

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка чат-боту (Python) та налаштування таргетингової реклами з метою вдосконалення каналу комунікації та просування продукції

Виконав: студент IV курсу, групи СН-41

спеціальності 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Вербіцький І.В.  
(підпис) (прізвище та ініціали)

Керівник Струтинська І.В.  
(підпис) (прізвище та ініціали)

Нормоконтроль Шимчук Г.В.  
(підпис) (прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.  
(підпис) (прізвище та ініціали)

Рецензент Гладь Ю.Б.  
(підпис) (прізвище та ініціали)

Тернопіль  
2022





## АНОТАЦІЯ

Розробка чат-боту (Python) та налаштування таргетингової реклами з метою вдосконалення каналу комунікації та просування продукції // Кваліфікаційна робота освітнього рівня «Бакалавр» // Вербіцький Іван Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2022 // С. – 49, рис. – 10, табл. – 1, кресл. – 0, додат. – 4, бібліогр. – 30.

**Ключові слова:** чат-бот, таргетована реклама, база даних, телеграм-бот, реклама, рекламна кампанія, код, телеграм.

Кваліфікаційна робота присвячена аналізу та дослідженню теми розробки чат-боа, використовуючи мову програмування Python та налаштуванню таргетованої реклами, щоб вдосконалити канал комунікації та просування продукції.

Мета роботи. Спроекувати і розробити чат-бот, налаштувати таргетовану рекламу для магазину картин, щоб вдосконалити канал комунікації та просування продукції.

В першому розділі кваліфікаційної роботи описано загальні положення розглянутої теми. Висвітлено теоретичні відомості та загальні положення стосовно розробки та підготовки до розробки чат-боту для магазину картин.

В другому розділі кваліфікаційної роботи наведена вся практична частина. Розглянуто опис практичної реалізації створення чат-боту та налаштування таргетованої реклами.

## ANNOTATION

Development of a chatbot (Python) and setting up targeted advertising to improve the communication channel and product promotion // Qualification work degree «Bachelor» // Verbitskyi Ivan // Ternopil Ivan Pul'uj National Technical University, Department of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2022 // P. – 49, Fig. – 10, Tables – 1, Annexes. – 4, References – 30.

The qualification work is devoted to the analysis and research of the topic of chat-bota development, using the Python programming language and setting up targeted advertising to improve the channel of communication and product promotion.

The purpose of the work is to Design and develop a chatbot, set up targeted advertising for a picture shop to improve the channel of communication and product promotion.

The first section of the qualification work describes the general provisions of the topic. Theoretical information and general provisions regarding the development and preparation for the development of a chatbot for a picture shop are highlighted.

The second section of the qualification work presents the whole practical part. The description of the practical implementation of creating a chatbot and setting up targeted advertising is considered.

**Keywords:** chatbot, targeted advertising, database, telegram bot, advertising, advertising campaign, code, telegram.

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ**

ПК – Персональний комп'ютер

БД – Бази даних

СКБД – Система керування базами даних

ІС – Інформаційна система

ТЗ – Технічне завдання

РК – Рекламна кампанія

API (англ. Application Programming Interface) – Інтерфейс програмування застосунків

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ КВАЛІФІКАЦІЙНОЇ РОБОТИ .....	10
1.1 Аналіз предметної області щодо розробки чат-боту та налаштування таргетованої реклами .....	10
1.2 Формування вимог до чат-боту для магазину картин .....	11
1.3 Пошук актантів та варіантів використання чат-боту для магазину картин .....	13
1.4 Оцінка методів розв’язання задачі розробки чат-боту для магазину картин .....	16
1.5 Життєвий цикл розроблюваного чат-боту для інтернет-магазину ....	17
1.6 Вибір середовища розробки чат-боту для онлайн магазину картин..	18
1.7 Обґрунтування інструментів для запуску таргетованої реклами.....	19
1.8 Висновок до першого розділу .....	21
РОЗДІЛ 2. РЕАЛІЗАЦІЯ ЧАТ-БОТУ ТА НАЛАШТУВАННЯ ТАРГЕТОВАНОЇ РЕКЛАМИ ДЛЯ МАГАЗИНУ КАРТИН .....	23
2.1 Проектування та створення бази даних для чат-боту магазину картин	23
2.2 Використання асинхронності для розробки телеграм-боту.....	26
2.3 Практична реалізація телеграм-боту для магазину картин.....	28
2.3.1 Створення та опис файлу сценарію для чат-боту .....	30
2.3.2 Робота з елементами проекту чат-боту для магазину картин .....	31
2.3.3 Розробка клієнтської та адмін-частини чат-боту для магазину картин .....	33
2.4 Налаштування та запуск таргетованої реклами для магазину картин	36
2.5 Висновок до другого розділу .....	38
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	39
3.1 Працездатність людини – оператора.....	39
3.2 Вимоги ергономіки до організації робочого місця оператора ПК, агрегату.....	42

3.3 Висновок до третього розділу .....	45
ВИСНОВКИ.....	46
ПЕРЕЛІК ДЖЕРЕЛ.....	47
ДОДАТКИ	



## ВСТУП

**Актуальність теми.** Внаслідок глобалізації питання переходу класичного бізнесу в онлайн, виникає потреба пошуку нових інструментів для знаходження каналів продажу власної продукції.

З кожним днем все більше і більше компаній вирішують проводити свою діяльність в інтернеті. Це спричиняє необхідність пошуку нових каналів комунікації та просування. Одні з таких каналів комунікації є чат-боти, а просування – таргетована реклама.

Ці два інструменти є дуже актуальними напрямками в сучасному світі. Попит на розробку чат ботів та налаштування таргетованої реклами росте з кожним днем, а це свідчить про те, що і попит на розробників також буде рости.

Спеціальності розробника чат-ботів для телеграму та таргетолога (людина, яка налаштовує таргетовану рекламу) є актуальними напрямками, оскільки попит на ринку праці на цих спеціалістів постійно росте.

**Мета** і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є створення чат-боту для магазину картин та налаштування таргетованої реклами для цього магазину, щоб вдосконалити канал комунікації та просування продукції.

Досягнення окресленої мети викликало необхідність виконання таких **завдань:**

- проаналізувати літературні та інтернет-джерела з метою виявлення каналів комунікації та просування продукції;
- проаналізувати канали інформації для таргетованої реклами;
- проаналізувати стан досліджень в сфері розробки чат-ботів для телеграм месенджерів;
- проаналізувати та зробити висновки стосовно ефективності таргетованої реклами для залучення нових клієнтів та продажів;

– завдяки розробці чат-боту для інтернет магазину картин вдосконалити канали комунікації клієнта з бізнесом.

**Практичне значення одержаних результатів.** Кінцевим практичним результатом виконання кваліфікаційної роботи бакалавра буде повністю розроблена програма, чат-бот інтернет магазину картин для телеграму та налаштована і запущена таргетована реклама.

## РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ КВАЛІФІКАЦІЙНОЇ РОБОТИ

### 1.1 Аналіз предметної області щодо розробки чат-боту та налаштування таргетованої реклами

У сучасному світі всі бізнеси переходять в діджитал простір, представляються в онлайні. Методи онлайн-бізнесу відрізняються від традиційного, класичного світу дуже суттєво. Те що було актуальним ще декілька років тому, вже не працює.

На ринку з кожним днем появляється все більше компаній, і кожна з них хоче виділитися на фоні інших чимось унікальним. Конкуренція неймовірно велика, і знайти той інструмент, який дозволить достукатися до потенційних клієнтів не так вже й легко.

Одним з таких інструментів є симбіоз (поєднання) маркетингових інструментів, що прийшли з традиційного бізнесу, та сучасних інформаційних технологій.

Актуальність розробки чат-боту картин полягає в тому, що конкуренція між бізнесами постійно росте, а телеграм-бот – це можливість виділитися на фоні інших. Також спостерігається тренд, де будь-яка людина хоче отримати рішення всіх своїх проблем з одного інструменту [7].

Таким інструментом можуть бути чат-боти, оскільки за допомогою них, можна створювати цілі екосистеми, в яких кожен може в одному місці отримати відповіді на всі свої запитання. Там можна і замовити товар і домовитися з продавцем про створення унікальності в цьому товарі.

Розробка ботів – це відносно новий напрямок у створенні програмних продуктів. Чат-боти вивели комунікацію з клієнтом на новий рівень. Це відбувається за рахунок того, що спілкування з клієнтом відбувається на високо-автоматизованому рівні. Втручання менеджерів практично не потрібне. А з іншої сторони, клієнт має змогу сам обрати що він хоче з добре структурованої інформації [2].

Чат-боти використовуються в різних сферах та інтегруються практично всюди.

В світі соціальних мереж про кожну особу система знає все, від найдрібніших вподобань до загального соціального статусу та психологічного типу людини. Для комп'ютерних програм це лише одиниця інформації в багатомільйонному потоці даних. Так, соціальна мережа Facebook знає про наші вподобання, оскільки постійно в телефоні чи комп'ютері щось шукаємо, те що нам подобається, спілкуємося з іншими користувачами всесвітньої мережі про те, що нам подобається. А кожен потік від нас записується і пізніше складається загальна характеристика [9].

Використовуючи всі зібрані дані про кожну людину, онлайн-маркетологи мають можливість цілити в точку своєю рекламою. Саме для цього використовується таргетована реклама. Завдяки цьому інструменту дуже велика кількість бізнесів знаходять клієнтів на покупки послуг чи товарів.

Таргетована реклама (таргет) – показ рекламних оголошень на основі демографічних даних цільової аудиторії та їх інтересів (Facebook, Instagram).

Варто сказати, що в цій галузі все поєднується, неможливо отримати результат використовуючи лише один інструмент. Саме тому для просування продукції картин в онлайн-магазині було вирішено розробити чат-бот та запустити таргетовану рекламу.

## **1.2 Формування вимог до чат-боту для магазину картин**

Для коректної реалізації чат-боту для онлайн-магазину картин слід описати декілька основних його функціональних елементів, відповідно до яких і буде сформоване технічне завдання.

У телеграм-боті для магазину мальованих картин буде реалізований функціонал спілкування з ботом, використовуючи кнопки та команди. Відповідно до цього повинні бути реалізовані функції та методи взаємодії з користувачем [15].

У чат-боті буде реалізовано три різних частини:

- Частина клієнта.
- Частина адміна.
- Інша частина.

Кожна з цих частин має відповідати за свій функціонал боту та призначена для своїх цілей. А саме, клієнтська частина – реалізує все, що стосується користувача, який не є адміном, який матиме доступ лише для спілкування з ботом та отримання від нього потрібних даних. Для клієнта має бути реалізоване меню, для навігації у боті. Також користувач повинен мати можливість бачити дані, що є у базі даних у вигляді карток товару

Адмін-частина чат-боту магазину картин повинна відповідати за все, що не бачить простий користувач. Адмін мусить мати прихований вхід в панель модератора. Там мають бути реалізовані функції роботи з базою даних, а саме створення та видалення карток товару (картин) з фото, назвою, описом та ціною.

Також слід відділити функціонал частини адміна від інших частин, щоб інтерфейси цих елементів не пересікалися на одній ділянці. Тобто, щоб простий користувач ніяким чином не бачив елементів керування панеллю модератора.

В іншій частину телеграм-боту інтернет магазину картин має бути реалізований код, для парсингу повідомлень та фільтрування нецензурної лексики. Якщо користувач буде вводити неприйнятні слова, то ці слова мають видалятися і повідомляти користувачу про його помилки.

Варто звернути особливу увагу на те, що проект чат-боту для інтернет-магазину повинен бути реалізованим в модульній структурі розміщення файлів. Не допускається написання коду в одному файлу, він має бути розбитим на різні модулі, де зберігаються методи, які відповідають за різний функціонал.

Для телеграм-боту магазину картин потрібно створити базу даних і реалізувати декілька функцій обробки її запитів. Потрібно використовувати СКБД для збереження інформації про представлені у онлайн-магазині товари.

### 1.3 Пошук актантів та варіантів використання чат-боту для магазину картин

Чат-бот для інтернет магазину картин має працювати за певними сценаріями що представляються у вигляді діаграми варіантів використання системи. Ця діаграма дасть краще розуміння структури проекту та основних його технічних можливостей.

В загальному, діаграму варіантів використання боту телеграм для магазину картин, що представлений в інтернеті можна подати у вигляді таблиці, де подано усіх акторів проекту, їх зони відповідальності та можливості до функціонування.

Реєстр варіантів використання веб-застосунку закритого тематичного клубу по саморозвитку подано у таблиці 1.1.

Таблиця 1.1 – Реєстр варіантів використання чат-боту для інтернет магазину картин

<b>Актор</b>	<b>Найменування</b>	<b>Формулювання</b>
<b>1</b>	<b>2</b>	<b>3</b>
Користувач	Звернення до бота	Дозволяє звернутися до бота у групі, командою виклику
	Перегляд вмісту	Дозволяє переглянути інформацію, та продукти які представлені у боті
Модератор, який не увійшов в режим адміністратора	Вхід в адмін-панель	Дозволяє звернутися до бота у чаті й увійти в адмін-панель

Продовження таблиці 1.1

1	2	3
Модератор, який увійшов в режим адміністратора	Перегляд та зміна вмісту бази даних	Дозволяє переглядати вміст бази даних та видаляти картки товару
	Створення вмісту	Дозволяє створювати нові карточки товару, які будуть відображені у боті

Отже, у таблиці можна побачити всіх акторів та їхні можливості використання чат-боту для інтернет магазину картин.

Далі описано розгорнуте бачення усіх можливих варіантів використання чат-боту для інтернет-магазину картин, а саме можливості користувача, модератора, який не увійшов в режим адміністратора та модератора, який увійшов в режим адміністратора.

Отже, спочатку розглянемо користувача чат ботом для магазину картин. Він має можливість звернутися до чат-боту у групі, в якій цей бот є адміністратором, за допомогою команди початку. В такому випадку бот напише йому в приватні повідомлення, або попросить добавитися і писати в приватні повідомлення.

Другий варіант – це звернутися до бота в приватних повідомленнях. Далі користувач має можливість переглянути інформацію про онлайн магазин картин, переглянути список товарів, написати в приватні повідомлення менеджеру, щоб замовити картину, або залишити заявку.

На рисунку 1.1 зображено UML діаграму для звичайного користувача чат-боту для магазину картин.



Рисунок 1.1 – UML діаграма для звичайного користувача чат-боту для магазину картин

Тепер перейдемо до наступного актора чат-боту для магазину картин – модератор, який не увійшов в режим адміністратора. Модератор, який не авторизувався, як адміністратор має можливість переглядати наявний товар чат-боту для магазину картин, отримувати інформацію про магазини, замовляти картини, перейти в чат з менеджером. Він на даному етапі являється простим користувачем, але його відмінність полягає в тому, що в будь-який момент він може ввести приховану команду в групі з ботом (команда «/moderator»), чим активує адмін-панель і зможе стати повноцінним модератором.

На рисунку 1.2 зображено UML діаграму для модератора, який ще не активував адмін-панель.

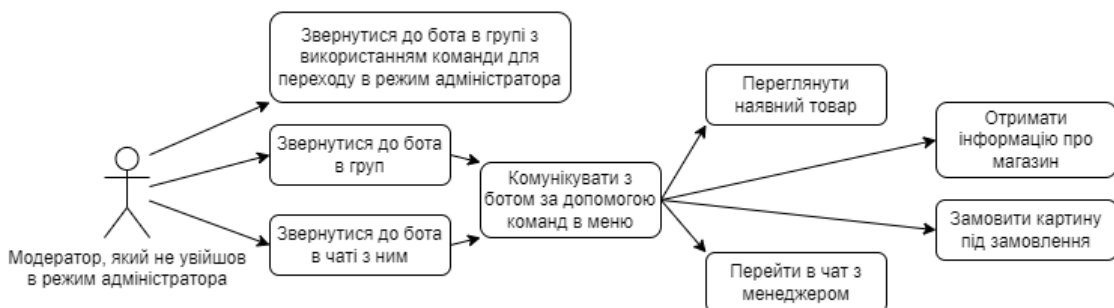


Рисунок 1.2 – UML діаграма для адміна чат-боту для онлайн магазину, що не авторизувався

Модератор, який авторизувався як адмін виконує ключову роль у чат-боті магазину картин. Він має доступ до адмін-панелі, де має змогу створювати та редагувати базу даних і карточки товарів, які пізніше висвітлюються



користувачам боту. Також адмін може видаляти створенні товари та інформацію про них. Саме адмін здійснює всю взаємодію з базою даних. Також варто зазначити, що адмін, після того, як він авторизувався в якості модератора, може також виконувати всі функції, що доступні простому користувачу

На рисунку 1.3 зображено UML діаграму для модератора, який ще не активував адмін-панель.



Рисунок 1.3 – UML діаграма для адміна, який авторизувався як модератор

Отже, вище можна побачити UML-діаграми для звичайного користувача, для модератора, що не активував адмін-панель та для модератора, що активував адмін-панель, і разом вони є акторами для чат-боту інтернет-магазину з картинками.

#### 1.4 Оцінка методів розв'язання задачі розробки чат-боту для магазину картин

Для розв'язування поставленої задачі необхідно провести розробку власного чат-боту телеграм для онлайн магазину картин. Для цього потрібно використати мову програмування Python та систему керування базами даних (СКБД) SQLite. База даних дозволить бути впевненим в зберіганні даних користувачів та статей, їх редагуванні та видаленні, тощо [16].

Також, окрім бази даних, буде продумано усі можливі пункти меню та керування чат-ботом для онлайн магазину. Це можуть бути інформація про

магазини, наші товари, розділ для замовлення картини, перехід у чат з менеджером, сторінка допомоги, тощо. Функціонал на цих вкладках буде найрізноманітнішим.

Збереження всіх даних, що використовуються у розробленому чат-боті для онлайн магазину картин буде здійснено у базі даних.

### **1.5 Життєвий цикл розроблюваного чат-боту для інтернет-магазину**

Життєвий цикл розробки чат-боту для інтернет магазину картин має шість етапів. Кожен з яких відповідає за свої процеси та має певні вимоги.

Перший етап відповідає за розробку інформаційної системи (ІС). На першому етапі також розглядаються: релевантність проекту, аналіз конкурентів, оцінка виконання вимог замовника на основі витрат ресурсів.

Після першої фази йде друга фаза-розробка технічного завдання для проекту чат-боту для магазину картин. На цьому етапі формуються технічні завдання, які стануть основою для власне розробки ІС та її подальшої діяльності. Крім того, у разі виникнення конфліктів при розробці програмного забезпечення цей документ може бути використаний як доказ суперечок між розробниками та замовниками.

На третьому етапі здійснюється проектування чат-боту магазину картин (розробка концепції). Крім того, існують рішення будь-яких проблем, пов'язаних з розробкою концепцій програмного забезпечення та вибором найбільш підходящої архітектури для моделі ІС.

Четвертий етап – це процес реалізації чат-боту для інтернет-магазину. Цей етап включає повну розробку та виконання всіх функцій відповідно до вимог, зазначених у технічних завданнях, створених на другому етапі [11].

П'ятим етапом стане загальне тестування та оптимізація розробленого телеграм-боту для магазину. Іншими словами, цей етап включає перевірку робіт, виконаних розробником, щоб проект відповідав задовільним умовам, зазначеним у технічному завданні. Крім того, зазвичай на цьому етапі

працівникам навчання при необхідності доведеться використовувати готові рішення.

Шостий етап, який є завершальним, полягає в підтримці та подальшому обслуговуванні готового чат-боту для інтернет магазину. Цей етап розпочинається після завершення всіх робіт по розробці програмного продукту, а завершення узгоджується з замовником.

## **1.6 Вибір середовища розробки чат-боту для онлайн магазину картин**

Для розробки чат-боту для просування представлення інтернет-магазину картин в інтернеті було вирішено використати інтерпретовану об'єктно-орієнтовану мову програмування Python, а саме версії Python 3.10.4 [21] (на момент розробки являється останньою версією). Насправді, чат-ботів для застосунку Telegram [7] можна створювати, використовуючи багато різних мов програмування, наприклад PHP, JavaScript, C# та інші, але основна перевага пайтона над всіма іншими – це зручність написання асинхронного коду.

Також для реалізації додаткового функціоналу та передавання структурованої інформації через мережу потрібно використати текстовий формат обміну даними між комп'ютерами JSON.

Для роботи з API Telegram [7] можна використовувати різні спеціалізовані бібліотеки, наприклад `pyTelegramBotApi`, але щоб реалізувати асинхронність чат-боту для магазину картин було вирішено використовувати бібліотеку `aiogram`.

Для збереження та подальшої обробки даних в чат-боті інтернет магазину варто реалізувати базу даних, для цього було використано СКБД SQLite. Для створення цих баз даних треба використати застосунок SQLiteStudio.

Відповідно до перелічених засобів було обрано середовище розробки, що дозволяє зручно писати код для чат-боту для магазину картин. Для цих цілей було використано Visual Studio Code.

## 1.7 Обґрунтування інструментів для запуску таргетованої реклами

Отже, ще одним дуже важливим етапом представлення бізнесу в інтернеті є його просування та знайомлення клієнта з новим товаром.

У даному пункті буде описано що таке реклама в загальному і що таке таргетована реклама. Які переваги має таргетована реклама перед класичною формою подання реклами.

В мережі Інтернет є дуже багато різних компаній і вони всі дуже схожі одна на одну. Але деякі мають продажі, деякі мають стабільний потік клієнтів в їхній бізнес, а дехто ні. Причина такого розподілу полягає в тому, що деякі люди використовують інструменти сучасного онлайн-маркетингу та направляють потік клієнтів у власний бізнес.

Є дуже багато інструментів сучасного маркетингу для представлення бізнесу в інтернеті. Це може бути і налаштування таргетованої реклами, email-маркетинг, реклама у блогерів та багато іншого [8].

За останній час найбільшого поширення набула таргетована реклама, оскільки даний інструмент маркетингу цілить саме у ту аудиторію, яку обирають для показу їм реклами.

Щоденно в інтернеті про кожного користувача збирається мільйони різноманітної інформації починаючи від його IP до побудови соціально демографічні характеристики з вподобаннями конкретної особи.

На основі запитів в пошукових системах на основі перегляду різноманітних веб-сторінок алгоритми фейсбуку описують загальне вподобання людини і після чого будь-який маркетолог може направляти свою рекламу на особу використовуючи ці дані. Потрапивши точно в ціль [27].

Таргетинг походить від англійського слова «target» - ціль. Цей термін чітко і ясно описує основну суть даного методу онлайн-маркетингу.

В загальному, щоб запуснути таргетовану рекламу потрібно просто скористатися таким інструментом, як ADS Manager. Але щоб зробити це ефективно та якісно, спершу треба розробити маркетингову стратегію для

таргетованої реклами. Кожен маркетолог складає її у такому форматі й так, як йому зручно.

Основним пунктом є сформувані цілі рекламної кампанії. Для інтернет-магазину картин – це збільшити кількість нових клієнтів, збільшити лояльність клієнтів до продукту, вивести в топ своєї ніші, виділитись на фоні конкурентів, збільшити продажі, формування потреби в продукті [26].

Отже першим етапом при налаштуванні таргетованої реклами є вибір цілі. Потрібно обрати зі списку, на яку ціль буде орієнтуватися рекламна кампанія. Це робиться задля підвищення ефективності реклами для кожного окремого випадку.

Слід зазначити, що налаштування рекламної кампанії відбувається у 3 етапи: налаштування кампанії, далі налаштування групи оголошень і налаштування оголошень. Кампанія може включати в себе групи оголошень, а ті відповідно – оголошення. Вигляд ієрархії створення рекламної кампанії зображено на рисунку 1.4

The screenshot shows a user interface for creating a campaign. At the top, it says 'Укажите название кампании • Необязательно' (Specify the campaign name • Optional). Below this, there are three main sections:

- Кампания (Campaign):** 'Здесь можно указать цель кампании и настроить оптимизацию бюджета.' (Here you can specify the campaign goal and set budget optimization). A text input field contains 'трафик, червень 2022' (traffic, June 2022).
- Группа объявлений (Ad Group):** 'Здесь определяются аудитория, места размещения, график и затраты.' (Here the audience, placement, schedule, and costs are defined). It includes a 'Создать групп...' (Create group...) button with a dropdown arrow and a text input field 'Назовите эту группу объявлений' (Name this ad group).
- Объявление (Ad):** 'Здесь находятся все настройки рекламного креатива.' (Here are all the settings for the ad creative). It includes a 'Создать о...' (Create ad...) button with a dropdown arrow and a text input field 'Укажите название этого объявления' (Specify the name of this ad).

Рисунок 1.4 – Ієрархія створення рекламної кампанії

На рівні налаштування рекламної кампанії можна вибрати тип закупки (зазвичай це аукціон). Також можна змінити ціль рекламної кампанії, яку було обрано раніше. На цьому етапі можна увімкнути А/В тестування та оптимізацію бюджету. А/В тестування перевіряє результати різних зображень, написів,

елементів реклами, аналізує їх і за допомогою цього інструменту в майбутньому піднімається ефективність реклами. При оптимізації бюджету кампанії кошти будуть розподілені між усіма групами оголошень, щоб могли отримати найкращі результати з урахуванням ваших рішень щодо оптимізації показу та стратегії ставок.

На етапі налаштування групи оголошень можна обрати куди ми будемо вести людей, які перейшли по рекламі. Це може бути сайт, інстаграм сторінка та багато іншого.

Також можна включити функцію динамічних креативів. На цьому етапі визначається бюджет рекламної компанії та графік його подання.

Один з ключових елементів створення рекламної компанії це вибір аудиторії. На цьому етапі ми можемо створити нову аудиторію або використовувати аудиторію що збережено раніше.

Також ми можемо обрати місця розміщення реклами це може бути сторінка Instagram, stories інстаграм, Facebook пост, Facebook stories та інше [1].

Також можна обрати оптимізацію для показу реклами стандартний показник це кліки по посиланню.

Останнім етапом налаштування рекламної компанії є налаштування оголошення. Тут можна обрати спочатку ідентифікацію компанії – сюди входить сторінка Facebook і сторінки Instagram.

Далі потрібно налаштувати саме рекламне оголошення це може бути одне зображення або відео кільцева галерея або збірка.

Інтерфейс ADS менеджера є дуже простим і з ним зможе розібратися будь-хто.

## **1.8 Висновок до першого розділу**

В першому розділі кваліфікаційної роботи було описано загальні теоретичні відомості стосовно розробки чат-боту для телеграм та налаштування таргетованої реклами.

Було описано вимоги для розробки чат-боту телеграм для інтернет магазину картин. Також описані дії користувачів, в уособленні різних акторів для телеграм-боту.

Висвітлено оцінку методів розв'язання задачі розробки чат-боту для інтернет-магазину картин.

У першому розділі також описано життєвий цикл розробки чат-боту та етапи створення даного проекту.

Також обґрунтовано актуальність сфери таргетованої реклами. Описано загальні положення та інструменти налаштування таргету та його запуску.

## РОЗДІЛ 2. РЕАЛІЗАЦІЯ ЧАТ-БОТУ ТА НАЛАШТУВАННЯ ТАРГЕТОВАНОЇ РЕКЛАМИ ДЛЯ МАГАЗИНУ КАРТИН

У даному розділі буде описано створення програмного комплексу на основі технологій програмування, відповідно до поставленої задачі. Буде детально представлено, як реалізувати програмне забезпечення чат-боту для магазину картин.

Буде детально показано, як проектувалася база даних створеного чат-боту інтернет магазину. Описано пройдені етапи стосовно роботи з SQLite та мовою програмування Python [21].

Також буде описано процес створення та налаштування таргетованої реклами.

### **2.1 Проектування та створення бази даних для чат-боту магазину картин**

Для розробки телеграм чат-боту інтернет магазину картин буде використано СКБД SQLite, що дозволить створити базу даних та керувати нею. Робота з базою даних буде включати в себе такі функції, як зберігання даних у таблицю, зчитування даних з бази, видалення певної інформації з БД, оновлення даних (зміна) [6].

Чат-бот для магазину картин міститиме базу даних, що записується в файлі з розширенням .db і має назву «svit\_kartyn\_db.db». Слід зазначити, що цей файл можна було створити через десктопний застосунок SQLite Studio. Але це було виконано шляхом запису SQL-запитів в код програми, і після запуску телеграм-бота база даних створилася автоматично. Далі представлено код для створення таблиці в базі даних:

```
base.execute('CREATE TABLE IF NOT EXISTS paintings(img TEXT, title TEXT PRIMARY KEY, description TEXT, price TEXT)')
```



Отже, для бази даних даного проекту було створено таблицю «paintings». Структура бази даних чат-боту для інтернет магазину картин представлена на рисунку 2.1.


	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated
1	img	TEXT							NULL
2	title	TEXT							NULL
3	description	TEXT							NULL
4	price	TEXT							NULL

Рисунок 2.1 – Структура БД чат-боту для магазину картин

Таблиця «paintings» містить усі дані про картку товару, яка в подальшому буде виведена для користувача, якщо він натисне на розділ «Наші товари». Ця таблиця служить для створення карток картин у базі даних. Ця таблиця також містить поля: img, title, description та price. Поле «img» - це зображення. Але тут тип не зображення і навіть не бінарний, це лише текстовий формат. Так реалізовано, оскільки в цьому полі записується посилання, яке парситься телеграмом і отримує дані в текстовому форматі. Наступне поле – «title» - це заголовок, або ж назва картини. Воно тут реалізоване як ключ. Зазвичай ще добавляють поле ідентифікатора, але в цьому варіанті саме title виконує його функціонал. Далі поле «description» - це опис картини. Використовується також текстовий тип даних. І останнє поле – це «price», тобто ціна товару-картини.

Оскільки визначено структуру бази даних чат-боту для інтернет магазину картин, можна перейти до її створення. Це можна зробити не одним варіантом. По-перше можна використати застосунок SQLite Studio, це зручно, оскільки розробник може бачити візуальний склад бази даних і відповідно керувати елементами бази даних безпосередньо у застосунку. Там є можливість створення нової бази даних, створення таблиць, проводити зв'язки між ними. При цьому використовуючи SQLite. З'єднання бази даних з програмним кодом здійснюється за допомогою SQL-запитів.

По-друге, створити базу даних та таблиці у ній можна за допомогою програмного коду. Це не є так зручно, як через застосунок SQLite Studio, але це набагато швидше.

Приклад коду, що створює нову базу даних та таблицю в ній представлено у лістингу 2.1.

Лістинг 2.1 – Створення бази даних та таблиці для чат боту інтернет магазину

```
def sql_start():
    global base, cur
    base = sq.connect('svit_kartyn_db.db')
    cur = base.cursor()
    if base:
        print('База даних підключилася успішно!')
        base.execute('CREATE TABLE IF NOT EXISTS paintings(img TEXT,
title TEXT PRIMARY KEY, description TEXT, price TEXT)')
        base.commit
```

Цей код загалом з'єднує вже створені бази даних, але якщо такої немає, то він створює нову.

Тепер, коли є файл бази даних потрібно знати про чотири базових операції і всі вони разом зазвичай називають CRUD. Create, Read, Update, Delete. І завдяки цим чотирьом операціям буде відбуватися маніпуляція даними в базі даних з коду. В коді робота з SQLite виглядає достатньо просто, тим більше в Python вже стандартно встановлений модуль SQLite 3 і нічого додатково через pip встановлювати не потрібно [21].

Просто викликається метод connect, вказуючи першим аргументом назву файлу бази даних, а далі отримуєте курсор-з'єднання і за допомогою нього робимо всі потрібні запити.

Слід зазначити, що було створено новий файл з назвою sqlite\_db.py. Цей файл буде зберігати функції для опису методів взаємодії з базою даних, тобто тут будуть методи по типу перевірки наявності картин в БД, додавання їх туди і так далі.

Такий підхід має багато переваг, як мінімум всі SQL-запити не будуть розкидані по коді, а будуть знаходитися в одному місці. Для створення малих, або середніх ботів такий метод повністю підходить і використовується багатьма розробниками. Якщо писати великого бота, то там вже бази даних описуються за допомогою моделей, ORM, але поки це не потрібно.

Найголовніше потрібно запам'ятати, що коли пишуться запити на мові SQL і відправляється їх на виконання, ні в якому випадку не потрібно конкатинувати `user input` в самий запит. Це потрібно робити за допомогою символу «?», або ж плейсхолдерів, як вони називаються. Якщо так не робити, то відкривається вікно для так званих SQL-ін'єкцій. Коли користувач за допомогою доступного йому інпуту буде видозмінювати ваш запит і зможе, наприклад, видалити всі таблиці. Тому треба бути обережним і завжди застосовувати плейсхолдери [23].

## 2.2 Використання асинхронності для розробки телеграм-боту

Для написання боту потрібно імпортувати класи `bot` та `types`. `Types` – це спеціальний тип даних для того, щоб мати змогу писати анотації типів в функціях. `Dispatcher` – це той клас, що дозволяє боту отримувати події в чаті, або відправку повідомлення користувачем боту і відповідним чином можна прописати його реакцію. `Executor` – для того, щоб запустити бота, щоб він вийшов в онлайн [24].

Також знадобиться імпортувати модуль `os`, для того, щоб ми могли прочитати токен з змінної середовища оточення.

Імпорт всіх цих модулів для чат-боту інтернет магазину картин представлено у лістингу 2.2.

## Лістинг 2.2 – Імпорт необхідних модулів для телеграм-бота

```
from aiogram import Bot, types
from aiogram.dispatcher import Dispatcher
from aiogram.utils import executor
import os
```

Бота можна запустити в двох різних режимах: LongPolling (запускає бота на локальній машині) та Webhook.

В режимі LongPolling бот постійно запитує сервер телеграм на наявність будь яких повідомлень до нього. Мінусом є те, що бот може часто вилітати та завершувати свою роботу. Webhook – бот деплоється на сервер, у бота появляється API [10], тобто є URL-адреса, і сервер телеграм посилає боту повідомлення, як тільки вони тут з'являються.

Для початку потрібно ініціалізувати бота та зчитати токен. Це робиться наступним чином:

```
bot = Bot(token=os.getenv('TOKEN'))
```

Далі потрібно ініціалізувати диспетчер, а вже тоді передати екземпляр боту. У коді це виглядає так:

```
bot = Bot(token=os.getenv("TOKEN"))
dp = Dispatcher(bot, storage=storage)
```

І як показано в прикладі далі буде записано команду запуску чат-боту для магазину картин.

```
executor.start_polling(dp, skip_updates = True, on_startup =
on_startup)
```

Так як aiogram – це асинхронна бібліотека, то перед кожною функцією потрібно писати `async def`, тобто визначати функцію і тут писати назву цієї функції [24].

Python є однопоточним, тобто всі команди виконуються тільки тоді, коли закінчилася попередня, але боту потрібно відповідати на багато повідомлень й робити це дуже швидко. Є варіант створювати багато потоків виконання.

Асинхронність – це коли в потоку виконання з'являється яка-небудь пауза, тобто працює функція, але в неї якийсь час очікування і в потоці з'являється проміжок часу, коли нічого не виконується. Асинхронність дозволяє в ці проміжки виконувати щось інше, чим пришвидшує роботу бота, починає працювати набагато швидше, швидко відповідає на всі повідомлення.

У лістингу 2.3 наведено приклад запису асинхронної функції для чат-боту інтернет магазину.

**Лістинг 2.3 – Асинхронна функція для отримання ID поточного модератора**

```
# отримуємо ID поточного модератора
async def make_changes_command(message: types.Message):
    global ID
    ID = message.from_user.id
    await bot.send_message(message.from_user.id, 'Ви у режимі
адміністратора! Створіть карточку товару',
reply_markup=admin_kb.kb_admin_add)
    await message.delete()
```

Для початку потрібно записати ключове слово `await`, що відноситься до асинхронності, тобто почекати, поки не з'явиться в потоці вільне місце для виконання цієї команди, далі потрібна подія `send_message` і в дужках передаємо те, що хочемо відправити.

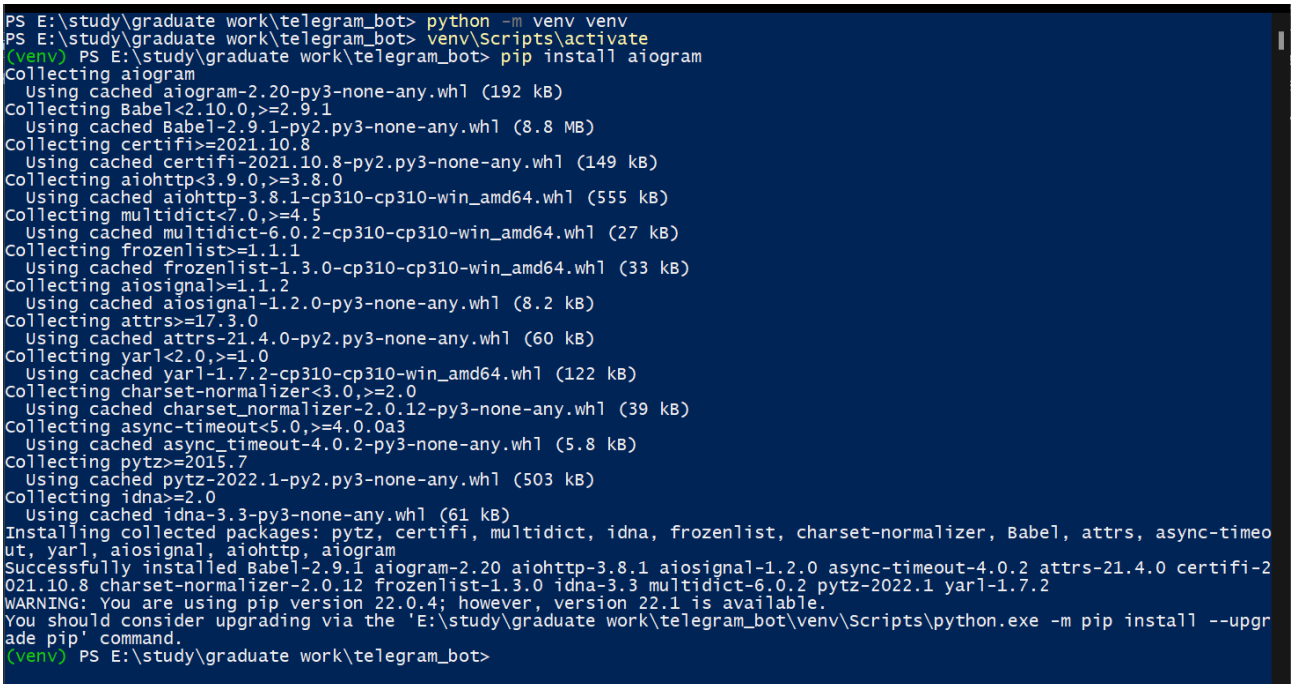
### **2.3 Практична реалізація телеграм-боту для магазину картин**

Отже, перейдемо власне до розробки чат-боту для магазину картин. У ньому користувач зможе переглядати добавлені товари, адмін зможе добавляти їх, редагувати та видаляти. Окрім того, буде реалізовано можливість авторизації адміністратора з правами модератора. Слід зазначити, що це буде

реалізовано не явно. Також буде реалізовано систему цензури. Це зроблено для того, щоб користувачі не могли вводити нецензурну лексику [18].

Перед реалізацією та побудовою загальної структури проекту чат-боту для продажу товарів було створено порожню папку і за допомогою терміналу створено та запущено віртуальне оточення. Воно використовується зазвичай, коли над проектом працюють декілька осіб, або ж спочатку працювали на одному комп'ютері а потім перейшли на інший. Це зроблено для того, щоб не було конфліктів версій бібліотек та самого Python. І в загальному для розробки чат-ботів телеграм прийнято використовувати не загальний інтерпретатор python, а віртуальне оточення venv [21].

Процес створення віртуального оточення для телеграм-боту зображено на рисунку 2.2.



```

PS E:\study\graduate work\telegram_bot> python -m venv venv
PS E:\study\graduate work\telegram_bot> venv\Scripts\activate
(venv) PS E:\study\graduate work\telegram_bot> pip install aiogram
Collecting aiogram
  Using cached aiogram-2.20-py3-none-any.whl (192 kB)
Collecting Babel<2.10.0,>=2.9.1
  Using cached Babel-2.9.1-py2.py3-none-any.whl (8.8 MB)
Collecting certifi>=2021.10.8
  Using cached certifi-2021.10.8-py2.py3-none-any.whl (149 kB)
Collecting aiohttp<3.9.0,>=3.8.0
  Using cached aiohttp-3.8.1-cp310-cp310-win_amd64.whl (555 kB)
Collecting multidict<7.0,>=4.5
  Using cached multidict-6.0.2-cp310-cp310-win_amd64.whl (27 kB)
Collecting frozenlist>=1.1.1
  Using cached frozenlist-1.3.0-cp310-cp310-win_amd64.whl (33 kB)
Collecting aiosignal>=1.1.2
  Using cached aiosignal-1.2.0-py3-none-any.whl (8.2 kB)
Collecting attrs>=17.3.0
  Using cached attrs-21.4.0-py2.py3-none-any.whl (60 kB)
Collecting yarl<2.0,>=1.0
  Using cached yarl-1.7.2-cp310-cp310-win_amd64.whl (122 kB)
Collecting charset-normalizer<3.0,>=2.0
  Using cached charset-normalizer-2.0.12-py3-none-any.whl (39 kB)
Collecting async-timeout<5.0,>=4.0.0a3
  Using cached async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Collecting pytz>=2015.7
  Using cached pytz-2022.1-py2.py3-none-any.whl (503 kB)
Collecting idna>=2.0
  Using cached idna-3.3-py3-none-any.whl (61 kB)
Installing collected packages: pytz, certifi, multidict, idna, frozenlist, charset-normalizer, Babel, attrs, async-timeout, yarl, aiosignal, aiohttp, aiogram
Successfully installed Babel-2.9.1 aiogram-2.20 aiohttp-3.8.1 aiosignal-1.2.0 async-timeout-4.0.2 attrs-21.4.0 certifi-2021.10.8 charset-normalizer-2.0.12 frozenlist-1.3.0 idna-3.3 multidict-6.0.2 pytz-2022.1 yarl-1.7.2
WARNING: You are using pip version 22.0.4; however, version 22.1 is available.
You should consider upgrading via the 'E:\study\graduate work\telegram_bot\venv\Scripts\python.exe -m pip install --upgrade pip' command.
(venv) PS E:\study\graduate work\telegram_bot>

```

Рисунок 2.2 – Створення віртуального оточення для телеграм-боту

Після чого створюється основний файл «bot\_telegram.py» та всі інші файли.

### 2.3.1 Створення та опис файлу сценарію для чат-боту

Для того, щоб не запускати бот постійно вводячи команду в термінал було вирішено створити файл з розширенням .bat – це файл запуску консольних команд, або ж файл сценарію. Цей файл містить код для запуску телеграм-бота. Також в ньому записаний токен для бота, це зроблено задля забезпечення захисту, щоб інші користувачі не отримали доступ до управління кодом чат-бота для магазину картин. Код файлу bot\_run.bat представлений у лістингу 2.4

#### Лістинг 2.4 – Код файлу bot\_run.bat для запуску чат-бота

```
@echo off
call %~dp0telegram_bot\venv\Scripts\activate
cd %~dp0telegram_bot
set TOKEN=5394763744:AAH_WQ9pn-A5aYnlxx6XBilQ1IW6lePR7q4
python bot_telegram.py
pause
```

Команда @echo off потрібна для того, щоб службова інформація самого bat файлу постійно не засмічувала консоль.

Тепер нам тут необхідно активувати віртуальне оточення. Так як файл активації віртуального оточення це також bat файл, то ми записуємо тут команду call, тобто нам потрібно його примусово викликати. Для того, щоб отримати розташування bat-файлу є спеціальна команда %~dp0 після чого потрібно записати шлях до файлу activates в папці віртуального оточення.

Далі, активувавши віртуальне оточення необхідно перейти командною стрічкою в папку проекту. Тут записуємо команду cd – це стандартна команда переходу по папках командної стрічки. Переходимо всередину папки проекту й запускаємо основний файл телеграм-боту.

І наступне, потрібно отримати токен авторизації бота від головного бота телеграму, за допомогою якого створюються всі боти - bot\_father. Токен програмою буде отримуватися з bat-файлу.

### 2.3.2 Робота з елементами проекту чат-боту для магазину картин

Оскільки в проекті чат-боту для інтернет магазину картин є багато різних файлів, які відповідають за різні функції та різні сторінки, слід ретельно дивитися та якісно налаштувати їх взаємодію між собою.

На рисунку 2.3 зображено файлову структуру веб-застосунку закритого тематичного клубу.

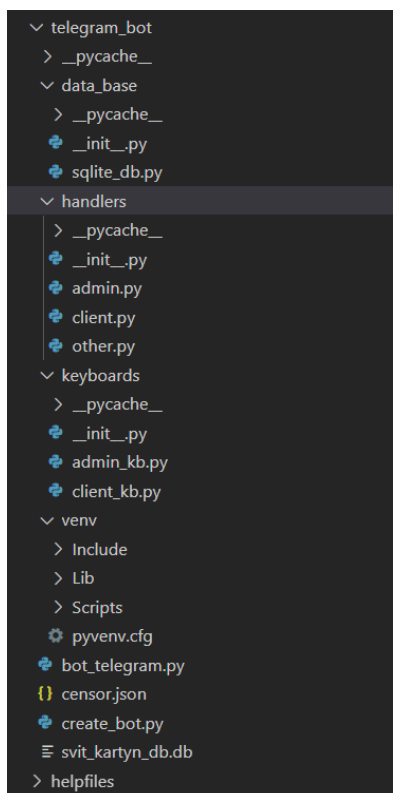


Рисунок 2.3 – Файлова структура застосунку чат-боту для магазину картин

Варто звернути увагу на те, що файли застосунку розкидані по різних блоках, це зроблено для реалізації модульної структури побудови чат-боту для магазину. Тут наявні 3 основних блоки – `data_base`, `handlers` та `keyboards`.

Так, наприклад, модуль `data_base` містить файли які відповідають за реалізацію бази даних, та за з'єднання її з основною програмою. В цьому блоці є файл ініціалізації – «`__init__.py`», який представляє елементи з цього модуля в основну папку.



Блок `keyboards` відповідає за побудову кнопок та меню для чат-боту інтернет магазину. Тут також окремо розділені елементи для клієнтської частини та для адміністратора. В цих файлах створюються макети кнопок, та цілих поєднань кнопок і даліше передаються в основні файли організації логіки програми [25].

Папка `handlers` містить файли, які відповідають за обробку загальної логіки програми. Тут також є окремо прописані клієнтська та частина адміністратора. Також ще наявний файл `other.py`, в якому є код для перевірки цензури.

В кожному з цих файлів реєструються холдери, і даліше передаються в основний файл програми. Приклад реєстрації холдерів представлений у лістингу 2.5.

**Лістинг 2.5 – Реєстрація функцій (холдерів) для клієнтської частини чат-боту для магазину картин**

```
def register_handlers_client(dp : Dispatcher):
    dp.register_message_handler(command_start, commands=['start',
    'help'])
    dp.register_message_handler(command_menu, commands=['меню'])
    dp.register_message_handler(command_shops,
    commands=['наші_магазини'])
    dp.register_message_handler(command_painting,
    commands=['наші_товари'])
    dp.register_message_handler(command_order,
    commands=['картина_під_замовлення'])
    dp.register_message_handler(command_chat_manager,
    commands=['чат_з_менеджером'])
```

Виклик цих функцій в основний файл проекту чат-боту для магазину картин відбувається наступним чином:

```
client.register_handlers_client(dp)
admin.register_handlers_admin(dp)
other.register_handlers_other(dp)
```

Варто зазначити, що саме такий порядок виклику цих функцій в файлі основної програми є коректним. Зміни порядку запису холдерів можуть призвести до виникнення помилки.

### 2.3.3 Розробка клієнтської та адмін-частини чат-боту для магазину картин

У файлі `client.py` описано всю логіку роботи меню та функцій, які доступні для простого користувача. Тут представлені асинхронні функції, кожна з яких відповідає за певний пункт меню.

Метод `send_message` виводить користувачу повідомлення різного типу в залежності від обраного пункту меню, а параметр `reply_markup` використовує значення відповідних кнопок, щоб дати змогу користувачу обрати наступний крок.

Варто ще сказати, що для пункту меню з переглядом товару, програма звертається до бази даних і відповідно використовуючи звернення до функції з sql-запитом виводить зображення та опис картин користувачу.

Ось це і є вся логіка клієнтської частини. Приклад функції представлений у лістингу 2.6.

#### Лістинг 2.6 – Функція виводу картки товару та її опис

```
async def command_painting(message : types.Message):
    await bot.send_message(message.from_user.id, "Тут ви можете побачити створені нами картини", parse_mode='html', reply_markup=kb_client_start)
    await sqlite_db.sql_read(message)

# реєструємо холдери
def register_handlers_client(dp : Dispatcher):
    dp.register_message_handler(command_painting, commands=['наші_товари'])
```

У розробленого бота є власник (адміністратор), який розробляє програмне забезпечення, редагує та керує ботом. Для нього буде розроблено

адмін-панель. Адмін-панелі можна розробляти багатьма способами, можна вивести на окремий сайт, а можна інтегрувати в самий телеграм.

Для чат-бота не важливо яка команда спрацювала першою, яка другою, те що йому в моменті надходить, те він і виконує. Але в даному випадку потрібно провести анкетування користувача (адміна). Бот пропонує йому завантажити зображення картини, потім ввести назву картини, опис та ціну. Бот це все має запам'ятати для того, щоб пізніше відправити в базу даних отримані записи. Для цього в програмі чот-боту для інтернет-магазину картин було використано технологію машинного стану FSM.

Основна суть полягає в тому, що бот використовує сховища для зберігання даних, одне з таких – це `storage.memory`, цей клас дозволяє зберігати дані в оперативній пам'яті. Щоб його підключити, потрібно ввести наступне:

```
from aiogram.contrib.fsm_storage.memory import MemoryStorage
```

Далі потрібно створити клас, який містить стани запам'ятовування програмою та елементи, які далі будуть записані в базу даних. Опис цього класу представлений у лістингу 2.7.

#### Лістинг 2.7 – Код опису класу для FSM адмін-панелі

```
class FSMAdmin(StatesGroup):
    photo = State()
    title = State()
    description = State()
    price = State()
```

Код, який демонструє використання машинного стану представлений у лістингу 2.8

## Лістинг 2.8 – Код, що завантажує фото і записує його в стан для FSM адмін-панелі

```
# Лапаємо першу відповідь і записуємо в словник
async def load_photo(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['photo'] = message.photo[0].file_id
        await FSMAdmin.next()
        await message.reply('Ведіть назву',
reply_markup=admin_kb.kb_admin_cancel)
```

Також в програмі реалізована система цензури яка забороняє ввведення користувачем нецензурної лексики, блокує її і повідомляє користувача про порушення правил.

Для цієї задачі був сформований текстовий файл з базою нецензурний лексику якою може вводитись користувачами. Далі цей файл переводився в JSON формат. Код файлу переведення текстового формату в JSON представлений у лістингу 2.9.

## Лістинг 2.9 – Код, що переводить текстовий файл в JSON

```
import json

ar = []

with open('censor.txt', encoding='utf-8') as r:
    for i in r:
        n = i.lower().split('\n')[0]
        if n != '':
            ar.append(n)

with open('censor.json', 'w', encoding='utf-8') as e:
    json.dump(ar, e)
```

У лістингу 2.10 наведений код функції, яка перевіряє введені значення користувачем і видаляє повідомлення в разі нецензурної лексики. Також результатом є повідомлення користувачу про порушення ним правил.

### Лістинг 2.10 – Код функції, яка блокує ненормативну лексику

```

async def censor_checker(message : types.Message):
    if {i.lower().translate(str.maketrans(' ', ' ',
string.punctuation)) for i in message.text.split(' ')}\
        .intersection(set(json.load(open('censor.json')))) !=
set():
    await message.reply('Нецензурна лексика заборонена!')
    await message.delete()

```

Отже, в кожній частині програми реалізований певний функціонал, який працює коректно та асинхронно.

Коди всіх файлів, що використовуються в дипломній роботі представлені в додатках

## 2.4 Налаштування та запуск таргетованої реклами для магазину картин

Отже, на першому етапі потрібно обрати ціль рекламної кампанії. За стратегією було визначено, що ціллю для цієї РК буде трафік (Див. рисунок 2.4).

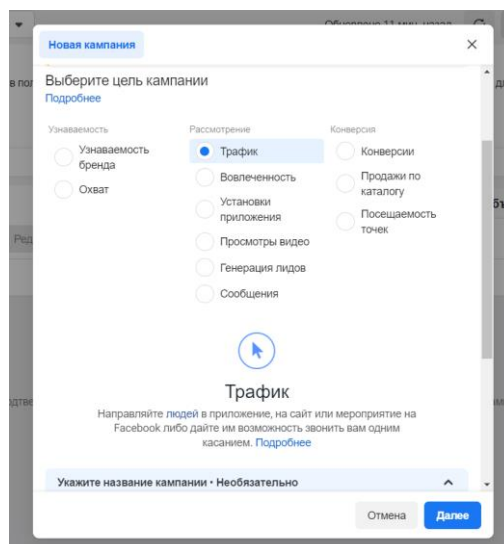


Рисунок 2.4 – Вибір цілі рекламної кампанії

Весь трафік рекламної кампанії буде переводитися на сайт, а в якості посилання буде посилання на Instagram-сторінку акаунту svit.kartyn. Рекламний бюджет буде виставлено 10\$ на добу.

Далі потрібно налаштувати аудиторію рекламної кампанії. Це найважливіший етап, оскільки саме від якості аудиторії залежить чи будуть купляти наш товар і чи реклама дасть результат.

Для аудиторії було обрано людей віком від 18 до 40 років, що живуть на території України, розмовляють українською мовою та мають зацікавленості живописом та дизайном інтер'єрів.

Варто зазначити, що можна обрати декілька аудиторій, створити декілька груп рекламних оголошень та тестувати, яка аудиторія спрацює краще, вже тоді масштабувати цей варіант.

Далі було обрано в якості місця розміщення реклами – пости фейсбук та інстаграм та сторіси фейсбук та інстаграм.

У якості оголошення було вирішено запустити кільцеву галерею з фото товарів та посиланням на сторінку інстаграм (Див. рисунок 2.5).

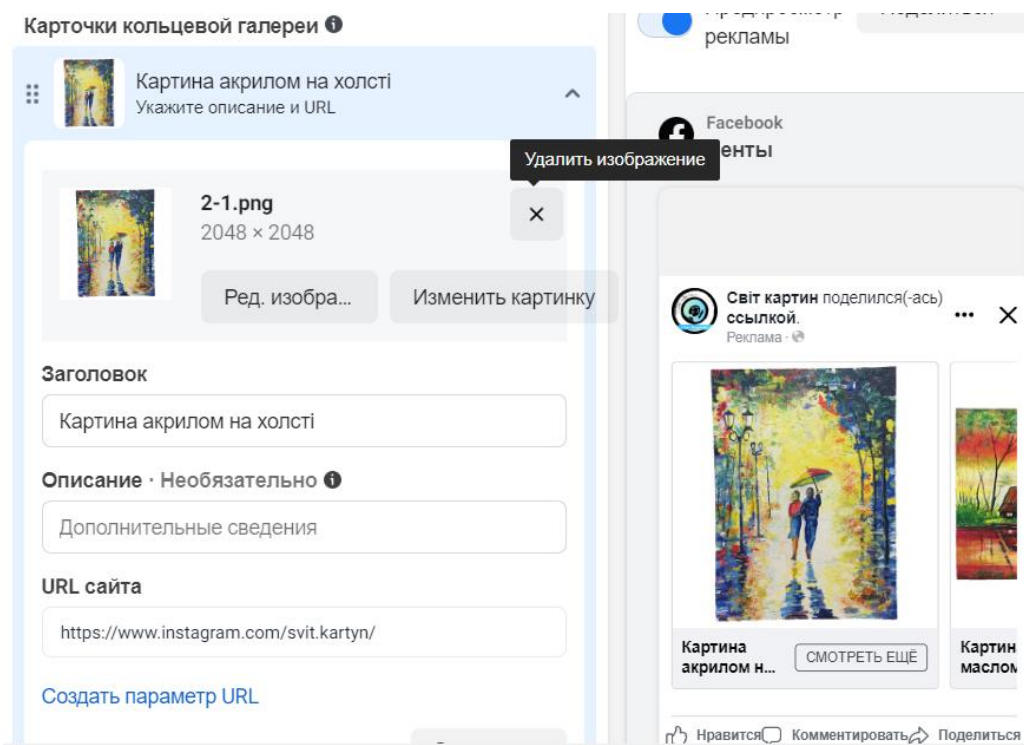


Рисунок 2.5 – Оформлення рекламного креативу

Після цього запускається рекламна кампанія і спостерігаються результати. Шляхом аналізу показників ефективності реклами робляться висновки і приймаються рішення стосовно подальших дій.

## **2.5 Висновок до другого розділу**

В другому розділі кваліфікаційної роботи було спроектовано та описано базу даних для чат-боту онлайн магазину картин. Висвітлено під'єднання бази даних до коду програми та їх взаємодію.

Загально описано та представлено на прикладах основну функцію асинхронності для написання чат-ботів мовою програмування Python.

Також продемонстровано та описано практичну розробку чат-боту для магазину картин. Описано основні функції та елементи створеного програмного забезпечення.

І на останок продемонстровано процес налаштування таргетованої реклами та описано ключові моменти стосовно цієї галузі.

## РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 3.1 Працездатність людини – оператора

У процесі розробки чат-боту, чи запуску таргетованої реклами людина переживає різні функціональні стани, які зумовлюють різні рівні її працездатності.

Під працездатністю людини розуміють можливість її виконувати роботу з необхідною якістю та в установлений час. Працездатність людини залежить як від зовнішніх чинників, так і від внутрішнього стану (внутрішні чинники).

До зовнішніх чинників належать: кількість та форма отриманої інформації, зручність робочого місця, характер взаємостосунків в колективі, вплив чинників середовища існування.

До внутрішніх чинників належать: рівень підготовки, тренованість людини та її емоційна стійкість.

У процесі роботи людина переживає різні функціональні стани, які зумовлюють різні рівні її працездатності.

На рисунку 3.1 наведено зміни функціонального стану та якості роботи людини у процесі одного трудового циклу (зміни).

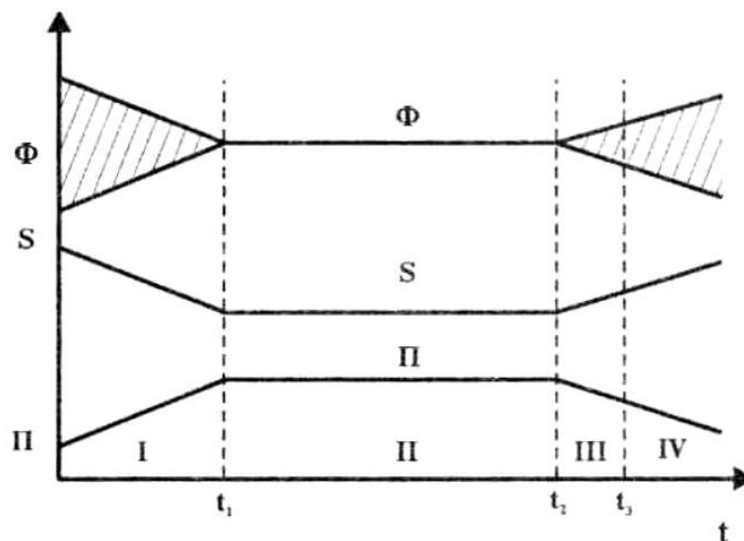


Рисунок 3.1 – Фази працездатності



Позначення на рисунку 3.1 розшифровуються наступним чином: Ф – показник функціонального стану, Б – помилки роботи, П – продуктивність праці.

Виділяють 4 фази працездатності: пристосування до праці, стійка працездатність, субкомпенсація та втома. Тривалість усіх фаз та усього циклу роботи залежить від рівня підготовки людини до роботи [30].

Фаза пристосування до праці – це час, протягом якого людина адаптується до майбутніх умов праці. Основний показник поступово досягає свого встановленого значення. Тривалість періоду пристосування організму до умов праці залежить від багатьох чинників, серед яких основними є інтенсивність роботи (чим інтенсивніша робота, тим цей період коротший) та рівень готовності людини до майбутньої роботи.

Значного скорочення фази пристосування до праці можна досягти за рахунок попередньої підготовки людини до роботи (виконання фізичних вправ, адаптації зору, слуху) та шляхом посиленого навчального навантаження. Суть останнього полягає в тому, що оператор перед початком роботи проводить короткочасне тренування щодо розв'язання однієї чи кількох задач підвищеної складності.

Фаза стійкої працездатності характеризується найвищою якістю праці при оптимальних рівнях функціонування фізіологічних систем організму. Тривалість цього періоду залежить від інтенсивності роботи. Чим інтенсивніша праця, тим коротший цей період. Найоптимальніша динамічна робота, коли цей період може бути в десятки разів довшим, ніж при статичній діяльності.

На процес стійкої працездатності великий вплив справляють емоції. Негативні (страх, невпевненість, поганий настрій) знижують працездатність. Позитивні (впевненість, спокій, бадьорий настрій) значно продовжують період стійкої працездатності [29].

Продовження періоду стійкої працездатності можна забезпечити:

- оптимальним рівнем напруги психофізіологічних функцій;
- комфортними умовами праці;

- Правильним поєднанням режимів праці та відпочинку;
- емоційним розвантаженням;
- використанням тонізуючих напоїв (кава, чай), фармакологічних засобів, зокрема препаратів рослинного походження (вітаміни, препарати, які впливають на енергетичні та метаболічні процеси);
- інформуванням людини про наслідки її діяльності, наглядом та контролем її роботи.

Практичний досвід свідчить, що вживання легких стимуляторів допомагає знизити сонливість, сприяє підвищенню працездатності на короткий період. Однак активні стимулятори на відповідальних видах робіт здатні викликати негативний ефект — погіршується самопочуття, знижується рухливість та швидкість реакцій. Поширене серед населення вживання транквілізаторів, викликаючи заспокоєння та запобігаючи розвитку неврозів, може знизити психічну активність, сповільнити реакції, спричинити апатію та сонливість [30].

Фаза субкомпенсації розглядається як початок розвитку втоми. В цей період якість праці ще зберігається на високому рівні, але тільки за рахунок перенапруг и відповідних функцій організму.

Фаза втоми характеризується чітко вираженим зниженням якості роботи при подальшому погіршенні функціонального стану людини. Об'єктивними показниками втоми є зміна частоти пульсу, дихання, зорової та слухової чутливості.

Наступною фазою життєдіяльності людини повинна бути фаза відновлення працездатності (відпочинку), яка може тривати від 3 до 5 хвилин; 60 — 90 хв. і навіть декілька діб.

### **3.2 Вимоги ергономіки до організації робочого місця оператора ПК, агрегату**

Організація робочого місця для розробників чат-ботів та для людей, які налаштовують таргетовану рекламу є дуже важливою, оскільки вся їхня робота відбувається сидячи за комп'ютером. І щоб підвищити ефективність роботи варто знати, як правильно організувати робоче місце.

Робоче місце – це зона простору, що оснащена необхідним устаткуванням, де відбувається трудова діяльність одного працівника чи групи працівників.

Раціональне планування робочого місця має забезпечувати: найкраще розміщення знарядь і предметів праці, не допускати загального дискомфорту, зменшувати втомлюваність працівника, підвищувати його продуктивність праці. Площа робочого місця має бути такою, щоб працівник не робив зайвих рухів і не відчував незручності під час виконання роботи. Важливо мати також можливість змінити робочу позу, тобто положення корпусу, рук, ніг. Проте доцільно виключати або мінімізувати всі фізіологічно неприродні і незручні положення тіла.

Проведені дослідження показують, що при раціональній організації робочих місць продуктивність праці зростає на 15-25% [30].

Гігієнічні вимоги визначають умови життєдіяльності і працездатності людини у процесі взаємодії з технікою і середовищем; показниками є рівень освітлення, температура, вологість, шум, вібрація, токсичність, загазованість тощо.

Антропометричні вимоги визначають відповідність конструкцій техніки антропометричним характеристикам людини (зріст, розміри тіла та окремі рухові ланки). Показниками є раціональна робоча поза, оптимальні зони досягнення, раціональні трудові рухи.

Фізіологічні та психофізіологічні вимоги визначають відповідність техніки і середовища можливостям працівника щодо сприйняття, переробки інформації, прийняття і реалізації рішень.

Організація робочого місця передбачає:

- правильне розміщення робочого місця у виробничому приміщенні;
- вибір ергономічно обґрунтованого робочого положення, виробничих меблів з урахуванням антропометричних характеристик людини;
- раціональне компонування обладнання на робочих місцях;
- урахування характеру та особливостей трудової діяльності.

Загальні принципи організації робочого місця:

- на робочому місці не повинно бути нічого зайвого. Усі необхідні для роботи предмети мають бути поряд із працівником, але не заважати йому;
- ті предмети, якими користуються частіше, розташовуються ближче, ніж ті предмети, якими користуються рідше;
- предмети, які беруть лівою рукою, повинні бути зліва, а ті предмети, які беруть правою рукою — справа;
- якщо використовують обидві руки, то місце розташування пристосувань вибирається з урахуванням зручності захоплення його двома руками;
- робоче місце не повинно бути захаращене;
- організація робочого місця повинна забезпечувати необхідну оглядовість [30].

Статичні напруження працівника в процесі праці пов'язані з підтриманням у нерухомому стані предметів і знарядь праці, а також підтриманням робочої пози.

Робоча поза – це основне положення працівника у просторі: зручна робоча поза має забезпечувати стійкість положення корпусу, ніг, рук, голови працівника під час роботи, мінімальні затрати енергії та максимальну результативність праці.

Найпоширенішими у процесі праці є пози сидячи і стоячи. Проектуючи робоче місце, потрібно враховувати, що при виконанні роботи з фізичним навантаженням бажана поза стоячи, а при малих зусиллях — сидячи [29].

Робоча поза стоячи втомлює людину більше, ніж сидяча. Вона вимагає на 10 % більше енергії, спричиняє підвищення артеріального і венозного тиску крові, розширення вен на ногах, пошкодження ступень, викривлення хребта.

Під час роботи сидячи нижня частина корпусу розслаблена, а основне статичне навантаження припадає на м'язи шиї, спини, таза, стегон. Неправильна сидяча поза може викликати застій крові в ногах, а якщо виконується великий обсяг роботи для пальців рук – запалення суглобів.

Організація робочого місця користувача комп'ютера повинна забезпечувати відповідність усіх елементів робочого місця та їх взаємного розташування ергономічним вимогам.

На рисунку 3.2 зображено робочий стіл з розміщенням користувача ПК та ергономічні вимоги до проектування робочого місця.

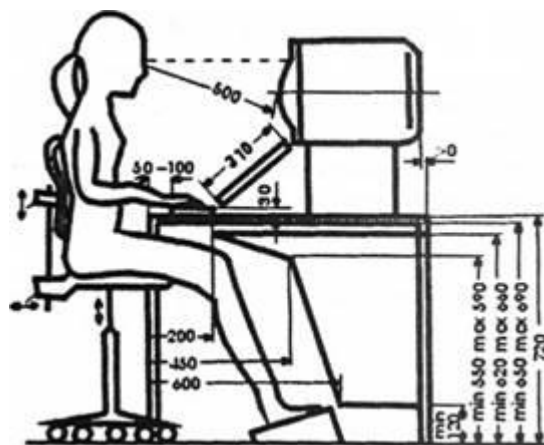


Рисунок 3.2 – Робочий стіл і розміщення користувача ПК

Ергономічні вимоги встановлюють основні параметри робочого місця, оснащеного дисплеєм, і враховують особливість виконуваних робіт.

Так, ергономічні вимоги передбачають наступні елементи: кут екрана, кут огляду (зору), відстань огляду, висота середини екрана, висота клавіатури, висота столу, відстань колін від столу, підставка для ніг, підставка для

документів, положення рук, кут ліктів, спинка крісла, підлокітник, опора для попереку, кут колін, кут спинки крісла, висота сидіння.

### **3.3 Висновок до третього розділу**

В третьому розділі кваліфікаційної роботи описано загальні відомості стосовно працездатності людини – оператора. Під працездатністю людини розуміють можливість її виконувати роботу з необхідною якістю та в установлений час.

Це є актуальним для будь-якої сфери діяльності людини, оскільки працювати потрібно, і якщо знати як правильно організувати роботу щоб підвищити працездатність людини, то і результат буде кращий. Питання працездатності щоденно використовується в розробників чат-ботів та людей, які налаштовують таргетовану рекламу. Вони думають як покращити власну працездатність і загальну базу знань даної теми потрібно знати.

В другій частині третього розділу описано вимоги ергономіки до організації робочого місця оператора ПК. Обрані питання тісно перетинаються, оскільки ергономіка робочого місця безпосередньо впливає і на працездатність людини. Вміння правильно організувати робоче місце піднімає ефективність людини вразі.

Ергономічні вимоги встановлюють основні параметри робочого місця, оснащеного дисплеєм, і враховують особливість виконуваних робіт, що значно впливає на працездатність виконавця роботи.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра було спроектовано та розроблено чат-бот інтернет магазину картин для телеграм. Також налаштовано таргетовану рекламу та запущено її.

В першому розділі кваліфікаційної роботи було описано загальні теоретичні відомості стосовно розробки чат-боту для телеграм та налаштування таргетованої реклами. Було описано вимоги для розробки чат-боту телеграм для інтернет магазину картин. Також описані дії користувачів, в уособленні різних акторів для телеграм-боту.

Висвітлено оцінку методів розв'язання задачі розробки чат-боту для інтернет-магазину картин. Описано життєвий цикл розробки чат-боту та етапи створення даного проекту.

Також обґрунтовано актуальність сфери таргетованої реклами. Описано загальні положення та інструменти налаштування таргету та його запуску.

В другому розділі кваліфікаційної роботи було спроектовано та описано базу даних для чат-боту онлайн магазину картин. Висвітлено під'єднання бази даних до коду програми та їх взаємодію.

Загально описано та представлено на прикладах основну функцію асинхронності для написання чат-ботів мовою програмування Python.

Також продемонстровано та описано практичну розробку чат-боту для магазину картин. Описано основні функції та елементи створеного програмного забезпечення.

І на останок продемонстровано процес налаштування таргетованої реклами та описано ключові моменти стосовно цієї галузі.

У розділі «Безпека життєдіяльності, основи охорони праці» висвітлено загальні відомості стосовно працездатності людини – оператора та описано вимоги ергономіки до організації робочого місця оператора ПК.

## ПЕРЕЛІК ДЖЕРЕЛ

- 1 Gregory A. 9 Tips for Providing Excellent Customer Service [Електронний ресурс] / Alyssa Gregory. – 2019. – Режим доступу до ресурсу: <https://www.thebalancesmb.com/providing-excellent-customer-service-2951744>
- 2 Акуліч М. Чат-боты и маркетинг / Маргарита Акуліч., 2018. – 82 с.
- 3 The History Of Chatbots – From ELIZA to ALEXA [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://onlim.com/en/the-history-of-chatbots/>.
- 4 Sean B. Chatbot History: What is Dr. Sbaitso [Електронний ресурс] / Sean. – 2020. – Режим доступу до ресурсу: <https://yakbots.com/chatbot-history-what-is-dr-sbaitso/>.
- 5 Perez S. Microsoft silences its new A.I. bot Tay, after Twitter users teach it racism [Електронний ресурс] / Sarah Perez. – 2016. – Режим доступу до ресурсу: <https://techcrunch.com/2016/03/24/microsoft-silences-its-new-a-i-bot-tay-after-twitter-users-teach-it-racism/>.
- 6 Rouhiainen L. How AI and Data Could Personalize Higher Education [Електронний ресурс] / Lasse Rouhiainen. – 2019. – Режим доступу до ресурсу: <https://hbr.org/2019/10/how-ai-and-data-could-personalize-higher-education>.
- 7 Юдин А. ТОП МЕССЕНДЖЕРОВ В УКРАИНЕ И МИРЕ 2020 [Електронний ресурс] / Антон Юдин. – 2020. – Режим доступу до ресурсу: <https://marketer.ua/top-messengers-in-ukraine-and-the-world/>.
- 8 Pope L. 45 SMS Marketing Statistics Your Customers Wish You Knew [Електронний ресурс] / Lauren Pope. – 2020. – Режим доступу до ресурсу: <https://learn.g2.com/sms-marketing-statistics>.
- 9 WHY CUSTOMERS PREFER TEXT MESSAGES TO PHONE CALLS [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.brightlink.com/why-customers-prefer-text-messages-to-phone-calls/>.
- 10 Most popular global mobile messaging apps 2021 Published by H. Tankovska, Feb 10, 2021 As of January 2021, two billion users were accessing the WhatsApp messenger on a monthly basis. The app’s reach is usage penetration is



particularly strong in markets [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/>.

11 Коцофанэ О. Все о чат-ботах: типы и примеры, какому бизнесу подойдет, список конструкторов для создания [Электронный ресурс] / Ольга Коцофанэ. – 2019. – Режим доступа до ресурсу: <https://web-promo.ua/blog/vse-o-chat-botah-tipy-i-primery-kakomu-biznesu-podojdet-spisok-konstruktorov-dlya-sozdaniya/>.

12 Wouters J. Flow XO Review [Электронный ресурс] / Joren Wouters. – 2021. – Режим доступа до ресурсу: <https://chatimize.com/reviews/flowxo/>.

13 BroBot – бот для продвижения в Инстаграм со встроенным автоответчиком [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://instagramsoft.ru/brobot-programma-dlya-prodvizheniya-instagram.html>.

14 BotKit — создание бота для мессенджера ТамТам без программирования [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: <https://vc.ru/tribuna/71759-botkit-sozdanie-bota-dlya-messendzhera-tamtam-bez-programmirovaniya>.

15 28 сервисов по созданию чат-ботов в Telegram [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://vc.ru/services/110661-28-servisov-po-sozdaniyu-chat-botov-v-telegram>.

16 Programming language [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.computerhope.com/jargon/p/programming-language.htm>.

17 PYPL PopularitY of Programming Language [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://pypl.github.io/PYPL.html>.

18 TIOBE Index for May 2021 [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.tiobe.com/tiobe-index/>.

19 Rauschmayer D. JavaScript for impatient programmers (ES2021 edition) / Dr. Axel Rauschmayer., 2021. – 319 с.

20 Schildt H. Java: A Beginner's Guide, Eighth Edition / Herbert Schildt., 2018. – (McGraw-Hill Education).

21 Shaw Z. Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code / Zed A. v Shaw., 2016. – 306 с.

22 What is Photoshop [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.agitraining.com/design-news/photoshop-training-news/what-photoshop>.

23 Malizia N. How To Create Your First Telegram Bot [Электронный ресурс] / Nicola Malizia. – 2018. – Режим доступа до ресурсу: <https://unnikked.ga/how-to-create-your-first-telegram-bot-9005c08a5aa5>.

24 Aiogram documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.aiogram.dev/en/latest/>.

25 Урок 5. Клавиатуры и кнопки [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: <https://surik00.gitbooks.io/aiogram-lessons/content/chapter5.html>.

26 Bhardawaj A. What is Marketing Management? [Электронный ресурс] / Bhardawaj. – 2019. – Режим доступа до ресурсу: <https://www.economicdiscussion.net/marketing-management/what-is-marketing-management/31788>.

27 Мілліган З. Міфи про брендинг / З. Мілліган, С. Бейлі., 2020. – 256 с. – (Фабула). – ISBN 978-617-09-6010-8.

28 Soloway. Ростерія "Soloway" [Электронный ресурс] / Soloway. – 2021. – Режим доступа до ресурсу: <https://solowaycoffee.com/>.

29 Управление охраной труда [Электронный ресурс] – Режим доступа до ресурсу: <https://stroilogik.ru/bezopasnost-truda/112-upravlenie-ohranoi-truda.html>.

30 Управление охраной труда в организации [Электронный ресурс] – Режим доступа до ресурсу: [https://studme.org/12461220/bzhd/upravlenie\\_ohranoy\\_truda\\_organizatsii](https://studme.org/12461220/bzhd/upravlenie_ohranoy_truda_organizatsii).

# ДОДАТКИ

## Лістинг файлів bot\_telegram.py та create\_bot.py

```
#file bot_telegram.py
from aiogram.utils import executor
from create_bot import dp
from data_base import sqlite_db

async def on_startup(_):
    print('Бот запустився успішно!')
    sqlite_db.sql_start()

from handlers import client, admin, other

client.register_handlers_client(dp)
admin.register_handlers_admin(dp)
other.register_handlers_other(dp)

executor.start_polling(dp, skip_updates = True, on_startup =
on_startup)

#file create_bot.py
from aiogram import Bot
from aiogram.dispatcher import Dispatcher
import os
from aiogram.contrib.fsm_storage.memory import MemoryStorage

storage=MemoryStorage()

bot = Bot(token=os.getenv("TOKEN"))
dp = Dispatcher(bot, storage=storage)
```

## Лістинг файлів admin.py, client.py та other.py

```
#file admin.py
from ast import Lambda
from cmd import IDENTCHARS
from typing import Text
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters.state import State, StatesGroup
from aiogram import types, Dispatcher
from create_bot import dp, bot
from aiogram.dispatcher.filters import Text
from data_base import sqlite_db
from keyboards import admin_kb
from aiogram.types import InlineKeyboardMarkup,
InlineKeyboardButton

from data_base.sqlite_db import sql_add_command,
sql_delete_command

ID = None

class FSMAdmin(StatesGroup):
    photo = State()
    title = State()
    description = State()
    price = State()

# отримуємо ID поточного модератора
async def make_changes_command(message: types.Message):
    global ID
    ID = message.from_user.id
    await bot.send_message(message.from_user.id, 'Ви у режимі
адміністратора! Створіть карточку товару',
reply_markup=admin_kb.kb_admin_add)
    await message.delete()

# Початок діалогу і загрузка нового пункту меню
async def cm_start(message : types.Message):
    if message.from_user.id == ID:
        await FSMAdmin.photo.set()
        await message.reply('Завантажте фото',
reply_markup=admin_kb.kb_admin_cancel)

# вихід з стану
async def cancel_handler(message: types.Message, state:
FSMContext):
    if message.from_user.id == ID:
        current_state = await state.get_state()
        if current_state is None:
            return
```

```

        await state.finish()
        await message.reply('Створення карточки товару
скасовано!', reply_markup=admin_kb.kb_admin_back)

# Лапаємо першу відповідь і записуємо в словник
async def load_photo(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['photo'] = message.photo[0].file_id
        await FSMAdmin.next()
        await message.reply('Ведіть назву',
reply_markup=admin_kb.kb_admin_cancel)

# Лапаємо другу відповідь
async def load_title(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['name'] = message.text
        await FSMAdmin.next()
        await message.reply("Введіть опис")

# Лапаємо третю відповідь
async def load_description(message: types.Message, state:
FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['description'] = message.text
        await FSMAdmin.next()
        await message.reply("Введіть ціну")

# Лапаємо четврту відповідь
async def load_price(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['price'] = float(message.text)
        await sqlite_db.sql_add_command(state)
        await state.finish()
        await message.reply('Новий товар успішно створений!',
reply_markup=admin_kb.kb_admin_back)

# @dp.callback_query_handler(lambda x: x.data and
x.data.startswith('del '))
async def del_callback_run(callback_query: types.CallbackQuery):
    await
sqlite_db.sql_delete_command(callback_query.data.replace('del ',
''))
    await
callback_query.answer(text=f'{callback_query.data.replace("del ",
"")} видалено!', show_alert=True)

# @dp.message_handler(commands='видалити')
async def delete_item(message: types.Message):

```

```

    if message.from_user.id == ID:
        read = await sqlite_db.sql_read2()
        for ret in read:
            await bot.send_photo(message.from_user.id, ret[0],
f'{ret[1]}\nОпис: {ret[2]}\nЦіна: {ret[-1]}')
            await bot.send_message(message.from_user.id,
text='^^^', reply_markup=InlineKeyboardMarkup().\
            add(InlineKeyboardButton(f'Видалити {ret[1]}',
callback_data=f'del {ret[1]}')))

# реєструємо холдери
def register_handlers_admin(dp : Dispatcher):
    dp.register_message_handler(cm_start,
commands=['завантажити'], state=None)
    dp.register_message_handler(cancel_handler, state="",
commands='скасувати')
    dp.register_message_handler(cancel_handler,
Text(equals='скасувати', ignore_case=True), state="")
    dp.register_message_handler(load_photo,
content_types=['photo'], state=FSMAdmin.photo)
    dp.register_message_handler(load_title, state=FSMAdmin.title)
    dp.register_message_handler(load_description,
state=FSMAdmin.description)
    dp.register_message_handler(load_price, state=FSMAdmin.price)
    dp.register_message_handler(make_changes_command,
commands=['moderator'], is_chat_admin=True)
    dp.register_callback_query_handler(del_callback_run, lambda x:
x.data and x.data.startswith('del '))
    dp.register_message_handler(delete_item, commands='видалити')

#file client.py
from aiogram import types, Dispatcher
from create_bot import dp, bot
from keyboards import kb_client_start, kb_client_menu,
kb_client_shops, kb_client_manager
from data_base import sqlite_db

async def command_start(message : types.Message):
    try:
        mess = f"Привіт, <b>{message.from_user.first_name}</b>.\n
Вас вітає Svit kartyn Bot.\n\nТут Ви можете отримати інформацію
про наші магазини, товари та зв'язатися з нами для замовлення
товару\n\nДля спілкування з нашим ботом, ви можете використати
наступні команди: \n/start \n/help"
        await bot.send_message(message.from_user.id, mess,
parse_mode='html', reply_markup=kb_client_start)
    except:

        await message.reply('Спілкування з ботом можливе через
приватні повідомлення, \nБудь ласка, напишіть йому:
@svit_kartyn_bot')

async def command_menu(message : types.Message):

```

```

        await bot.send_message(message.from_user.id, "Вітаємо в
меню!\n\nБудь ласка оберіть пункт, щоб перейти далі",
parse_mode='html', reply_markup=kb_client_menu)

async def command_shops(message : types.Message):
    await bot.send_message(message.from_user.id, "Поки ми маємо
лише онлайн магазини, згодом відкриємо також офлайн точки",
parse_mode='html', reply_markup=kb_client_start)
    await bot.send_message(message.from_user.id, "Переходьте у наш
Instagram чи OLY", parse_mode='html',
reply_markup=kb_client_shops)

async def command_painting(message : types.Message):
    await bot.send_message(message.from_user.id, "Тут ви можете
побачити створені нами картини", parse_mode='html',
reply_markup=kb_client_start)
    await sqlite_db.sql_read(message)

async def command_order(message : types.Message):
    await bot.send_message(message.from_user.id, "Для створення
картини під замовлення, будь ласка, напишіть нам, що ви хочете
бачити та надішліть приблизний приклад картини",
parse_mode='html', reply_markup=kb_client_start)
    await bot.send_message(message.from_user.id, "Напишіть нашому
менеджеру, або залиште свої контакти, ми зв'яжемося з вами",
parse_mode='html', reply_markup=kb_client_manager)

async def command_chat_manager(message : types.Message):
    await bot.send_message(message.from_user.id, "Це наш менеджер:
@vverbitskyi.\nЗадайте йому будь-яке питання, що вас цікавить",
parse_mode='html', reply_markup=kb_client_start)

# реєструємо холдери
def register_handlers_client(dp : Dispatcher):
    dp.register_message_handler(command_start, commands=['start',
'help'])
    dp.register_message_handler(command_menu, commands=['меню'])
    dp.register_message_handler(command_shops,
commands=['наші_магазини'])
    dp.register_message_handler(command_painting,
commands=['наші_товари'])
    dp.register_message_handler(command_order,
commands=['картина_під_замовлення'])
    dp.register_message_handler(command_chat_manager,
commands=['чат_з_менеджером'])

#file other.py
from aiogram import types, Dispatcher
from create_bot import dp
import json, string

async def censor_checker(message : types.Message):
    if {i.lower().translate(str.maketrans('', '',
string.punctuation)) for i in message.text.split(' ')}\

```



```
        .intersection(set(json.load(open('censor.json')))) !=
set():
    await message.reply('Нецензурна лексика заборонена!')
    await message.delete()

def register_handlers_other(dp : Dispatcher):
    dp.register_message_handler(censor_checker)
```

## Лістинг файлів admin\_kb.py, client\_kb.py та sqlite\_db.py

```
#file admin_kb.py
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton

button_add = KeyboardButton('/завантажити')
button_cancel = KeyboardButton('/скасувати')
button_delete = KeyboardButton('/видалити')
button_back = KeyboardButton('/меню')

kb_admin_add = ReplyKeyboardMarkup(resize_keyboard=True)
kb_admin_cancel = ReplyKeyboardMarkup(resize_keyboard=True)
kb_admin_back = ReplyKeyboardMarkup(resize_keyboard=True)

kb_admin_add.add(button_add).add(button_delete)
kb_admin_cancel.add(button_cancel)
kb_admin_back.add(button_add).add(button_delete).add(button_back)

#file client_kb.py
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton,
InlineKeyboardMarkup, InlineKeyboardButton

b1 = KeyboardButton('/меню')
b2 = KeyboardButton('/наші_магазини')
b3 = KeyboardButton('/наші_товари')
b4 = KeyboardButton('/картина_під_замовлення')
b5 = KeyboardButton('/чат_з_менеджером')
b6 = KeyboardButton('/help')
b7 = InlineKeyboardButton('Instagram',
url="https://www.instagram.com/svit.kartyn/")
b8 = InlineKeyboardButton('OLX',
url="https://www.olx.ua/uk/list/user/Mwq0P/")
b9 = InlineKeyboardButton("Написати менеджеру",
url="https://t.me/vverbitskyi")

kb_client_start = ReplyKeyboardMarkup(resize_keyboard=True)
kb_client_menu = ReplyKeyboardMarkup(resize_keyboard=True)
kb_client_shops = InlineKeyboardMarkup()
kb_client_manager = InlineKeyboardMarkup()

kb_client_start.add(b1)
kb_client_menu.add(b2).add(b3).add(b4).add(b5).add(b6)
kb_client_shops.row(b7, b8)
kb_client_manager.add(b9)

#file sqlite_db.py
from os import curdir
import sqlite3 as sq
from create_bot import dp, bot
```

```
def sql_start():
    global base, cur
    base = sq.connect('svit_kartyn_db.db')
    cur = base.cursor()
    if base:
        print('База даних підключилася успішно!')
        base.execute('CREATE TABLE IF NOT EXISTS paintings(img TEXT,
title TEXT PRIMARY KEY, description TEXT, price TEXT)')
        base.commit

async def sql_add_command(state):
    async with state.proxy() as data:
        cur.execute('INSERT INTO paintings VALUES (?, ?, ?, ?)',
tuple(data.values()))
        base.commit()

async def sql_read(message):
    for ret in cur.execute('SELECT * FROM paintings').fetchall():
        await bot.send_photo(message.from_user.id, ret[0],
f'{ret[1]}\nОпис: {ret[2]}\nЦіна: {ret[-1]}')

async def sql_read2():
    return cur.execute('SELECT * FROM paintings').fetchall()

async def sql_delete_command(data):
    cur.execute('DELETE FROM paintings WHERE title == ?', (data,))
    base.commit()
```

## Приклади результату створення чат-боту для магазину картин

На рисунку Г.1 зображено перше звернення до бота в групі, в яку бота добавлено й назначено адміністратором.

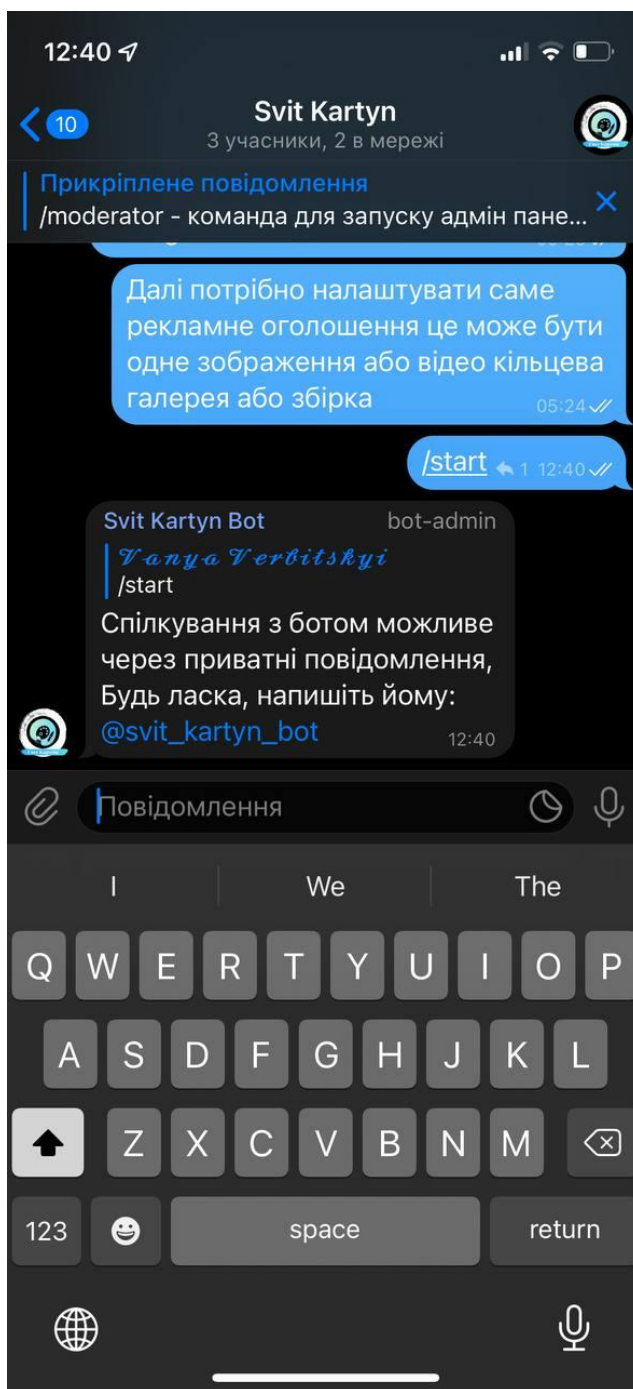


Рисунок Г.1 – Перше звернення до бота

На рисунку Г.2 зображено виконання початкової команди /start.

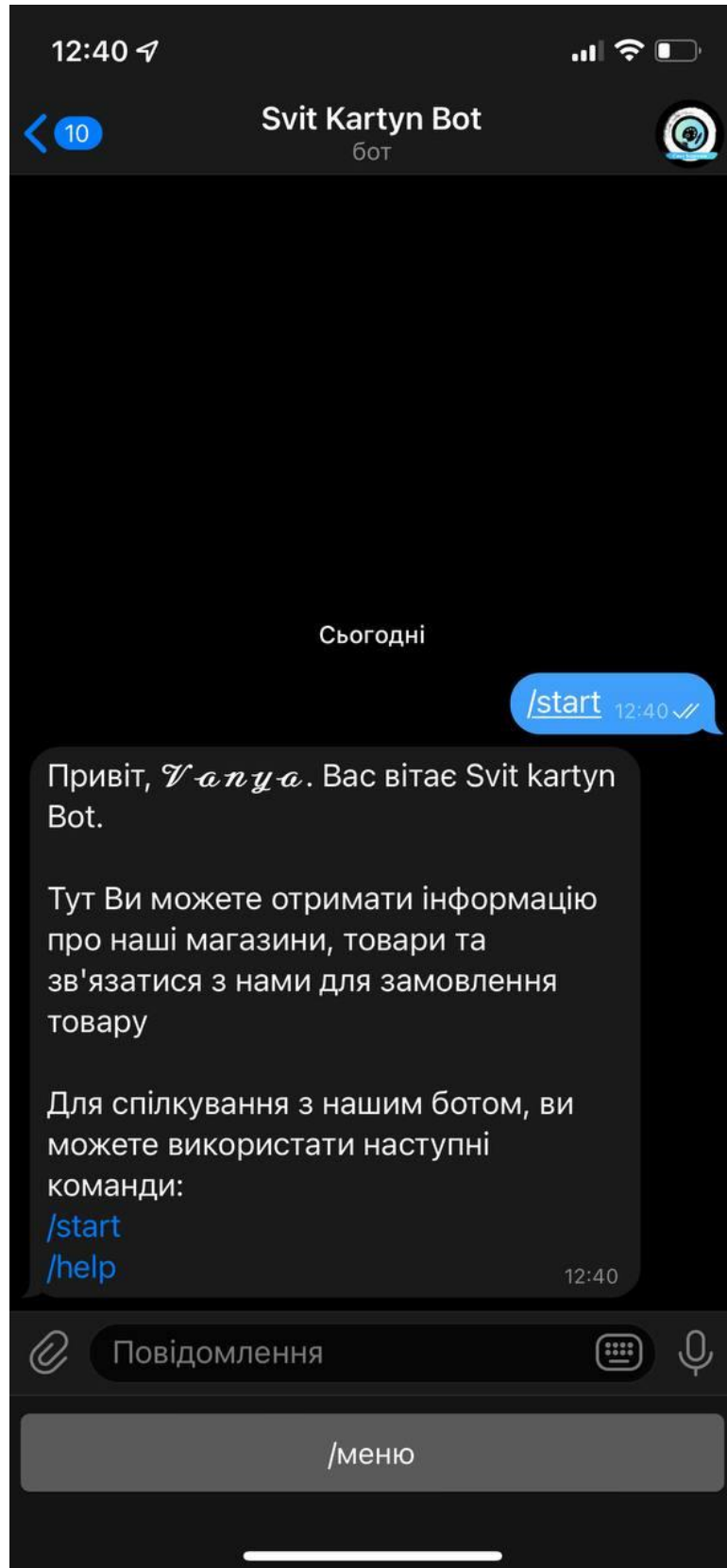


Рисунок Г.2 – Виконання команди /start

На рисунку Г.3 зображено перехід до меню чат-боту. Далі виведено пункти, за допомогою яких можна комунікувати з ботом.

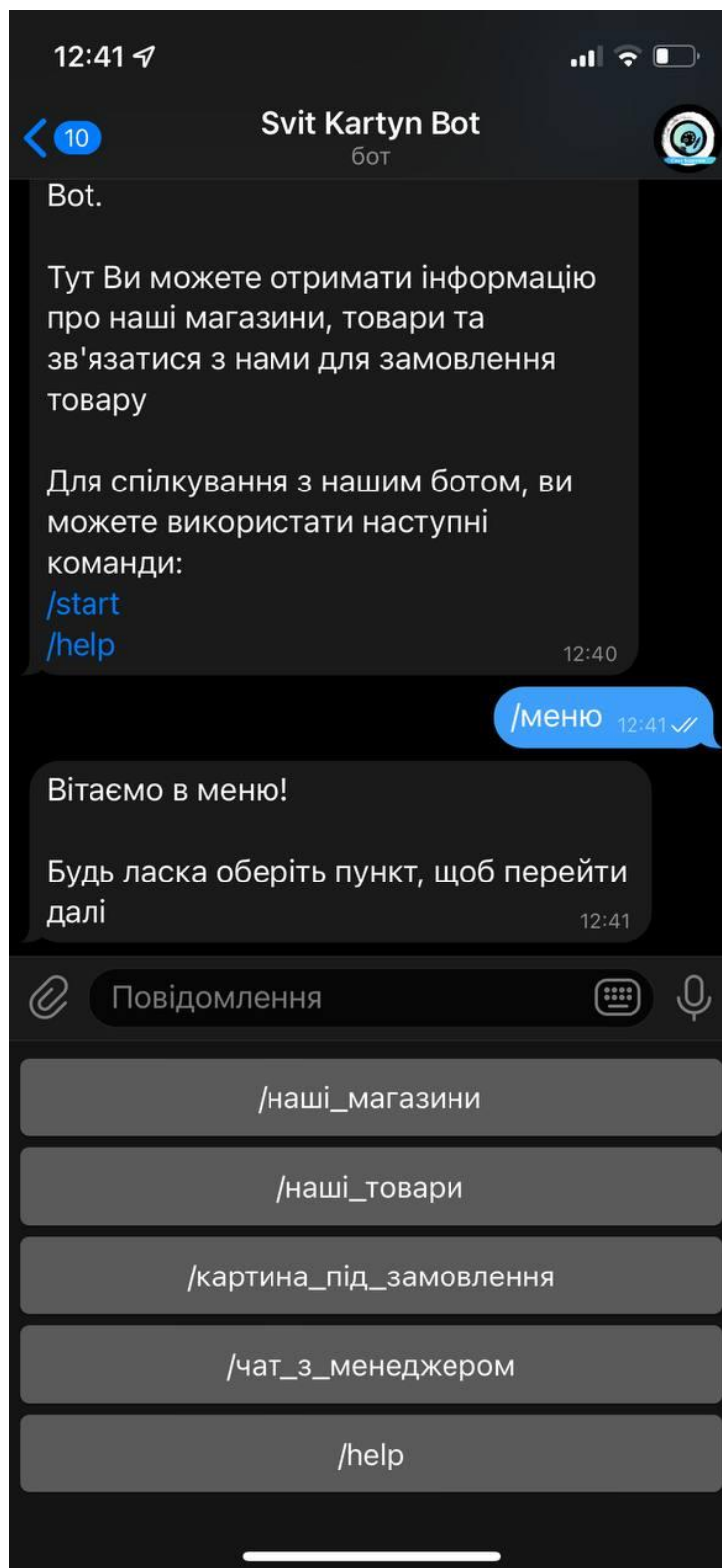


Рисунок Г.3 – Меню чат-бота

На рисунку Г.4 зображено вкладку наші магазини. Де виводиться інформація про магазини та пропонує перейти на сайт і в інстаграм

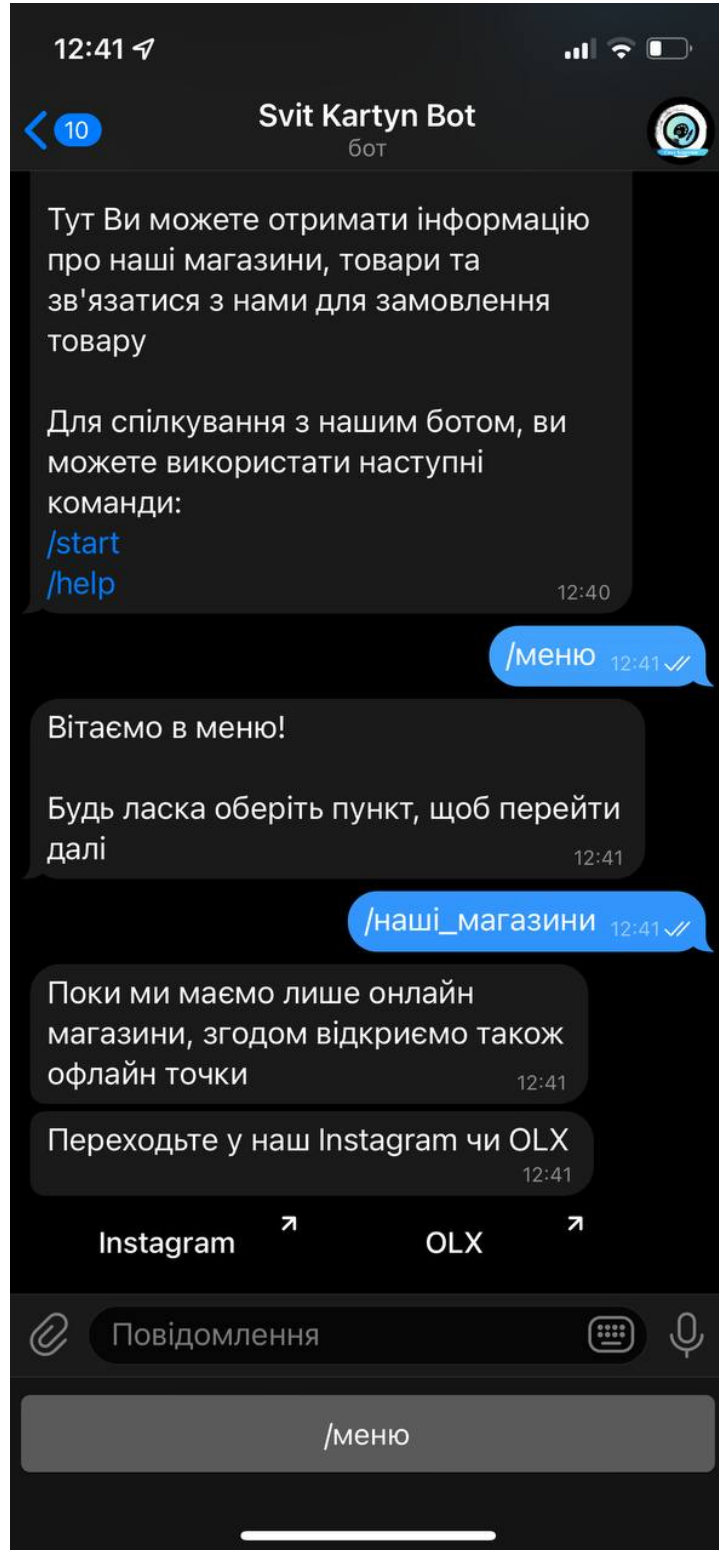


Рисунок Г.4 – Пункт меню «наші магазини»

На рисунку Г.5 зображено пункт меню «Картини під замовлення»

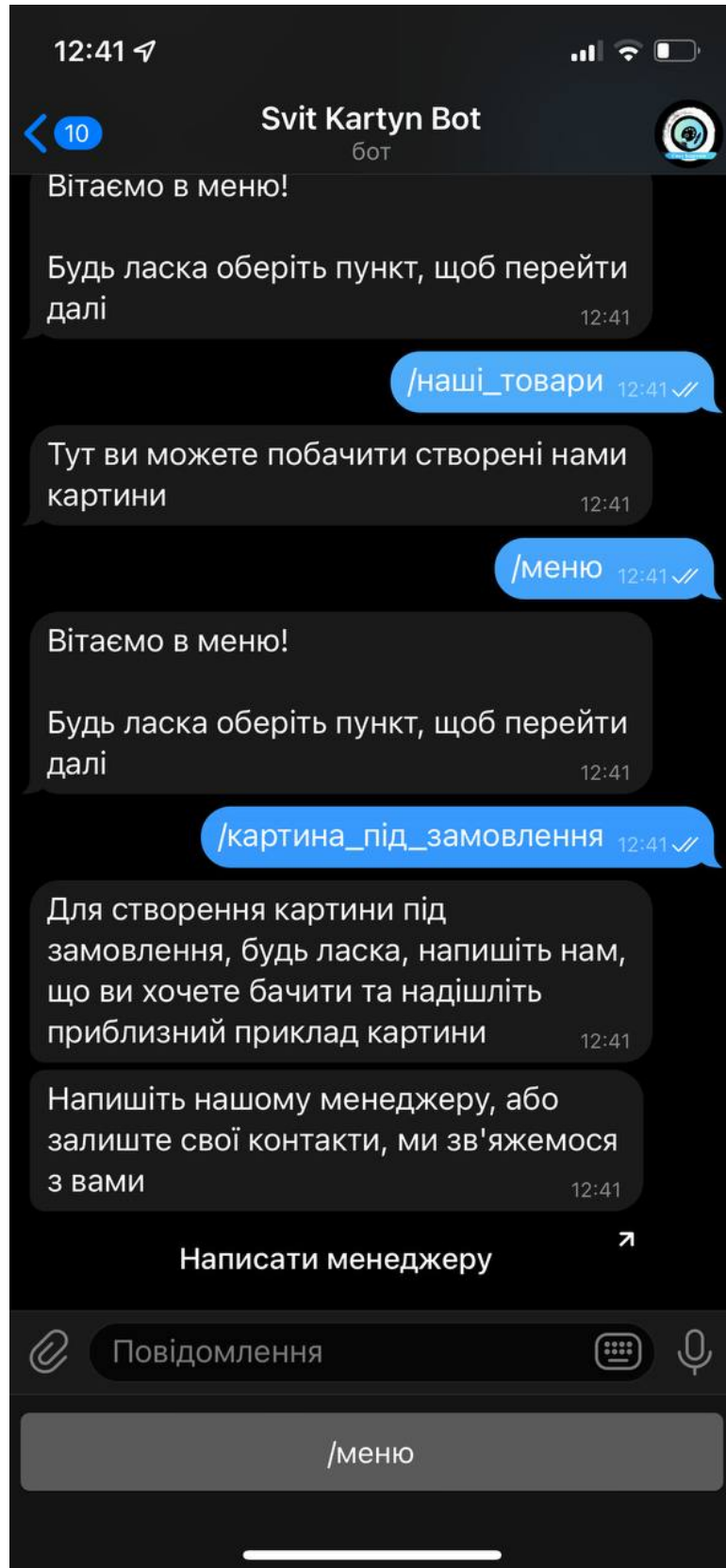


Рисунок Г.5 – Пункт меню «Картини під замовлення»



На рисунку Г.6 зображено пункт меню «Чат з менеджером»

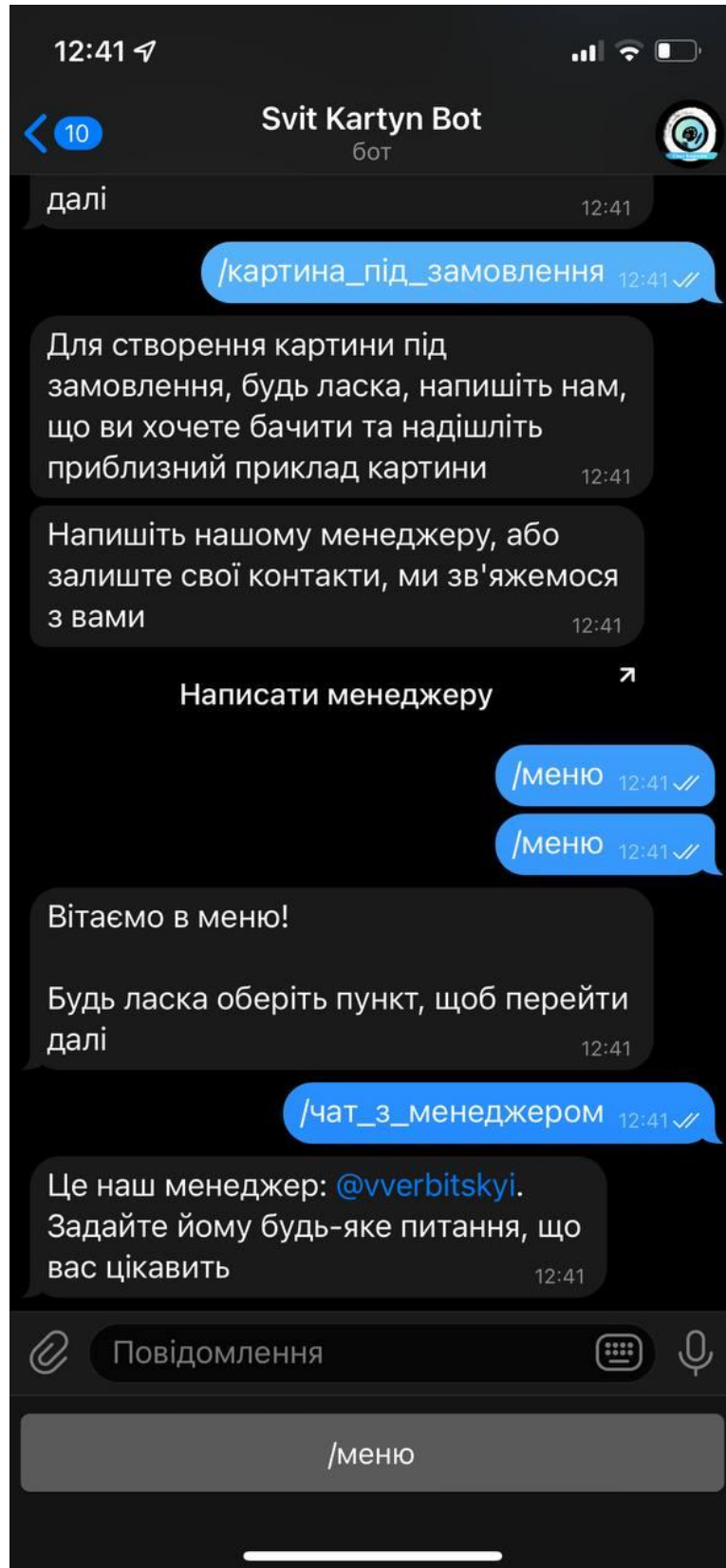


Рисунок Г.6 – Пункт меню «Чат з менеджером»

На рисунку Г.7 зображено пункт меню «Допомога»

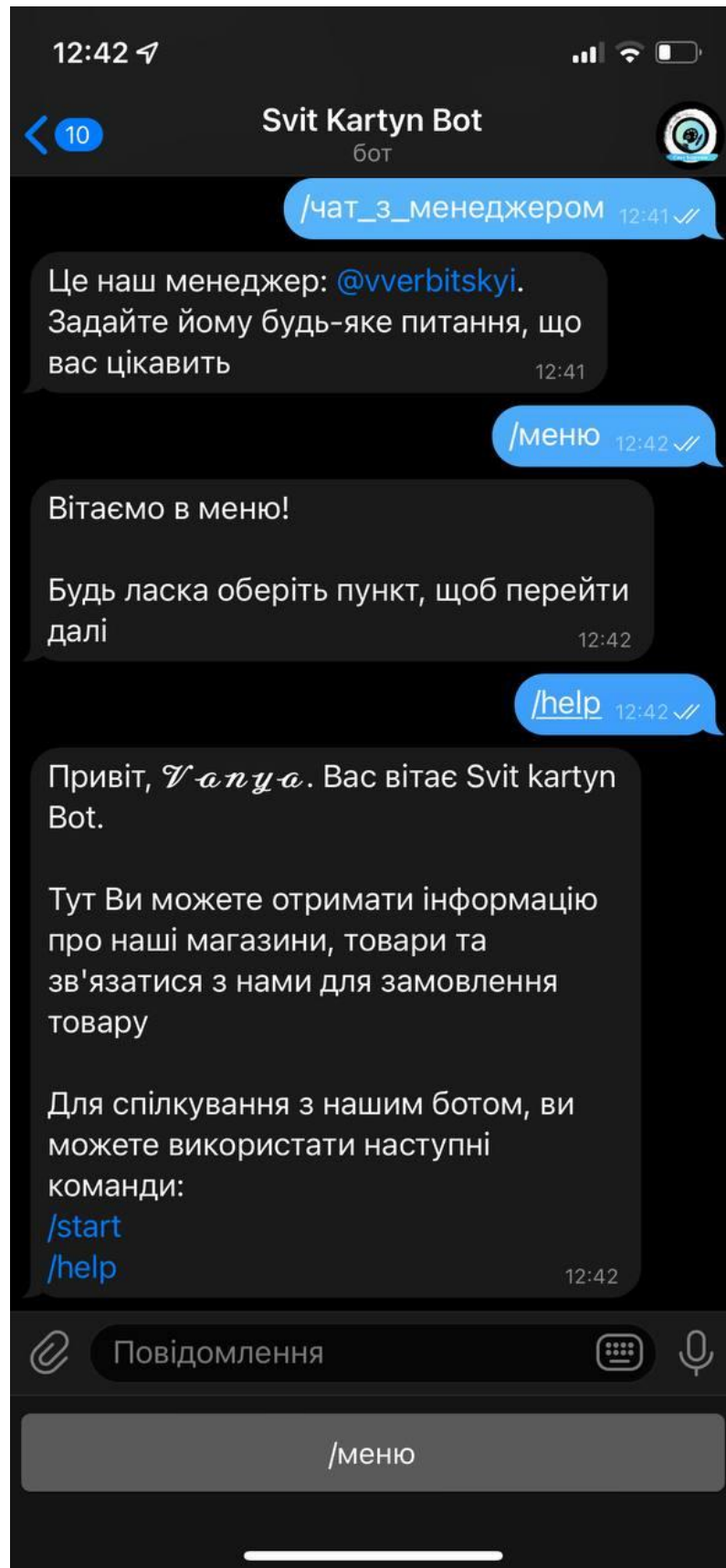


Рисунок Г.7 – Пункт меню «Допомога»

На рисунку Г.8 зображено виконання бота після вводу користувачем ненормативної лексики.

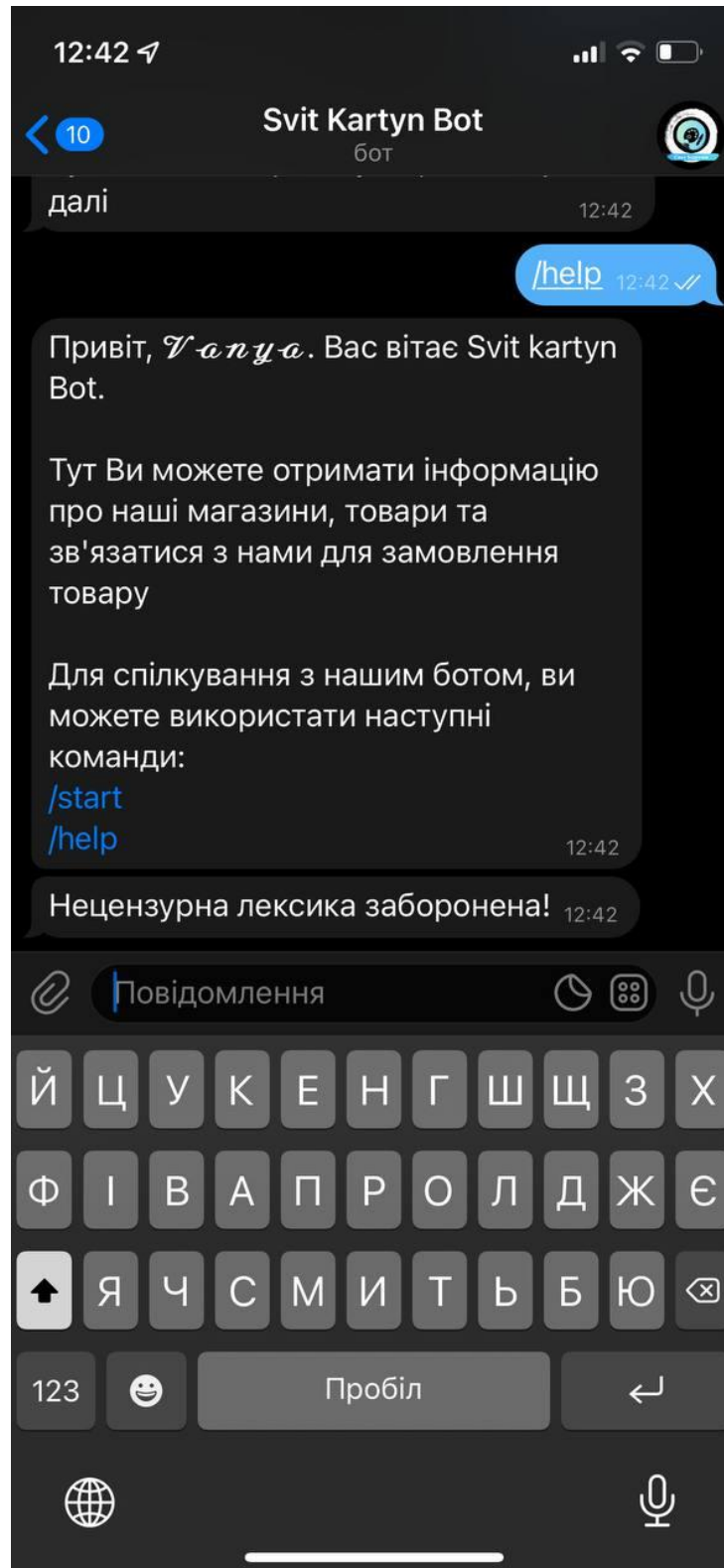


Рисунок Г.8 – Реакція бота на нецензурну лексику

На рисунку Г.9 зображено вхід у режим адміністратора. Для цього в групі, куди добавлено бота потрібно було ввести команду /moderator. Ще слід зауважити, що так можуть зробити лише адміністратори цієї групи.

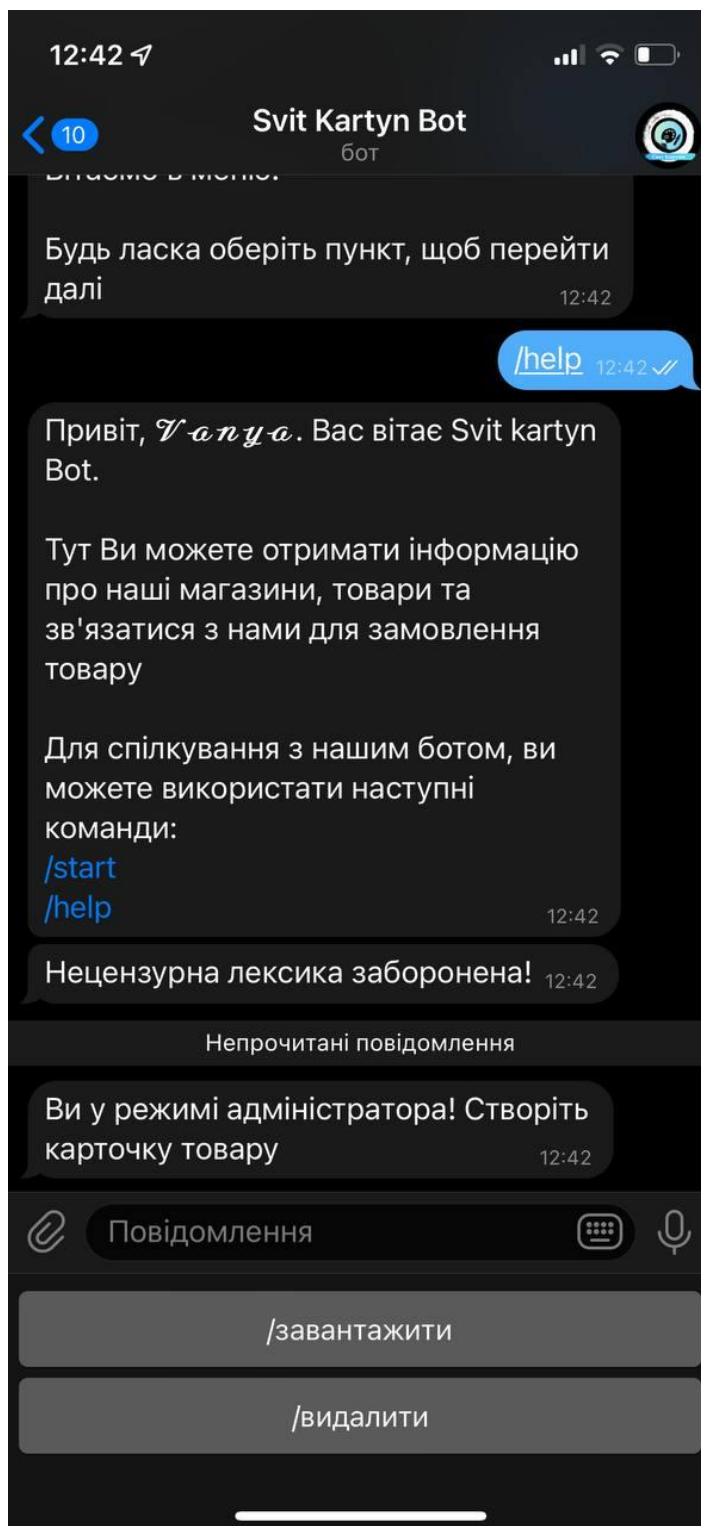


Рисунок Г.9 – Вхід в адмін-панель

На рисунку Г.10 зображено процес додавання нової картки з картиною, та запис її в базу даних.

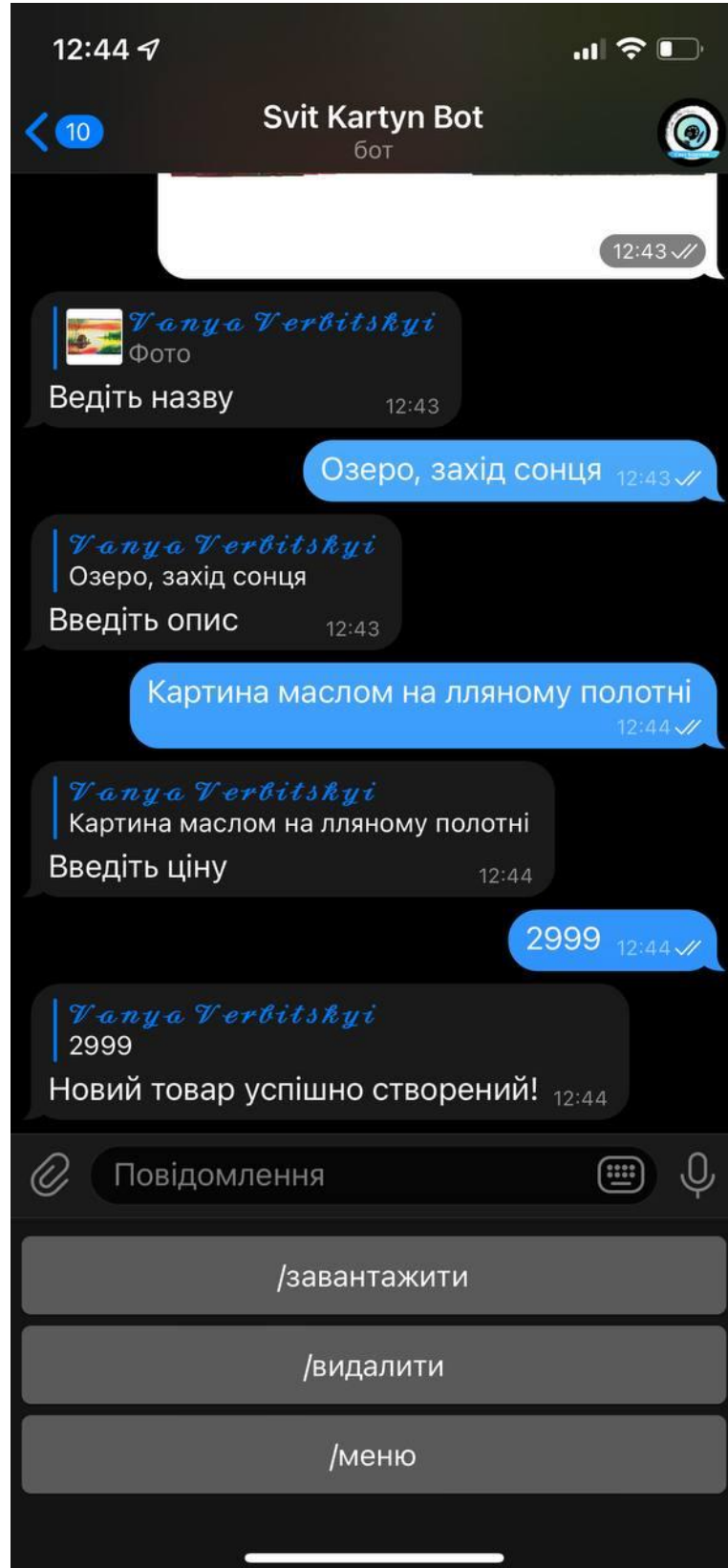


Рисунок Г.10 – Створення нової картки товару

На рисунку Г.11 зображено пункт «Видалення товару»

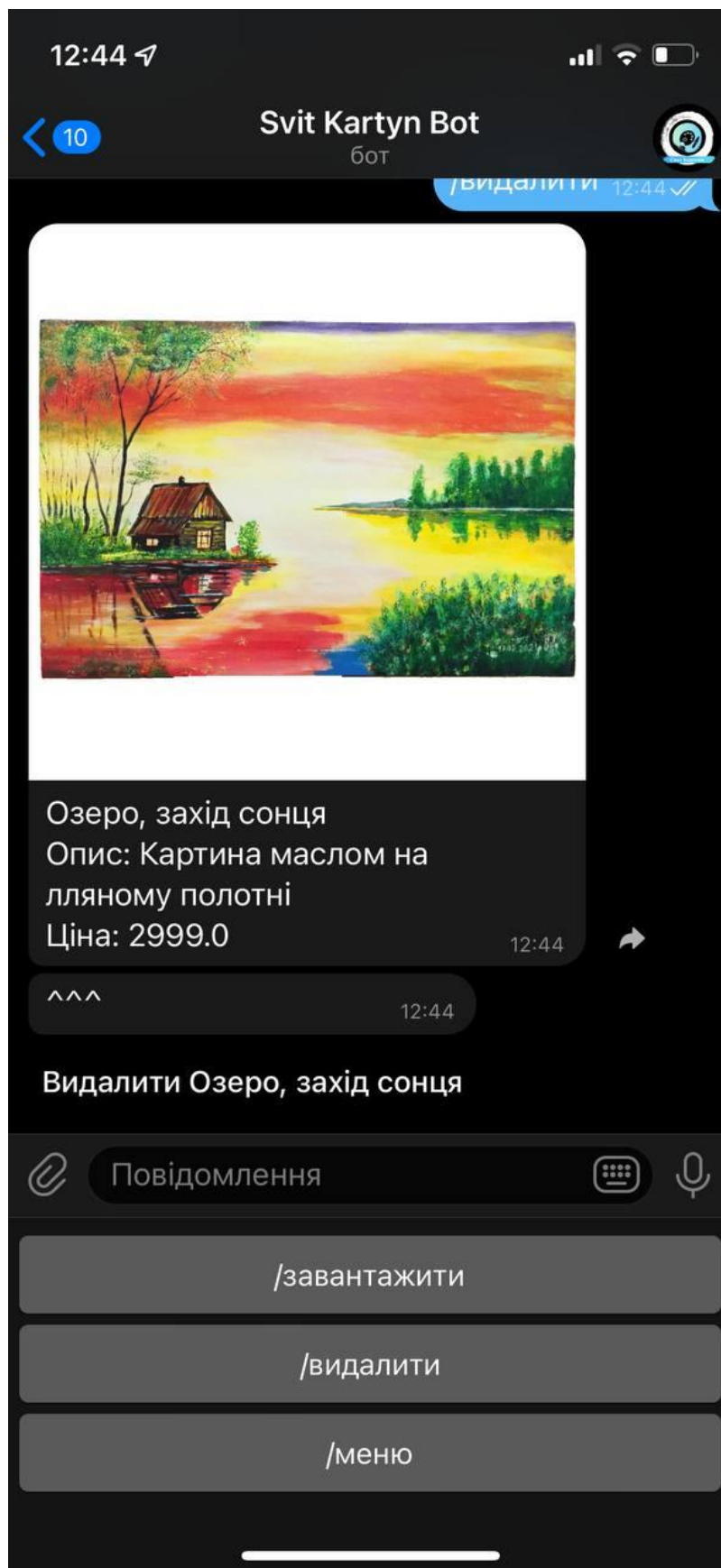


Рисунок Г.11 – Видалення товару