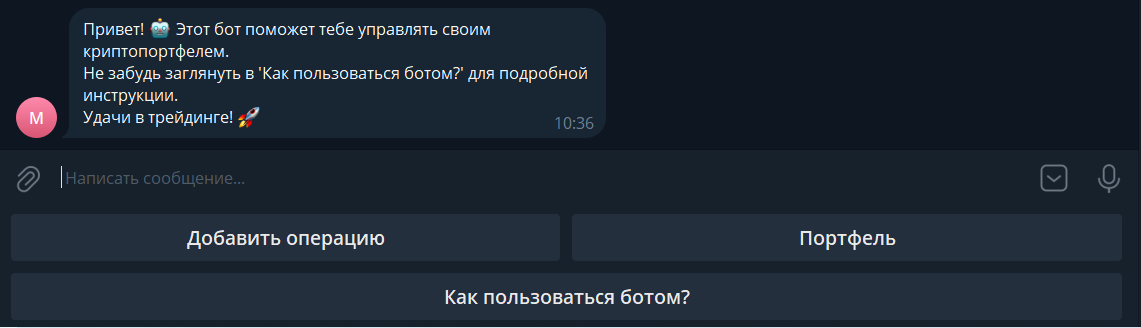
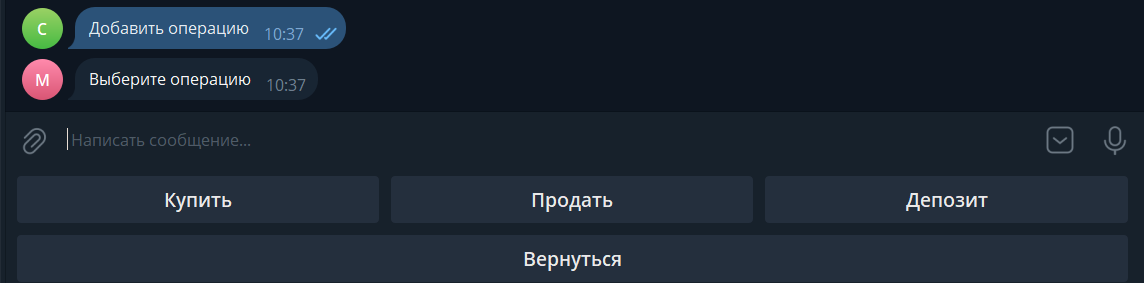
**Отчет по выполненной работе**

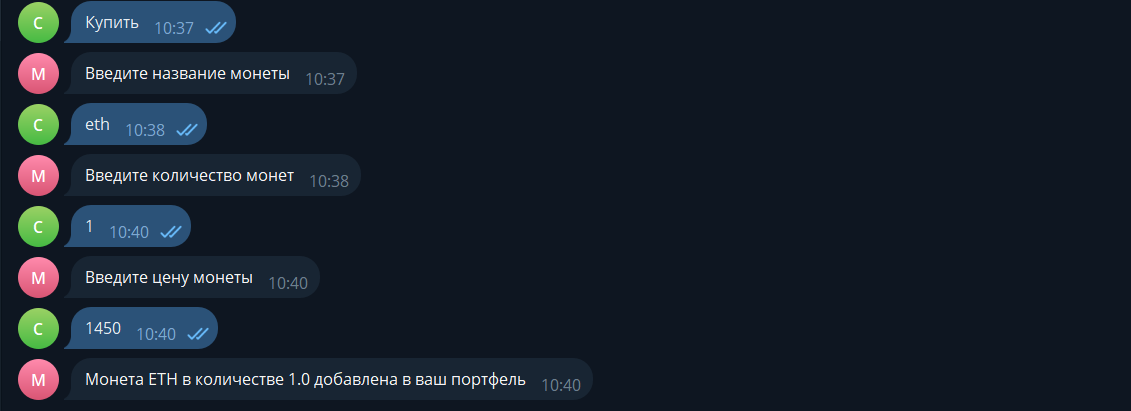
1. **Описание бота:**
   * **Название бота:** @My\_Crypto\_Wallet\_1\_Bot
   * **Язык программирования:** Python
   * **Библиотеки:** aiogram (Telegram бот), SQLite (хранение данных), ccxt (Запросы бирже Binance),
   * **Описание:** Бот предоставляет функционал криптопортфеля для учета финансов пользователя. Вся информация хранится в базе данных SQLite, а взаимодействие с пользователем осуществляется через интерфейс Telegram.
2. **Основной функционал бота:**
   * **/start:** Команда для начала взаимодействия с ботом. Пользователь получает приветственное сообщение от бота. Появляется основная клавиатура с тремя кнопками: "Добавить операцию", "Портфель", "Как пользоваться ботом?".



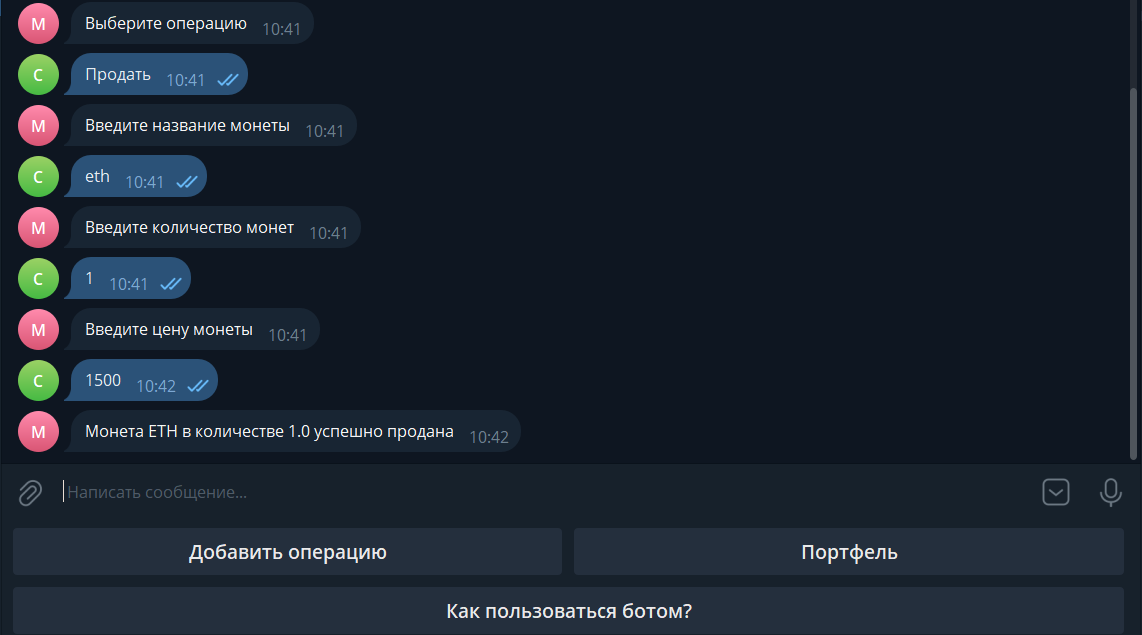
* + **Добавить операцию:**



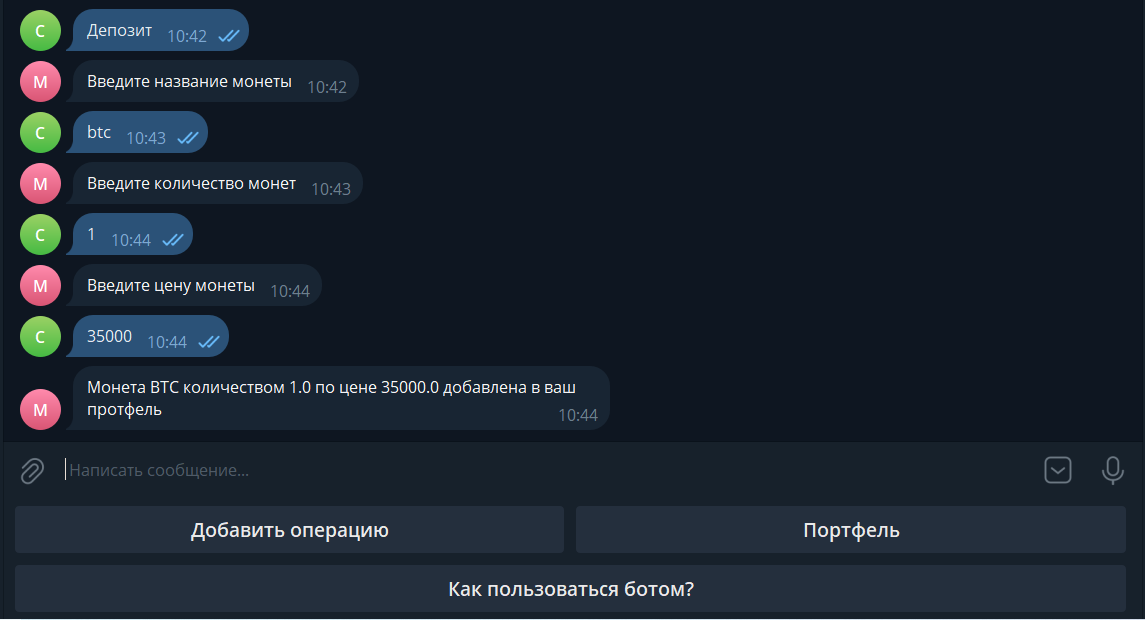
* + - **"Купить":** Покупка новой монеты за счет USDT в портфеле. Указывается название, количество и цена покупки.



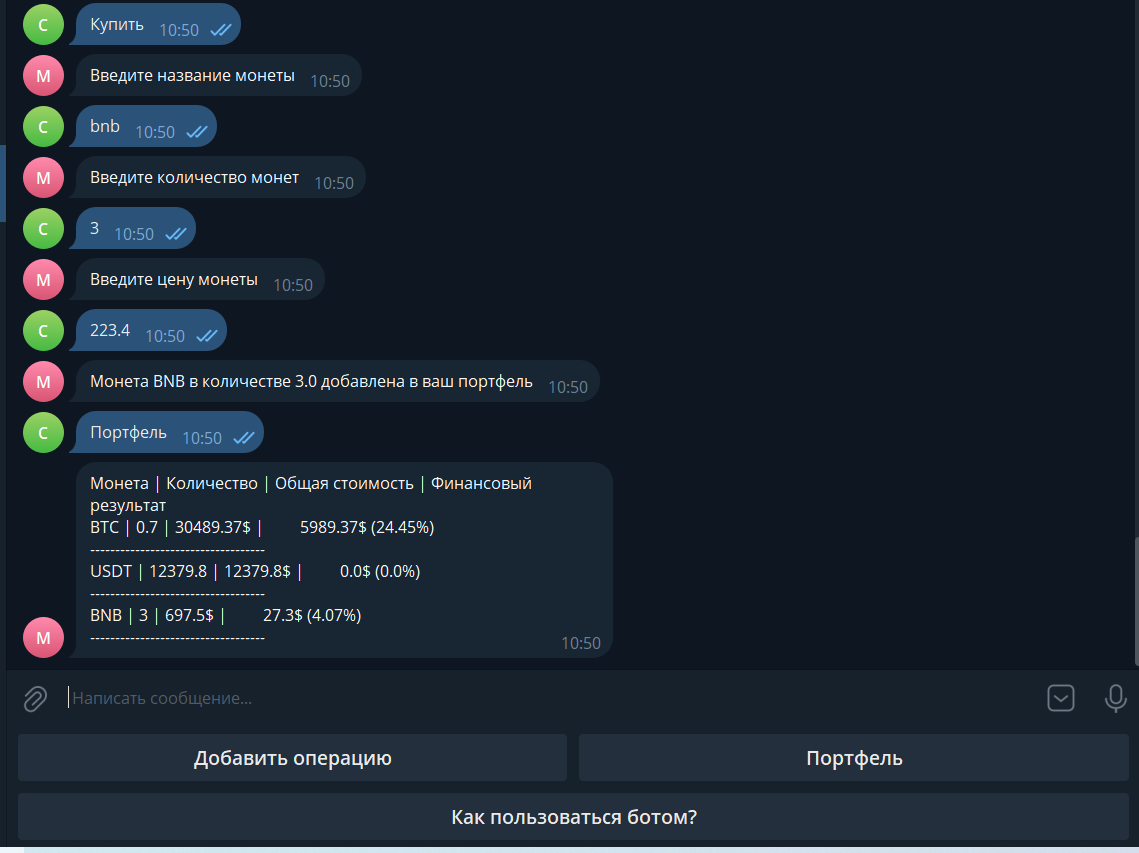
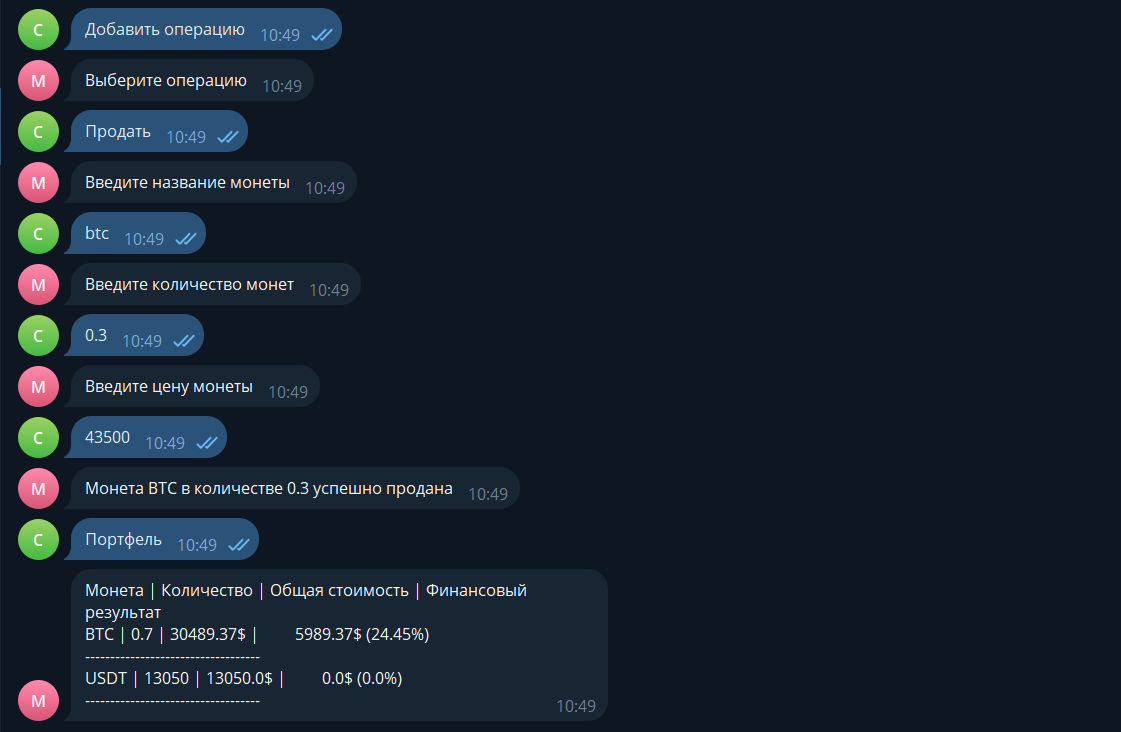
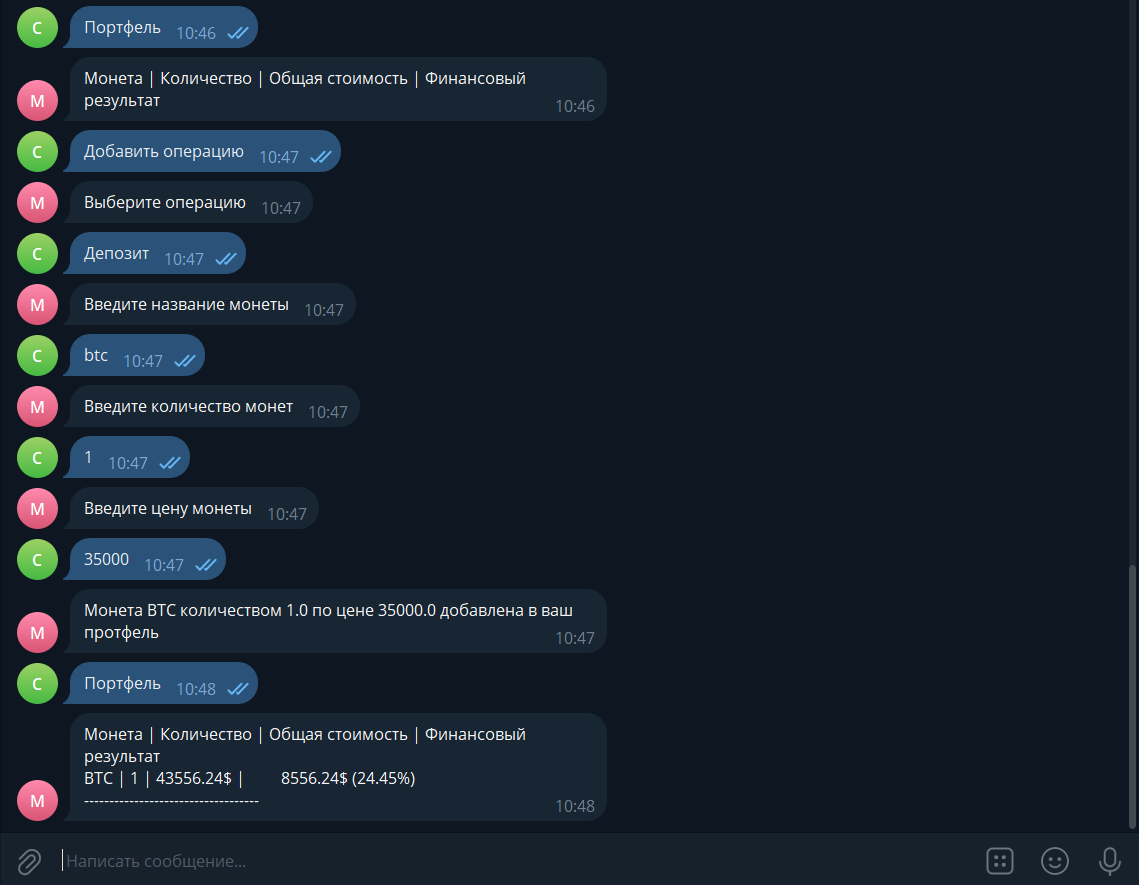
* + - **"Продать":** Продажа имеющейся монеты за USDT. Указывается название монеты, количество и цена продажи.



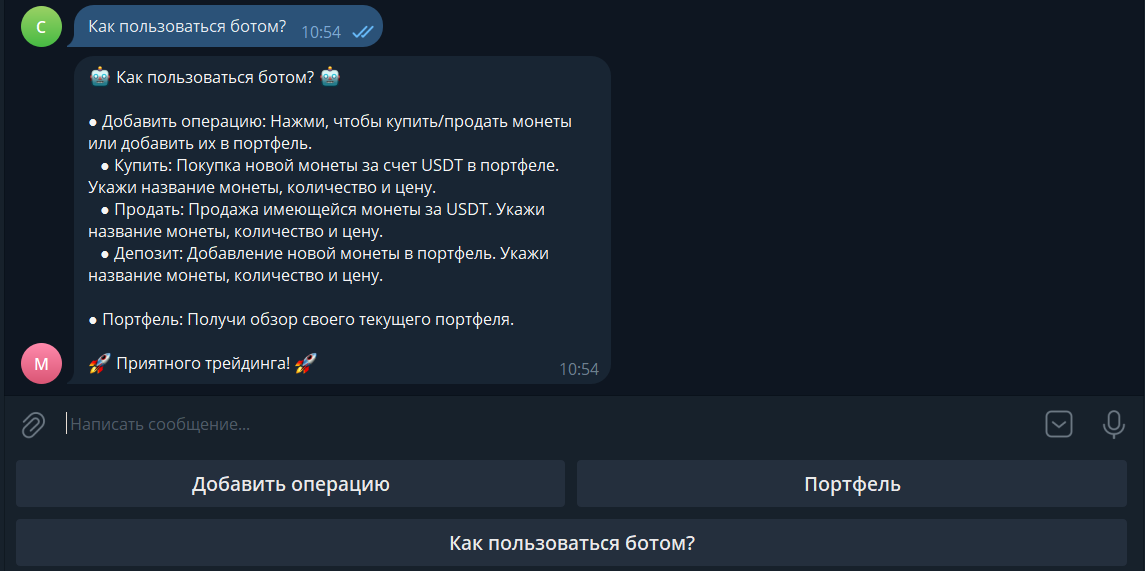
* + - **"Депозит":** Добавление новой монеты в портфель. Указывается название, количество и цена монеты в момент приобретения.



* + **Портфель:** Получение обзора текущего состояния портфеля.



* + **Как пользоваться ботом?:** Краткое руководство по использованию функционала бота.



1. **Файлы и структура проекта:**
   * **main.py:** Основной файл бота.
   * **keyboards.py**: Клавиатуры для бота
   * **config.py:** Системные настройки бота
   * **check\_token.py:** Модуль проверки монеты на бирже
   * **create\_db.py:** Модуль создание и обновление базы данных
   * **parse\_token\_price.py:** Модуль получения актуального курса валют и обновление базы данных
   * **crypto.db:** База данных**Начало формы**

**4. Ошибки и отклонения в ходе работы:**

1. Возможность уйти в отрицательное значение монеты в портфеле. Продавая монету, программа проверяет, есть ли продаваемая монета в портфеле пользователя. После этой проверки, программа собирает оставшиеся данные. Затем получает имеющееся количество монет из портфеля и вычитает из них продаваемое значение. Если продаваемое количество монет больше, чем есть в портфеле, то значение уйдет в минус, однако программа не вылетит с ошибкой. Решается обычной проверкой имеющееся количество >= продаваемое количество.
2. По плану, кнопка «Портфель» должна выводить финансовый результат по каждой монете. Формула расчета выглядит так: Общая стоимость на данный момент – Стоимость закупки = Профит. Общая стоимость на данный момент рассчитывается просто, цена в данный момент \* на количество. Проблема заключалась в нахождении курса монеты для расчета Стоимости закупки. Входе тестирования и поиска решений, самым подходящим вариантом оказался перерасчет средней цены закупки, после каждого пополнения монеты. Финансовый результат должен корректно покрывать возможность закупки монеты «ступеньками», то есть в несколько этапов, по разному курсу. Расчет средней цены с этим справляется
3. Вывод состояния портфеля пользователя. Функционал программы следующий: из бота в функцию передается id пользователя, после чего достаются все его монеты из портфеля. Циклом перебираются названия, добавляя необходимые данные в словарь. Одни из данных это «Общая стоимость» и «Финансовый результат». Для расчета этих данных в цикле происходит запрос на биржу Binance для получения последней цены монеты. Используя такой алгоритм, пользователь слишком долго ожидает отклика программы, после нажатия на кнопку «Портфель». Формирование отчета на 4-5 монет занимает 15-20 секунд. Для того, чтобы отчет был моментальным, в базе данных была создана таблица с курсами валют. Далее, написан парсер, который достает все уникальные названия монет из таблица с портфелями пользователей, затем циклом перебирает каждое имя, запросом на биржу Binance получает последнюю цену. Сформировав словарь вида имя\_токена : курс\_валюты, передает значения в базу данных с курсами валют. Процедура бесконечно зациклена и повторяется каждые 5 секунд. Благодаря такому подходу, в момент формирования отчета, программа моментально достает последний курс валюты из таблицы. Отклик программы менее 1 секунды