

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач випускової кафедри  
\_\_\_\_\_ Аліна САВЧЕНКО  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР  
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ  
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ»

**Тема:** «Застосування технології чат-ботів для онлайн-магазину з  
продажу геймерських товарів мовою Python»

Виконавець: Рустам НАЗІМОВ

Керівник: д.т.н., доцент Аліна САВЧЕНКО

Нормоконтролер: к.т.н., доцент Олена ТОЛСТІКОВА

# 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)

« \_\_\_\_\_ » \_\_\_\_\_ 2023 р.

## ЗАВДАННЯ

на виконання кваліфікаційної роботи

*Назімова Рустама Євгеновича*

(ПІБ випусника )

1. Тема роботи: «Застосування технології чат-ботів для онлайн-магазину з продажу геймерських товарів мовою Python» затверджена наказом ректора № 623/ст від 01.05.2023р.
2. Термін виконання роботи: з 15 травня 2023 року по 25 червня 2023 року.
3. Вихідні дані до роботи: чат-бот для онлайн-магазину з продажу геймерських товарів мовою Python.
4. Зміст пояснювальної записки: 1. Обґрунтування розроблення чат-бота. 2. Проектування чат-бота. 3. Розроблення чат-бота.
5. Перелік обов'язкового ілюстративного матеріалу: 1. Аналіз предметної області існуючих товарів для геймінгу. 2. Методи підходу до розробки чат-ботів на python. 3. Інструменти розробки. 4. Мови розмітки та програмування. 5. Аналіз мов програмування для розробки телеграм-ботів. 6. Аналіз розширень та бібліотек Python для розробки чат-ботів. 7. Огляд бібліотек та фреймворків для розробки чат-ботів на Python. 8. Проектування програмного забезпечення для телеграм-боту. 9. Написання коду. 10. Демонстрація готового продукту.

## 6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Ознайомитись з постановкою задачі дипломного проектування	15.05.2023- 20.05.2023	
2.	Вивчити спеціальну літературу і технічну документацію	21.05.2023- 27.05.2023	
3.	Написати розділ 1.	28.05.2023- 04.06.2023	
4.	Написати розділ 2.	05.06.2023- 10.06.2023	
5.	Виконати розробку чат-боту. Написати розділ 3.	10.06.2023- 12.06.2023	
6.	Оформити пояснювальну записку	12.06.2023- 15.06.2023	
7.	Підготувати графічний демонстраційний матеріал	16.06.2023- 18.06.2023	

7. Дата видачі завдання \_\_\_\_\_ 15.05.2023р.

Керівник кваліфікаційної роботи \_\_\_\_\_ Аліна САВЧЕНКО  
(підпис керівника)

Завдання прийняв до виконання \_\_\_\_\_ Рустам НАЗІМОВ  
(підпис випускника)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Застосування технології чат-ботів для онлайн-магазину з продажу геймерських товарів мовою Python» містить : 50 сторінок, 27 рисунків, 14 літературних джерел.

**Об'єкт дослідження** – процес продажу геймерських товарів.

**Предмет дослідження** – автоматизація продажу геймерських товарів онлайн-магазином за допомогою чат-боту.

**Мета кваліфікаційної роботи** – Створення і розробка чат-бота, що забезпечує автоматизований та персоналізований підхід до обслуговування клієнтів в онлайн-магазині геймерських товарів.

**Методи дослідження** – логічний, синтезу, аналізу, порівняльний, обробка літературних джерел та моделювання.

Результати дипломного проектування рекомендується використовувати в потребах покупки геймерських товарів.

Для розробки чат-боту знайдено та використано найефективніше програмне забезпечення, одна з найпопулярніших бібліотек для створення ботів в Телеграм, готові матеріали та сучасні технології.

ЧАТ-БОТ, PYTHON, API TELEGRAM, БІБЛІОТЕКА TELEBOT, PYCHARM, JSON, REQUESTS.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ .....	6
ВСТУП .....	7
РОЗДІЛ 1. ОБГРУНТУВАННЯ РОЗРОБЛЕННЯ ЧАТ-БОТА .....	8
1.1. Аналіз предметної області існуючих товарів для геймінгу .....	8
1.2. Методи підходу до розробки чат-ботів на python .....	13
1.3. Інструменти розробки .....	14
ВИСНОВКИ ДО РОЗДІЛУ 1 .....	16
РОЗДІЛ 2. ПРОЕКТУВАННЯ ЧАТ-БОТА .....	17
2.1. Мови розмітки та програмування .....	17
2.2. Аналіз мов програмування для розробки телеграм-ботів .....	18
2.3. Аналіз розширень та бібліотек Python для розробки чат-ботів .....	20
2.4. Огляд бібліотек та фреймворків для розробки чат-ботів на Python .....	24
ВИСНОВКИ ДО РОЗДІЛУ 2 .....	26
РОЗДІЛ 3. РОЗРОБЛЕННЯ ЧАТ-БОТА .....	27
3.1. Проектування програмного забезпечення для телеграм-боту .....	27
3.2. Написання коду .....	30
3.2.1. Встановлення Pycharm .....	30
3.2.2. Інсталяція бібліотеки .....	32
3.2.3. Реалізація боту .....	33
3.3. Демонстрація готового продукту .....	44
ВИСНОВКИ ДО РОЗДІЛУ 3 .....	48
ВИСНОВКИ .....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	50

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

QA	–	Quality Assurance.
DOS	–	Disk Operating System.
DDOS	–	Distributed Denial-Of-Service attack.
JSON	–	JavaScript Object Notation.
HTML	–	HTML.
API	–	Application Programming Interface.
ООП	–	Об'єктно-орієнтоване програмування.
ORM	–	Object-Relational Mapping.
CSS	–	Cascading Style Sheets.
IDE	–	Integrated Development Environment.

## ВСТУП

Сучасні бізнеси не можуть ігнорувати важливість інтернет-магазинів, оскільки кожен день все більше людей вибирають онлайн-шопінг. Таким чином, компанії, що діють у цій сфері, постійно шукають нові способи поліпшення своїх послуг. Одним з ефективних рішень є використання телеграм-боту для продажу геймерських товарів, який може бути створений на мові програмування Python.

У цьому проекті важливу роль відіграє QA (англ. Quality Assurance) - забезпечення якості продукту. Комплекс процесів QA включає в себе весь цикл розробки - від опису проекту до тестування, релізу і пост-релізного обслуговування. Фахівці QA створюють і реалізують різні тактики для підвищення якості на всіх стадіях виробництва.

Об'єктом дослідження є процес продажу геймерських товарів. Основна мета даного проекту - створити телеграм-бота для онлайн магазину геймерських товарів, який забезпечить швидкий і зручний доступ клієнтів до продуктів, а також підвищить якість обслуговування. Для досягнення цієї мети потрібно виконати кілька завдань. Один з важливих етапів розробки телеграм-боту - автоматизація тестування.

Методи дослідження включають аналіз та синтез, щоб знайти найбільш оптимальні рішення для автоматизації тестування телеграм-боту. Завдання полягає в створенні зручного та ефективного засобу для виявлення помилок в телеграм-боті онлайн магазину геймерських товарів, таких як помилки доступу, помилки UI та функціональні помилки.

## РОЗДІЛ 1

### ОБГРУНТУВАННЯ РОЗРОБЛЕННЯ ЧАТ-БОТА

#### 1.1. Аналіз предметної області існуючих товарів для геймінгу

Телеграм-бот - це програма, яка автоматизує взаємодію користувачів з платформою Телеграм. Такі боти можуть мати різноманітні функції, наприклад, вони можуть відповідати на запити користувачів, відправляти повідомлення та навіть збирати і обробляти дані.

Python - це одна з найбільш популярних мов програмування для розробки Телеграм-ботів. Вона має простий синтаксис, багату стандартну бібліотеку та багато сторонніх бібліотек, які дозволяють розширювати можливості ботів.

1. Необхідно розробити архітектуру телеграм-бота, який буде взаємодіяти з клієнтами і забезпечувати їм доступ до продуктів. Для цього можна використати бібліотеку `python-telegram-bot`, яка містить в собі набір класів і методів для розробки телеграм-ботів.

2. Потрібно створити набір тестів для перевірки функціональності телеграм-бота. Ці тести можуть бути написані з використанням бібліотек `Pytest` та `Selenium`, які дозволяють автоматизувати процес тестування.

3. Необхідно забезпечити захист від зламу телеграм-бота та захист персональної інформації клієнтів. Для цього можна використати бібліотеку `python-telegram-bot`, яка містить в собі вбудований захист від DDOS-атак та інші функції безпеки.

Загалом, розробка телеграм-боту для онлайн магазину геймерських товарів та його автоматизоване тестування дозволять забезпечити клієнтам швидкий та зручний доступ до продуктів, а також підвищити якість обслуговування та забезпечити безпеку клієнтської інформації.

Кафедра КІТ				НАУ 23 21 46 000 ПЗ			
	ПІБ			РОЗДІЛ 1. ОБГРУНТУВАННЯ РОЗРОБЛЕННЯ ЧАТ-БОТА	Літ.	Аркуш	Аркушів
Розроб.	Назімов Р.Є.					8	9
Керівник	Савченко А.С.				ТП-416Б – 122		
Н. Контр.	Толстікова О.В.						



Крім того, автоматизація процесу тестування дозволить швидко виявляти та виправляти помилки у функціональності телеграм-бота, що дозволить уникнути неприємних ситуацій з клієнтами та зберегти репутацію компанії. Застосування Python та його бібліотек дозволяє зробити розробку та тестування більш ефективним та швидким процесом, що також може знизити витрати на розробку та підтримку проекту.

На сьогоднішній день геймінг є однією з найпопулярніших розважальних галузей, тому існує велика кількість товарів, спеціально створених для геймерів.

Розглянемо деякі з них:

- Клавіатури - Клавіатури для геймерів мають спеціальну конструкцію та механізм клавіш, що забезпечують швидку та точну реакцію. Деякі з них мають підсвічування, що дозволяє грати у темному приміщенні, а інші мають програмування клавіш для зручності гри.

- Миші - Миші для геймерів мають спеціальні сенсори, що забезпечують високу точність та швидкість відгуку. Деякі з них мають програмовані клавіші, що дозволяють виконувати швидкі команди.

- Навушники - Навушники для геймерів мають високоякісний звук та можуть мати вбудований мікрофон для комунікації з іншими гравцями в онлайн-іграх.

- Геймпади - Геймпади є ідеальним вибором для тих, хто більше любить грати на консолях. Вони забезпечують більш зручний та реалістичний контроль за грою.

- Відеокарти - Відеокарти забезпечують більш високу продуктивність та якість графіки в грі, що робить їх необхідними для геймерів, які хочуть отримати найкращий досвід гри.

- Монітори - Монітори з високою частотою оновлення зображення, великою діагоналлю та роздільною здатністю, забезпечують більш комфортний та реалістичний досвід гри.

- Комп'ютерні крісла - Крісла для геймерів мають спеціальну конструкцію, що забезпечує комфортну посадку та можливість налаштування кута нахилу спинки та підголів'я. Деякі з них також мають вбудовані динаміки, що підсилюють звуковий ефект гри.

Усі ці товари можуть бути продаваними за допомогою телеграм-ботів. Наприклад, бот може пропонувати каталог товарів, спілкуватися з клієнтами та обробляти замовлення. Бот може мати також інші корисні функції, наприклад, повідомляти про знижки, розсилати новини та інформацію про нові продукти.

Товари для геймінгу можуть відрізнятися від звичайних товарів за своїми специфічними властивостями та призначенням.



Рис. 1.1. Приклад геймерської миші для багатofункціональних “МОВА” ігор

Наприклад, ігрова клавіатура або миша має спеціальні кнопки або додаткові функції, які дозволяють гравцеві краще керувати грою. Комп'ютерні компоненти, такі як відеокарти або процесори, можуть бути підібрані спеціально для оптимального функціонування ігор на комп'ютері.

Ігрові консолі, такі як PlayStation або Xbox, мають свої унікальні функції та ігри, які не можуть бути відтворені на звичайних консолях або комп'ютерах.

Крім того, важливо враховувати, що геймери можуть бути більш вимогливими щодо якості та функціональності товарів, особливо якщо вони використовуються для професійних цілей або конкурсів. Тому, при розробці та просуванні товарів для геймінгу, важливо враховувати ці специфічні вимоги та потреби клієнтів.

Феномен набуття популярності продажу товарів через платформи, такі як Instagram та Telegram, в останні роки можна пояснити кількома факторами. Ці платформи стали надзвичайно популярними через свою простоту використання та широку аудиторію, яка їх використовує щодня.

Instagram є одним з найпопулярніших соціальних медіа на сьогоднішній день, з мільярдами активних користувачів по всьому світу. Він спеціалізується на візуальному змісті, що робить його ідеальним місцем для продажу товарів. Багато бізнесів створюють свої профілі на Instagram, де вони показують свої товари, розміщують фотографії та відео, діляться історіями та спілкуються зі своїми клієнтами. Крім того, Instagram пропонує функцію "Shop Now" (Купити зараз), яка дозволяє користувачам купувати товари безпосередньо з додатку, що значно полегшує процес покупки.

Telegram, з іншого боку, є месенджером, який набув великої популярності завдяки своїм функціям безпеки та конфіденційності. Багато бізнесів створюють свої канали або групи в Telegram, де вони публікують інформацію про свої товари, акції, новини та інше. Користувачі можуть підписатися на ці канали та отримувати оновлення безпосередньо у своїх чатах. Крім того, Telegram має функцію "платежі", яка дозволяє бізнесам приймати платежі безпосередньо через платформу.

Основна перевага використання Instagram та Telegram для продажу товарів полягає в їхній широкій аудиторії та можливості прямого взаємодії з клієнтами.

Бізнеси можуть досягти великої кількості потенційних покупців, показуючи їм свої товари через привабливі візуальні зображення або повідомлення. Крім того, ці платформи дозволяють створювати активні спільноти навколо бренду, залучати клієнтів до дискусій, проводити опитування та отримувати цінні відгуки.

Таким чином, Instagram та Telegram стали популярними платформами для продажу товарів через свою зручність використання, широку аудиторію та можливість прямого взаємодії з клієнтами. Вони дозволяють бізнесам ефективно просувати свої товари та залучати нових клієнтів, відкриваючи нові можливості для електронної комерції.

Телеграм-боти відіграють важливу роль у продажу товарів через платформу Telegram, особливо в онлайн-магазинах. Вони є автоматизованими асистентами, які можуть виконувати різні функції та завдання, пов'язані з продажем та обслуговуванням клієнтів.

Ось кілька способів, які Телеграм-боти можуть використовуватися в продажу товарів:

- Каталог товарів: Бот може мати каталог товарів з фотографіями, описами та цінами. Користувачі можуть переглядати товари та отримувати детальну інформацію про них безпосередньо у чаті з ботом.

- Оформлення замовлення: Бот може надати можливість користувачам оформляти замовлення прямо у Telegram. Користувачі можуть вибирати товари, обирати варіанти доставки та сплачувати за допомогою інтегрованих платіжних систем.

- Підтримка клієнтів: Бот може надавати підтримку клієнтам, відповідаючи на їх запитання щодо товарів, доставки, оплати та інших деталей. Він може автоматично відповідати на загальні запитання або передавати запити живим операторам у разі потреби.

- Оповіщення та акції: Бот може надсилати користувачам сповіщення про нові товари, розпродажі, знижки та інші акції. Це дозволяє залучати увагу клієнтів та стимулювати їх до покупок.

- Аналітика та статистика: Деякі Телеграм-боти можуть надавати аналітичну інформацію про продажі, відстежувати популярні товари, аналізувати поведінку користувачів та надавати статистику для прийняття рішень щодо стратегії продажу.

Телеграм-боти дозволяють автоматизувати процес продажу товарів, спрощуючи взаємодію з клієнтами та підвищуючи ефективність бізнесу. Вони забезпечують зручну комунікацію та швидку обробку замовлень, що дозволяє бізнесам зосередитися на якісному обслуговуванні клієнтів та збільшенні продажів.

## **1.2. Методи підходу до розробки чат-ботів на python**

Існує декілька підходів до розробки Телеграм-ботів на Python, які можуть бути використані залежно від потреб розробника. Нижче наведено деякі з них:

Використання бібліотеки `pyTelegramBotAPI`: Цей підхід є дуже популярним серед розробників Python для створення Телеграм-ботів. Бібліотека `pyTelegramBotAPI` надає простий та зрозумілий інтерфейс для взаємодії з Телеграм API, що дозволяє швидко створювати ботів та додавати до них функціональність.

Використання фреймворку Flask: Flask є одним з найпопулярніших фреймворків Python для веб-розробки, але він також може бути використаний для розробки Телеграм-ботів. Flask дозволяє розробникам створювати власний сервер, який можна інтегрувати з Телеграм API.

Використання фреймворку Django: Django є ще одним популярним фреймворком Python для веб-розробки, який може бути використаний для розробки Телеграм-ботів. Django надає багато функціональності, що дозволяє розробникам швидко створювати веб-додатки, в тому числі і Телеграм-боти.

Використання платформи Heroku: Heroku є хмарною платформою, яка дозволяє розробникам швидко та легко розгорнути веб-додатки, в тому числі і Телеграм-боти. Heroku може бути інтегрована з різними фреймворками та бібліотеками Python, що дозволяє розробникам використовувати їх для створення своїх ботів.

Ці підходи можуть бути використані окремо або в комбінації один з одним для розробки Телеграм-ботів на Python. Вибір підходу залежить від потреб розробника та особливостей проекту. Якщо потрібно швидко розробити простий Телеграм-бот з мінімальною функціональністю, то можна скористатися бібліотекою `pyTelegramBotAPI`. Якщо проект складніший та вимагає використання власного сервера, то можна розглянути використання фреймворків Flask або Django. Якщо потрібно швидко розгорнути веб-додаток без потреби власного сервера, то можна використовувати платформу Heroku.

Незалежно від підходу, для розробки Телеграм-бота на Python потрібно мати розуміння роботи з API Телеграм та Python-бібліотеками, зокрема `requests`, `json`, а також розуміння роботи з базами даних, якщо вони використовуються в проекті. Також потрібно мати досвід у програмуванні на Python та розуміння основних концепцій ООП.

### **1.3. Інструменти розробки**

Для подальшого дослідження в дипломній роботі розглянуто метод `backend` та `frontend` розробки телеграм-бота на мові Python. Для цього необхідно створити програмний код бота з використанням бібліотеки `pyTelegramBotAPI` або інших відповідних інструментів. Для розробки та тестування програмного коду бота можна використовувати зручні редактори коду, такі як PyCharm або Visual Studio Code.

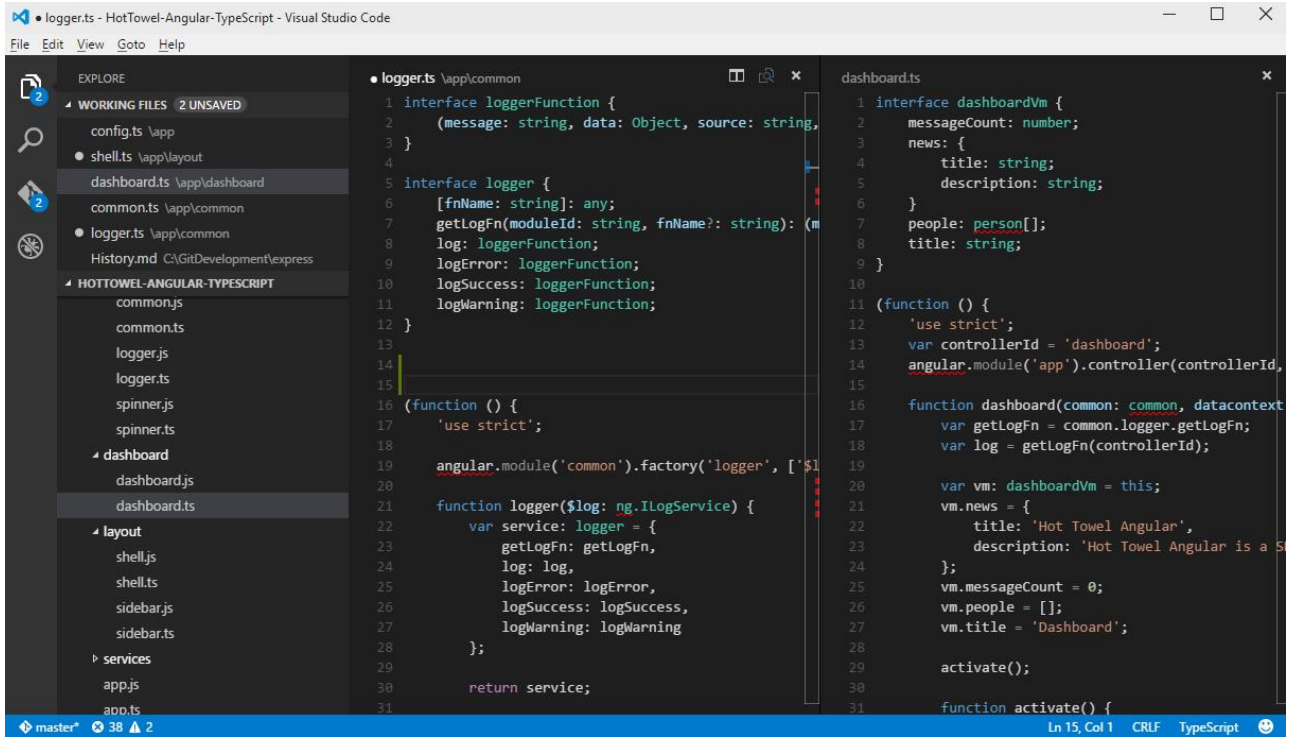


Рис. 1.2. Програма Visual Studio Code

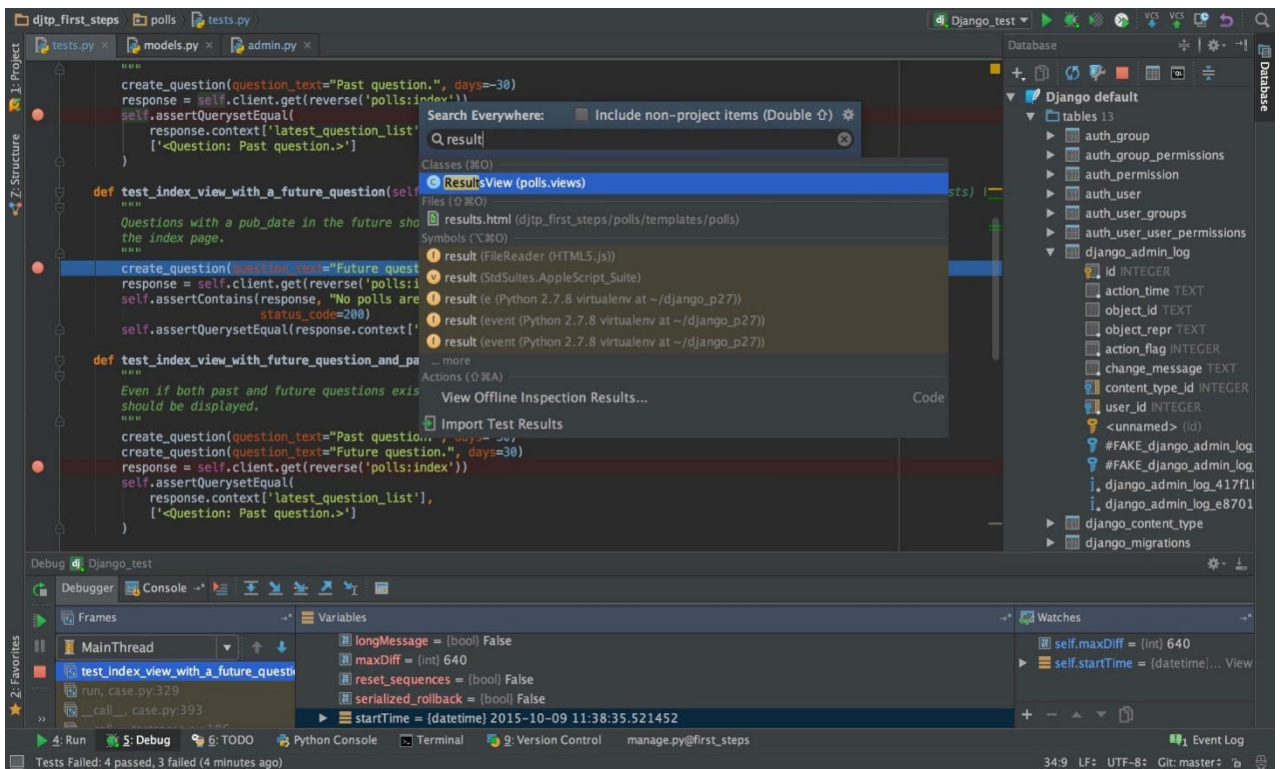


Рис 1.3 Програма ручарм

## ВИСНОВКИ ДО РОЗДІЛУ 1

1. Аналіз геймерських товарів підтверджує потребу в ефективному способі продажу та обслуговування клієнтів в цій галузі.

2. Продаж товарів через Telegram є актуальним способом, що знаходиться на перетині простоти розробки у порівнянні з вебсайтом з однієї сторони та охоплення великої аудиторії через низький поріг входу та зручний стандартизований інтерфейс.

3. Телеграм-боти відіграють важливу роль у продажу товарів через платформу Telegram, особливо в онлайн-магазинах. Вони є автоматизованими асистентами, які можуть виконувати різні функції та завдання, пов'язані з продажем та обслуговуванням клієнтів.

4. Методи розробки чат-ботів на Python надають гнучкість та можливості для створення потужних та функціональних чат-ботів.

5. Вибір відповідних інструментів розробки допомагає забезпечити ефективну та продуктивну розробку чат-бота на основі Python.

Усі ці фактори підтверджують, що розробка Телеграм-боту для онлайн-магазину продажу геймерської продукції є важливим та перспективним кроком. Вона дозволяє забезпечити зручну та ефективну платформу для продажу, підтримки клієнтів та розбудови взаємовідносин з геймерською аудиторією. За умови правильної розробки та вдосконалення, Телеграм-бот може стати потужним інструментом для збільшення продажів та досягнення успіху в галузі продажу геймерських товарів.



## РОЗДІЛ 2

### ПРОЕКТУВАННЯ ЧАТ-БОТА

#### 2.1. Мови розмітки та програмування

Python - це високорівнева мова програмування, яка зазвичай використовується для розробки різноманітних програм. Для створення Телеграм-ботів на Python часто використовуються такі інструменти, як бібліотека pyTelegramBotAPI та фреймворки Flask та Django.

Для додавання взаємодії з користувачем до бота можуть використовуватися різні мови розмітки, наприклад, Markdown або HTML. Якщо потрібно відображати складніші елементи, такі як таблиці чи графіки, можна використовувати спеціалізовані бібліотеки, наприклад, Plotly для побудови графіків.

Окрім цього, Python має велику кількість різноманітних бібліотек та інструментів, що значно спрощують розробку Телеграм-ботів. Наприклад, для роботи з картинками можна використовувати бібліотеку Pillow, для відправки повідомлень з зображеннями або відео - бібліотеку pyTelegramBotAPI.

Python також має багатий екосистему бібліотек та фреймворків, які дозволяють розробникам швидко створювати різноманітні додатки, включаючи і Телеграм-ботів.

Щодо мов розмітки, HTML та CSS є стандартними мовами для розробки веб-додатків та сайтів, але вони також можуть бути використані для створення користувацького інтерфейсу Телеграм-бота.

HTML використовується для створення структури сторінки, тоді як CSS відповідає за її оформлення та стилізацію.

Кафедра КІТ				НАУ 23 21 46 000 ПЗ			
	ПІБ			РОЗДІЛ 2. ПРОЕКТУВАННЯ ЧАТ-БОТА	Літ.	Аркуш	Аркушів
Розроб.	Назімов Р.Є.					17	10
Керівник	Савченко А.С.				ТП-416Б – 122		
Н. Контр.	Толстікова О.В.						

Використання HTML та CSS дозволяє розробникам створювати красивий та зрозумілий користувацький інтерфейс для Телеграм-бота, що забезпечує зручну та приємну взаємодію з користувачами.

Однак, для розробки бізнес-логіки Телеграм-бота та взаємодії з API Телеграму необхідно використовувати Python. Python забезпечує можливість звернення до API Телеграму, обробку та аналіз даних, створення логіки бота та інші можливості, що дозволяють розробникам створювати потужні та функціональні Телеграм-боти.

Отже, для розробки Телеграм-ботів на Python можна використовувати Python для бізнес-логіки та взаємодії з API, а також HTML та CSS для створення користувацького інтерфейсу.

## **2.2. Аналіз мов програмування для розробки телеграм-ботів**

При розробці телеграм-ботів можна використовувати різні мови програмування. Вибір мови програмування залежить від ряду факторів, таких як зручність розробки, доступність ресурсів для навчання, швидкість виконання та інші. Далі ми розглянемо деякі з найпопулярніших мов програмування для розробки телеграм-ботів.

- Python є дуже зручною мовою для початківців. Мова має простий та зрозумілий синтаксис, який допомагає знизити кількість помилок при написанні коду. Більш того, Python має велику кількість бібліотек, які значно спрощують розробку. Деякі з найпопулярніших бібліотек для розробки телеграм-ботів включають `python-telegram-bot` та `Telepot`. Крім того, Python має гарну швидкість виконання та забезпечує доступ до багатої екосистеми та підтримки спільноти.

- JavaScript є популярною мовою для розробки веб-додатків, що дає можливість створювати телеграм-ботів, які взаємодіють з веб-сторінками. JavaScript має швидку швидкість виконання та доступ до багатої екосистеми бібліотек. Для розробки телеграм-ботів на JavaScript можна використовувати бібліотеку `Telegraf`.

- Ruby є мовою програмування, яка також має зрозумілий синтаксис та широку екосистему бібліотек. Для розробки телеграм-ботів на Ruby можна використовувати бібліотеку telegram-bot-ruby.

- PHP є мовою програмування, яка також підходить для розробки телеграм-ботів. PHP має швидку швидкість виконання та доступ до великої кількості бібліотек для розробки. Для розробки телеграм-ботів на PHP можна використовувати бібліотеку Telegram Bot API.

- Go є мовою програмування, розробленою компанією Google. Ця мова має швидку швидкість виконання та ефективність, що особливо важливо для великих проектів. Для розробки телеграм-ботів на Go можна використовувати бібліотеку telegram-bot-api.

- Java є мовою програмування, яка підходить для розробки великих проектів. Java має широку екосистему бібліотек та інструментів для розробки. Для розробки телеграм-ботів на Java можна використовувати бібліотеку TelegramBots.

- C# є мовою програмування, розробленою компанією Microsoft. Ця мова має зрозумілий синтаксис та велику кількість бібліотек. Для розробки телеграм-ботів на C# можна використовувати бібліотеку Telegram.Bot.

У кожній з цих мов програмування є свої переваги та недоліки, тому вибір мови програмування залежить від ваших потреб.

Python є однією з найпопулярніших мов програмування для розробки Телеграм-ботів і має кілька переваг, які роблять його привабливим вибором для цієї задачі:

- Простота та зручність: Python має простий і зрозумілий синтаксис, що дозволяє розробникам швидко створювати функції та логіку бота. Він є лаконічним і легкочитається, що полегшує розробку та підтримку коду.

- Велика спільнота та багата екосистема: Python має широку спільноту розробників, що активно підтримує мову і надає безліч корисних бібліотек та інструментів.

У випадку Телеграм-ботів, існує популярна бібліотека `python-telegram-bot`, яка забезпечує простий та зручний інтерфейс для взаємодії з API Телеграму.

- Розширюваність та гнучкість: Python є мовою з великою кількістю модулів та фреймворків, що дозволяють легко розширювати функціональність бота. Наявність готових рішень та інструментів спрощує розробку складних функцій, таких як інтеграція з базами даних, веб-скрапінг, робота з зображеннями тощо.

- Підтримка асинхронного програмування: Python 3.7 і вище надає підтримку асинхронного програмування за допомогою асинхронного фреймворку `asyncio`. Це дозволяє створювати ботів, які можуть обробляти багато запитів одночасно, забезпечуючи більшу продуктивність та відповідальність.

- Широкий спектр інтеграцій: Python має багато готових бібліотек і модулів для роботи з різними сервісами та API. Це дає можливість легко інтегрувати бота з іншими сервісами, такими як бази даних, соціальні мережі, платіжні системи тощо.

Загалом, Python забезпечує зручність, простоту, гнучкість та розширюваність, що робить його відмінним вибором для розробки Телеграм-ботів. Його широка спільнота та багата екосистема також забезпечують підтримку та доступ до потужних інструментів для створення функціональних та ефективних ботів.

### **2.3. Аналіз розширень та бібліотек Python для розробки чат-ботів**

Існує величезна кількість бібліотек та розширень, які дозволяють легко розробляти та впроваджувати чат-ботів на основі Python. Розглянемо деякі з цих бібліотек та розширень:

- `Telegram Bot API`: Ця бібліотека дозволяє легко створювати та впроваджувати телеграм-ботів на основі Python.

Ця бібліотека надає можливість створювати різні типи повідомлень, зберігати користувацьку інформацію та дізнаватися про різні події, такі як нові повідомлення.

- ChatterBot: Ця бібліотека дозволяє створювати чат-ботів з використанням машинного навчання. Вона навчається на основі попередньо зібраних даних та може відповідати на різні запити. ChatterBot має вбудовані інструменти для побудови нейронних мереж та генерації випадкових відповідей.

- NLTK: Ця бібліотека дозволяє використовувати природну мову для розробки чат-ботів. NLTK забезпечує різноманітні функції, такі як синтаксичний аналіз, лематизація, знаходження іменованих сутностей та багато іншого.

- TensorFlow: Ця бібліотека дозволяє розробляти чат-ботів з використанням глибинного навчання. TensorFlow має велику кількість функцій та інструментів, що дозволяють легко створювати та навчати нейронні мережі для розпізнавання мови та інших завдань.

- PyTorch: Це інша бібліотека глибинного навчання, яка стала дуже популярною останнім часом. Вона дозволяє створювати та навчати нейронні мережі з високою швидкістю та продуктивністю. PyTorch має простий та зрозумілий інтерфейс, що дозволяє розробникам швидко та легко розуміти та розробляти складні моделі глибинного навчання. Крім того, вона має велику кількість розширень та інструментів, які дозволяють з легкістю візуалізувати та аналізувати роботу нейронних мереж.

- SpaCy: Це бібліотека для обробки природної мови, яка містить в собі різноманітні інструменти для токенизації, стемінгу, частиномовного аналізу та інших задач. SpaCy дозволяє розробникам легко та ефективно обробляти текстові дані та використовувати їх для розробки чат-ботів.

- Rasa: Це фреймворк для розробки чат-ботів, який забезпечує повний стек інструментів для побудови та впровадження чат-ботів.

Rasa дозволяє легко навчати та впроваджувати чат-ботів, використовуючи різні інтерфейси, такі як текстовий, голосовий та інший.

Порівняємо деякі з наведених бібліотек та розширень для розробки чат-ботів на основі Python:

- Telegram Bot API: Ця бібліотека надає базові функціональні можливості для створення та взаємодії з телеграм-ботами. Вона легка у використанні і дозволяє швидко підключатись до Telegram API та обробляти повідомлення від користувачів. Однак, вона не надає вбудованих інструментів для розробки складних логік бота або обробки природної мови.

- ChatterBot: Ця бібліотека зосереджена на використанні машинного навчання для створення чат-ботів. Вона має вбудовані алгоритми навчання, що дозволяють боту відповідати на запити, враховуючи контекст та попередні взаємодії. ChatterBot має гнучку архітектуру, але для досягнення кращих результатів потребує досить значної кількості даних для навчання.

- NLTK: Ця бібліотека спеціалізується на обробці природної мови і надає широкий набір інструментів для аналізу тексту. Вона дозволяє виконувати такі завдання, як токенізація, стемінг, лематизація та знаходження іменованих сутностей. NLTK підтримує велику кількість мов та має багато корисних модулів, але вимагає більшої ручної настройки та кодування для створення повноцінного чат-бота.

- TensorFlow та PyTorch: Ці бібліотеки глибинного навчання часто використовуються для створення складних чат-ботів, що здатні до розпізнавання мови та вирішення складних завдань. Вони мають потужні інструменти для роботи з нейронними мережами, включаючи генерацію відповідей, переклад та розпізнавання образів. Однак, вони вимагають більшої експертизи в глибинному навчанні та складнішої настройки, ніж інші бібліотеки.

- SpaCy: Ця бібліотека також спеціалізується на обробці природної мови і має широкий набір інструментів для аналізу тексту.

Вона відрізняється високою швидкістю та продуктивністю обробки тексту, що робить її популярним вибором для великих обсягів даних. SpaCy також має підтримку різних мов та деякі вбудовані моделі для розпізнавання елементів мови.

- Rasa: Цей фреймворк надає повний стек інструментів для розробки та впровадження чат-ботів. Rasa включає в себе інструменти для навчання, створення діалогових моделей та інтеграції з різними каналами зв'язку. Цей фреймворк забезпечує гнучкість і розширюваність у розробці чат-ботів, але може вимагати більшого обсягу роботи для настройки та навчання моделей.

Кожна з наведених бібліотек та розширень має свої переваги і недоліки, і вибір залежить від конкретних потреб та обмежень проекту. Для простих ботів, які виконують основні завдання, Telegram Bot API може бути швидким та зручним варіантом. Для більш складних чат-ботів, які потребують гнучкості та можливостей машинного навчання, ChatterBot або Rasa можуть бути кращими варіантами. TensorFlow та PyTorch підходять для проектів з високими вимогами до глибинного навчання, а NLTK та SpaCy дозволяють зосередитися на обробці природної мови. Важливо враховувати власні потреби та знання в області, щоб вибрати найбільш підходящу бібліотеку для свого проекту.

Python є неймовірно ефективна мова: ваші програми зроблять більше за меншу кількість рядків коду, ніж потребує багато інших мов. Синтаксис Python також допоможе вам написати «чистий» код. Ваш код буде легко читати, легко налагоджувати, і легко розширювати та надбудовувати порівняно з іншими мовами[1].

Загалом, розробка чат-ботів на основі Python стає все більш популярною, завдяки великій кількості інструментів та бібліотек, що дозволяють легко розробляти та впроваджувати чат-ботів на основі машинного навчання та глибинного навчання.

## 2.4. Огляд бібліотек та фреймворків для розробки чай-ботів на Python

Проаналізуємо найпопулярніші бібліотеки та фреймворки для розробки телеграм-ботів :

- pyTelegramBotAPI є однією з найпопулярніших бібліотек для розробки телеграм-ботів на мові Python. Вона надає простий та зрозумілий інтерфейс для взаємодії з API Телеграма. Бібліотека дозволяє легко отримувати та обробляти повідомлення, відповіді, фотографії, відео та інші типи даних. Вона також підтримує створення клавіатур для взаємодії з користувачем та надає можливість обробляти команди та події.

- aiogram є потужним фреймворком для розробки телеграм-ботів на Python з підтримкою асинхронного програмування. Він надає розширені можливості для розробки ботів, включаючи обробку повідомлень, клавіатур, команд, розсилку повідомлень, роботу з базами даних та інші функції. aiogram пропонує зручний та ефективний спосіб створення ботів з використанням асинхронних корутин.

- python-telegram-bot є іншою популярною бібліотекою для розробки телеграм-ботів на Python. Вона надає простий інтерфейс для взаємодії з API Телеграма та включає в себе багато корисних функцій. Бібліотека дозволяє обробляти повідомлення, створювати клавіатури, відправляти зображення, відео, аудіо та інші медіафайли. Вона також підтримує обробку команд та реакцій на повідомлення.

- Telebot є простою і легкою у використанні бібліотекою для розробки телеграм-ботів на Python. Вона надає базові функції для роботи з API Телеграма, такі як отримання та обробка повідомлень, відправка тексту та медіафайлів, створення клавіатур тощо. Telebot також підтримує обробку команд та дозволяє налаштовувати взаємодію з ботом.

- Telethon є потужною бібліотекою для розробки телеграм-клієнтів на Python.



Вона надає повний доступ до всіх функцій Телеграма, включаючи взаємодію з повідомленнями, групами, каналами, контактами та іншими аспектами месенджера. Telethon дозволяє розробляти різноманітні додатки, включаючи телеграм-ботів з багатофункціональними можливостями.

Кожна з цих бібліотек та фреймворків має свої переваги та особливості. Вибір залежить від потреб, рівня досвіду та вимог до функціоналу телеграм-бота. Ретельне вивчення документації та спроба роботи з різними бібліотеками можуть допомогти знайти найкращий інструмент для розробки чат-бота.

Вибір бібліотеки telebot для написання Телеграм-боту на мові Python може бути обґрунтований кількома причинами:

- Простота використання: Бібліотека telebot має простий та зрозумілий інтерфейс, що дозволяє швидко розпочати роботу зі створенням Телеграм-боту.

- Велика спільнота та підтримка: Бібліотека telebot має широку спільноту розробників, яка активно підтримує бібліотеку та надає допомогу у вирішенні питань та проблем. Це означає, що ви можете швидко знайти відповіді на свої питання, документацію та приклади коду, що спрощує розробку Телеграм-боту.

- Функціональні можливості: Бібліотека telebot надає широкий набір функціональних можливостей для створення потужного Телеграм-боту. Вона підтримує обробку текстових повідомлень, створення клавіатур, роботу з мультимедійним контентом, розсилку повідомлень, роботу з базами даних та багато іншого.

- Актуальні оновлення: Бібліотека telebot активно оновлюється та підтримується, що забезпечує безпеку, стабільність та доступ до нових функціональних можливостей.

- Документація та приклади коду: Бібліотека telebot має добре документовану API та надає приклади коду для різних сценаріїв використання.

Отже, вибір бібліотеки telebot для написання Телеграм-боту на мові Python обґрунтований його простотою використання, підтримкою спільноти, функціональністю, актуальністю оновлень та наявністю добре документованої API та прикладів коду

## ВИСНОВКИ ДО РОЗДІЛУ 2

Порівнявши різні мови програмування, ми прийшли до висновку, що Python найкраще відповідає нашим потребам з великою кількістю переваг. Python не лише має простий синтаксис, але й володіє широким спектром функціональності та отримує широку підтримку від активної спільноти розробників. Це сприяє ефективній розробці та підтримці нашого чат-бота, забезпечуючи потужні та гнучкі можливості.

Python є потужною та гнучкою мовою програмування з багатим набором бібліотек і фреймворків, що полегшують розробку та розширення функціоналу бота. Завдяки цьому набору інструментів, ми можемо використовувати широкий спектр функціональних можливостей для побудови потрібного функціоналу нашого чат-бота.

Під час аналізу розширень та бібліотек для Python, ми звернули увагу на бібліотеку telebot, яку обрали як основний інструмент для розробки нашого телеграм-бота. telebot вирізняється багатьма корисними функціями та простим інтерфейсом, що дозволяє нам легко взаємодіяти з API Телеграма та розробляти різноманітний функціонал, необхідний для нашого проекту.

У процесі проектування програмного забезпечення для нашого телеграм-бота, ми докладно врахували потреби користувачів і специфіку нашого онлайн-магазину, що спеціалізується на продажу геймерських товарів. Ми розробили функціонал, який дозволяє нашим користувачам зручно та швидко переглядати, вибирати та замовляти товари, що відповідає їхнім потребам та сприяє позитивному враженню від користування чат-ботом.

В цілому, детальний огляд різних аспектів проектування та розробки чат-бота на основі Python дозволив нам зробити виважені рішення та обрати оптимальні інструменти для успішної реалізації чат-бота в онлайн-магазині, спеціалізованому на продажу геймерських товарів.

## РОЗДІЛ 3 РОЗРОБЛЕННЯ ЧАТ-БОТА

### 3.1. Проектування програмного забезпечення для телеграм-боту

Проектування програмного забезпечення для телеграм-боту на Python - це процес створення програмного забезпечення, яке взаємодіє з користувачами через чат-бот в Telegram.

Для початку, необхідно визначити функціональні вимоги до програмного забезпечення, які повинен виконувати чат-бот. У моєму випадку з онлайн-магазином з продажу геймерських товарів, це можуть бути такі функції, як перегляд асортименту товарів, замовлення товарів, оплата замовлення, підтримка клієнтів тощо.

Далі, необхідно вибрати фреймворк для розробки телеграм-боту. Наприклад, одним з популярних фреймворків на Python є `pyTelegramBotAPI`. Цей фреймворк дозволяє легко підключатись до Telegram API, отримувати та оброблювати повідомлення від користувачів, надсилати повідомлення та медіафайли, працювати з клавіатурами тощо.

Далі, потрібно розробити логіку роботи телеграм-боту. Це може включати визначення акторів, що взаємодіють з ботом, визначення функцій боту, створення діаграми варіантів використання, налаштування обробки замовлень та оплати, налаштування системи підтримки користувачів тощо.

Також, потрібно додати до чат-боту функціональність, яка забезпечить зручність користування і полегшить комунікацію з клієнтами. Наприклад, можна реалізувати систему налагодження налаштувань боту.

Далі, необхідно забезпечити безпеку та захист користувачів та їх даних.

Кафедра КІТ				НАУ 23 21 46 000 ПЗ						
	<i>ПІБ</i>			РОЗДІЛ 3. РОЗРОБЛЕННЯ ЧАТ-БОТА			<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>	
<i>Розроб.</i>	Назімов Р.Є.								27	22
<i>Керівник</i>	Савченко А.С.						ТП-416Б – 122			
<i>Н. Контр.</i>	Толстікова О.В.									

Це можна зробити за допомогою шифрування даних, валідації введення користувачів, забезпечення захищеного з'єднання до API та збереження даних в безпечному місці.

Крім того, важливим етапом є тестування програмного забезпечення для телеграм-боту. Тестування повинно бути проведено на різних етапах розробки для перевірки правильності роботи та виявлення помилок та недоліків.

У результаті розробки програмного забезпечення для телеграм-боту на Python для онлайн-магазину з продажу геймерських товарів, користувачі матимуть зручний і простий інструмент для замовлення та оплати товарів, а також для звернення до підтримки клієнтів. Розробка програмного забезпечення на Python для телеграм-боту забезпечить ефективну взаємодію з користувачами та високу якість обслуговування, що дозволить збільшити продажі та покращити репутацію магазину.

Проектування програмного забезпечення для телеграм-боту на Python відіграє важливу роль у створенні ефективного та зручного сервісу для онлайн-магазину з продажу геймерської продукції.

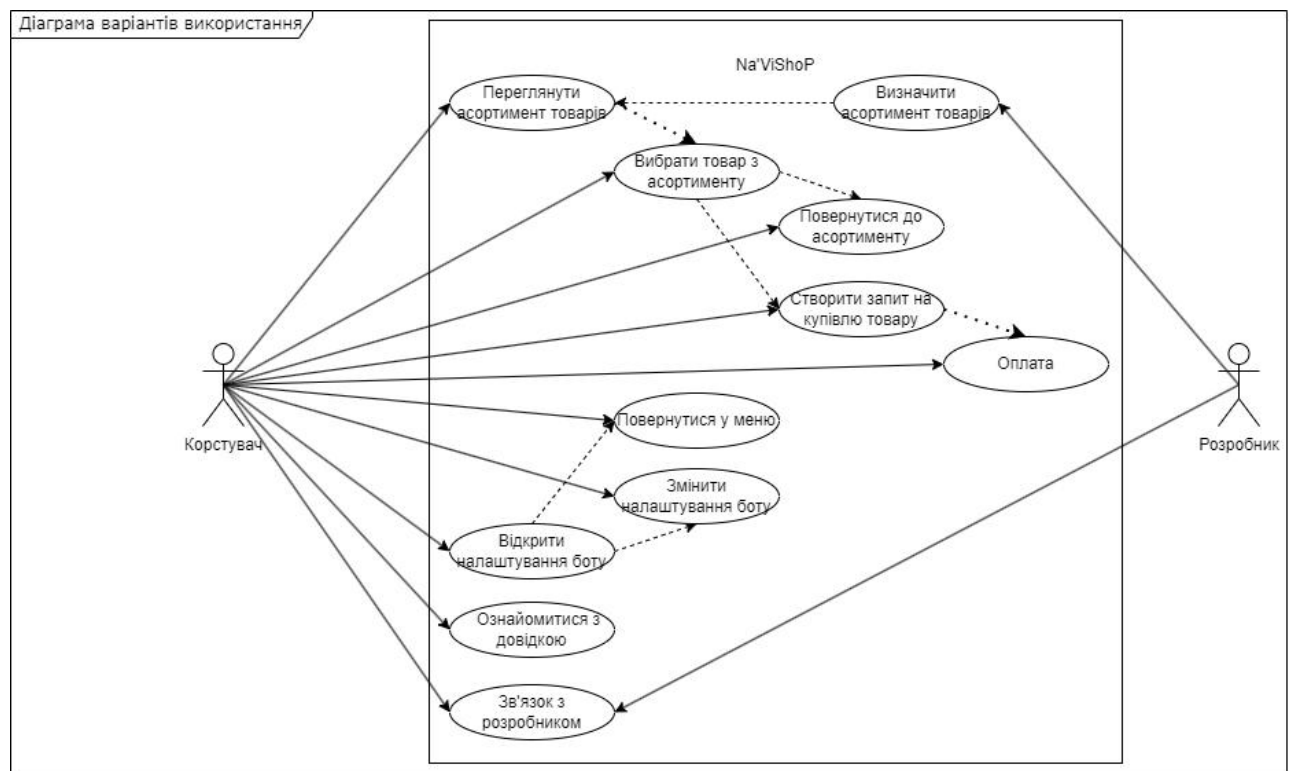


Рис. 3.1. Діаграма варіантів використання боту.

Процес розробки починається з визначення функціональних вимог до боту, таких як замовлення товарів, оплата, зміна налаштувань боту та підтримка клієнтів.

Вибір популярного фреймворку, такого як `pyTelegramBotAPI`, спрощує підключення до Telegram API та обробку повідомлень, надсилання медіафайлів та роботу з клавіатурами. Далі, розробник повинен створити логіку роботи боту, включаючи налаштування бази даних для зберігання інформації про товари та замовлення, обробку замовлень та оплату, а також систему повідомлень про статус замовлення для полегшення комунікації з клієнтами.

Забезпечення безпеки та захисту користувачів та їх даних є надзвичайно важливим аспектом розробки. Шифрування даних, валідація введення користувачів, забезпечення захищеного з'єднання до API та збереження даних в безпечному місці є ключовими елементами для забезпечення конфіденційності та безпеки.

Тестування програмного забезпечення на різних етапах розробки допомагає виявити помилки та недоліки, забезпечуючи високу якість та надійність роботи боту. Це дозволяє перевірити правильність функціонування та забезпечити належну реакцію на вхідні дані.

Результатом розробки програмного забезпечення для телеграм-боту на Python для онлайн-магазину з продажу геймерської продукції є зручний та простий інструмент для замовлення товарів, оплати та звернення до підтримки клієнтів. Розробка на основі Python дозволяє забезпечити ефективну взаємодію з користувачами та підвищити якість обслуговування, що сприяє зростанню продажів та підвищенню репутації магазину.

## 3.2. Написання коду

Розробка чат-боту для онлайн-магазину з продажу геймерських товарів мовою Python включає в себе деякі дії, які я описав нижче:

### 3.2.1. Встановлення Pycharm

Спочатку я завантажив виконавчий файл інсталяції PyCharm з офіційного веб-сайту JetBrains, розробника цього інтегрованого середовища розробки (IDE).

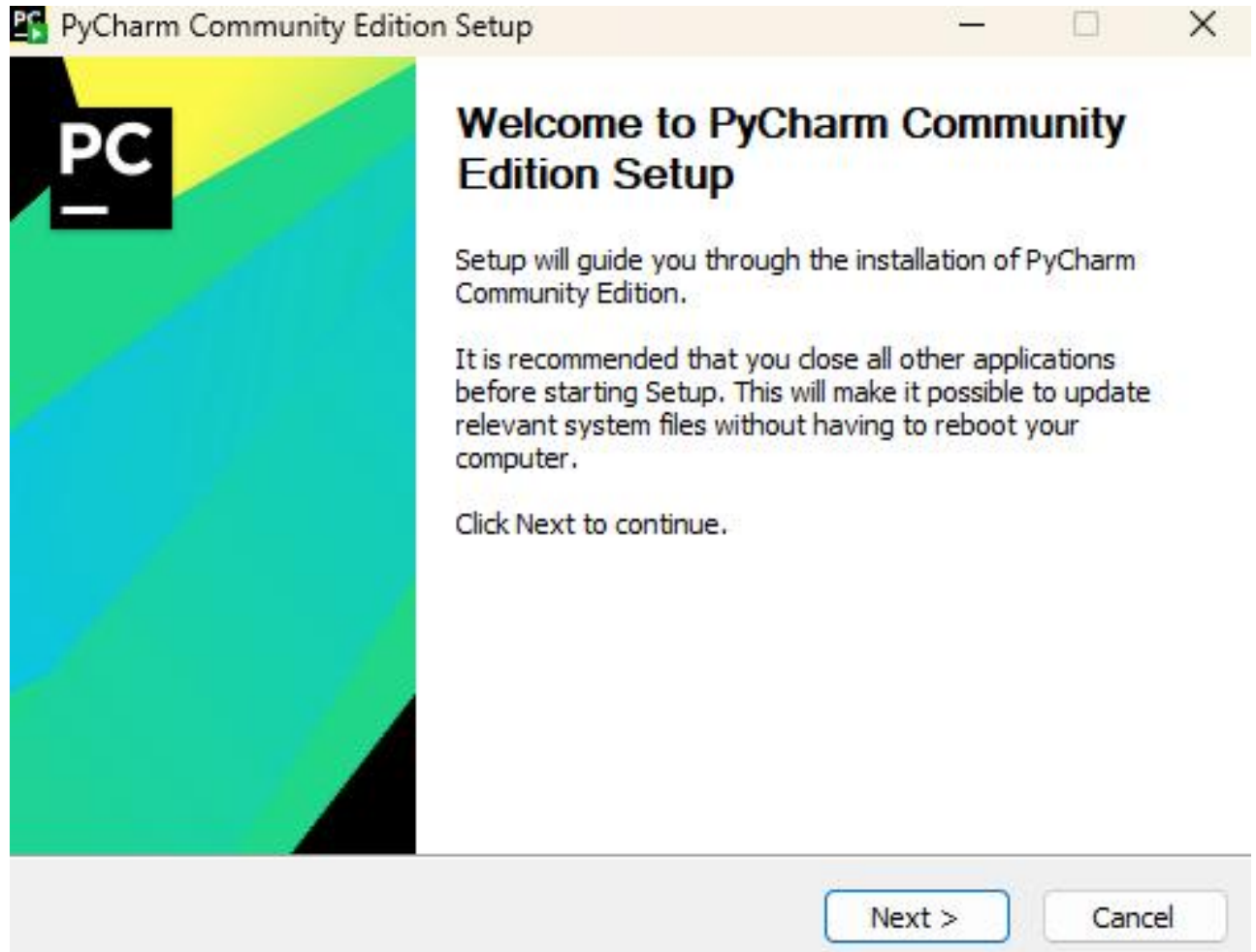


Рис. 3.2. Вікно встановлення PyCharm

Після завантаження файлу виконання я подвійно клацнув на ньому, щоб розпочати процес інсталяції. З'явилося вікно встановлення з вітанням, яке привітало мене. Далі мені запропонувало вибрати тип інсталяції: Community або Professional. Я обрав Community Edition, оскільки вона є безкоштовною і має всі основні можливості, які мені потрібні для роботи з Python.

Після вибору типу інсталяції я погодився з ліцензійною угодою та вибрав шлях для встановлення PyCharm на моєму комп'ютері. За замовчуванням було запропоновано шлях, але я міг вибрати інший шлях, якщо бажаю.

Потім я вибрав компоненти, які хотів встановити разом з PyCharm. Це включало в себе додаткові плагіни та інструменти, які можна використовувати в IDE. Я встановив деякі корисні плагіни для роботи з Git та віртуальними середовищами Python.

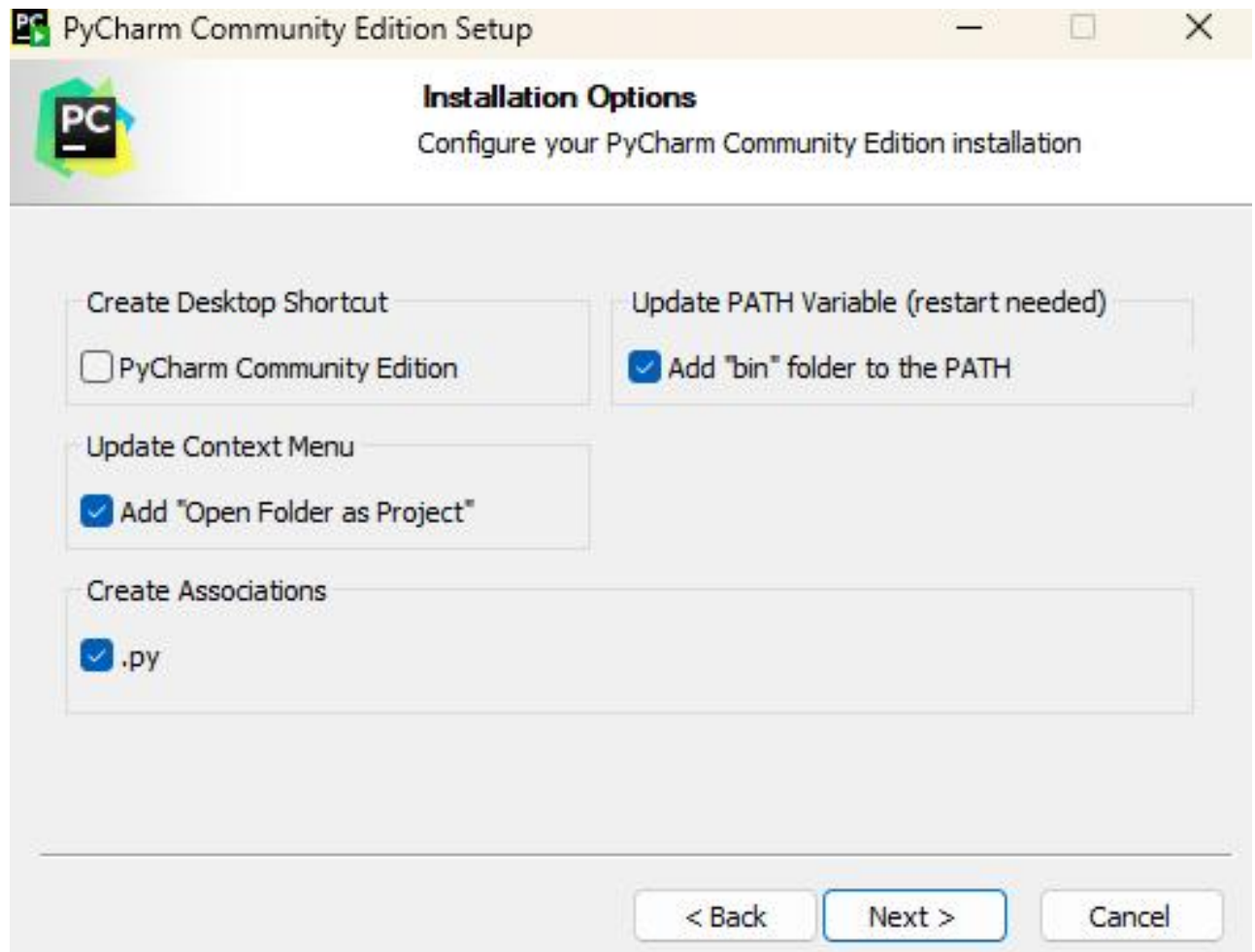


Рис. 3.3. Вибір додаткових компонентів (плагінів та інструментів для роботи з Python)

Після того, як я обрав компоненти, я натиснув кнопку "Встановити", і процес інсталяції розпочався. Це зайняло кілька хвилин, після чого я побачив повідомлення, що інсталяція успішно завершена.

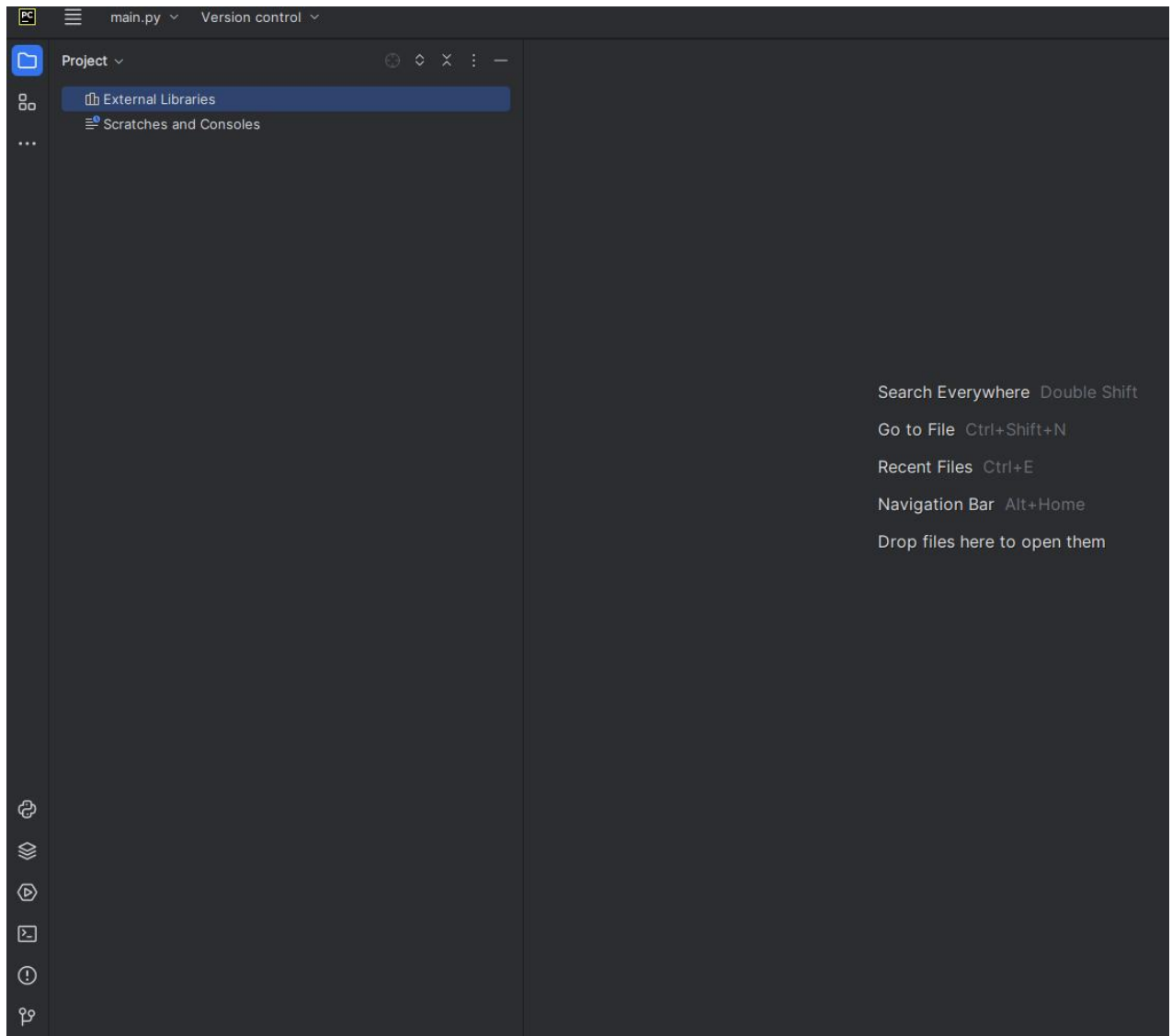


Рис. 3.4. Запуск PyCharm

### 3.2.2. Інсталяція бібліотеки

Після встановлення PyCharm, я створив новий проект, обрав мову програмування Python і налаштував віртуальне середовище для мого проекту.

Починаємо з інсталяції бібліотеки telebot:

```
PS C:\Users\user\Desktop\13123123> pip3 install pyTelegramBotAPI
```

Рис. 3.5. Інсталяція бібліотеки telebot



```
Terminal Local + v
+ CategoryInfo          : Ошибка безопасности: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : UnauthorizedAccess

PS C:\Users\user\Desktop\13123123> pip3 install pyTelegramBotAPI
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pyTelegramBotAPI in c:\users\user\appdata\roaming\python\python311\site-packages (4.12.0)
Requirement already satisfied: requests in c:\users\user\appdata\roaming\python\python311\site-packages (from pyTelegramBotAPI) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\user\appdata\roaming\python\python311\site-packages (from requests->pyTelegramBotAPI) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in c:\users\user\appdata\roaming\python\python311\site-packages (from requests->pyTelegramBotAPI) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\user\appdata\roaming\python\python311\site-packages (from requests->pyTelegramBotAPI) (2.0.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\user\appdata\roaming\python\python311\site-packages (from requests->pyTelegramBotAPI) (2023.5.7)

[notice] A new release of pip available: 22.3.1 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\user\Desktop\13123123>
```

Рис. 3.6. Успішна інсталяція бібліотеки

### 3.2.3. Реалізація боту

Далі переходимо до написання коду:

```
import random
import telebot
import webbrowser
from telebot import types
```

Рис. 3.7. Початок написання коду

```
# Зчитую свій токен із файлу mytoken.txt, у твоєму
випадку це буде не потрібно
file = open('./mytoken.txt')
mytoken = file.read()
# Передаємо сюди токен, який отримали від FatherBot
bot =
telebot.TeleBot('6099909092:AAGCJVByZS74GfiNW0tW5lpL9q
uNpNm14o')
# Варіанти відповідей користувачеві, якщо той ввів
незрозуміле боту повідомлення
answers = ['Я не зрозумів, що ти хочеш сказати.',
'Вибач, я тебе не розумію. ', 'Я не знаю такої
команди.', 'Мій розробник не казав, що відповідати в
такій ситуації... >_<']
```

Рис. 3.8. Створюємо токен API за допомогою чат-боту botfather

Далі ми переходимо до функціоналу бота та додаємо кнопки які допоможуть нам взаємодіяти з покупцем

```
# Обробка команди/start
@bot.message_handler(commands=['start'])
def welcome(message):
    # Додаємо кнопки, які з'являтимуться після
    # введення команди /start
    markup =
types.ReplyKeyboardMarkup(resize_keyboard=True)
    button1 = types.KeyboardButton('')
    button2 = types.KeyboardButton('⚙️ Налаштування')
    button3 = types.KeyboardButton('')
    # Розділяю кнопки по рядках так, щоб товари були
    # окремо від інших кнопок
    markup.row(button1)
    markup.row(button2, button3)

    if message.text == '/start':
        # Надсилаю вітальний текст
        bot.send_message(message.chat.id, f'Привіт,
{message.from_user.first_name}!\nУ мене ти зможеш
купити деякі товари!\nКонтакт мого розробника::
https://t.me/Candynut', reply_markup=markup)
    else:
        bot.send_message(message.chat.id, 'Перекинув
тебе в головне меню! Вибирай!', reply_markup=markup)
```

Рис. 3.9. Додаємо кнопки

Цей код обробляє команду "/start", яка викликається користувачем у чаті з ботом. Коли бот отримує цю команду, виконується функція welcome, яка приймає параметр message (повідомлення від користувача).

У функції welcome створюється клавіатура з кнопками, які з'являються після введення команди "/start". Це досягається за допомогою класу ReplyKeyboardMarkup з бібліотеки python-telegram-bot. Кнопки створюються за допомогою об'єктів KeyboardButton і додаються до клавіатури.

У результуючій клавіатурі кнопки розташовані на різних рядках, щоб товари були окремо від інших кнопок. Це досягається за допомогою методу `row` класу `ReplyKeyboardMarkup`, який додає кнопки до різних рядків клавіатури.

Якщо текст повідомлення, отриманого від користувача, дорівнює `"/start"`, то бот відправляє привітальне повідомлення разом з клавіатурою на чатовий ідентифікатор користувача. У привітальному повідомленні використовується форматування рядка для підстановки значення `message.from_user.first_name`, яке представляє ім'я користувача. Також в повідомленні є посилання на контакт розробника.

Якщо отримане повідомлення не дорівнює `"/start"`, то бот відправляє повідомлення "Перекинув тебе в головне меню! Вибирай!" разом з клавіатурою на чатовий ідентифікатор користувача.

Далі написання функцій для кнопок:

```
# Обробка звичайних текстових команд, описаних у
кнопках
@bot.message_handler()
def info(message):
    if message.text == ':
        goodsChapter(message)
    elif message.text == ' Налаштування':
        settingsChapter(message)
    elif message.text == ':
        infoChapter(message)
    elif message.text == ':
        markup =
types.ReplyKeyboardMarkup(resize_keyboard=True)
        button1 = types.KeyboardButton(')
        button2 = types.KeyboardButton(' Назад')
        markup.row(button1, button2)
        bot.send_message(message.chat.id, 'Ігрова
клавіатура : HATOR Rockfall EVO TKL',
```

Рис. 3.10. Написання функцій для кнопок

```

reply_markup=markup)
    elif message.text == ':
        markup =
types.ReplyKeyboardMarkup(resize_keyboard=True)
        button1 = types.KeyboardButton(')
        button2 = types.KeyboardButton(' Назад')
        markup.row(button1, button2)
        bot.send_message(message.chat.id, 'Ігрова
миша : HyperX Pulsefire Haste', reply_markup=markup)
    elif message.text == ':
        markup =
types.ReplyKeyboardMarkup(resize_keyboard=True)
        button1 = types.KeyboardButton(')
        button2 = types.KeyboardButton(' Назад')
        markup.row(button1, button2)
        bot.send_message(message.chat.id, 'Ігрові
навушники : Logitech G PRO X Gaming Headset',
reply_markup=markup)
    elif message.text == ':
        markup =
types.ReplyKeyboardMarkup(resize_keyboard=True)
        button1 = types.KeyboardButton(')
        button2 = types.KeyboardButton(' Назад')
        markup.row(button1, button2)
        bot.send_message(message.chat.id, 'Ігрове
крісло : DXRacer Air PRO', reply_markup=markup)
    elif message.text == ' Налаштування #1':
        # Функціонал кнопки
        bot.send_message(message.chat.id,
'Налаштування номер 1...')
    elif message.text == ' Налаштування #2':
        # Функціонал кнопки
        bot.send_message(message.chat.id,
'Налаштування номер 2...')
    elif message.text == ' or message.text == '
Написати розробнику':

```

Рис. 3.11. Продовження написання функцій для кнопок

```
webbrowser.open('https://t.me/Candynut')
elif message.text == ' Назад':
    goodsChapter(message)
elif message.text == ' Назад у меню':
    welcome(message)
```

Рис. 3.12. Продовження написання функцій для кнопок

Цей код визначає функцію `info`, яка обробляє повідомлення від користувача, коли не співпадає з попередніми визначеними командами.

У функції `info` використовується об'єкт `message`, який містить інформацію про отримане повідомлення від користувача.

Код виконує перевірку значення `message.text`, що представляє текстове повідомлення від користувача. Залежно від значення `message.text` виконується певна дія.

Якщо значення `message.text` дорівнює певному рядку, наприклад `'`, то викликається функція `goodsChapter(message)`, яка обробляє вибір товару. Аналогічно, для команд `' Налаштування'`, `' '`, `' '` викликаються відповідні функції `settingsChapter(message)`, `infoChapter(message)`, `gameKeyboard(message)`, `chairChapter(message)`.

У кожному з цих блоків коду створюється клавіатура з кнопками за допомогою класу `ReplyKeyboardMarkup`, і відповідне повідомлення разом з клавіатурою відправляється користувачу за допомогою `bot.send_message`. Кнопки додаються до клавіатури за допомогою методу `row`.

Додатково, код обробляє команди `' Налаштування #1'`, `' Налаштування #2'`, `' Написати розробнику'`, `' Назад'`, `' Назад у меню'` та виконує відповідні дії. Наприклад, відправляє повідомлення з налаштуваннями, відкриває посилання на розробника у браузері або повертає користувача до попереднього розділу.

Отже, цей код обробляє повідомлення, які не співпадають з попередніми визначеними командами у клавіатурі. Це дає можливість реагувати на довільні повідомлення, що не відповідають конкретним кнопкам.

Крім того, код містить додаткові перевірки для конкретних команд. Наприклад, якщо значення `message.text` дорівнює ' Налаштування #1', відправляється повідомлення з налаштуваннями номер 1. Аналогічно, для команди ' Налаштування #2' відправляється повідомлення з налаштуваннями номер 2.

У рядках, де значення `message.text` порівнюється з ' Написати розробнику', використовується функція `webbrowser.open()` для відкриття посилання на розробника у браузері.

Крім того, є перевірки для команд ' Назад' і ' Назад у меню'. У разі отримання цих команд викликаються відповідні функції для повернення користувача до попередніх розділів.

Загалом, цей код надає можливість обробляти повідомлення, які не відповідають заданим командам у клавіатурі, і виконувати відповідні дії в залежності від отриманого повідомлення. Він також дозволяє взаємодіяти з користувачем за допомогою клавіатури, надсилати повідомлення з текстом та реалізовувати додатковий функціонал на основі отриманих команд.

Користувач може отримати відповідь, якщо напише власне повідомлення боту, він отримає рандомно згенеровану відповідь із варіантів, які ми йому надамо при бажанні (на рис 3.2. бот не надає відповіді) :

```
# Якщо користувач написав своє повідомлення, то бот генерує один із можливих варіантів відповіді
# Додавати і редагувати варіанти відповідей можна в списку answers
else:
    bot.send_message(message.chat.id,
answers[random.randint(0, 3)])
```

Рис. 3.13. Функція для відповіді користувачеві

Також нам потрібен каталог товарів:

```
# Функція, що відповідає за розділ товарів
def goodsChapter(message):
    # Кнопки для товарів
    markup =
types.ReplyKeyboardMarkup(resize_keyboard=True)
    button1 = types.KeyboardButton('')
    button2 = types.KeyboardButton('')
    button3 = types.KeyboardButton('')
    button4 = types.KeyboardButton('')
    button5 = types.KeyboardButton(' Назад у меню')
    markup.row(button1, button2)
    markup.row(button3, button4)
    markup.row(button5)
```

Рис. 3.14. Функція для вибору товарів

Цей код визначає функцію `goodsChapter`, яка відповідає за розділ товарів у боті.

У функції створюється клавіатура з кнопками для товарів за допомогою класу `ReplyKeyboardMarkup`. Кожна кнопка представлена об'єктом класу `KeyboardButton` з відповідним текстом.

У даному випадку визначено п'ять кнопок для товарів. Змінні `button1`, `button2`, `button3`, `button4` та `button5` містять об'єкти `KeyboardButton` з текстами товарів.

Далі кнопки додаються до клавіатури за допомогою методу `row` класу `ReplyKeyboardMarkup`. Кнопки `button1` та `button2` розміщуються в одному рядку, `button3` та `button4` - в другому рядку, а `button5` розміщується окремо у третьому рядку.

Параметр `resize_keyboard=True` в конструкторі класу `ReplyKeyboardMarkup` дозволяє автоматично змінювати розмір клавіатури залежно від екрану користувача.

Отримана клавіатура разом з кнопками використовується для надсилання повідомлення користувачеві за допомогою методу `bot.send_message`.

Клавіатура встановлюється як параметр `reply_markup`, щоб відобразити кнопки разом з повідомленням.

Таким чином, цей код створює клавіатуру з кнопками для товарів у розділі товарів бота. Користувач може вибрати один з товарів, натискаючи на відповідну кнопку.

Далі створив функцію, для кнопок налаштування, які поки не функціонують, але за потребою вони можуть відповідати за вибір конкретної периферії (клавіатури, мишки, гарнітури і т.д.)

```
# Функція, що відповідає за розділ налаштувань
def settingsChapter(message):
    markup =
types.ReplyKeyboardMarkup(resize_keyboard=True)
    button1 = types.KeyboardButton('  Налаштування #1')
    button2 = types.KeyboardButton('  Налаштування #2')
    button3 = types.KeyboardButton('  Назад в меню')
    markup.row(button1, button2)
    markup.row(button3)
    bot.send_message(message.chat.id, 'Розділ
налаштувань.\nОбери один із варіантів:',
reply_markup=markup)
```

Рис. 3.15. Функція кнопок налаштування

Цей код визначає функцію `settingsChapter`, яка відповідає за розділ налаштувань у боті.

У функції створюється клавіатура з кнопками для налаштувань за допомогою класу `ReplyKeyboardMarkup`. Кожна кнопка представлена об'єктом класу `KeyboardButton` з відповідним текстом.

У даному випадку визначено три кнопки для налаштувань. Змінні `button1`, `button2` та `button3` містять об'єкти `KeyboardButton` з текстами налаштувань.

Далі кнопки додаються до клавіатури за допомогою методу `row` класу `ReplyKeyboardMarkup`. Кнопки `button1` та `button2` розміщуються в одному рядку, а `button3` розміщується окремо у другому рядку.



Параметр `resize_keyboard=True` в конструкторі класу `ReplyKeyboardMarkup` дозволяє автоматично змінювати розмір клавіатури залежно від екрану користувача.

Отримана клавіатура разом з кнопками використовується для надсилання повідомлення користувачеві за допомогою методу `bot.send_message`. Клавіатура встановлюється як параметр `reply_markup`, щоб відобразити кнопки разом з повідомленням.

Таким чином, цей код створює клавіатуру з кнопками для налаштувань у розділі налаштувань бота. Користувач може вибрати одне з налаштувань, натискаючи на відповідну кнопку.

Звісно нам потрібна функція для зворотнього зв'язу з користувачем, кнопка довідки та кнопка повернення до меню.

```
# Функція, що відповідає за розділ допомоги
def infoChapter(message):
    markup =
types.ReplyKeyboardMarkup(resize_keyboard=True)
    button1 = types.KeyboardButton(' Написати
розробнику')
    button2 = types.KeyboardButton(' Назад у меню')
    markup.row(button1, button2)
    bot.send_message(message.chat.id, 'Розділ
довідки.\nТут ти можеш написати моєму розробнику.',
reply_markup=markup)
```

Рис. 3.16. Кнопки зворотнього зв'язку, довідки та повернення

Цей код визначає функцію `infoChapter`, яка відповідає за розділ допомоги в боті.

У функції створюється клавіатура з кнопками для розділу допомоги за допомогою класу `ReplyKeyboardMarkup`. Кожна кнопка представлена об'єктом класу `KeyboardButton` з відповідним текстом.

У даному випадку визначено дві кнопки: `button1` з текстом " Написати розробнику" та `button2` з текстом " Назад у меню".

Далі кнопки додаються до клавіатури за допомогою методу `row` класу `ReplyKeyboardMarkup`. Кнопки `button1` та `button2` розміщуються в одному рядку.

Параметр `resize_keyboard=True` в конструкторі класу `ReplyKeyboardMarkup` дозволяє автоматично змінювати розмір клавіатури залежно від екрану користувача.

Отримана клавіатура разом з кнопками використовується для надсилання повідомлення користувачеві за допомогою методу `bot.send_message`. Клавіатура встановлюється як параметр `reply_markup`, щоб відобразити кнопки разом з повідомленням.

Таким чином, цей код створює клавіатуру з кнопками для розділу допомоги у боті. Користувач може вибрати одну з опцій, натискаючи на відповідну кнопку. Зокрема, кнопка " Написати розробнику" дозволяє користувачу надіслати повідомлення розробнику, а кнопка " Назад у меню" дозволяє повернутися до головного меню.

Користувач матиме змогу дізнатись, які товари в наявності:

```
# Відправляємо повідомлення з прикріпленими до нього  
кнопками товарів  
bot.send_message(message.chat.id, 'Ось усі товари,  
які зараз перебувають у продажу:', reply_markup=markup)
```

Рис. 3.17. Кнопка отримання інформації щодо товарів

Цей код на мові програмування Python відправляє повідомлення користувачеві з прикріпленими до нього кнопками товарів.

У цьому коді використовується функція `bot.send_message` для відправки повідомлення. Перший аргумент `message.chat.id` вказує ідентифікатор чату або користувача, до якого буде відправлене повідомлення.

Другий аргумент 'Ось усі товари, які зараз перебувають у продажу:' містить текст повідомлення, яке буде відображатись у чаті.

Третій аргумент `reply_markup=markup` встановлює клавіатуру `markup` як прикріплену клавіатуру до повідомлення. Це означає, що разом з повідомленням будуть відображені кнопки, які були створені та налаштовані у клавіатурі `markup`.

Таким чином, цей код надсилає повідомлення з текстом "Ось усі товари, які зараз перебувають у продажу:" і прикріпленою клавіатурою `markup`, яка містить кнопки товарів. Користувач зможе вибрати одну з цих кнопок, натиснувши на неї у чаті.

Для того, щоб наш бот працював завжди ми задаємо йому наступних параметрів :

```
# Рядок, щоб програма не зупинялася  
bot.polling(none_stop=True)
```

Рис. 3.18. Надання можливості безперервної роботи боту

Цей викликає функцію `bot.polling(none_stop=True)`, яка починає постійно (без зупинки) отримувати та обробляти нові повідомлення.

Функція `bot.polling()` використовується в телеграм-ботах для отримання повідомлень в реальному часі. Вона запускає цикл, який постійно перевіряє наявність нових повідомлень та обробляє їх. При отриманні нового повідомлення, бот виконує відповідну функцію обробки для цього повідомлення.

Аргумент `none_stop=True` вказує, що функція `bot.polling()` повинна працювати без зупинки і повторюватись у випадку виникнення помилок чи відключення з'єднання. Це дозволяє боту постійно бути активним та готовим обробляти повідомлення незалежно від зовнішніх умов.

Отже, цей рядок коду забезпечує постійну роботу бота та обробку нових повідомлень у режимі "non-stop" (без зупинки).

### 3.3. Демонстрація готового продукту

Основні принципи тестування стосовно телеграм-ботів включають наступні аспекти:

**Функціональне тестування:** Слід переконатися, що бот правильно реагує на команди користувачів, обробляє запити та надає очікувані результати. Проведіть перевірку всіх основних функцій, таких як відображення та вибір товарів, обробка замовлень та взаємодія з клієнтами.

**Тестування взаємодії з API Телеграма:** Оскільки телеграм-боти взаємодіють з API Телеграма, слід переконатися, як боти обробляють та відправляють повідомлення, обробляють клавіатурні натискання та взаємодіють з іншими функціями API. Переконайтеся, що бот правильно виконує ці дії та передає відповідні дані.

**Тестування на різних сценаріях взаємодії:** Під час тестування слід враховувати різні сценарії взаємодії з ботом. Спробуйте різні типи запитів, введіть некоректні дані та переконайтеся, як бот відповідає на них. Тестування на різних сценаріях допоможе виявити можливі проблеми та забезпечити очікувану роботу бота.

**Тестування з реальними користувачами:** Щоб отримати більш точні результати тестування, можна залучити реальних користувачів до процесу. Запросіть їх скористатися ботом та надати повідомлення про свої спостереження, враження та можливі проблеми. Відгуки реальних користувачів можуть допомогти виявити недоліки та покращити взаємодію з ботом.

**Тестування на різних пристроях та платформах:** Слід переконатися, що бот працює на різних пристроях та платформах, таких як смартфони, планшети та комп'ютери. Перевірте, як виглядає інтерфейс бота на різних розмірах екрану та чи функціонує він без проблем.

**Тестування безпеки:** Безпека є важливим аспектом тестування телеграм-ботів. Слід переконатися, що боти не мають вразливостей, які можуть бути використані для несанкціонованого доступу або шкідливих дій.

## Ілюстрації готового чат-боту та його функціоналу

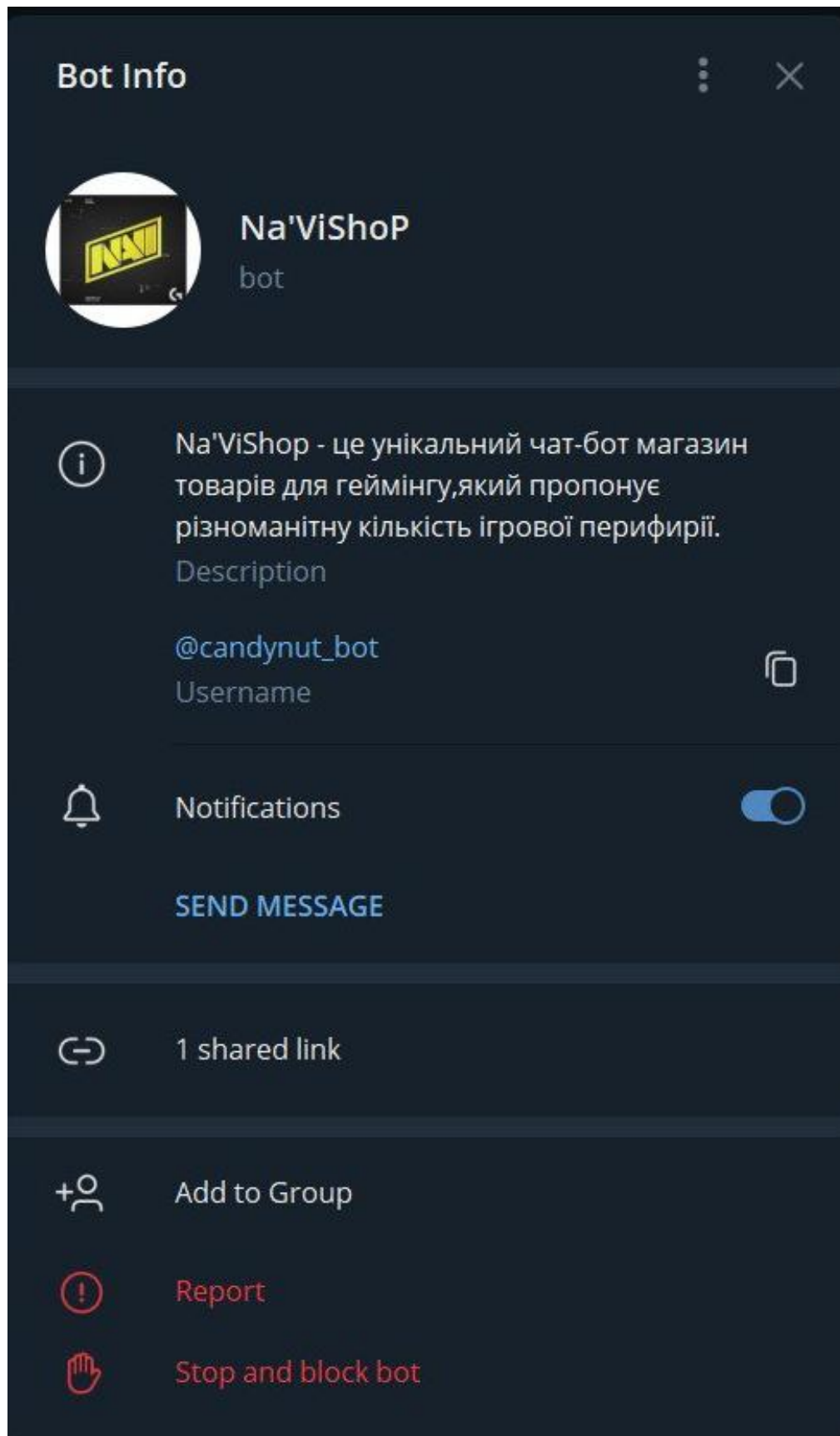


Рис. 3.19. Інформація чат-бота

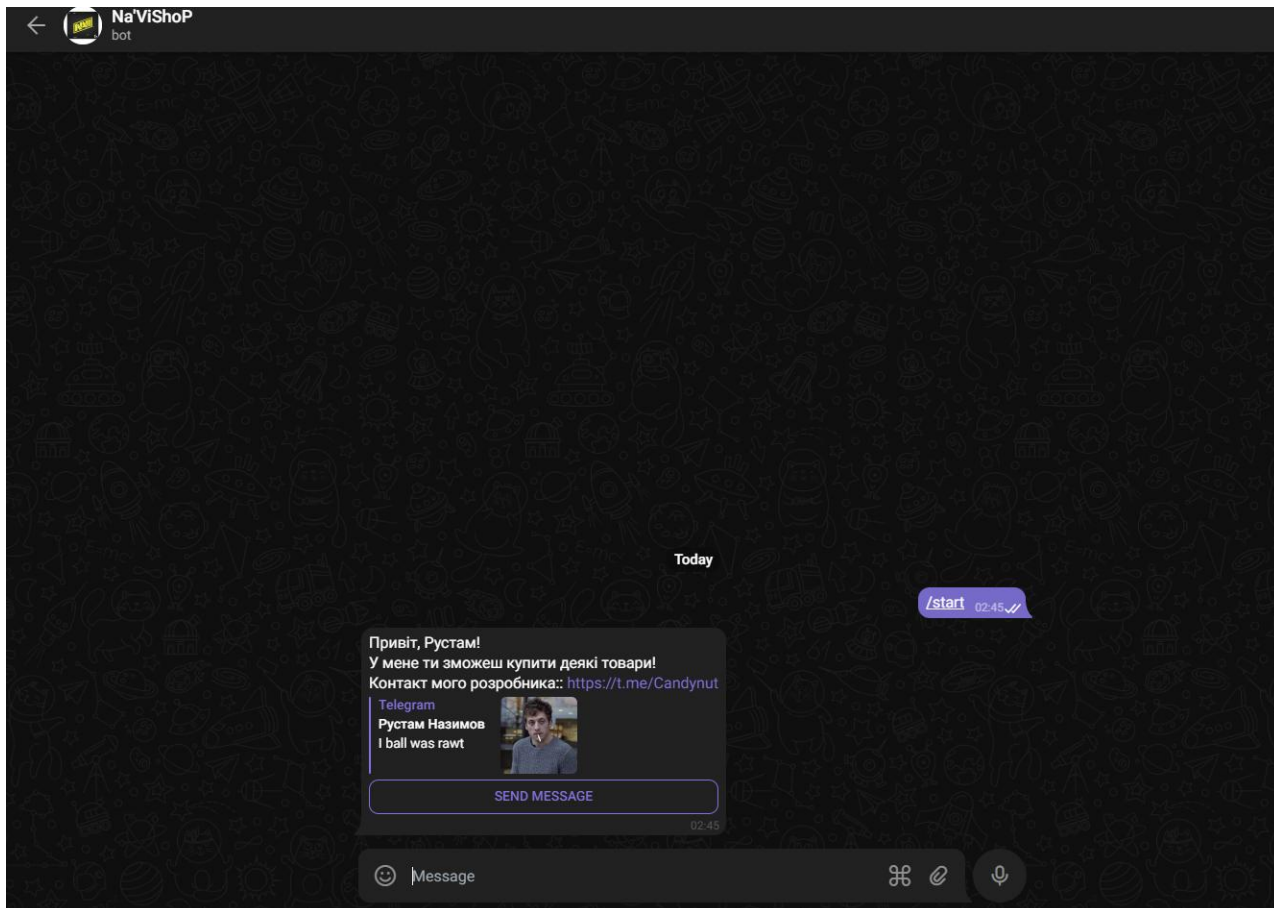


Рис. 3.20. Початок взаємодії з чат-ботом

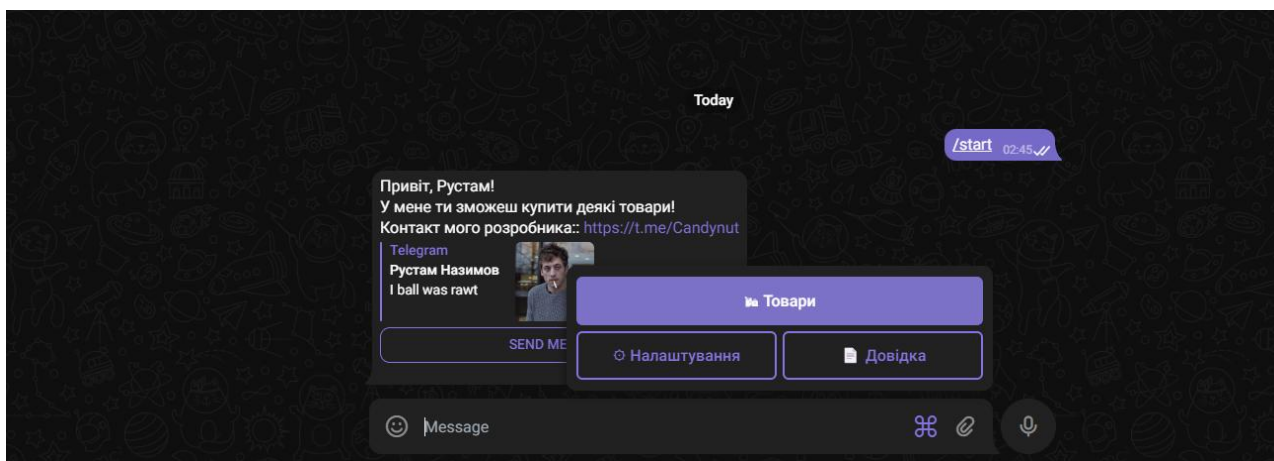


Рис. 3.21. Головне Меню



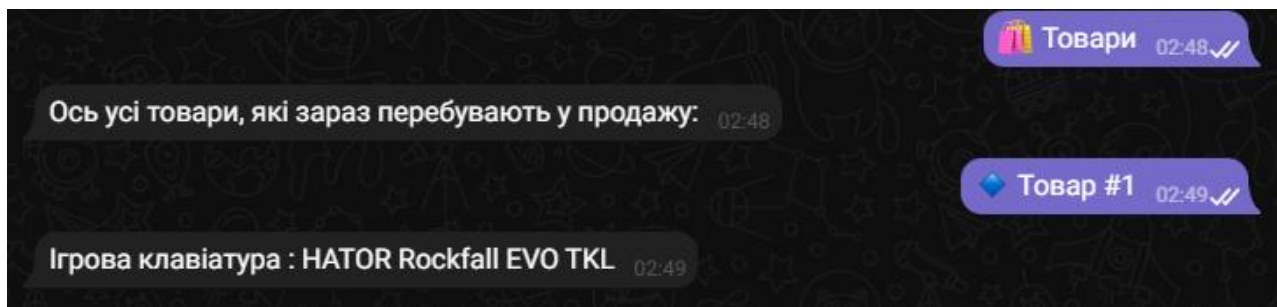


Рис. 3.22. Можливість вибору товарів

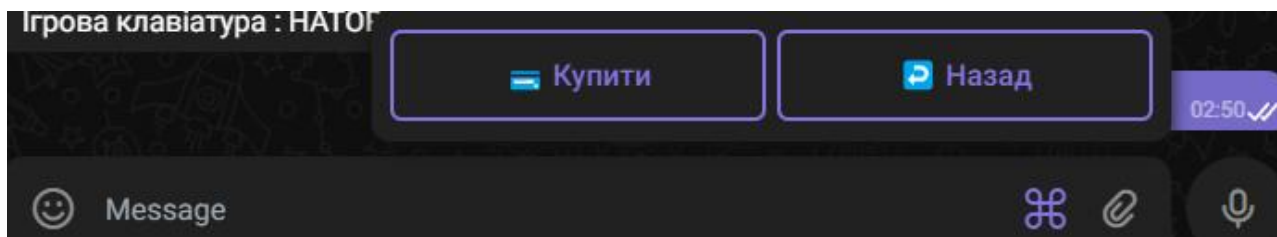


Рис. 3.23. Після вибору товару, з'являється можливість його придбати

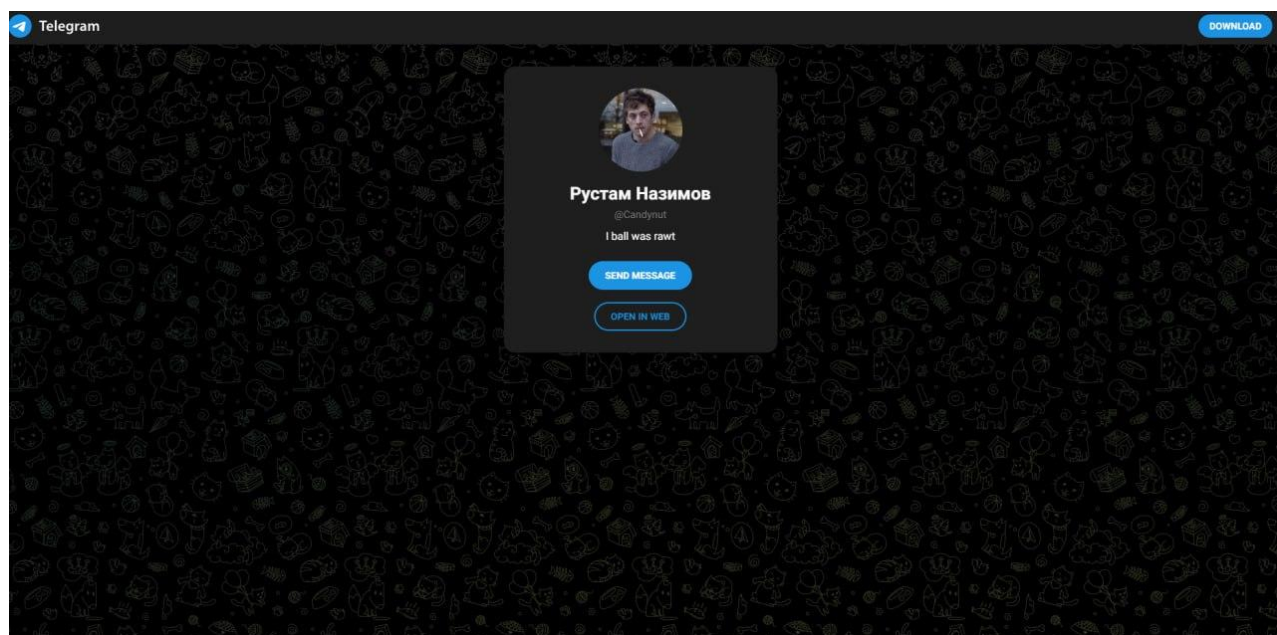


Рис. 3.24. Кнопка "Купити" перенаправляє вас власника для можливості придбання товару

### ВИСНОВКИ ДО РОЗДІЛУ 3

Було докладно розглянуто процес завантаження програмного забезпечення, яке буде використовуватись для подальшої розробки нашого чат-бота. Ми провели аналіз різних інструментів та технологій, що можуть бути використані в процесі розробки, і прийшли до висновку, що Python є найоптимальнішим вибором для наших потреб.

Python вражає своїм простим синтаксисом та багатofункціональністю, що дозволяє нам ефективно реалізувати потрібний функціонал нашого чат-бота. Крім того, Python має широку підтримку спільноти, що є важливим фактором для успішної розробки та підтримки проекту.

Наш чат-бот, завдяки використанню Python та деяких ключових бібліотек і фреймворків, буде мати значний функціонал та розширені можливості. Ми розглянули різні аспекти його функціоналу, забезпечили йому постійну доступність та функціонування, а також передбачили можливість зворотного зв'язку з користувачами.

Завдяки нашій ретельній розробці та увазі до деталей, наш чат-бот буде готовий зустрічати потреби користувачів, надавати їм зручні та швидкі способи взаємодії, а також забезпечувати безперервне та надійне функціонування.

У практичній частині роботи було докладено зусиль для розробки та впровадження нашого чат-бота з високоякісним функціоналом і зручним інтерфейсом. Наша робота була спрямована на створення ефективного та надійного чат-бота, який задовольнить потреби наших користувачів та підтримуватиме наш онлайн-магазин геймерських товарів на високому рівні.

В даному розділі було розглянуто завантаження програмного забезпечення для подальшу роботу над написанням чат-бота. Також розглянули функціонал та можливості бота, надали йому безперервну можливість функціонування, зворотнього зв'язку тощо.



## ВИСНОВКИ

У даній кваліфікаційній роботі було досліджено та розроблено застосування технології чат-ботів для онлайн-магазину з продажу геймерських товарів з використанням мови програмування Python. Отримані результати підтвердили ефективність та перспективи використання чат-ботів для покращення процесу продажу товарів і поліпшення взаємодії з клієнтами.

Під час дослідження були визначені основні вимоги та функціональні можливості, які повинен мати чат-бот для ефективної роботи в онлайн-магазині геймерських товарів. Було проведено аналіз існуючих рішень та технологій для розробки чат-ботів і вибрано мову програмування Python як основний інструмент реалізації проекту.

За допомогою інтегрованого середовища розробки PyCharm, було створено чат-бот, надає можливість перегляду та придбання геймерських товарів, надсилає повідомлення про акції та новини, а також має можливість здійснювати підтримку клієнтів шляхом відповіді на запитання та допомоги вирішенні проблем.

На основі отриманих результатів можна зробити висновок, що застосування технології чат-ботів в онлайн-магазинах з продажу геймерських товарів є дієвим інструментом для покращення процесу продажу та забезпечення зручної комунікації з клієнтами. Використання мови програмування Python дозволяє розробникам ефективно та швидко реалізувати функціональність чат-ботів і забезпечити їх надійну та стабільну роботу.

Майбутнім розвитком проекту може бути розширення функціональних можливостей чат-бота, впровадження системи рекомендацій товарів, інтеграція з платіжними шлюзами та вдосконалення системи підтримки клієнтів. Крім того, можливість розширення проекту на інші платформи месенджерів та інтеграція з іншими сервісами може сприяти подальшому розширенню бази клієнтів та збільшенню обсягів продажів геймерських товарів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Маттез Е. Python Crash Course: A Hands-On, Project-Based Introduction to Programming: підручник / Е. Маттез. - Сан-Франциско, СА: No Starch Press, 2019. - 560 с.
2. freeCodeCamp: Як створити телеграм-бота за допомогою Python. [Електронний ресурс] Режим доступу: <https://www.freecodecamp.org/ukrainian/news/yak-stvoryty-telehram-bota-za-dopomohoyu-python/>(дата звернення 25.04.2023).
3. Evergreen: Розробка чат-ботів. [Електронний ресурс] Режим доступу: <https://evergreens.com.ua/ua/articles/bot-reviews.html> (дата звернення 25.04.2023).
4. Wikipedia: Інструменти веброзробки. [Електронний ресурс] Режим доступу: [https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B8\\_%D0%B2%D0%B5%D0%B1%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B8](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B8_%D0%B2%D0%B5%D0%B1%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B8) (дата звернення 25.04.2023).
5. NOSC-UA Hub: TensorFlow. [Електронний ресурс] Режим доступу : [http://cloud-5.bitp.kiev.ua/?page\\_id=598](http://cloud-5.bitp.kiev.ua/?page_id=598) (дата звернення 25.04.2023).
6. DUO : Перші кроки в NLP: розглядаємо Python-бібліотеку NLTK в реальному завданні. [Електронний ресурс] Режим доступу: <https://dou.ua/lenta/articles/first-steps-in-nlp-nltk/> (дата звернення 25.04.2023).
7. GitHub: Pytorch. [Електронний ресурс] Режим доступу: <https://github.com/pytorch/pytorch> (дата звернення 25.04.2023).
8. GitHub: spaCy. [Електронний ресурс] Режим доступу: <https://github.com/explosion/spaCy> (дата звернення 25.04.2023).
9. qudata: ML Чатбот RASA Введення. [Електронний ресурс] Режим доступу: [https://qudata.com/ml/ru/NLP\\_RASA.html](https://qudata.com/ml/ru/NLP_RASA.html) (дата звернення 25.04.2023).
10. Service Desk: Як створити Телеграм-Бота правильно. [Електронний ресурс] Режим доступу:

<https://www.servicedesk.site/uk/2019/08/09/%D1%8F%D0%BA-%D1%81%D1%82%D0%B2%D0%BE%D1%80%D0%B8%D1%82%D0%B8-%D1%82%D0%B5%D0%BB%D0%B5%D0%B3%D1%80%D0%B0%D0%BC-%D0%B1%D0%BE%D1%82%D0%B0-%D0%BF%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%D1%8C%D0%BD%D0%BE/>(дата звернення 25.04.2023).

11. Друкарня: Telegram-боти на Python: огляд п'яти найкращих фреймворків/бібліотек. [Електронний ресурс] Режим доступу: <https://drukarnia.com.ua/articles/telegram-boti-na-python-oglyad-p-yati-naikrashikh-freimvorkiv-bibliotek-L7UA7> (дата звернення 25.04.2023).

12. Codeguida: Створюємо Telegram бота на Python. Частина 1. [Електронний ресурс] Режим доступу: <https://codeguida.com/post/410> (дата звернення 25.04.2023).

13. Wikipedia: Python. [Електронний ресурс] Режим доступу : <https://uk.wikipedia.org/wiki/Python> (дата звернення 25.04.2023).

14. VEST: ГЕЙМЕРСЬКІ ДЕВАЙСИ: ІСТОРІЯ СОЗДАННЯ І ЯКІ ВАРТО КУПУВАТИ. [Електронний ресурс] Режим доступу: <https://vest.in.ua/ua/interesnaya-informacuya/gejmerskie-devajsy-istoriya-sozdaniya-i-kakie-stoit-pokupat> (дата звернення 25.04.2023).

15. GINCORE: Переваги продажів через чат-бот. [Електронний ресурс] Режим доступу: <https://gincore.net/uk/blog/preimuschestva-prodazh-cherez-chat-bot> (дата звернення 25.04.2023).