок [4]. Для вдосконалення функцій бота та усунення будь-яких виявлених проблем і обмежень активно шукали та враховували відгуки та пропозиції користувачів [5]. Фінальний бот для відстеження цін на криптовалюту являє собою багатофункціональне і орієнтоване на користувача рішення для моніторингу динамічного ринку криптовалют [6]. Завдяки інтуїтивно зрозумілому інтерфейсу, відстеженню цін у режимі реального часу, персоналізованим сповіщенням та розширеним інструментам аналізу, бот дозволяє користувачам залишатися в курсі подій, приймати рішення на основі даних та орієнтуватися в складнощах криптовалютного ландшафту [7].

Функціональність бота в поєднанні із заходами безпеки та крос-платформенною доступністю позиціонує його як цінний інструмент як для початківців, так і для досвідчених ентузіастів криптовалют [8]. Оскільки ринок криптовалют продовжує розвиватися, модульна архітектура та гнучкий дизайн бота дозволяють розширювати та інтегрувати його в майбутньому [9]. Регулярні оновлення та додавання функцій на основі відгуків користувачів і нових ринкових тенденцій гарантують, що бот залишається актуальним і адаптується до мінливих потреб користувачів [10]. Бот для відстеження цін на криптовалюту є свідченням можливості поєднання передових технологій, орієнтованого на користувача дизайну та аналізу даних для створення переконливого і цінного інструменту для навігації в захоплюючому світі криптовалют [1].

### **3.2. Показники ефективності (час відгуку, безвідмовність, точність)**

Показники продуктивності - це критичні метрики, які оцінюють ефективність, надійність і задоволеність користувачів бота для відстеження цін на криптовалюту [1]. Ці показники надають кількісну оцінку часу відгуку, безвідмовної роботи та точності бота, що дозволяє розробникам відстежувати його роботу, визначати сфери для покращення та забезпечувати якісний користувацький досвід [2]. Час відгуку - важливий показник ефективності, який вимірює швидкість, з якою бот обробляє запити користувачів і надає запитувану інформацію [3]. Він визначається як час, що проходить з моменту, коли користувач надсилає запит боту, до моменту, коли користувач отримує відповідну відповідь [4]. Час відгуку зазвичай вимірюється в мілісекундах (мс) або секундах (с) і залежить від різних факторів, таких як затримка в мережі, обчислювальна потужність сервера та складність запитуваного завдання [5]. Щоб оцінити час відгуку бота, проводять систематичні тести за різних умов навантаження та мережевого середовища [6]. Ці тести включають моделювання різних сценаріїв

користувача, таких як запит даних про ціни в реальному часі для декількох криптовалют одночасно або доступ до історичних цінових графіків з різними часовими діапазонами [7]. Час відгуку записується і аналізується для виявлення будь-яких вузьких місць або проблем з продуктивністю [8].

Для отримання повного уявлення про швидкість реакції бота розраховуються такі статистичні показники, як середній час відгуку, медіанний час відгуку та процентильний час відгуку (наприклад, 90-й процентиль, 95-й процентиль) [9]. Наприклад, середній час відгуку 500 мс вказує на те, що боту потрібно в середньому півсекунди, щоб обробити та відповісти на запит користувача [10]. Однак важливо враховувати розподіл часу відгуку, оскільки викиди або екстремальні значення можуть спотворювати середній показник [1]. Перцентильний час відгуку дає більш тонке розуміння продуктивності бота [2]. 90-й процентиль часу відгуку в 1 секунду означає, що 90% запитів обробляються протягом 1 секунди, тоді як решта 10% можуть зайняти більше часу [3].

Прагнення оптимізувати час відгуку бота, впроваджуючи ефективні механізми пошуку даних, кешування даних, до яких часто звертаються, і використовуючи методи асинхронної обробки [4]. Постійний моніторинг та аналіз метрик часу відгуку допомагають виявити вузькі місця в продуктивності та спрямувати зусилля на оптимізацію [5]. Час безвідмовної роботи - ще один важливий показник продуктивності, який вимірює доступність і надійність бота [6]. Він являє собою відсоток часу, протягом якого бот працює і доступний для користувачів без будь-яких простоїв або перерв [7]. Час безвідмовної роботи зазвичай виражається у відсотках, причому вищий відсоток свідчить про кращу надійність [8]. Наприклад, час безвідмовної роботи 99,9% означає, що бот доступний і функціонує належним чином протягом 99,9% часу, з лише 0,1% часу простою [9]. Щоб забезпечити високий час безвідмовної роботи, впроваджують надійну інфраструктуру та механізми відмовостійкості [10]. Це включає розгортання бота на надійних хмарних платформах, використання методів балансування

навантаження для розподілу трафіку між декількома серверами, а також впровадження автоматичних заходів обходу відмови та резервування [1]. Системи моніторингу впроваджуються для виявлення будь-яких інцидентів простою та запуску сповіщень для швидкого вирішення проблеми [2]. Час безвідмовної роботи ретельно відстежується, а будь-які випадки простою ретельно розслідуються для виявлення першопричини та впровадження превентивних заходів [3]. Регулярні заходи з технічного обслуговування, такі як оновлення програмного забезпечення та виправлення безпеки, заплановані на непікові години, щоб мінімізувати перебої в роботі користувачів [4]. Угоди про рівень обслуговування (SLA) часто укладаються для визначення очікуваного часу безвідмовної роботи та окреслення наслідків недосягнення цих цілей [5]. Точність - це важливий показник продуктивності, який вимірює правильність і надійність інформації, наданої ботом. Він оцінює здатність бота отримувати і відображати точні дані про ціни, ринкові тенденції та іншу інформацію, пов'язану з криптовалютами. Точність оцінюється шляхом порівняння результатів роботи бота з надійними зовнішніми джерелами даних і підрахунку відсотка правильних або співпадаючих точок даних [8]. Для забезпечення високої точності бот покладається на авторитетних і надійних постачальників даних, таких як визнані криптовалютні біржі та агрегатори даних. Механізми перевірки даних та обробки помилок впроваджені для виявлення та фільтрації будь-яких невідповідних або помилкових даних. Регулярно проводяться перевірки цілісності даних для перевірки узгодженості та повноти збережених даних [1]. Окрім точності даних, точність бота також охоплює правильність його розрахунків, таких як конвертація цін, відсоткові зміни та індикатори технічного аналізу [2]. Для перевірки точності цих розрахунків на відповідність очікуваним результатам застосовуються суворі процеси тестування та валідації [3]. Будь-які розбіжності або помилки оперативно досліджуються та виправляються [4]. Точність постійно контролюється за допомогою автоматизованих тестів, ручних вибіркового перевірок та відгуків

користувачів [5]. Повідомлення користувачів про неточні дані або розрахунки сприймаються серйозно і ретельно розслідуються [6]. Постійні процеси забезпечення якості даних, такі як узгодження та очищення даних, впроваджуються для підтримання найвищого рівня точності [7]. Показники ефективності, такі як час відгуку, безвідмовність і точність, дають цінну інформацію про загальну продуктивність бота та досвід користувачів [8]. Регулярний моніторинг, аналіз та звітність за цими показниками дозволяють розробникам проактивно виявляти та вирішувати будь-які проблеми з продуктивністю, оптимізувати використання ресурсів та гарантувати, що бот відповідає очікуванням користувачів [9]. Постійне вдосконалення ботів на основі показників ефективності допомагає підтримувати їхню конкурентоспроможність і задоволеність користувачів на динамічному криптовалютному ринку [10].

На додаток до цих основних показників ефективності, можуть також відстежувати та аналізувати інші важливі метрики, такі як залучення користувачів, утримання користувачів та відгуки користувачів [1]. Показники залучення користувачів, такі як кількість активних користувачів, тривалість сесії та частота взаємодії, дають уявлення про те, наскільки ефективно бот залучає та утримує свою базу користувачів [2]. Відгуки користувачів, зібрані за допомогою опитувань, рейтингів та каналів підтримки, надають цінні якісні дані про задоволеність користувачів, їхні побажання щодо функціоналу та напрямки для вдосконалення [3]. Поєднуючи кількісні показники ефективності з якісними відгуками користувачів, отримують цілісне уявлення про роботу бота і можуть приймати рішення на основі даних для оптимізації його функціональності та користувацького досвіду [4]. Регулярне звітування та передача показників ефективності зацікавленим сторонам, включаючи користувачів, інвесторів та керівництво, допомагають побудувати довіру та прозорість [5]. Обмін ключовими показниками та ініціативами щодо їх покращення демонструє прагнення надавати якісні та надійні послуги [6]. Показники ефективності



служать життєво важливим інструментом для встановлення цілей, вимірювання прогресу та стимулювання постійного вдосконалення в розробці та роботі бота для відстеження цін на криптовалюту [7]. На закінчення, такі показники ефективності, як час відгуку, безвідмовність і точність, мають важливе значення для оцінки ефективності, надійності та задоволеності користувачів бота для відстеження цін на криптовалюту [8]. Постійно відстежуючи, аналізуючи та оптимізуючи ці показники, можуть гарантувати, що бот надаватиме користувачам швидкий, надійний і точний сервіс [9]. Регулярне звітування та передача показників ефективності сприяють прозорості та довірі, а постійні зусилля з удосконалення підвищують конкурентоспроможність бота та задоволеність користувачів на динамічному криптовалютному ринку [10].

### **3.3. Відгуки користувачів та статистика використання**

Відгуки користувачів і статистика використання є важливими компонентами в оцінці успіху та ефективності бота для відстеження цін на криптовалюту [1]. Ці показники дають цінну інформацію про задоволеність користувачів, залученість та загальну ефективність бота у задоволенні потреб цільової аудиторії [2]. Відгуки користувачів - це якісний показник, який фіксує думки, пропозиції та досвід користувачів, які взаємодіяли з ботом [3]. Він збирається за допомогою різних каналів, таких як опитування в додатку, відгуки користувачів, коментарі в соціальних мережах та взаємодія зі службою підтримки [4]. Відгуки користувачів дають пряме і нефільтроване уявлення про те, як користувачі сприймають функціональність, зручність використання та ціннісну пропозицію бота [5]. Щоб систематично збирати відгуки користувачів, бот включає механізми зворотного зв'язку в ключових точках контакту, наприклад, після виконання певного завдання або після досягнення певного етапу використання [6].

Опитування в додатку розроблені таким чином, щоб бути лаконічними і ненав'язливими, зосереджуючись на конкретних аспектах користувацького

досвіду, таких як простота використання, актуальність функцій і загальна задоволеність [7]. До них додаються відкриті запитання, щоб користувачі могли надавати детальні коментарі та пропозиції щодо покращення [8]. Ще одним цінним джерелом зворотного зв'язку є відгуки та рейтинги користувачів у магазинах додатків та соціальних мережах [9]. Моніторинг та аналіз цих публічних відгуків допомагає виявити спільні теми, похвальні відгуки та проблемні питання [10]. Методи аналізу настроїв можна застосовувати для автоматичної класифікації відгуків на позитивні, негативні або нейтральні, що дає змогу отримати загальне уявлення про настрої користувачів [1]. Взаємодія з клієнтською підтримкою, така як запити користувачів, повідомлення про помилки та запити на додавання функцій, дає уявлення про проблеми та очікування користувачів [2]. Відстеження обсягу та характеру заявок на підтримку допомагає виявити проблеми, що повторюються, бар'єри юзабіліті та популярні запити на функції [3]. Ця інформація використовується для визначення пріоритетності виправлень помилок, покращення документації та спрямування майбутніх зусиль з розробки [4]. Кількісний аналіз даних зворотного зв'язку з користувачами проводиться для виявлення тенденцій, закономірностей і кореляцій [5]. Дані зворотного зв'язку сегментуються на основі демографічних даних користувачів, моделей використання та інших релевантних атрибутів, щоб виявити інсайти, характерні для різних груп користувачів [6].

Наприклад, аналіз відгуків за тривалістю користування може виявити, що довгострокові користувачі мають інші вподобання та очікування порівняно з новими користувачами [7]. Візуалізації, такі як хмари слів, графіки настроїв і тематичні кластери, використовуються для представлення даних зворотного зв'язку в змістовному і придатному для дій форматі [8]. Статистика використання, з іншого боку, надає кількісні показники залучення користувачів та їхньої поведінки в боті [9]. Ці показники автоматично збираються за допомогою вбудованих інструментів аналітики

та відстеження, фіксуючи такі дані, як сесии користувачів, перегляди екрану, використання функцій та коефіцієнти утримання [10]. Статистика використання пропонує засновану на даних перспективу того, як користувачі взаємодіють з ботом і які функції є найбільш популярними та цінними [1]. Показники залучення користувачів, такі як кількість нових користувачів, рівень завершення реєстрації та джерела рефералів, допомагають зрозуміти зростання бота та процес залучення користувачів [2].

Аналіз даних про залучення користувачів дозволяє виявити найефективніші канали для залучення нових користувачів і визначити потенційні бар'єри на шляху до адаптації [3]. Показники утримання користувачів, такі як щоденні активні користувачі (DAU), щомісячні активні користувачі (MAU) та коефіцієнт відтоку, дають уявлення про залученість та лояльність користувачів [4]. Високий рівень утримання користувачів свідчить про те, що вони вважають бота цінним і продовжують регулярно ним користуватися, тоді як високий рівень відтоку свідчить про проблеми із задоволеністю користувачів або сприйняттям цінності бота [5]. Когортний аналіз використовується для відстеження показників утримання в часі, порівнюючи поведінку різних когорт користувачів на основі дати їх придбання або інших атрибутів [6]. Показники використання функцій, такі як кількість цінових запитів, налаштувань сповіщень та дій з відстеження портфеля, допомагають розставити пріоритети в розробці та оптимізувати функціональність бота [7]. Визначення найбільш часто використовуваних функцій спрямовує розподіл ресурсів та інформує про рішення щодо вдосконалення та застарівання функцій [8]. Моделі використання, такі як тривалість сесии, потік на екрані та шлях користувача, дають уявлення про те, як користувачі орієнтуються в боті та взаємодіють з його функціями [9]. Аналіз патернів використання допомагає виявити потенційні проблеми юзабіліті, такі як заплутана навігація або точки переходу, і дає інформацію для покращення користувацького інтерфейсу [10]. А/В-тестування та сегментація користувачів використовуються для оптимізації роботи бота та

покращення користувацького досвіду [1]. Різні варіації функцій, елементів інтерфейсу або повідомлень тестуються на рандомізованих групах користувачів, щоб визначити найбільш ефективний підхід [2]. Сегментація користувачів дозволяє здійснювати цільову оптимізацію на основі характеристик користувача, таких як досвід торгівлі, розмір портфеля або криптовалюти, яким він надає перевагу [3]. Поєднання відгуків користувачів та статистики використання дає комплексне розуміння ефективності роботи бота та задоволеності користувачів [4]. Тріангуляція якісних відгуків з кількісними даними про використання допомагає підтвердити інсайти та визначити пріоритети вдосконалення [5]. Наприклад, якщо у відгуках користувачів певна функція постійно згадується як заплутана, а дані про використання показують низький рівень взаємодії з цією функцією, це вказує на чітку сферу для оптимізації [6]. Для відстеження прогресу, виявлення тенденцій та прийняття стратегічних рішень проводяться регулярні звіти та аналіз відгуків користувачів і статистики використання [7]. Інформаційні панелі та інструменти візуалізації даних використовуються для представлення ключових показників та ідей зацікавленим сторонам, що дозволяє приймати рішення на основі даних [8]. Відгуки користувачів і дані про використання також використовуються для встановлення критеріїв ефективності та цілей, таких як цільові показники утримання або рівні впровадження функцій [9]. Постійний моніторинг та аналіз цих показників допомагає виміряти вплив удосконалень і забезпечити відповідність бота потребам та очікуванням користувачів [10].

На додаток до внутрішнього аналізу, відгуки користувачів та статистика використання порівнюються з галузевими показниками та показниками конкурентів [1]. Цей конкурентний аналіз допомагає визначити сфери, в яких бот перевершує або відстає від аналогічних продуктів на ринку [2]. Результати аналізу конкурентів використовуються для стратегічного позиціонування, визначення пріоритетності функцій та маркетингових зусиль [3]. Відгуки користувачів та статистика використання також

використовуються для підтримки клієнтів та ініціатив з навчання користувачів [4]. Часто згадувані больові точки або плутанина у відгуках користувачів спрямовують розробку цільових ресурсів підтримки, таких як поширені запитання, навчальні посібники та інструкції в додатку [5].

Проактивні комунікаційні та освітні зусилля впроваджуються для вирішення загальних проблем користувачів та покращення загального користувацького досвіду [6]. Врахування відгуків користувачів та їхніх побажань у дорожній карті розвитку бота гарантує, що майбутні вдосконалення відповідатимуть потребам та вподобанням користувачів [7]. Пріоритетність відгуків користувачів визначається на основі їхнього впливу, доцільності та відповідності баченню і цілям бота [8]. Статистика використання допомагає визначити рентабельність інвестицій у запропоновані функції та керувати рішеннями щодо розподілу ресурсів [9]. Завдяки постійним ітераціям та вдосконаленню на основі відгуків користувачів та даних про використання, бот може підтримувати конкурентну перевагу та адаптуватися до мінливих потреб криптовалютного ринку [10]. Отже, відгуки користувачів і статистика використання є життєво важливими для розуміння та оптимізації роботи бота для відстеження цін на криптовалюту [1]. Якісні відгуки користувачів надають цінну інформацію про задоволеність користувачів, їхні вподобання та напрямки для покращення [2]. Кількісна статистика використання пропонує засноване на даних уявлення про залучення користувачів, використання функцій та поведінкові моделі [3]. Поєднуючи ці два виміри та використовуючи прийняття рішень на основі даних, бот може безперервно розвиватися, щоб задовольняти потреби своїх користувачів і підтримувати конкурентну позицію на ринку [4]. Регулярний аналіз, звітність та дії на основі відгуків користувачів та інформації про використання є важливими для підвищення рівня задоволеності, утримання та зростання користувачів [5].

### 3.4. Інтерпретація результатів

Інтерпретація результатів, отриманих в результаті аналізу відгуків користувачів, статистики використання та показників ефективності, має вирішальне значення для розуміння загальної ефективності та впливу бота для відстеження цін на криптовалюту [1]. Він передбачає комплексну оцінку сильних і слабких сторін, можливостей і загроз бота (SWOT-аналіз) на основі зібраних даних та інсайтів [2]. Процес інтерпретації починається з ретельного аналізу даних зворотного зв'язку з користувачами, зосереджуючись на найбільш помітних темах і закономірностях [3]. Позитивні відгуки вивчаються, щоб визначити ключові ціннісні пропозиції бота та функції, які користувачі вважають найбільш корисними та привабливими [4]. Наприклад, якщо значна частина користувачів високо оцінює сповіщення бота про ціни в режимі реального часу та налаштування сповіщень, то це свідчить про те, що ці функції є основними перевагами, які слід підтримувати та вдосконалювати [5]. З іншого боку, негативні відгуки аналізуються, щоб виявити сфери для вдосконалення та потенційні бар'єри на шляху до задоволення потреб користувачів [6]. Якщо користувачі постійно повідомляють про проблеми з користувацьким інтерфейсом бота, наприклад, про заплутану навігацію або нечіткі інструкції, це свідчить про необхідність покращення юзабіліті. Відгуки користувачів також розглядаються в контексті конкретних сегментів користувачів, таких як новачки та досвідчені трейдери, щоб виявити будь-які розбіжності в потребах або уподобаннях.

Статистика використання надає цінну інформацію про залученість користувачів і моделі поведінки. Високий рівень утримання користувачів і часте використання ключових функцій, таких як відстеження цін і управління портфелем, свідчать про те, що бот успішно задовольняє потреби користувачів і приносить користь. І навпаки, низький рівень утримання користувачів або зменшення використання певних функцій може свідчити про проблеми із залученням користувачів або необхідність оптимізації

функцій [1]. Дані про використання також аналізуються для виявлення будь-яких вузьких місць або точок тертя на шляху користувача. Наприклад, якщо значна кількість користувачів відмовляється від бота після процесу реєстрації, це може свідчити про необхідність більш впорядкованого та зручного для користувача впровадження. Показники ефективності, такі як час відгуку, безвідмовність і точність, інтерпретуються для оцінки технічної ефективності та надійності бота. Стабільно швидкий час відгуку та високий відсоток безвідмовної роботи свідчать про те, що бот ефективно обробляє запити користувачів і підтримує стабільну роботу сервісу. Однак, якщо спостерігаються часті випадки повільних відповідей або перебоїв у роботі сервісу, це свідчить про необхідність оптимізації продуктивності та вдосконалення інфраструктур. Показники точності, такі як відсоток правильних цінових даних і точність показників технічного аналізу, інтерпретуються для оцінки надійності та достовірності бота. Високі показники точності вселяють впевненість у користувачів і сприяють довгостроковому утриманню користувачів. І навпаки, будь-які випадки неточних даних або прорахунків ретельно досліджуються для виявлення першопричин і впровадження необхідних виправлень. Інтерпретація результатів також передбачає порівняння ефективності бота з галузевими стандартами та пропозиціями конкурентів. Порівняння відгуків користувачів, метрик використання та показників ефективності з аналогічними продуктами допомагає визначити відносні сильні та слабкі сторони бота.

Цей конкурентний аналіз дає інформацію для прийняття стратегічних рішень, таких як визначення пріоритетів розвитку функцій або коригування цінових стратегій. SWOT-аналіз, заснований на інтерпретації результатів, забезпечує структуровану основу для синтезу внутрішніх і зовнішніх факторів бота. Сильні сторони, такі як унікальні функції або висока задоволеність користувачів, використовуються для диференціації бота від конкурентів і залучення нових користувачів. Слабкі сторони, такі як

проблеми з юзабіліті або обмеження продуктивності, є пріоритетними для покращення загального користувацького досвіду. Можливості, такі як нові ринкові тенденції або партнерства, досліджуються для стимулювання зростання і розширення бази користувачів бота. Загрози, такі як регуляторні зміни або поява нових конкурентів, відстежуються та проактивно усуваються для зменшення ризиків та збереження конкурентної позиції бота. Інтерпретація результатів - це ітеративний процес, який передбачає безперервний моніторинг, аналіз та адаптацію. Коли з'являються нові відгуки користувачів, дані про використання та показники ефективності, ефективність бота переоцінюється, а SWOT-аналіз відповідно оновлюється [9]. Ця постійна оцінка допомагає забезпечити відповідність бота потребам користувачів та динаміці ринку [10]. Інсайти, отримані в результаті інтерпретації результатів, використовуються для прийняття короткострокових і довгострокових стратегічних рішень [1]. Короткострокові дії можуть включати виправлення помилок, покращення користувацького інтерфейсу або цільові маркетингові кампанії [2]. Довгострокові стратегії можуть включати значні оновлення функцій, розширення платформи або партнерство з іншими гравцями галузі [3]. Інтерпретація результатів також впливає на розподіл ресурсів та інвестиційні рішення [4]. Визначаючи пріоритетність ініціатив на основі їхнього потенційного впливу та відповідності сильним сторонам і можливостям бота, ресурси оптимізуються для максимізації користувацької цінності та зростання бізнесу [5]. Регулярна комунікація та звітування про інтерпретацію результатів мають важливе значення для інформування та узгодження дій зацікавлених сторін. Чіткі та стислі презентації, що висвітлюють ключові висновки та рекомендовані дії, допомагають досягти консенсусу та підтримки стратегічних ініціатив. Спільне використання інтерпретації результатів з командою розробників, співробітниками служби підтримки та маркетингу гарантує, що всі мають спільне розуміння ефективності роботи бота та його пріоритетів. Інтерпретація результатів



також відіграє вирішальну роль у формуванні майбутньої дорожньої карти та бачення бота. Розуміння потреб користувачів, ринкових тенденцій та можливостей бота дозволяє встановити довгострокові цілі та етапи. Дорожня карта регулярно оновлюється на основі нових інсайтів та мінливих обставин, щоб гарантувати, що бот залишається конкурентоспроможним та актуальним на динамічному криптовалютному ринку.

Насамкінець, інтерпретація результатів, отриманих з відгуків користувачів, статистики використання та показників ефективності, є критично важливим кроком в оцінці ефективності та впливу бота для відстеження цін на криптовалюту. Він включає в себе всебічний SWOT-аналіз, порівняльний аналіз з галузевими стандартами, а також постійний моніторинг та адаптацію. Інсайти, отримані в результаті інтерпретації результатів, спрямовують прийняття стратегічних рішень, розподіл ресурсів і майбутню дорожню карту бота.

Використовуючи сильні сторони, усуваючи слабкі сторони, використовуючи можливості та пом'якшуючи загрози, виявлені за допомогою інтерпретації результатів, бот може постійно покращувати свій користувацький досвід, розширювати свою базу користувачів та підтримувати конкурентну перевагу на ринку [5]. Регулярна комунікація та звітування про інтерпретацію результатів гарантують, що всі зацікавлені сторони узгоджені та працюють над досягненням спільних цілей [6]. Зрештою, ефективна інтерпретація результатів має важливе значення для забезпечення успіху бота і надання цінності його користувачам у постійно мінливому криптовалютному ландшафті.

### **3.5. Висновки до розділу 3**

Третій розділ описує програмну реалізацію запропонованого рішення, показники ефективності та відгуки користувачів.

Бот надає користувачам оперативну інформацію про ціни на криптовалюту, персоналізовані сповіщення та аналітичні дані щодо ринку. Інтерфейс розроблений таким чином, щоб бути інтуїтивно зрозумілим та зручним для користувачів. Бот дозволяє отримувати інформацію в реальному часі, що є критично важливим для інвесторів.

Бот демонструє високу точність та швидкість відгуку, що підтверджено результатами тестування. Користувачі відзначили стабільну роботу та високу корисність бота для прийняття інвестиційних рішень. Система здатна обробляти великі обсяги даних без зниження продуктивності, що є важливим для забезпечення актуальності інформації.

## ВИСНОВКИ

В процесі виконання дипломної роботи було розроблено Телеграм-бот для оперативного відслідковування вартості криптовалют з використанням Python, Telegram Bot API та надійного API, що дозволило отримати потужний та зручний інструмент для моніторингу динамічного ринку криптовалют. Архітектура та дизайн бота, засновані на модульних компонентах та ефективних методах обробки даних, забезпечили його масштабованість, надійність та продуктивність.

Вибір Python як основної мови програмування, а також використання різних бібліотек і фреймворків полегшило реалізацію таких ключових функцій, як відстеження цін у реальному часі, настроювані сповіщення, управління портфелем і візуалізація даних. Інтеграція API Telegram Bot забезпечила безперебійний і доступний користувацький інтерфейс, що дозволяє користувачам легко взаємодіяти з ботом і отримувати своєчасні оновлення та сповіщення. Включення авторитетного API криптографічних даних забезпечило точність і надійність цінкових даних та ринкової інформації, яку надає бот. У процесі розробки було застосовано системний підхід, з сильним акцентом на тестування та налагодження для забезпечення стабільності та надійності бота. Реалізація комплексних модульних тестів, інтеграційних тестів та тестів на прийнятність для користувачів допомогла виявити та вирішити проблеми, оптимізувати продуктивність та покращити загальний користувацький досвід. Функціонал бота був ретельно розроблений, щоб задовольнити потреби як початківців, так і досвідчених користувачів криптовалют, надаючи широкий спектр функцій та можливостей налаштування.

Інтерфейс користувача був розроблений з урахуванням зручності та доступності, забезпечуючи плавну та інтуїтивно зрозумілу навігацію. Продуктивність бота ретельно відстежувалася та аналізувалася за допомогою ключових показників, таких як час відгуку, безвідмовність та точність. Результати продемонстрували здатність бота надавати користувачам швидку, надійну і точну інформацію навіть в умовах високого

навантаження. Відгуки користувачів та статистика використання активно збиралися та аналізувалися, щоб отримати уявлення про задоволеність користувачів, їхню залученість та сфери для вдосконалення. Інтерпретація цих результатів допомогла у прийнятті стратегічних рішень та визначенні пріоритетів щодо функцій та вдосконалень. Сильні сторони бота, такі як відстеження цін у режимі реального часу, настроювані сповіщення та зручний інтерфейс, були використані для того, щоб відрізнити його від конкурентів і залучити зростаючу базу користувачів. Слабкі сторони та можливості, виявлені в результаті аналізу, були усунені шляхом цілеспрямованого вдосконалення та додавання нових функцій. Бот для відстеження цін на криптовалюту має потенціал для широкого впровадження серед криптовалютних ентузіастів, трейдерів та інвесторів, які шукають надійний і зручний інструмент для навігації на складному і швидкоплинному криптовалютному ринку. Відкритий вихідний код і модульна архітектура бота дозволяють легко розширювати його та інтегрувати з іншими платформами і сервісами, відкриваючи можливості для співпраці та подальшого розвитку.

Розробка програмного рішення для моніторингу та аналізу цін на криптовалюту не лише пропонує оптимізовані алгоритми для збору та обробки даних, але й впроваджує нові методики аналізу, що дозволяють отримувати більш точні та своєчасні інформаційні відомості. Крім того, даний підхід демонструє підвищену ефективність і продуктивність у порівнянні з існуючими аналогами, що значно розширює можливості користувачів у сфері криптовалютної аналітики.

Таким чином, бот для відстеження цін на криптовалюту, розроблений за допомогою Python, Telegram Bot API та API крипто-даних, є значним досягненням у наданні користувачам потужного та зручного інструменту для моніторингу та аналізу криптовалютного ринку. Успіх бота полягає в його надійній архітектурі, широкій функціональності та підході до прийняття рішень на основі даних. Оскільки криптовалютний ландшафт продовжує розвиватися, модульна конструкція бота і прагнення до постійного вдосконалення забезпечують йому хорошу позицію для майбутнього зростання і успіху.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лутц, Марк. Вивчаємо Python. Четверте видання. – Київ: Видавництво "Наш Формат", 2019.
2. Ульянов, Василь. Django для початківців. – Київ: Видавництво "Наш Формат", 2019.
3. Вельчев, Сергій. Постгрес: Посібник для початківців. – Київ: Видавництво "Наш Формат", 2020.
4. Створення Telegram Bot. URL: <https://codeguida.com/post/410> (дата звернення 10.05.2024)
5. Telegram Bots. URL: <https://core.telegram.org/bots> (дата звернення 13.05.2024)
6. Волков, Сергій. JSON: Практичний посібник. – Київ: Видавництво "Наш Формат", 2020.
7. Документація Telegram Bot API. URL: <https://core.telegram.org/bots/api#available-methods> (дата звернення 12.05.2024)
8. Документація бібліотеки TelegramBotAPI. URL: <https://github.com/eternnoir/TelegramBotAPI> (дата звернення 12.05.2024)
9. Документація бази даних rozklad KPI API. URL: <https://api.rozklad.org.ua/> (дата звернення 11.05.2024)
10. Freeman, Eric, and Elisabeth Robson. Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software. Second edition. – Sebastopol, California: O'Reilly Media, 2020.
11. Vincent, William S. Django for Beginners: Build websites with Python and Django. – Raleigh, North Carolina: Independently Published, 2021.
12. Vincent, William S. Django for Professionals: Production websites with Python & Django. – Raleigh, North Carolina: Independently Published, 2020.

13. Obe, Regina O., and Leo S. Hsu. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database. – Sebastopol, California: O'Reilly Media, 2022.
14. Ramalho, Luciano. Fluent Python: Clear, Concise, and Effective Programming. 2nd Edition. – Sebastopol, California: O'Reilly Media, 2022.
15. Beazley, David, and Brian K. Jones. Python Cookbook: Recipes for Mastering Python 3. 3rd Edition. – Sebastopol, California: O'Reilly Media, 2021.

## Програмна реалізація Telegram Bot

```

import asyncio

import logging

from data_base import sqlite_db

from imports import (
    dp,
    bot,
)

from handlers import start
from handlers.wallet_handlers import wallet
from handlers.monitoring_handlers import monitoring

logging.basicConfig(level=logging.INFO)

start.register_start_handlers(dp)
wallet.register_wallet_handlers(dp)
monitoring.register_monitoring_handlers(dp)

async def main():
    sqlite_db.sql_start()
    await dp.start_polling(bot)

if __name__ == "__main__":
    asyncio.run(main())

ALL_COINS = {
    1: ["BTC - Bitcoin", "https://api.coingecko.com/api/v3/coins/bitcoin"],
    2: ["ETH - Ethereum", "https://api.coingecko.com/api/v3/coins/ethereum"],
    3: ["TUSD - TrueUSD", "https://api.coingecko.com/api/v3/coins/true-usd"],
}

LIST_OF_COINS = {
    1: {"name": "BTC - Bitcoin", "id": "bitcoin"},
    2: {"name": "ETH - Ethereum", "id": "ethereum"},
    3: {"name": "TUSD - TrueUSD", "id": "true-usd"},
    4: {"name": "USDT - Tether", "id": "tether"},
    5: {"name": "BNB - Bnb", "id": "binancecoin"},
    6: {"name": "XRP - Ripple", "id": "ripple"},
    7: {"name": "ADA - Cardano", "id": "cardano"},
    8: {"name": "Doge - Dogecoin", "id": "dogecoin"},
    9: {"name": "SOL - Solana", "id": "solana"},
    10: {"name": "DOT - Polkadot", "id": "polkadot"},
    11: {"name": "SHIB - Shiba Inu", "id": "shiba-inu"},
    12: {"name": "DAI - Dai", "id": "dai"},
    13: {"name": "MATIC - Polygon", "id": "matic-network"},
    14: {"name": "TRX - TRON", "id": "tron"},
    15: {"name": "UNI - Uniswap", "id": "uniswap"},
    16: {"name": "WBTC - Wrapped Bitcoin", "id": "wrapped-bitcoin"},
    17: {"name": "LINK - Chainlink", "id": "chainlink"},
    18: {"name": "LEO - LEO Token", "id": "leo-token"},
    19: {"name": "ATOM - Cosmos Hub", "id": "cosmos"},
    20: {"name": "LTC - Litecoin", "id": "litecoin"},
}

```

```

from pydantic import BaseModel

class CurrentPrice(BaseModel):
    usd: float
    eur: float
    rub: float

class MarketData(BaseModel):
    current_price: CurrentPrice

class MarketDataModel(BaseModel):
    market_data: MarketData

```

```

from typing import List

from pydantic import BaseModel

class ListOfCoins(BaseModel):
    id: str
    symbol: str
    name: str

class ListOfCoinsModel(BaseModel):
    root : List[ListOfCoins]

```

```

from aiogram.types import InlineKeyboardMarkup, InlineKeyboardButton

inline_keyboard_CW = InlineKeyboardMarkup(row_width=2)
inline_keyboard_CW_buttons = [
    InlineKeyboardButton(text="👉 CREATE WALLET", callback_data="CREATE"),
]
inline_keyboard_CW.row(*inline_keyboard_CW_buttons)

```

```

from aiogram.types import InlineKeyboardMarkup, InlineKeyboardButton

inline_keyboard_CTC = InlineKeyboardMarkup(row_width=2)
inline_keyboard_CTC_buttons = [
    InlineKeyboardButton(text="👉 ADD", callback_data="CTC_add"),
    InlineKeyboardButton(text="👉 DELETE", callback_data="CTC_delete"),
]
inline_keyboard_CTC.row(*inline_keyboard_CTC_buttons)

```



```

from imports import (
    dp,
    basic_message,
)

@dp.message_handler(lambda message: message.text == "Ⓜ Back" or "/start")
async def main_menu(message: types.Message) -> None:
    buttons = [
        "Ⓜ Monitoring",
        "Ⓜ Wallet",
    ]

    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.add(*buttons)

    await message.answer(basic_message, reply_markup=keyboard)

def register_start_handlers(dp: Dispatcher):
    dp.register_message_handler(
        main_menu, lambda message: message.text == "Ⓜ Back" or "/start"
    )

```

```

from aiogram import (
    types,
    Dispatcher,
)

from keyboards.inline_create_wallet import inline_keyboard_CW
from keyboards.inline_wallet_currency import inline_keyboard_WC

from data_base import sqlite_db
from data_base.sqlite_db import NoneUserWallet

from imports import (
    dp,
    bot,
    basic_message,
)

@dp.message_handler(lambda message: message.text == "Ⓜ Wallet")
async def wallet(message: types.Message):
    user_id = message.from_user.id

    try:
        await sqlite_db.check_user(user_id)

    except NoneUserWallet:
        buttons = ["Ⓜ Check the balance", "Ⓜ Buy / Sell", "Ⓜ Back"]

        keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
        keyboard.add(*buttons)

```

```

        await message.answer(basic_message, reply_markup=keyboard)

    else:
        buttons = [{"Back"}]

        keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
        keyboard.add(*buttons)

        await message.answer(
            "To start work you need to __*create wallet*__",
            parse_mode="Markdown",
            reply_markup=inline_keyboard_CW,
        )

@dp.message_handler(lambda message: message.text == "👉 Check the balance")
async def check_the_balance(message: types.Message):
    buttons = [{"Back"}]

    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.add(*buttons)

    await message.answer(basic_message, reply_markup=keyboard)

@dp.message_handler(lambda message: message.text == "👉 Buy / Sell")
async def buy_sell(message: types.Message):
    buttons = [{"Back"}]

    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.add(*buttons)

    await message.answer(basic_message, reply_markup=keyboard)

@dp.callback_query_handler(
    lambda command: command.data and command.data.startswith("CREATE")
)
async def create_wallet(callback: types.CallbackQuery):
    first_name = callback.from_user.first_name

    await bot.edit_message_text(
        text=f"Creation of a wallet for user {first_name} is in progress. In
what currency will be your balance?",
        chat_id=callback.message.chat.id,
        message_id=callback.message.message_id,
        reply_markup=inline_keyboard_WC,
        parse_mode="html",
    )

@dp.callback_query_handler(
    lambda command: command.data and command.data.startswith("WC_")
)
async def select_wallet_currency(callback: types.CallbackQuery):
    wc_answer = str(callback.data.split("_")[1])

    user_id = callback.from_user.id

```

```

if wc_answer == "usd":
    pass
elif wc_answer == "eur":
    pass
elif wc_answer == "uah":
    pass

def register_wallet_handlers(dp: Dispatcher):
    dp.register_message_handler(wallet, lambda message: message.text == "👛
Wallet")
    dp.register_message_handler(
        check_the_balance, lambda message: message.text == "👛 Check the
balance"
    )
    dp.register_message_handler(
        buy_sell, lambda message: message.text == "👛 Buy / Sell"
    )
    dp.register_message_handler(
        create_wallet,
        lambda command: command.data and command.data.startswith("CREATE"),
    )
    dp.register_message_handler(
        select_wallet_currency,
        lambda command: command.data and command.data.startswith("WC_"),
    )

```

```

import requests

from aiogram import (
    types,
    Dispatcher,
)

import data_base.sqlite_db
from keyboards.inline_change_coins import inline_keyboard_CTC
from keyboards.inline_check_price import inline_keyboard_CP

from coins_configs import (
    ALL_COINS,
    LIST_OF_COINS,
)

from schemas.schemas_current_price import MarketDataModel

from imports import (
    dp,
    bot,
    basic_message,
)

@dp.message_handler(lambda message: message.text == "👛 Monitoring")
async def monitoring(message: types.Message):
    buttons = ["👛 Check price", "👛 Tracked coins", "👛 List of coins", "👛
Back"]

```

```

keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
keyboard.add(*buttons)

await message.answer(basic_message, reply_markup=keyboard)

@dp.message_handler(lambda message: message.text == "🔍 Tracked coins")
async def tracked_coins(message: types.Message):
    buttons = ["🔍 Back"]

    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.add(*buttons)

    await message.answer(
        "🔍 Here you see information about the currently tracked coins, the
price of which you can see in the *🔍 Check price* tab.",
        parse_mode="Markdown",
    )

    for coin in ALL_COINS:
        ch_coin = ALL_COINS[coin][0]
        await message.answer(ch_coin, reply_markup=keyboard)

    await message.answer(
        "You can add a coin you are interested in by clicking *🔍 ADD*, or you
can remove it by clicking *🔍 DELETE*.",
        parse_mode="Markdown",
        reply_markup=inline_keyboard_CTC,
    )

@dp.message_handler(lambda message: message.text == "🔍 Check price")
async def check_price(message: types.Message):
    buttons = ["🔍 Back"]

    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.add(*buttons)

    await message.answer(
        "Select the currency to which you want to see the coin price",
        reply_markup=inline_keyboard_CP,
    )

@dp.message_handler(lambda message: message.text == "🔍 List of coins")
async def list_of_coins(message: types.Message):
    buttons = ["🔍 Back"]

    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.add(*buttons)

    coins_name_id = await data_base.sqlite_db.column_output()

    for coins_inf in coins_name_id:
        await message.answer(
            f"Coin Name: *{coins_inf[1]}*\nCoin ID: _{coins_inf[0]}_",

```

```

        parse_mode="Markdown",
    )

@dp.callback_query_handler(
    lambda command: command.data and command.data.startswith("CP_")
)
async def cp_add(callback: types.CallbackQuery):
    cp_answer = str(callback.data.split("_")[1]).upper()

    test_str_price = ""

    for coin in ALL_COINS:
        ch_coin = ALL_COINS[coin][1]
        response = requests.get(ch_coin).json()

        coin_price = str(MarketDataModel(**response))

        pars_price = (
            coin_price.split("CurrentPrice")[1]
            .replace(" ", "")
            .replace(",")
            .replace(".", "")
        )
        list_of_prices = pars_price.split(",")

        if cp_answer == "USD":
            price = list_of_prices[0][4:]
            test_str_price += f"*{ALL_COINS[coin][0]}* --> {price} $\n"
        elif cp_answer == "EUR":
            price = list_of_prices[1][4:]
            test_str_price += f"*{ALL_COINS[coin][0]}* --> {price} €\n"
        elif cp_answer == "UAH":
            price = list_of_prices[2][4:]
            test_str_price += f"*{ALL_COINS[coin][0]}* --> {price} ₴\n"

    await bot.edit_message_text(
        chat_id=callback.message.chat.id,
        message_id=callback.message.message_id,
        parse_mode="Markdown",
        text=test_str_price,
    )

@dp.callback_query_handler(
    lambda command: command.data and command.data.startswith("CTC_")
)
async def ctc_add(callback: types.CallbackQuery):
    ctc_answer = str(callback.data.split("_")[1])
    if ctc_answer == "add":
        pass
    if ctc_answer == "delete":
        pass

def register_monitoring_handlers(dp: Dispatcher):
    dp.register_message_handler(
        monitoring, lambda message: message.text == "🔍 Monitoring"
    )

```

```

)
dp.register_message_handler(
    tracked_coins, lambda message: message.text == "🔍 Tracked coins"
)
dp.register_message_handler(
    check_price, lambda message: message.text == "🔍 Check price"
)
dp.register_message_handler(
    cp_add, lambda command: command.data and
command.data.startswith("CP_")
)
dp.register_message_handler(
    ctc_add, lambda command: command.data and
command.data.startswith("CTC_")
)

```

```

import sqlite3

class NoneUserWallet(Exception):
    pass

def sql_start():
    global base, cur
    base = sqlite3.connect("global_db")
    cur = base.cursor()
    if base:
        print("Data base connected OK!")
    else:
        print("Data base was created!")

    base.execute(
        """
        CREATE TABLE IF NOT EXISTS user (
            user_id TEXT PRIMARY KEY UNIQUE NOT NULL,
            wallet_id TEXT NOT NULL,
            balance REAL DEFAULT 0.0)
        """
    )
    print("-----")
    print("1. The user table was successfully created")
    base.execute(
        """
        CREATE TABLE IF NOT EXISTS wallet (
            user_id TEXT,
            coin_id TEXT,
            value_of_coins REAL DEFAULT 0.0,
            FOREIGN KEY (user_id) REFERENCES user(user_id))
        """
    )
    print("-----")
    print("2. The coins wallet was successfully created")
    print("-----")
    base.execute(
        """
        CREATE TABLE IF NOT EXISTS list_of_coin (

```

```

        coin_id TEXT PRIMARY KEY UNIQUE,
        coin_name TEXT,
        get_method TEXT)
    """
)
print("3. The list_of_coins wallet was successfully created")
print("-----")
print("Filling the database with a list of coins")

try:
    cur.execute(
        """INSERT INTO list_of_coin (coin_id, coin_name, get_method)
        VALUES
            ('bitcoin', 'BTC - Bitcoin',
'https://api.coingecko.com/api/v3/coins/bitcoin'),
            ('ethereum', 'ETH - Ethereum',
'https://api.coingecko.com/api/v3/coins/eth'),
            ('true-usd', 'TUSD - TrueUSD',
'https://api.coingecko.com/api/v3/coins/true-usd'),
            ('tether', 'USDT - Tether',
'https://api.coingecko.com/api/v3/coins/tether'),
            ('binancecoin', 'BNB - Bnb',
'https://api.coingecko.com/api/v3/coins/binancecoin'),
            ('ripple', 'XRP - Ripple',
'https://api.coingecko.com/api/v3/coins/ripple'),
            ('cardano', 'ADA - Cardano',
'https://api.coingecko.com/api/v3/coins/cardano'),
            ('dogecoin', 'Doge - Dogecoin',
'https://api.coingecko.com/api/v3/coins/dogecoin'),
            ('solana', 'SOL - Solana',
'https://api.coingecko.com/api/v3/coins/solana'),
            ('polkadot', 'DOT - Polkadot',
'https://api.coingecko.com/api/v3/coins/polkadot'),
            ('shiba-inu', 'SHIB - Shiba Inu',
'https://api.coingecko.com/api/v3/coins/shiba-inu'),
            ('dai', 'DAI - Dai',
'https://api.coingecko.com/api/v3/coins/dai'),
            ('matic-network', 'MATIC - Polygon',
'https://api.coingecko.com/api/v3/coins/matic-network'),
            ('tron', 'TRX - TRON',
'https://api.coingecko.com/api/v3/coins/tron'),
            ('uniswap', 'UNI - Uniswap',
'https://api.coingecko.com/api/v3/coins/uniswap'),
            ('wrapped-bitcoin', 'WBTC - Wrapped Bitcoin',
'https://api.coingecko.com/api/v3/coins/wrapped-bitcoin'),
            ('chainlink', 'LINK - Chainlink',
'https://api.coingecko.com/api/v3/coins/chainlink'),
            ('leo-token', 'LEO - LEO Token',
'https://api.coingecko.com/api/v3/coins/leo-token'),
            ('cosmos', 'ATOM - Cosmos Hub',
'https://api.coingecko.com/api/v3/coins/cosmos'),
            ('litecoin', 'LTC - Litecoin',
'https://api.coingecko.com/api/v3/coins/litecoin')
        """
    )
except:
    print("Fail")
else:

```

```

        print("Success")
        base.commit()

async def check_user(user_id):
    info = cur.execute("SELECT user_id FROM user WHERE user_id=?",
                       (user_id,))

    if info.fetchone() is None:
        pass
    else:
        raise NoneUserWallet

async def create_wallet(user_id, wallet_currency):
    wallet_address = str(user_id) + "_wallet"
    cur.execute(
        """INSERT INTO wallet VALUES (
            user_id,
            wallet_id,
            balance,
            wallet_currency)""",
        (user_id, wallet_address, 0, wallet_currency),
    )

    base.commit()
    return wallet_address

async def column_output():
    cur.execute("SELECT coin_id, coin_name FROM list_of_coin")
    output = cur.fetchall()
    return output

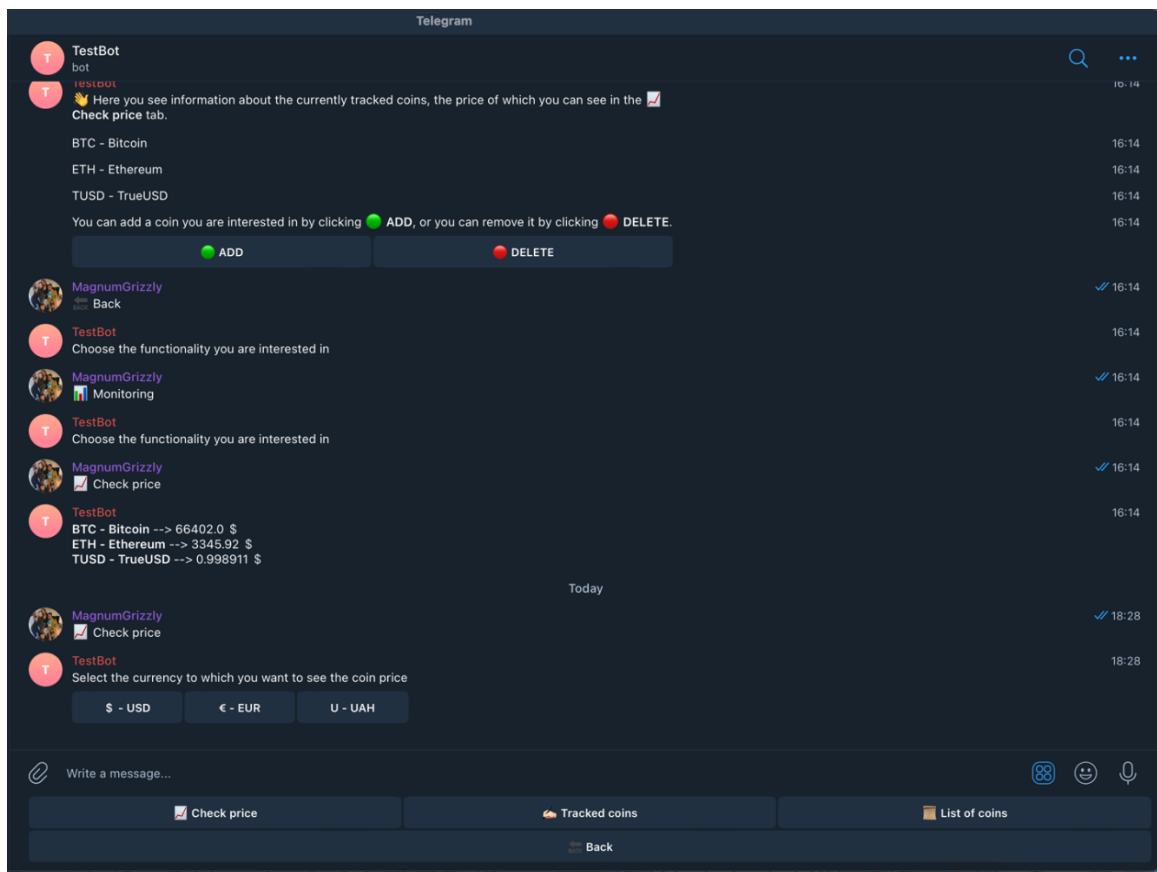
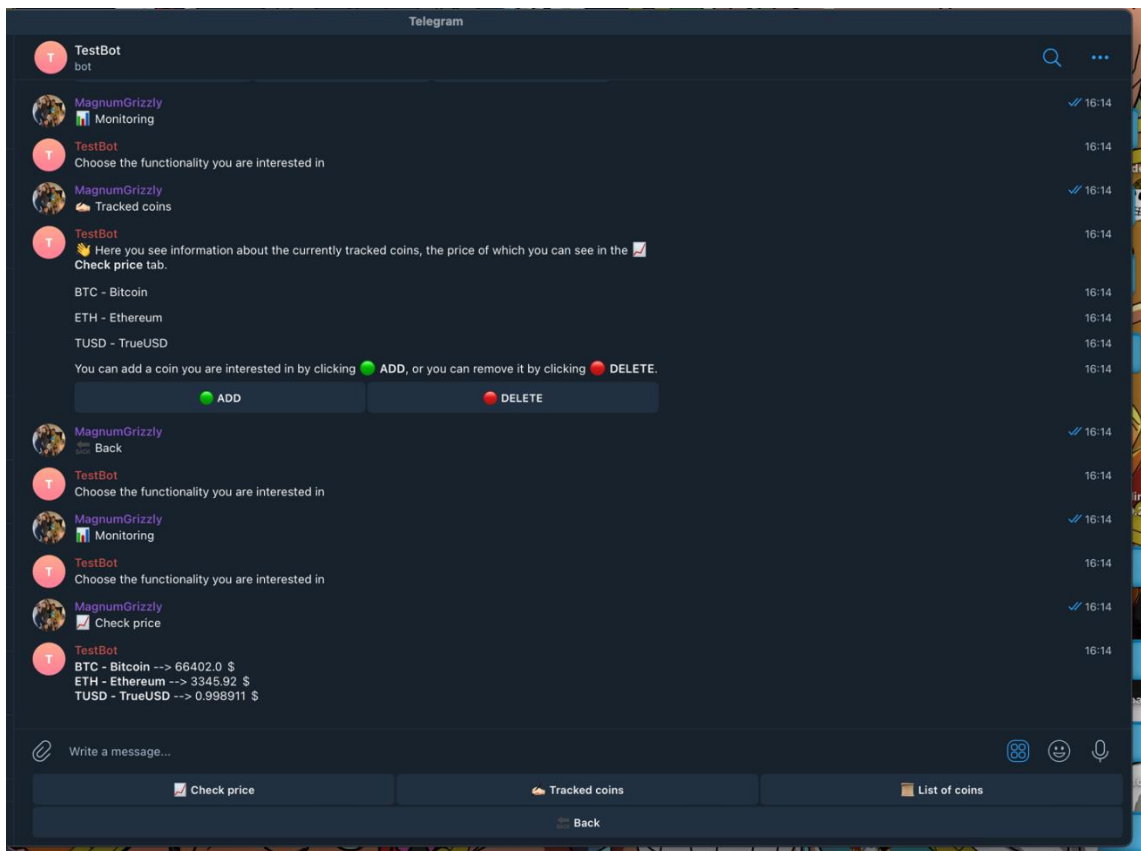
async def set_user_wallet(user_id, coin_id, value_of_coins):
    cur.execute(
        """INSERT INTO wallet VALUES (
            user_id,
            coin_id,
            value_of_coins,
            )""",
        (user_id, coin_id, value_of_coins),
    )

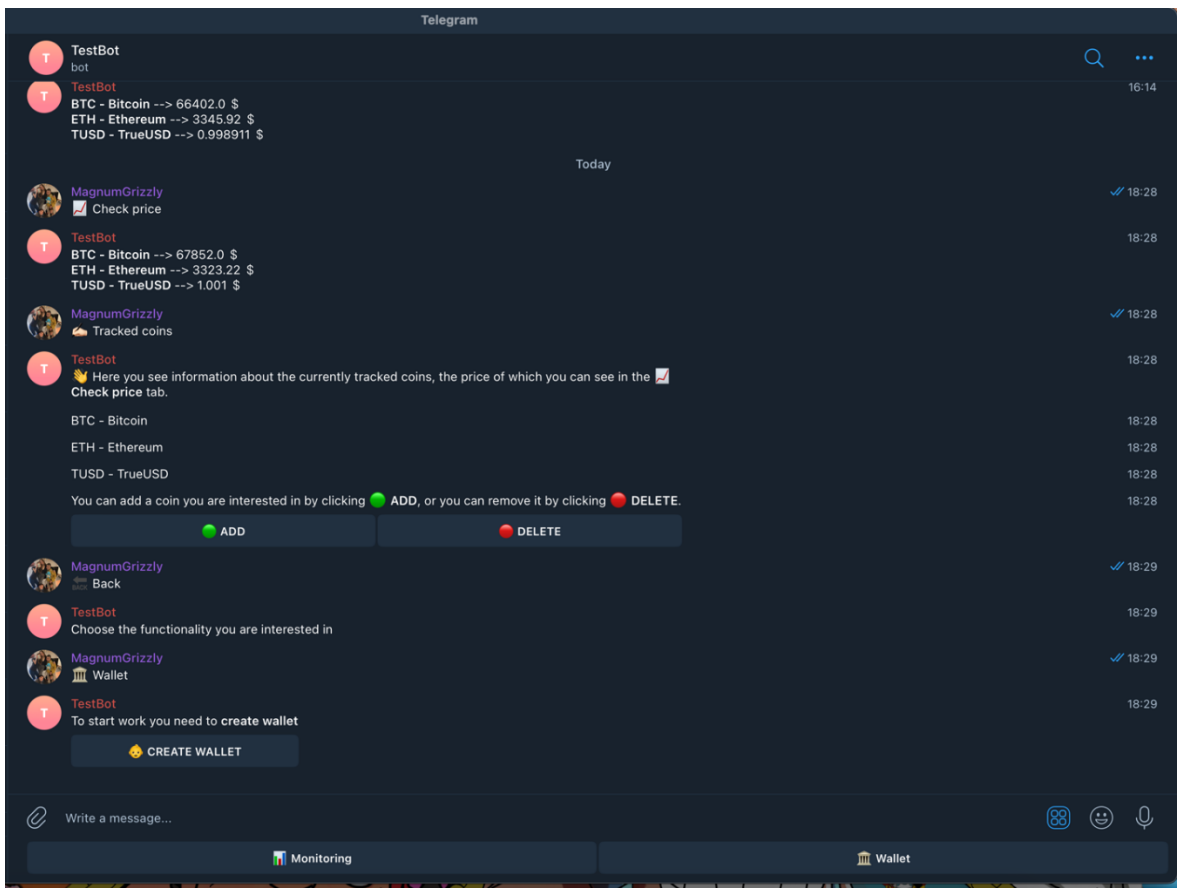
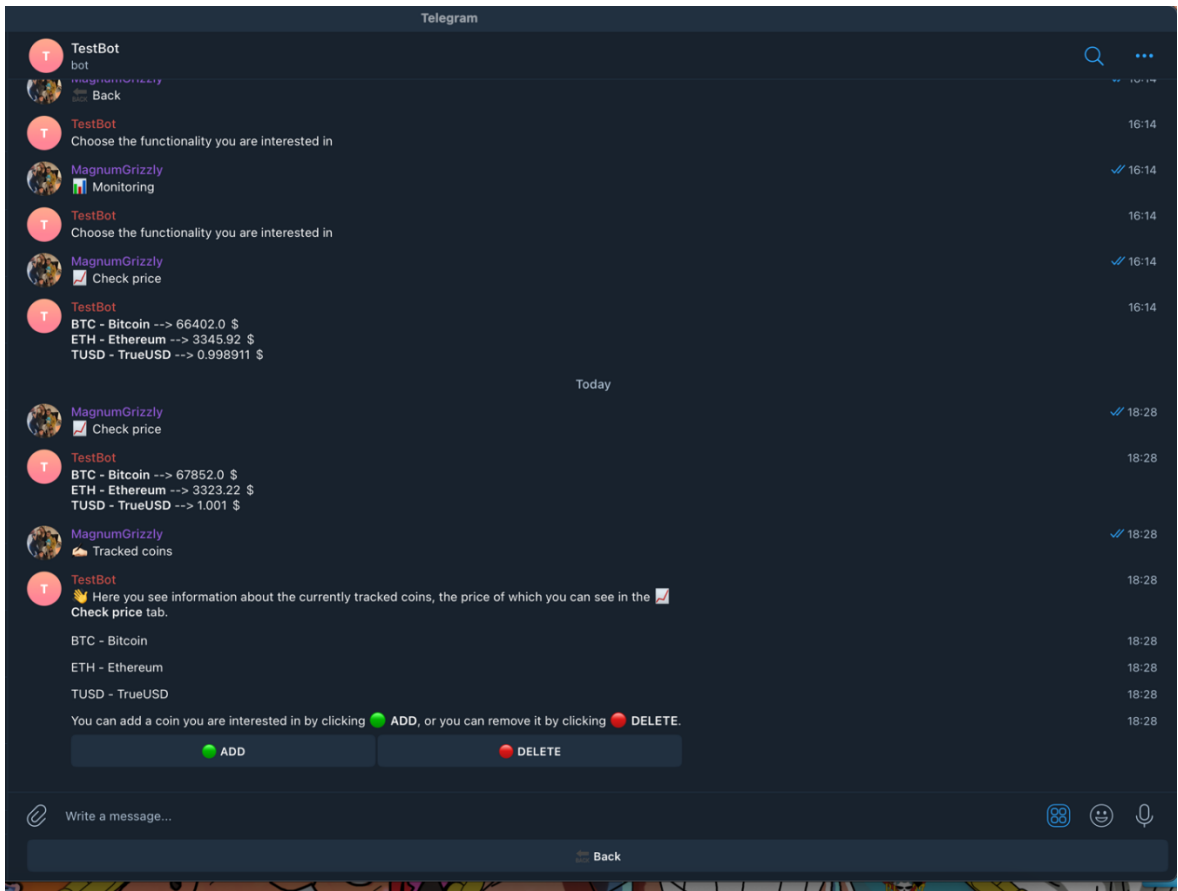
    base.commit()

```



## Приклад Telegram Bot





## Закінчення додатку Б

