

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦІЇ ОБСЛУГОВУВАННЯ
КЛІЄНТІВ У СФЕРІ ЕЛЕКТРОННОЇ КОМЕРЦІЇ З ВИКОРИСТАННЯМ
PYTHON, NODE.JS**

**DEVELOPMENT OF A CHATBOT FOR AUTOMATING CUSTOMER
SERVICE IN E-COMMERCE USING PYTHON AND NODE.JS**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ПЗс-21
Приходько Максим Ігорович

Керівник:
Бойко Лев Степанович

Кваліфікаційну роботу
допущено до захисту
«10» 06 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна



Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри



«10» 12 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Приходьку Максиму Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка чат-бота для автоматизації обслуговування клієнтів у сфері електронної комерції з використанням Python і Node.js».

Керівник роботи: к.т.н., доцент Бойко Л. С.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

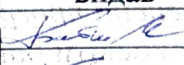
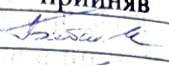
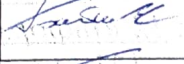
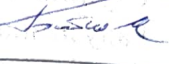
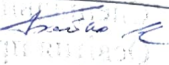
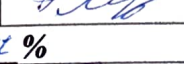
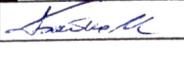
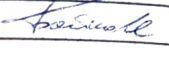
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «10» червня 2025 р.

3. Вихідні дані до роботи: Telegram API, Node.js, код на мові програмування Python, скріншоти інтерфейсу, методичні вказівки до виконання кваліфікаційної роботи бакалавра.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): аналіз існуючих рішень та технологій для розробки, проектування архітектури, реалізація функціоналу, вибір оптимальних інструментів та технологій, тестування та оптимізація роботи, оцінка ефективності впровадження.

5. Перелік графічного матеріалу: кількість рисунків : 16, кількість таблиць: 2.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Бойко Л. С.		
Специфікація вимог до розробленої системи	Бойко Л. С.		
Розробка об'єкта проектування	Бойко Л. С.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		1,07 %	
Академічна доброчесність	Бойко Л. С.		

7. Дата видачі завдання «20» грудня 2025 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	Виконано
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 01.03.2025 р.	Виконано
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	Виконано
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 29.03.2025 р.	Виконано
5	Практична реалізація об'єкта проектування та розробка бази даних	до 26.04.2025 р.	Виконано
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	Виконано
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 10.06.2025 р.	Виконано

Здобувач вищої освіти



Максим ПРИХОДЬКО

Керівник кваліфікаційної роботи



Лев БОЙКО

АНОТАЦІЯ

Приходько М. І. Розробка чат-бота для автоматизації обслуговування клієнтів у сфері електронної комерції з використанням Python, Node.js. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота бакалавра складається з вступу, трьох розділів, висновків та списку використаних джерел.

У першому розділі здійснено аналіз теоретичних основ створення чат-ботів, їх класифікації, переваг, обмежень, сучасних інструментів і сформульовано завдання дослідження. У другому розділі обґрунтовано вибір технологій, спроектовано архітектуру чат-бота та розроблено структуру бази даних. У третьому розділі реалізовано функціональні модулі чат-бота, створено базу даних, проведено тестування та розглянуто можливості подальшого розвитку. У висновках підсумовано результати роботи, оцінено ефективність розробленого рішення та визначено перспективи його використання.

Ключові слова: чат-бот, електронна комерція, Telegram API, Python, Node.js, автоматизація обслуговування, хмарні технології, база даних.

ABSTRACT

Prykhodko M. Development of a Chatbot for Automating Customer Service in E-commerce Using Python and Node.js. Manuscript. Qualification work of the bachelor's program "Software engineering" in the specialty "Software engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's qualification thesis consists of an introduction, three chapters, conclusions and a list of references.

The first section analyses the theoretical foundations of chatbots, their classification, advantages, limitations, modern tools, and formulates the objectives of the study. The second section justifies the choice of technologies, designs the architecture of the chatbot and develops the structure of the database. In the third section, the functional modules of the chatbot are implemented, the database is created, testing is carried out, and the possibilities for further testing, and the possibilities for further development. In the conclusion, the results of the work are summarised, the effectiveness of the developed solution is evaluated, and the prospects for its use are outlined.

Keywords: chatbot, e-commerce, Telegram API, Python, Node.js, customer service automation, cloud technologies, database.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	19
Висновки до розділу 1	20
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ ..	21
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	21
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання ...	23
Висновки до розділу 2	27
РОЗДІЛ 3 РОЗРОБКА ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦІЇ ОБСЛУГОВУВАННЯ КЛІЄНТІВ У СФЕРІ ЕЛЕКТРОННОЇ КОМЕРЦІЇ З ВИКОРИСТАННЯМ PYTHON, NODE.JS.....	30
3.1 Практична реалізація об'єкта проектування	30
3.2 Тестування та налагодження інформаційно-комп'ютерної системи.....	33
3.3 Розробка бази даних.....	37
3.4 Розробка інсталяційного пакета	39
Висновки до розділу 3	46
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50

ВСТУП

Актуальність теми. Зумовлена необхідністю підвищення рівня обслуговування клієнтів, зменшення навантаження на персонал, а також забезпечення цілодобової підтримки користувачів. Чат-боти дозволяють автоматизувати рутинні процеси, надаючи відповіді на типові запитання, допомагаючи з оформленням замовлень, консультуванням тощо.

Метою роботи є створення ефективного, масштабованого, функціонального та зручного у використанні чат-бота, який забезпечить автоматизоване обслуговування клієнтів у сфері електронної комерції, працюватиме на базі стеку Python та Node.js, використовуватиме API месенджера Telegram і хмарні технології для зберігання та обробки даних.

Сучасний бізнес дедалі частіше інтегрує цифрові технології для оптимізації обслуговування клієнтів. В умовах стрімкого розвитку електронної комерції та зростання обсягів онлайн-замовлень виникає потреба в ефективних інструментах комунікації між продавцем і покупцем. Одним із таких інструментів є чат-боти – автоматизовані програмні системи, які імітують спілкування з користувачем у режимі реального часу.

Об'єктом кваліфікаційної роботи бакалавра є процес автоматизації взаємодії між електронною торговою платформою та клієнтом за допомогою чат-бота.

Предметом кваліфікаційної роботи бакалавра виступають методи та засоби проектування, реалізації та розгортання інтелектуального чат-бота з підтримкою функціоналу обслуговування клієнтів.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- аналіз існуючих рішень та технологій для розробки чат-ботів, визначення їхніх переваг та недоліків;

- вибір оптимальних інструментів та технологій для реалізації поставленого завдання з урахуванням специфіки бізнес-процесів у сфері електронної комерції;
- проектування архітектури чат-бота, яка забезпечить масштабованість, надійність та зручність використання;
- реалізація функціоналу чат-бота, що включає каталог товарів, кошик покупок, оформлення замовлень, відстеження статусу замовлення, підтримку клієнтів та сповіщення;
- тестування та оптимізація роботи чат-бота, забезпечення його інтеграції з існуючими системами підприємства.

Структура роботи включає три основних розділи. У першому розділі розглядаються теоретичні аспекти, класифікація та переваги чат-ботів, аналізуються сучасні інструменти для їх створення. У другому розділі подано опис етапів проектування архітектури чат-бота, вибір технологій та побудову структури бази даних. Третій розділ присвячено безпосередньо реалізації функціоналу, розробки бази даних, тестуванню роботи бота та опису можливостей для подальшого розвитку.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Чат-боти – незамінні інструменти для сучасного бізнесу. Вони забезпечують швидку та ефективну комунікацію з клієнтами, автоматизують рутинні завдання та оптимізують витрати.

У загальному розумінні, чат-бот – це програмне забезпечення, що використовує штучний інтелект (ШІ) та алгоритми обробки природної мови для імітації людської взаємодії в текстовому або голосовому форматі [1].

Розвиток чат-ботів став можливим завдяки еволюції технологій машинного навчання та обробки великих даних. Основна мета чат-бота автоматизація рутинних завдань, зокрема обслуговування клієнтів, технічної підтримки, продажів та маркетингу.

У сфері електронної комерції чат-боти відіграють важливу роль: вони спрощують процес обслуговування споживачів, сприяють оптимізації бізнес-процесів і підвищують рівень задоволеності клієнтів.

Чат-боти можна класифікувати за кількома критеріями. За типом взаємодії вони поділяються на текстові, голосові та мультимодальні. Текстові чат-боти працюють через месенджери, веб-чати або SMS і використовують текстові повідомлення як основний спосіб комунікації. Голосові чат-боти взаємодіють через голосові інтерфейси, такі як розумні колонки або телефонні служби. Мультимодальні чат-боти комбінують текст і голос, забезпечуючи інтерактивний досвід для користувача [1].

За рівнем складності розрізняють прості та розумні чат-боти.

Прості чат-боти функціонують за принципом скриптів, виконуючи заздалегідь визначені сценарії. Такий вид чат-ботів обмежений у своїх

можливостях, оскільки не здатні аналізувати складні запити чи адаптуватися до змін у поведінці користувача.

Розумні чат-боти базуються на алгоритмах штучного інтелекту і здатні навчатися, аналізуючи попередні взаємодії, та надавати більш точні й персоналізовані відповіді [2].

За сферою застосування виділяють декілька типів чат-ботів. Інформаційні чат-боти використовуються для надання користувачам загальної інформації про продукти, послуги або компанії. Комерційні чат-боти орієнтовані на підтримку продажів і маркетингу: вони допомагають обирати товари, оформлювати замовлення та проводити транзакції. Сервісні чат-боти забезпечують технічну підтримку та допомагають у вирішенні проблем клієнтів. Розважальні чат-боти створені для забезпечення інтерактивного дозвілля, наприклад, у вигляді ігор чи жартівливих діалогів [2].

Особливе місце займають гібридні чат-боти, що поєднують функції кількох типів. У сфері електронної комерції вони можуть одночасно інформувати клієнтів про товари, підтримувати продажі та надавати технічну підтримку. Такі боти, завдяки інтеграції з CRM-системами, аналізують поведінку користувачів, їхні вподобання та історію покупок, забезпечуючи високий рівень персоналізації.

Інтеграція чат-ботів у електронну комерцію супроводжується низкою викликів. Серед них варто виділити необхідність забезпечення конфіденційності даних користувачів, адаптації до різноманітних мов і культур, а також забезпечення зручності використання. Проте їхня популярність продовжує зростати завдяки численним перевагам, серед яких доступність 24/7, скорочення витрат на обслуговування клієнтів, підвищення швидкості обробки запитів і зниження рівня помилок.

Сучасний бізнес стикається з постійною необхідністю оптимізації процесів, підвищення ефективності роботи та забезпечення високого рівня обслуговування клієнтів. Впровадження чат-ботів дозволяє значно знизити

витрати, автоматизувати рутинні завдання та покращити взаємодію зі споживачами. Завдяки розвитку технологій штучного інтелекту, обробки природної мови та машинного навчання, сучасні чат-боти стали інтелектуальними помічниками, здатними не лише надавати інформацію, а й вирішувати складні бізнес-задачі.

У глобальному масштабі чат-боти знаходять застосування у різних сферах: від електронної комерції та фінансів до охорони здоров'я та освіти. У електронній комерції вони допомагають клієнтам обирати товари, здійснювати покупки, відстежувати доставку та отримувати технічну підтримку [2].

У фінансовій сфері чат-боти використовуються для управління рахунками, консультацій щодо кредитування та надання інформації про транзакції. Системи охорони здоров'я інтегрують чат-ботів для запису пацієнтів на прийом, нагадувань про прийом ліків та первинної консультації.

Однак, попри їхню популярність, рівень впровадження чат-ботів значною мірою залежить від розміру бізнесу та його галузі. Великі компанії активно використовують чат-ботів для масштабування бізнесу та зниження витрат на обслуговування клієнтів. Малі й середні підприємства також починають усвідомлювати переваги цієї технології, зокрема можливість покращення клієнтського досвіду при відносно низьких витратах на впровадження [3].

Сучасні чат-боти розвиваються у напрямку персоналізації та багатофункціональності. Використання даних клієнтів дозволяє їм надавати індивідуальні рекомендації, прогнозувати потреби та адаптуватися до змін у поведінці споживачів. Інтеграція з CRM-системами, соціальними мережами та іншими цифровими платформами дає змогу чат-ботам працювати як частина єдиної екосистеми бізнесу.

Серед перспектив використання чат-ботів слід виділити подальший розвиток технологій голосового розпізнавання та емоційного інтелекту. Майбутні чат-боти зможуть розпізнавати емоційний стан користувачів і адаптувати стиль спілкування, підвищуючи рівень задоволеності клієнтів. Також

можливе подальше поширення багатомовних чат-ботів, що дозволить компаніям ефективніше працювати на міжнародних ринках.

Попри значні переваги, використання чат-ботів супроводжується з низкою труднощів. Серед них можна виділити необхідність забезпечення конфіденційності даних, складність налаштування для специфічних галузей та високу вартість розробки для багатофункціональних рішень. Однак з огляду на стрімкий розвиток технологій, ці труднощі можуть бути подолані у найближчому майбутньому [4].

Тенденції використання чат-ботів у бізнесі:

- широке впровадження в електронній комерції;
- використання в обслуговуванні клієнтів;
- інтеграція з CRM та іншими бізнес-системами;
- розвиток голосових чат-ботів;
- підвищення рівня автоматизації процесів.

Переваги для бізнесу:

- зниження витрат на обслуговування клієнтів;
- підвищення швидкості відповіді на запити;
- персоналізація взаємодії з клієнтами;
- залучення нових каналів комунікації;
- оптимізація внутрішніх бізнес-процесів.

Чат-боти стали важливим інструментом цифрової трансформації бізнесу, забезпечуючи автоматизацію процесів, зниження витрат і покращення взаємодії з клієнтами.

Їхнє впровадження у різних сферах, зокрема електронній комерції, дозволяє значно підвищити рівень персоналізації та ефективності обслуговування. Незважаючи на існуючі виклики, сучасні технології продовжують вдосконалювати функціональність і можливості чат-ботів, відкриваючи нові перспективи для їх використання у бізнесі. Враховуючи

швидкий розвиток цієї технології, чат-боти стануть ще більш важливою складовою успішних бізнес-стратегій у майбутньому.

Однією з головних переваг чат-ботів є можливість цілодобової роботи. Завдяки цьому клієнти можуть отримувати відповіді на свої запити в будь-який час, незалежно від робочого графіка компанії. Чат-боти здатні обробляти велику кількість запитів одночасно, що дозволяє уникати черг та знижувати навантаження на персонал.

Завдяки інтеграції з CRM-системами та використанню даних про попередні взаємодії, чат-боти можуть пропонувати індивідуальні рекомендації, адаптовані до потреб кожного користувача. Чат-боти можуть допомогти обрати товар, базуючись на історії покупок клієнта, або запропонувати знижку на підставі його поведінки на сайті [5].

Чат-боти також допомагають оптимізувати витрати бізнесу. Вони скорочують необхідність утримання великого штату працівників для обслуговування клієнтів, виконуючи рутинні завдання автоматично. Завдяки цьому компанії можуть спрямовувати ресурси на стратегічні напрями розвитку, такі як вдосконалення продуктів чи маркетингові компанії [5].

Основні переваги чат-ботів у сфері електронної комерції:

- обслуговування клієнтів у будь-який час доби, незалежно від робочого графіка компанії;
- можливість одночасної роботи з багатьма клієнтами без зниження якості обслуговування;
- чат-боти можуть надавати індивідуальні рекомендації та адаптуватися до потреб клієнтів;
- зниження витрат на персонал і обслуговування рутинних завдань;
- мінімізація часу очікування відповіді, що підвищує задоволеність клієнтів;
- збір даних клієнтів для вдосконалення продуктів і маркетингових стратегій;

– інтеграція з різними платформами, такими як месенджери, соціальні мережі та веб-сайти.

Окрім цього, чат-боти сприяють покращенню аналітики бізнесу. Взаємодія з клієнтами через чат-ботів дозволяє збирати дані про їхні вподобання, запити та поведінку, що допомагає компаніям краще розуміти своїх споживачів. Інформація може використовуватися для вдосконалення продуктів, персоналізації пропозицій та підвищення ефективності маркетингових стратегій.

Попри численні переваги, використання чат-ботів у сфері електронної комерції має й певні обмеження. Одним із основних проблем є недостатня гнучкість простих ботів, які працюють за заздалегідь визначеними сценаріями. Чат-боти не здатні розуміти складні чи нестандартні запити, що може призводити до незадоволеності клієнтів.

Іншою проблемою є мовний бар'єр. Чат-боти можуть мати труднощі з розумінням специфіки регіональних діалектів, жаргонів або культурних особливостей, що може вплинути на точність відповідей та якість взаємодії з клієнтами. Розв'язання цієї проблеми потребує додаткових витрат на адаптацію бота до різних ринків.

Основні проблеми чат-ботів у сфері електронної комерції:

- чат-боти можуть не впоратися зі складними або нестандартними запитами;
- чат-боти не здатні проявляти емпатію або повністю замінити людське спілкування;
- труднощі з розумінням регіональних діалектів, жаргонів і культурних особливостей;
- загроза витоку особистої інформації клієнтів та необхідність дотримання регуляторних вимог;
- створення багатофункціональних систем потребує значних фінансових вкладень;
- потреба в постійному оновленні;

– можливість технічних збоїв.

Оскільки чат-боти працюють з особистою інформацією клієнтів, існує ризик її витоку або неправомірного використання. Компанії повинні інвестувати в безпеку даних, щоб відповідати регуляторним вимогам і підтримувати довіру клієнтів.

Вартість розробки та підтримки чат-ботів також може бути високою, особливо для багатофункціональних систем, інтегрованих з іншими платформами бізнесу. Малий і середній бізнес іноді стикається з фінансовими обмеженнями, які не дозволяють реалізувати повноцінні рішення.

Крім того, важливим викликом є необхідність постійного оновлення та вдосконалення чат-ботів. Технології розвиваються швидкими темпами, і бізнесу потрібно регулярно інвестувати у модернізацію своїх систем, щоб залишатися конкурентоспроможним.

Створення чат-ботів є складним процесом, який вимагає використання сучасних інструментів, мов програмування, фреймворків та платформ. Вибір технологій залежить від функціональності, яку планується реалізувати, цільової аудиторії та особливостей бізнесу. У цьому пункті розглянемо основні технології, інструменти та мови програмування, які використовуються для розробки чат-ботів.

Мова програмування має важливе значення на початковому етапі розробки.

Ось найпопулярніші мови програмування, які використовуються для створення чат-ботів:

1) Python – одна з найбільш універсальних мов для розробки чат-ботів. Завдяки численним бібліотекам, таким як NLTK, SpaCy та TensorFlow, Python забезпечує можливості обробки природної мови та інтеграцію з алгоритмами штучного інтелекту;

2) JavaScript – використовується для створення чат ботів завдяки популярності серед розробників і підтримці платформою Node.js, яка дозволяє створювати високопродуктивні серверні застосунки;

3) Java – підходить для створення корпоративних чат-ботів, особливо у фінансовому та банківському секторах, завдяки своїй стабільності та масштабованості;

4) PHP – популярна для розробки веб-додатків, часто використовується для інтеграції чат-ботів із веб-сайтами;

5) C# – ідеальна для створення чат-ботів на платформах Microsoft завдяки фреймворку Bot Framework;

6) Ruby – забезпечує швидкий процес розробки для невеликих або середніх проектів завдяки простоті та читабельності коду.

Для спрощення розробки та підвищення ефективності використовуються спеціалізовані фреймворки та платформи:

1) Microsoft Bot Framework – потужний набір інструментів для створення інтерактивних чат-ботів. Підтримує інтеграцію з різними каналами, такими як Skype, Teams, Facebook Messenger;

2) IBM Watson Assistant – забезпечує підтримку складних функціональних можливостей завдяки інтеграції з сервісами штучного інтелекту IBM Watson;

3) Botpress – ще одна платформа з відкритим кодом, орієнтована на створення ботів із простим інтерфейсом;

4) Node.js – це сучасна серверна платформа, яка дозволяє розробляти масштабовані, швидкі та ефективні веб-додатки та сервіси. Ідеально підходить для створення чат-ботів, які повинні швидко реагувати на численні запити користувачів у режимі реального часу;

5) Amazon Lex – платформа від Amazon, що забезпечує інтеграцію з Alexa та підтримує голосових помічників [5].

Особливості та переваги популярних фреймворків зображені в таблиці 1.1.

Таблиця 1.1 – Особливості та переваги популярних фреймворків [6]

Фреймворк/Платформа	Особливості	Переваги
Microsoft Bot Framework	Інтеграція з Microsoft Teams та іншими каналами	Потужний API, інтеграція з Azure
Dialogflow	ШІ-алгоритми від Google	Проста інтеграція, підтримка багатомовності
Rasa	Відкритий вихідний код, локалізоване рішення	Гнучкість, локальне використання
IBM Watson Assistant	Потужні можливості NLP	Висока точність, інтеграція з Watson AI
Botpress	Зручність для новачків	Відкритий код, інтуїтивний інтерфейс
Amazon Lex	Інтеграція з Alexa та іншими сервісами Amazon	Підтримка голосових інтерфейсів

Розробка сучасних чат-ботів часто включає впровадження алгоритмів обробки природної мови (NLP).

До популярних інструментів природної мови належать:

- NLTK (Natural Language Toolkit) – набір бібліотек для обробки тексту, класифікації та аналізу мовлення;
- SpaCy – високопродуктивний інструмент для аналізу тексту, що підтримує різні мови;
- TensorFlow і PyTorch – фреймворки для побудови моделей машинного навчання, які використовуються для аналізу тексту та створення моделей діалогів.

При розробці чат-ботів особливу увагу слід приділяти вибору інструментів для інтеграції з різними каналами комунікації. Сучасні платформи надають можливість створювати ботів, які працюють у популярних месенджерах, таких як Facebook Messenger, WhatsApp, Telegram, а також на веб-сайтах і мобільних додатках. Інтеграція з кількома каналами дозволяє бізнесу охоплювати ширшу аудиторію та забезпечувати зручність для клієнтів. Наприклад, Dialogflow і

Microsoft Bot Framework мають вбудовану підтримку багатоканального підходу, що спрощує підключення чат-ботів до різних платформ [4].

Більшість сучасних платформ, таких як Amazon Lex, IBM Watson Assistant і Microsoft Azure Bot Services, пропонують хмарні рішення для створення та розгортання ботів. Чат-бот забезпечує масштабованість, високу доступність і можливість обробки великої кількості запитів одночасно.

Щодо обробки природної мови, технології NLP (Natural Language Processing) відіграють ключову роль у забезпеченні точності та природності взаємодії. Завдяки таким інструментам, як SpaCy, NLTK, TensorFlow і PyTorch, розробники можуть створювати моделі, які аналізують текст, розпізнають наміри користувача та адаптують відповіді відповідно до контексту [5].

Dialogflow і IBM Watson Assistant використовують власні алгоритми NLP для розпізнавання та класифікації намірів, що значно полегшує розробку ботів із мінімальним досвідом програмування.

Інтеграція з CRM-системами, такими як Salesforce або HubSpot, дозволяє ботам отримувати доступ до історії взаємодій клієнтів, їхніх вподобань і попередніх запитів. Боти в електронній комерції можуть пропонувати персоналізовані рекомендації на основі історії покупок клієнта або його активності на веб-сайті [5].

У розробці чат-ботів важливо також враховувати аспект тестування та підтримки. Тестування чат-ботів має бути ретельним, щоб переконатися в коректності їхньої роботи, точності розпізнавання запитів і відповідності очікуванням клієнтів. Для цього використовуються спеціалізовані інструменти, такі як Botium або Rasa Test, які дозволяють перевіряти ефективність роботи бота в різних сценаріях [5].

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Метою даної кваліфікаційної роботи є розробка чат-бота для сфери електронної комерції з використанням Python, Node.js, який забезпечуватиме ефективну взаємодію з клієнтами, автоматизацію процесів продажу та підтримки, а також підвищення конкурентоспроможності підприємства. Для досягнення цієї мети необхідно вирішити низку завдань, серед яких:

- аналіз існуючих рішень та технологій для розробки чат-ботів, визначення їхніх переваг та недоліків;
- вибір оптимальних інструментів та технологій для реалізації поставленого завдання з урахуванням специфіки бізнес-процесів у сфері електронної комерції;
- проектування архітектури чат-бота, яка забезпечить масштабованість, надійність та зручність використання;
- реалізація функціоналу чат-бота, що включає каталог товарів, кошик покупок, оформлення замовлень, відстеження статусу замовлення, підтримку клієнтів та сповіщення;
- тестування та оптимізація роботи чат-бота, забезпечення його інтеграції з існуючими системами підприємства.

Для реалізації поставлених завдань буде використано мову програмування Python, оскільки вона є зручною для розробки чат-ботів завдяки простоті, універсальності та великій кількості бібліотек для роботи з API та обробки даних.

Бібліотека `python-telegram-bot` буде застосована для роботи з API Telegram, що дозволить створити бота, доступного для широкої аудиторії користувачів цього месенджера. Для зберігання даних про товари, кошик, замовлення та статуси планується використання СУБД MySQL, яка забезпечить надійне та ефективне управління інформацією.

Очікуваним результатом роботи є повнофункціональний чат-бот для соціальної мережі Telegram та забезпечує повний цикл взаємодії з клієнтом у

сфері електронної комерції. Це рішення має сприяти підвищенню ефективності бізнес-процесів компанії, збільшенню рівня задоволеності клієнтів та зміцненню конкурентних позицій на ринку.

В процесі виконання кваліфікаційної роботи будуть досліджені актуальні тенденції у сфері розробки чат-ботів, проаналізовані існуючі інструменти та технології, а також розроблені практичні рекомендації щодо впровадження подібних рішень у бізнес.

Висновки до розділу 1

Чат-боти продемонстрували себе як потужний інструмент для автоматизації бізнес-процесів, покращення взаємодії з клієнтами та підвищення конкурентоспроможності компаній. Їхня здатність забезпечувати швидке реагування на запити, персоналізувати досвід користувачів і оптимізувати витрати робить їх важливим компонентом цифрової трансформації бізнесу.

Аналіз переваг і обмежень чат-ботів показав, що їхнє впровадження має як значні перспективи, так і певні виклики. Серед основних переваг виділено доступність 24/7, швидкість обробки запитів, зниження витрат і можливість інтеграції з різними платформами. Проте існують обмеження, пов'язані з мовними бар'єрами, технічними складнощами, конфіденційністю даних і потребою в постійному оновленні технологій.

Окремо було розглянуто сучасні інструменти для розробки чат-ботів, включаючи мови програмування, фреймворки, бібліотеки та бази даних.

Вибір Python як основної мови програмування обґрунтовано її універсальністю, простотою та підтримкою бібліотек для обробки природної мови та взаємодії з API. Фреймворки, такі як python-telegram-bot, забезпечують зручність розробки та надійність системи.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Основне завдання чат-бота для електронної комерції з використанням Python, Node.js – забезпечення користувачів можливістю перегляду каталогу товарів, додавання товарів у кошик, оформлення замовлення та відстеження статусу покупки [7].

Оскільки бот взаємодіє з базою даних, необхідно забезпечити створення відповідних таблиць у базі даних, зокрема для каталогу товарів, кошика покупця та історії замовлень.

З точки зору користувацького досвіду, бот має бути зручним у використанні, а тому команди повинні бути зрозумілими та логічними. Наприклад, команда `catalog` дозволяє переглянути категорії товарів, а команда `cart` – перевірити вміст кошика.

Для керування замовленнями передбачені команди `orders` та `purchase_history`. Використання інтерактивних кнопок через `CallbackQueryHandler` допоможе зробити інтерфейс більш зручним, наприклад, для збільшення або зменшення кількості товарів у кошику чи підтвердження замовлення.

Важливою вимогою до функціоналу є можливість пошуку товарів, що передбачає реалізацію обробки користувацьких запитів для знаходження відповідних пропозицій у каталозі.

Адміністратор повинен мати доступ до розширених функцій, таких як додавання категорій товарів та управління статусами замовлень. Для цього система перевіряє роль користувача перед виконанням відповідних команд. Важливо також забезпечити коректну обробку помилок, наприклад, якщо користувач вводить некоректні дані під час додавання товару або замовлення.

З точки зору архітектури, застосовується модульний підхід із розподілом відповідальності між різними модулями: каталог товарів, кошик, замовлення та управління користувачами.

Оскільки бот працює в режимі реального часу, важливо забезпечити стабільне з'єднання з базою даних і коректну обробку одночасних запитів від користувачів.

Telegram-бот працює на основі Python і бібліотеки `python-telegram-bot`, що дозволяє реалізувати обробку команд та інтерактивних повідомлень. Основні функціональні модулі включають каталог товарів, кошик покупця, систему замовлень та управління користувачами.

Система має передбачати використання бази даних MySQL для збереження інформації про товари, категорії, користувачів, кошики та замовлення.

Робота з базою здійснюється через SQL-запити, що дозволяє виконувати операції з отримання, оновлення та видалення даних у реальному часі.

Компонент управління бізнес-логікою буде розділений на окремі модулі, такі як:

- каталог товарів, який забезпечує можливість пошуку та перегляду категорій;
- кошик користувача, який дозволяє додавати або видаляти товари;
- модуль замовлень, який відповідає за їх створення, зміну статусу та перегляд історії покупок.

Взаємодія між модулями здійснюється через виклики функцій, що дозволяє легко розширювати можливості бота без значного переписування коду.

Використання інтерактивних кнопок дозволяє покращити зручність взаємодії з ботом, надаючи можливість змінювати параметри товарів у кошику або оформлювати замовлення без необхідності введення складних команд.

Для стабільної роботи системи необхідно передбачити обробку помилок, перевірку дій та забезпечення безперебійної взаємодії між ботом і базою даних.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Чат-бот буде написаний на мові програмування Python, оскільки вона має широкий набір бібліотек для роботи з Telegram API, а також ефективні засоби для взаємодії з базами даних. В якості бази даних буде використана MySQL, яка дозволяє зберігати велику кількість даних, має потужний механізм транзакцій і забезпечує високу швидкість обробки запитів.

Використання MySQL дає можливість легко виконувати складні вибірки, зберігати взаємопов'язані дані та підтримувати велику кількість одночасних підключень. Взаємодія між Python-додатком і MySQL буде здійснюватися через драйвер MySQL Connector, що дозволить абстрагувати роботу з базою та зменшити ризик SQL-ін'єкцій [8].

Робота бота базується на використанні бібліотеки python-telegram-bot, яка підтримує як синхронну, так і асинхронну роботу з API Telegram. Обрано саме цю бібліотеку через її зручність у створенні обробників команд, роботу з callback-запитами та підтримку інтерактивних кнопок.

Завдяки широкому вибору бібліотек і модулів, зокрема node-telegram-bot-api, розробка Telegram-ботів на Node.js стала простою та доступною навіть для початківців. Ця бібліотека надає зручний інтерфейс для взаємодії з Telegram Bot API, дозволяючи швидко реалізувати базовий функціонал бота: обробку повідомлень, команд, відправку відповідей, інтеграцію з базами даних та зовнішніми сервісами.

Node.js також забезпечує гнучкість у розробці, можна легко додавати нові функції, підключати сторонні API, масштабувати систему під зростаючі навантаження. Це робить платформу оптимальним вибором для реалізації чат-ботів, які мають обробляти великий потік запитів, наприклад, у сфері електронної комерції.

Node.js дозволяє створювати ефективних Telegram-ботів для електронної комерції, забезпечуючи високу продуктивність, масштабованість та зручність інтеграції з іншими системами. Використання цієї платформи разом із Telegram API відкриває широкі можливості для автоматизації бізнес-процесів і підвищення якості клієнтського сервісу.

Архітектурно система буде поділена на кілька модулів, що дозволить спростити підтримку та майбутнє розширення додатка.

Оскільки Telegram-бот є основним засобом взаємодії з користувачем, важливо забезпечити його швидкодію та коректну роботу з базою даних. Для цього буде реалізовано механізм кешування запитів, що зменшить навантаження на сервер бази даних. Крім того, необхідно забезпечити безпеку збережених даних, тому в роботі з MySQL будуть використовуватися параметризовані запити, що унеможливають SQL-ін'єкції [8].

Серверна частина буде працювати як окремий процес, що отримуватиме повідомлення через Telegram API та оброблятиме їх згідно з визначеною логікою. Всі запити до бази даних будуть виконуватися асинхронно для уникнення блокувань і затримок у відповіді бота. Використання MySQL дозволить створити структуровану схему з окремими таблицями для користувачів, товарів, кошика та замовлень, що забезпечить високу продуктивність навіть при зростанні кількості користувачів.

Для зручності адміністрування передбачено механізм перевірки ролей користувачів. При надсиланні команди, що вимагає прав адміністратора, бот перевірятиме статус користувача в базі даних та, у разі відсутності необхідних прав, відхилятиме запит. Після створення замовлення воно може отримати статус «очікує підтвердження», після чого адміністратор змінюватиме його на «виконується» або «відправлено».

Буде передбачено перевірку можливих сценаріїв використання, таких як додавання товарів до кошика, оформлення замовлення, оновлення статусів та

адміністрування каталогу. Оскільки система працюватиме в реальному часі, необхідно забезпечити її стійкість до збоїв і неправильного введення даних.

Користувач може спробувати додати товар до кошика, якого вже немає в наявності, або відправити некоректну команду. У таких випадках бот повинен відповідати відповідними повідомленнями та запобігати помилкам.

Проектування структури бази даних для електронної комерції передбачає створення кількох основних таблиць, що забезпечують збереження інформації про користувачів, товари, категорії, замовлення та кошик покупця. Основна мета проектування бази даних – створити ефективну, масштабовану та зручну систему, що дозволяє швидко отримувати дані, обробляти замовлення та взаємодіяти з користувачами через Telegram-бота [9].

Таблиця `users`, яка міститиме інформацію про всіх користувачів системи. Основні колонки цієї таблиці включають унікальний ідентифікатор користувача, який використовується як первинний ключ, `chat_id` для зв'язку з Telegram, ім'я користувача та його роль у системі. Остання колонка дозволяє розмежовувати звичайних користувачів та адміністраторів, які мають розширені можливості, наприклад, додавання нових товарів чи зміну статусу замовлень.

Наступною таблицею є `categories`, яка містить список категорій товарів. Вона складається з унікального ідентифікатора категорії, її назви та опису. Категорії дозволяють структурувати каталог товарів, спрощуючи навігацію для користувачів.

Таблиця `products` зберігатиме дані про товари, що доступні в магазині. Вона міститиме унікальний ідентифікатор товару, його назву, опис, ціну, кількість у наявності та посилання на відповідну категорію через зовнішній ключ.

Для реалізації можливості додавання товарів у кошик використовується таблиця `cart_items`. Вона містить унікальний ідентифікатор запису, ідентифікатор користувача, ідентифікатор товару та кількість одиниць товару, доданих у кошик.

Щоб зберігати інформацію про здійснені замовлення, створена таблиця `orders`. Вона містить унікальний ідентифікатор замовлення, ідентифікатор користувача, статус замовлення та дату його створення. Статус замовлення відображає поточний стан покупки, наприклад, «оформлено», «в обробці», «відправлено» або «доставлено».

Оскільки одне замовлення може містити кілька різних товарів, необхідно створити допоміжну таблицю `order_items`, що містить детальну інформацію про кожну позицію в замовленні.

Вона включає унікальний ідентифікатор запису, ідентифікатор замовлення, ідентифікатор товару, кількість замовленого товару та його ціну на момент покупки. Наявність цієї таблиці дозволяє зберігати історію покупок та відстежувати, які товари входили в кожне окреме замовлення.

Всі таблиці пов'язані між собою через зовнішні ключі, що забезпечує логічну цілісність даних. Таблиця `products` має зовнішній ключ `category_id`, що вказує на відповідну категорію товару у таблиці `categories`. Таблиця `cart_items` містить посилання на користувача та товар, що дозволяє зберігати товари в кошику конкретного користувача.

Аналогічно, `orders` пов'язана з `users`, що дозволяє відстежувати, кому належить кожне замовлення. Таблиця `order_items` зв'язана з `orders` і `products`, що дозволяє чітко визначити, які товари були замовлені в межах кожного окремого замовлення.

Щоб забезпечити швидкість доступу до даних, необхідно правильно налаштувати індекси. Первинні ключі автоматично створюють унікальні індекси, що дозволяють швидко здійснювати пошук за ними. Однак для покращення продуктивності запитів необхідно також створити додаткові індекси на зовнішніх ключах, наприклад, `user_id` у таблицях `cart_items` та `orders`, `product_id` у `order_items` та `cart_items`, а також `category_id` у `products`.

З точки зору масштабованості, запропонована структура дозволяє з легкістю розширювати функціонал системи. Якщо згодом буде потрібно додати

нові категорії товарів або змінити їх опис, це можна зробити без необхідності змінювати інші таблиці. Також можна легко розширювати асортимент товарів, додаючи нові позиції у `products`. Оскільки статуси замовлень можуть змінюватися в процесі їх обробки, цей механізм реалізовано у `orders`, де адміністратор може оновлювати статус замовлення в міру його виконання.

Перед розгортанням бази даних необхідно ретельно протестувати її роботу, включаючи перевірку зв'язку між таблицями, коректність виконання SQL-запитів і продуктивність при збільшенні навантаження. Використання індексів і оптимізованих запитів дозволяє значно зменшити час виконання операцій, що критично важливо для стабільної роботи додатка в реальному часі. Завдяки реляційній структурі бази даних і чітко визначеним зв'язкам можна бути впевненими в коректності та узгодженості даних, що дозволить користувачам отримувати актуальну інформацію про товари, замовлення та їх статуси.

Висновки до розділу 2

Розробка чат-бота для електронної комерції в Telegram передбачає детальне проектування архітектури системи, вибір технологій і побудову структури бази даних. У результаті аналізу вимог було визначено основні функціональні можливості бота, серед яких перегляд каталогу товарів, можливість додавання товарів у кошик, оформлення замовлення та відстеження його статусу.

Для реалізації цих функцій система має включати механізм управління користувачами, що дозволяє розрізняти звичайних покупців та адміністраторів, які мають право додавати нові товари, редагувати каталог і змінювати статуси замовлень. Оскільки бот працює в режимі реального часу, важливо забезпечити його ефективну взаємодію з базою даних і можливість обробки декількох запитів одночасно.

Для забезпечення гнучкості та простоти розробки буде обрана реляційна база даних MySQL, яка дозволяє ефективно зберігати інформацію про товари, категорії, користувачів, замовлення та історію покупок. Робота з базою даних здійснюватиметься через SQL-запити.

Вибір інструментів для реалізації базується на доцільності використання Python, що має потужну екосистему бібліотек для створення Telegram-ботів і взаємодії з базами даних. Бібліотека `python-telegram-bot` забезпечує зручний спосіб обробки команд, використання інтерактивних кнопок і роботи з `callback`-запитами. Також важливою частиною розробки буде реалізація механізмів кешування, що дозволить зменшити навантаження на базу даних і покращити швидкодію додатка. Окрему увагу буде приділено безпеці, включаючи використання налаштованих запитів, механізмів перевірки ролей користувачів і обмеження доступу до адміністративних команд.

Після реалізації основної логіки на Python, для окремих компонентів системи використовується Node.js. Node.js ідеально підходить для обробки великої кількості одночасних запитів, що особливо актуально для Telegram-ботів із високим навантаженням. Для створення ботів застосовується бібліотека `telegraf`, яка забезпечує зручний інтерфейс для роботи з Telegram API, підтримує інтерактивні елементи та `callback`-запити.

Node.js також використовується для організації кешування даних із використанням Redis, що дозволяє зменшити кількість звернень до основної бази даних та підвищити швидкодію сервісу. Для взаємодії з базами даних використовуються сучасні ORM та драйвери, які забезпечують безпечну та ефективну роботу з даними.

Окрему увагу приділено питанням безпеки: реалізовано перевірку прав доступу користувачів, захист від несанкціонованого виконання адміністративних команд. Таким чином, Node.js дозволяє створювати масштабовані, швидкі та безпечні сервіси у складі загальної архітектури проекту.

Структура бази даних розроблена таким чином, щоб забезпечити оптимальне зберігання та швидкий доступ до необхідних даних.

РОЗДІЛ 3

РОЗРОБКА ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦІЇ ОБСЛУГОВУВАННЯ КЛІЄНТІВ У СФЕРІ ЕЛЕКТРОННОЇ КОМЕРЦІЇ З ВИКОРИСТАННЯМ PYTHON, NODE.JS

3.1 Практична реалізація об'єкта проєктування

Основні модулі програми включають реалізацію команд, обробку кнопок зворотного зв'язку, взаємодію з базою даних, керування кошиком, оформлення замовлень, пошук товарів, адміністрування каталогу та замовлень [10].

Усі компоненти побудовані на основі Telegram Bot API з використанням бібліотеки `python-telegram-bot`, що дозволяє керувати структурою повідомлень і взаємодією з користувачем.

Першим функціональним блоком є реалізація команди `start`, що відіграє роль точки входу до системи.

Користувачу виводиться текст з описом доступних команд, серед яких:

- перегляд каталогу;
- доступ до кошика;
- оформлення замовлень;
- перевірка історії покупок;
- пошук товарів;
- доступ до адміністративних функцій.

Команда `catalog` надає можливість переглянути категорії товарів (ліст. 3.1).

Лістинг 3.1 – Перегляд каталогу товарів `catalog`

```
async def catalog(update, context):
    keyboard = show_categories()
    await update.message.reply_text("Оберіть категорію:",
    reply_markup=keyboard)
```

кінець лістингу 3.1

Команда `cart` надає можливість переглянути кошик (ліст. 3.2).

Лістинг 3.2 – Перегляд кошика cart

```

async def cart(update, context):
    chat_id = update.message.chat_id
    cart_text, keyboard = show_cart(chat_id)
    if keyboard:
        await update.message.reply_text(cart_text,
reply_markup=keyboard)
    else:
        await update.message.reply_text(cart_text)

```

кінець лістингу 3.2

Команда orders надає можливість переглянути замовлення (ліст. 3.3).

Лістинг 3.3 – Перегляд замовлень orders

```

async def orders(update, context):
    chat_id = update.message.chat_id
    orders_text = show_orders(chat_id)
    await update.message.reply_text(orders_text)

```

кінець лістингу 3.3

Команда add_catalog дозволяє адміністратору створювати нові категорії, передаючи назву та опис через аргументи команди (ліст. 3.4).

Лістинг 3.4 – Додавання категорій add_catalog

```

async def add_catalog(update, context):
    chat_id = update.message.chat_id
    user_id, role = get_or_create_user_id(chat_id)
    if role != "admin":
        await update.message.reply_text("У вас немає прав для виконання цієї команди.")
        return

    if len(context.args) < 2:
        await update.message.reply_text("Формат: /add_catalog Назва Опис")
        return

    category_name = context.args[0]
    category_description = " ".join(context.args[1:])
    if add_category(category_name, category_description):
        await update.message.reply_text(f"Категорію '{category_name}'

```

кінець лістингу 3.4

Команда `add_products`, у свою чергу, забезпечує додавання нових товарів до бази даних з відповідною прив'язкою до категорії, з перевіркою усіх введених параметрів. Завдяки цьому адміністратор має змогу швидко наповнювати каталог новими позиціями, не використовуючи сторонній інтерфейс, а лише командний рядок бота (ліст. 3.5).

Лістинг 3.5 – Додавання товарів `add_products`

```

async def add_products(update, context):
    name = context.args[0]
    price = float(context.args[-4])
    stock = int(context.args[-3])
    category_id = int(context.args[-2])
    sku = context.args[-1]
    description = " ".join(context.args[1:-4])
    if add_product_to_db(name, description, price, stock,
category_id, sku):
        await update.message.reply_text(f"Товар '{name}' успішно
додано.")

```

кінець лістингу 3.5

Для користувачів передбачено можливість перевірки історії покупок через команду `purchase_history`. Бот отримує статус замовлення з бази даних, групує його і виводить у структурованому вигляді з відображенням кожного товару, кількості та дати створення (ліст. 3.6).

Лістинг 3.6 – Історія покупок `purchase_history`

```

async def purchase_history(update, context):
    completed_orders = get_purchase_history(user_id)
    if not completed_orders:
        await update.message.reply_text("У вас немає завершених
замовлень.")
    return
    for order_id, data in grouped_orders.items():

        history_text += f"Замовлення #{order_id} - {data['status']}..."

```

кінець лістингу 3.6

Функціональність оновлення статусу замовлень реалізована у вигляді окремої команди `update_order_status` (ліст. 3.7), що дозволяє адміністраторам змінювати статус виконання замовлення (наприклад, «в обробці», «відправлено», «доставлено» тощо). При введенні команди бот очікує два аргументи – ID замовлення та новий статус. У випадку помилок, таких як некоректний формат або недостатні права, бот надає відповідні підказки.

Лістинг 3.7 – Оновлення статусу замовлення `update_order_status`

```

async def update_order_status_handler(update, context):
    if len(context.args) != 2:
        await update.message.reply_text("Формат: /update_order_status [id] [статус]")
        return
    order_id = int(context.args[0])
    new_status = context.args[1]
    result = update_order_status(order_id, new_status)
    await update.message.reply_text(result)

```

кінець лістингу 3.7

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

У процесі розробки чат-бота для автоматизації обслуговування клієнтів у сфері електронної комерції особливу увагу було приділено тестуванню та налагодженню системи, що дозволило забезпечити її надійність, стабільність і відповідність функціональним вимогам.

Для реалізації періодичних завдань у проекті було обрано використання бібліотеки APScheduler (Advanced Python Scheduler), яка є гнучким і надійним інструментом для планування фонових задач у Python.

Бібліотека дозволяє створювати різні типи завдань: одноразові, повторювані з певним інтервалом, а також завдання, що виконуються у визначений час. Використання APScheduler дозволяє зберігати інформацію про заплановані завдання у базі даних.

Одним із перших завдань, реалізованих у рамках цієї підсистеми, стало щоденне формування звіту про нові замовлення. Завдання виконується кожного дня о заданій годині (наприклад, о 9:00 ранку) і надсилає адміністратору список нових замовлень, створених за останні 24 години [11].

Механізм дозволяє адміністраторам швидко реагувати на дії користувачів без необхідності постійно перевіряти систему вручну.

Реалізовано завдання, що перевіряє всі замовлення зі статусом «в очікуванні» протягом тривалого часу (наприклад, понад 48 годин) і автоматично змінює їхній статус на «відхилено» або надсилає користувачу нагадування з проханням підтвердити або завершити оформлення [11].

Ще одним прикладом використання періодичних завдань є автоматичне очищення кошиків користувачів, які залишалися неактивними певний час.

Механізм дозволяє підтримувати актуальність інформації у базі даних та звільняти ресурси, особливо у випадках, коли користувачі додають товари до кошика, але не здійснюють покупку.

У реалізованому боті це завдання запускається кожні 12 годин і перевіряє дати останньої активності користувача. Якщо користувач не здійснював дій протягом більше ніж 3 діб, його кошик автоматично очищається [12].

Також була реалізована автоматична генерація резервних копій бази даних. Завдання запускається щодоби вночі й створює повну копію структури та вмісту таблиць, зберігаючи її в хмарному середовищі (у хмарному сховищі Dropbox або Google Drive, інтеграція з якими реалізується через API) [13].

Функція аналітики – раз на тиждень бот може обчислювати кількість нових користувачів, загальну кількість замовлень, популярні категорії товарів та обсяг виручки, надсилаючи цю статистику адміністратору для прийняття управлінських рішень. Для цього дані беруться з різних таблиць і формуються у вигляді текстового звіту, який надсилається у повідомленні в Telegram.

З технічного погляду, реалізація підсистеми періодичних завдань включає створення окремого модуля (`scheduler.py`), у якому ініціалізується об'єкт

планувальника BackgroundScheduler. Кожне завдання реєструється через функцію `add_job`, де задається метод, що виконується, тип тригера (інтервальний або одноразовий), а також відповідні параметри [14].

Для збереження стабільності системи кожне завдання виконується у try-ехсерт блоці з веденням перевірок, що дозволяє фіксувати всі помилки та винятки.

Всі завдання виконуються у фоновому режимі, що не перешкоджає основному циклу бота. Бібліотека APScheduler сумісна з асинхронною обробкою, тому всі фонові задачі можуть працювати паралельно з основною логікою Telegram-бота, не блокуючи роботу з користувачами.

Важливою частиною реалізації є підтримка callback-подій, які дозволяють оновлювати вміст повідомлень у відповідь на дії користувача, не створюючи нових повідомлень.

При натисканні кнопки «Назад до товарів» бот завантажує список доступних категорій товарів, а кнопка «Додати до кошика» додає обраний товар у базу, що відповідає кошику користувача (рис. 3.1).

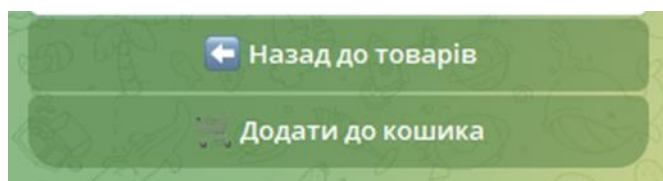


Рисунок 3.1 – Кнопка назад до товарів та додати до кошика

Функціональність оновлення статусу замовлень реалізована у вигляді окремої команди `update_order_status`, що дозволяє адміністраторам змінювати статус виконання замовлення (наприклад, «в обробці», «відправлено», «доставлено» тощо). При введенні команди бот очікує два аргументи – ID замовлення та новий статус. У випадку помилок, таких як некоректний формат або недостатні права, бот надає відповідні підказки (рис. 3.2).

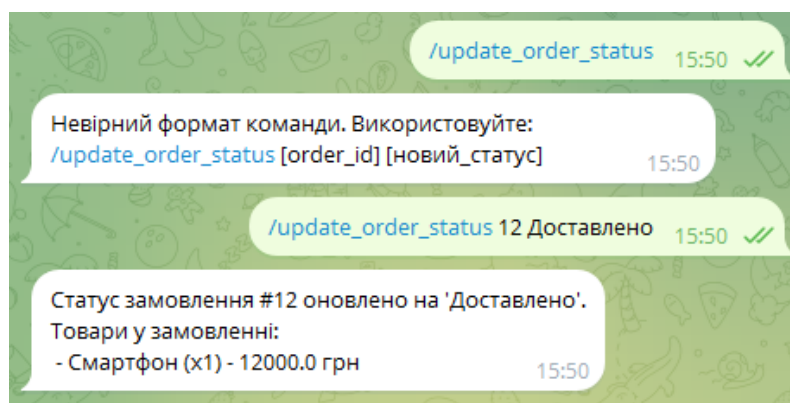


Рисунок 3.2 – Команда `update_order_status`

Функціонал бота, розробленого для управління інтернет-магазином у Telegram, може бути суттєво розширений відповідно до потреб користувачів і адміністраторів. Поточна реалізація охоплює основні можливості: перегляд каталогу товарів, додавання нових категорій і товарів, роботу з кошиком, оформлення замовлень та оновлення їх статусу. Однак із врахуванням сучасних тенденцій електронної комерції та зручності взаємодії з користувачем можна запропонувати низку напрямів для подальшого вдосконалення системи [15].

Перш за все, актуальною буде реалізація системи фільтрації товарів за різними параметрами, такими як ціна, наявність на складі, популярність, рейтинг або за ключовими словами в описі.

Кожен користувач після покупки міг би залишити короткий коментар або оцінити товар за п'ятибальною шкалою. Зібрані оцінки дозволили б формувати список найпопулярніших товарів, які можуть відображатися окремо.

Також можна реалізувати бонусну систему або програму лояльності. Користувач за кожну покупку буде отримувати бали, які згодом можна обміняти на знижку або подарунок. Система може автоматично підраховувати накопичені бонуси і пропонувати варіанти їх використання при оформленні наступного замовлення.

3.3 Розробка бази даних

База даних створена для підтримки роботи інтернет-магазину. Вона дозволяє керувати товарами, категоріями, кошиком та замовленнями. Основна мета структури бази даних – забезпечити зберігання й обробку інформації про товари, клієнтів та їх покупки з можливістю фільтрації, аналізу та перегляду замовлень.

Основні таблиці розроблюваної бази даних зображенні нижче в таблиці 3.1.

Таблиця 3.1 – Основні таблиці розроблюваної бази даних [10]

Таблиця	Опис
users	Зберігає інформацію про користувачів, включаючи їхні ідентифікатори в месенджері, імена користувачів і ролі (користувач або адміністратор)
categories	Містить категорії товарів (наприклад, електроніка, одяг, книги) з описом кожної
products	Містить дані про товари: назва, опис, ціна, кількість на складі, а також категорію, до якої належить товар
cart_items	Зберігає товари, які користувач додав до свого кошика, включаючи кількість
orders	Містить інформацію про замовлення: користувача, дату створення та поточний статус (наприклад, оформлено, доставлено)
order_items	Містить перелік товарів, які входять у конкретне замовлення, включаючи їх кількість та ціну на момент покупки

Основний з елементів бази даних є таблиця користувачів users, яка пов'язана з таблицею замовлень orders у співвідношенні один до багатьох, що означає, що один користувач може створити багато замовлень.

База даних має визначену реляційну структуру, що забезпечує зберігання та обробку інформації, необхідної для функціонування інтернет-магазину. Вона складається з кількох пов'язаних між собою таблиць, які реалізують основну бізнес-логіку платформи [10].

Користувач може мати кілька товарів у кошику, що реалізовано через зв'язок між таблицями users та cart_items, також у співвідношенні «один до багатьох» [16]. Таблиця cart_items, у свою чергу, пов'язана з таблицею products,

адже один товар може бути доданий у кошик багатьма користувачами – знову ж таки, зв'язок один до багатьох.

ER-діаграма зображена на рисунку 3.3.

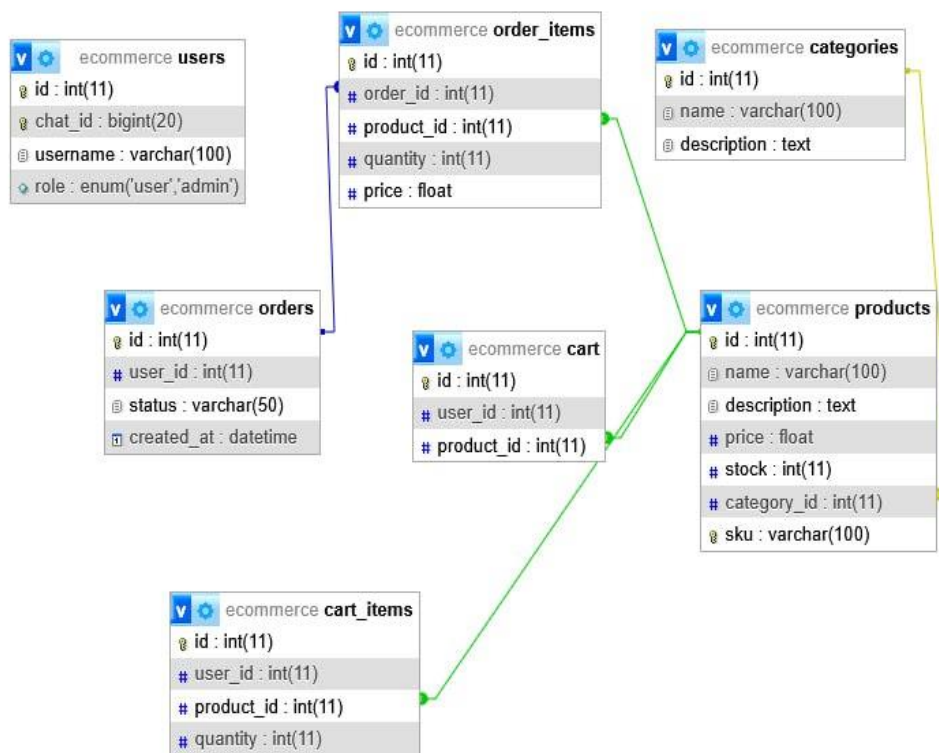


Рисунок 3.3 – ER-діаграма бази даних [10]

Таблиця products пов'язана і з таблицею order_items, оскільки один товар може входити до складу багатьох замовлень. Самі замовлення, представлені в таблиці orders, можуть включати кілька товарів, про що свідчить зв'язок між orders та order_items – також один до багатьох. Кожен товар належить до певної категорії, а кожна категорія може включати багато товарів, що забезпечується зв'язком між таблицями products та categories.

Усі ці зв'язки побудовані за принципом «один до багатьох», що є типовим для реляційних баз даних, оскільки дозволяє ефективно організувати зберігання великої кількості пов'язаних даних із забезпеченням цілісності та оптимізації запитів [17].

3.4 Розробка інсталяційного пакета

Розглянемо покрокову взаємодію користувача з Telegram-ботом.

Спочатку користувач надсилає команду start, після чого бот виводить повідомлення з переліком доступних команд. Приклад взаємодії користувача з Telegram-ботом показаний на рисунку 3.4.

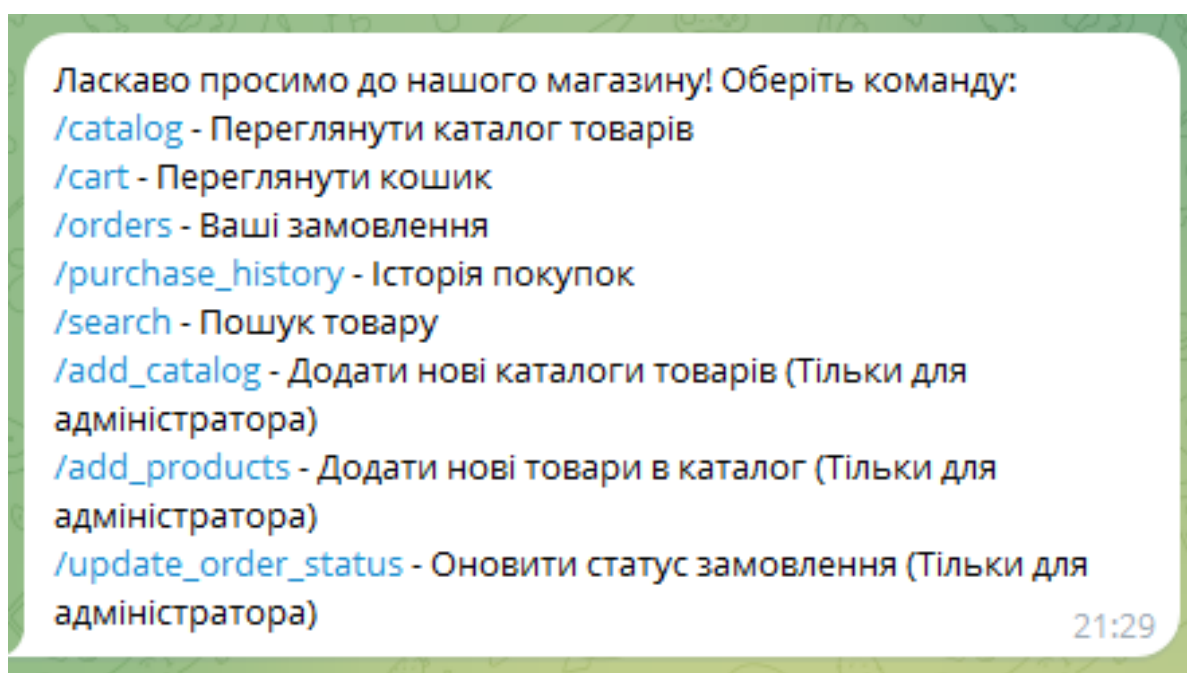


Рисунок 3.4 – Меню чат-бота

Далі пропонують обрати відповідь бота на введення команди catalog. Користувачу надсилається перелік категорій товарів у вигляді інтерактивних кнопок. У даному прикладі доступні категорії – «Електроніка», «Одяг», «Книги» та «Продукти». Натискання на одну з кнопок запускає наступний етап взаємодії.

Коли користувач вводить команду catalog, йому пропонують обрати одну з категорій товарів. Вони відображаються у вигляді інтерактивних кнопок з назвами, наприклад: «Електроніка», «Одяг», «Книги», «Продукти». Користувач має змогу натиснути на одну з них для перегляду товарів у цій категорії. Приклад введення команди catalog показаний на рисунку 3.5.

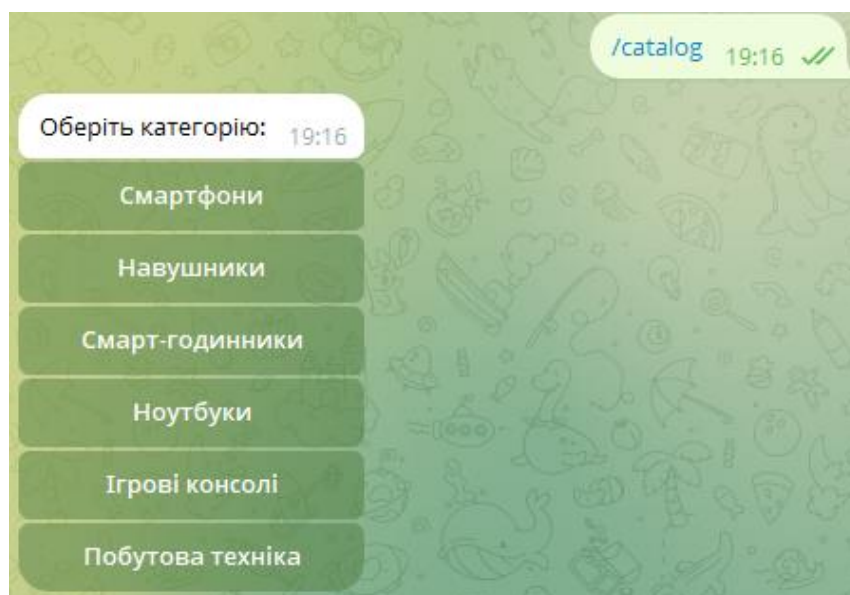


Рисунок 3.5 – Вибір категорії

Після натискання на категорію відображається список товарів, що до неї належать. Товари подаються у вигляді кнопок із назвами і зображені на рисунку 3.6.



Рисунок 3.6 – Вибір товару

При виборі конкретного товару, наприклад «Навушники», користувачу відображається детальна інформація: назва товару, короткий опис, ціна, кількість на складі. Під цим повідомленням розміщено дві кнопки – одна дозволяє повернутись до попереднього списку товарів, а інша – додати товар до кошика. Інформація про товар відображена на рисунку 3.7.

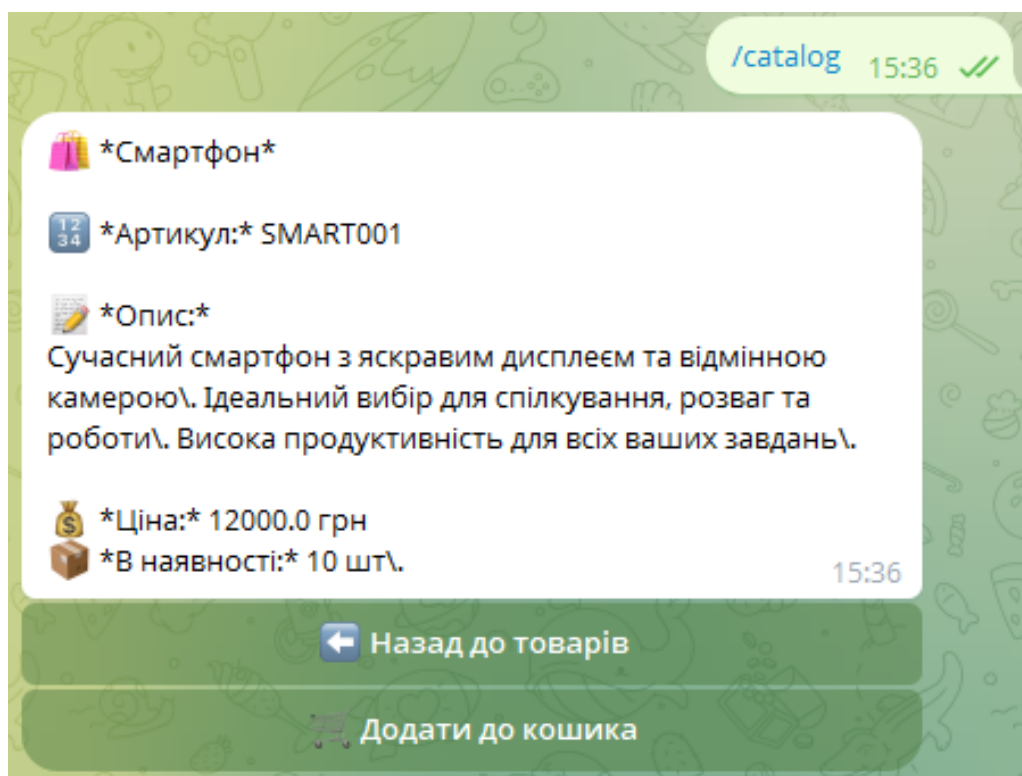


Рисунок 3.7 – Інформація про товар

Після натискання кнопки додавання до кошика бот надсилає коротке повідомлення з підтвердженням про те, що товар було успішно додано. Приклад додавання товару до кошика зображений на рисунку 3.8.



Рисунок 3.8 – Додавання товару до кошика

Далі, при введенні команди cart, користувачу відображається поточний вміст кошика. Для кожного товару зазначається його назва, кількість і сума. Також під повідомленням розташовані кнопки для керування товарами: зменшення або збільшення кількості, повне видалення товару з кошика, а також кнопка для оформлення замовлення. Вміст кошика зображений на рисунку 3.9.

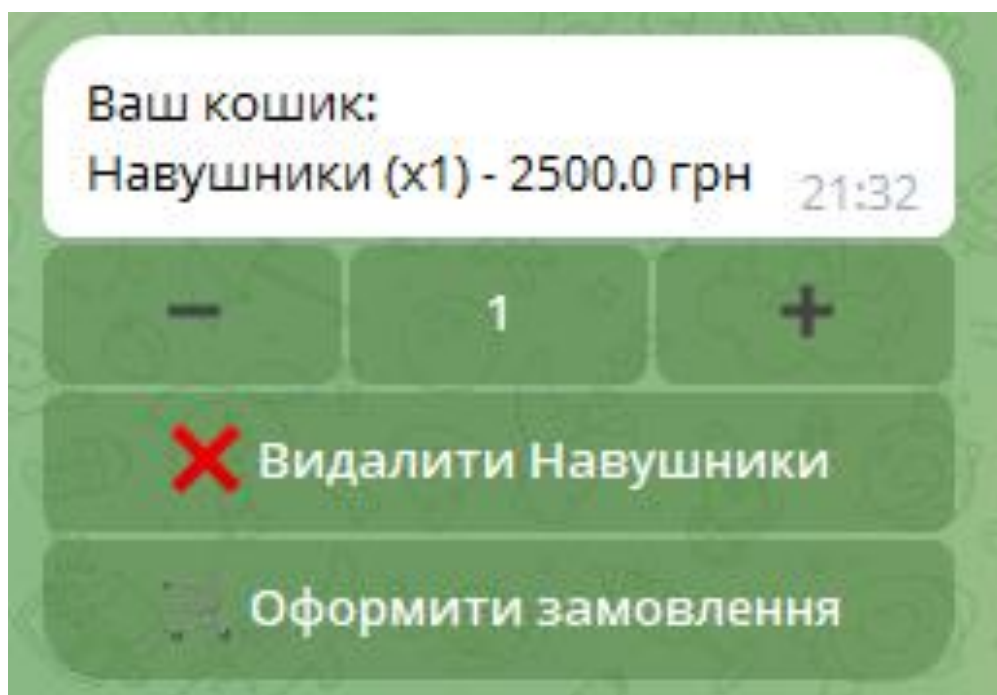


Рисунок 3.9 – Додавання товару до кошика

Після натискання кнопки оформлення замовлення система створює нове замовлення. У відповідь користувач отримує повідомлення з номером щойно сформованого замовлення, що свідчить про успішне завершення операції. Створення нового замовлення зображене на рисунку 3.10.

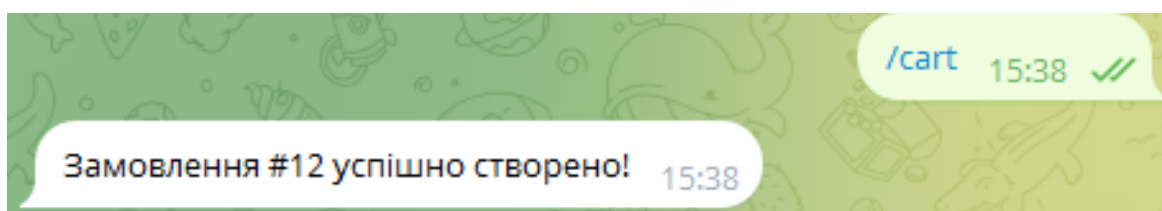


Рисунок 3.10 – Створення нового замовлення

Далі, за допомогою команди `orders`, користувач отримує список усіх своїх активних замовлень. У кожному замовленні вказано номер, статус, дата створення, ім'я користувача та перелік замовлених товарів разом із кількістю та вартістю кожного з них. Приклад введення команди `orders` показаний на рисунку 3.11.

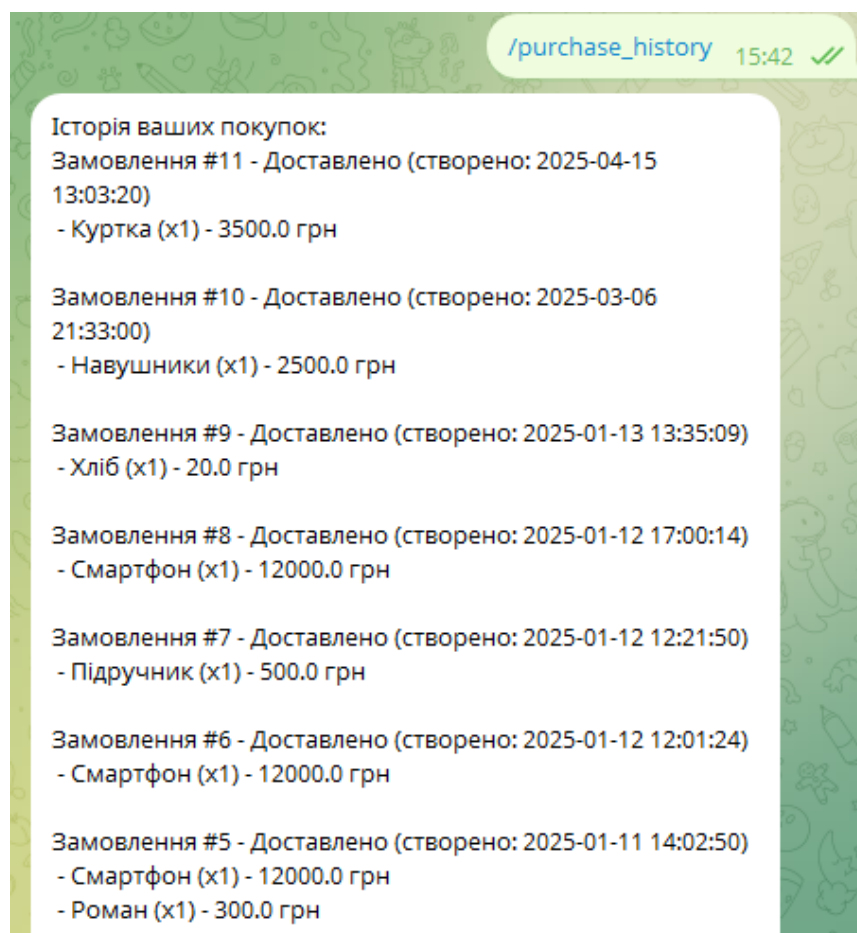


Рисунок 3.11 – Команда orders

Якщо користувач вводить команду `purchase_history`, йому відображаються всі завершені замовлення зі статусом «Доставлено». У кожному такому записі зазначено дату, перелік придбаних товарів та суму замовлення. Приклад введення команди `purchase_history` показаний на рисунку 3.12.

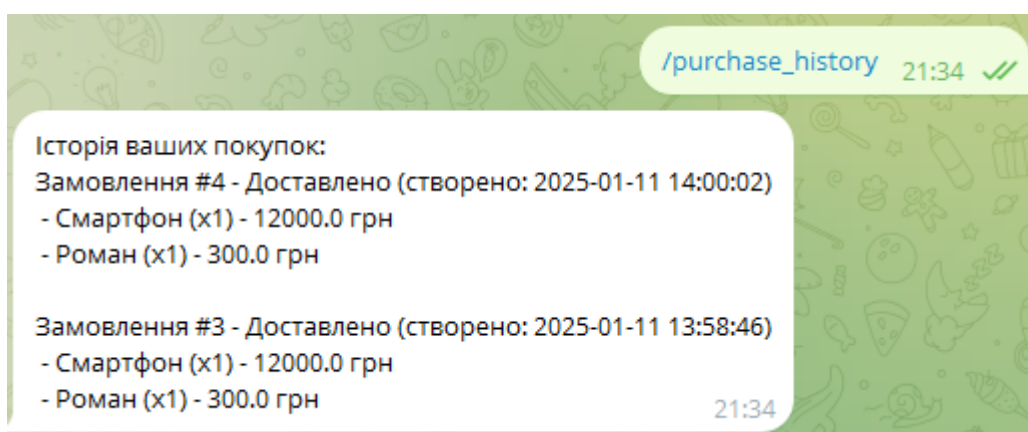


Рисунок 3.12 – Команда purchase_history

Бот підтримує функцію пошуку. При введенні, наприклад, `search смартфон`, користувачу пропонується список знайдених товарів, що відповідають пошуковому запиту. Приклад введення команди `search` показаний на рисунку 3.13.

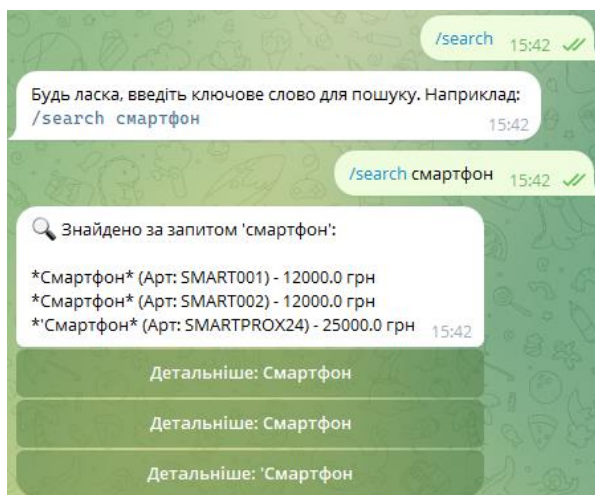


Рисунок 3.13 – Команда `search`

Також бот підтримує можливість додавання нових категорій товарів. Для цього адміністратор вводить команду у форматі `add_catalog`, вказуючи назву категорії та її опис. Приклад використання команди `add_catalog` показаний на рисунку 3.14.



Рисунок 3.14 – Команда `add_catalog`

Адміністратор може додавати нові товари до існуючих категорій. Для цього використовується команда `add_products`, після якої послідовно зазначаються назва товару, опис, ціна, кількість та ідентифікатор категорії. Приклад додавання товару наведено на рисунку 3.15.

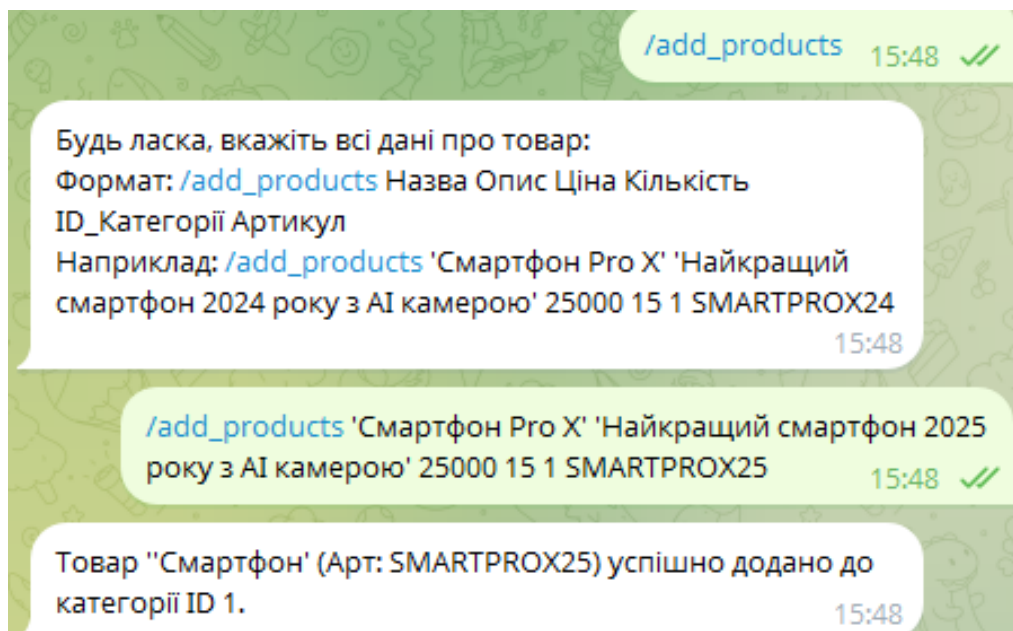


Рисунок 3.15 – Приклад додавання замовлення

Бот підтримує оновлення статусу замовлення. Функцію можуть використовувати лише адміністратори за допомогою команди `update_order_status`, вказуючи номер замовлення та новий статус. Приклад оновлення статусу замовлення показаний на рисунку 3.16.

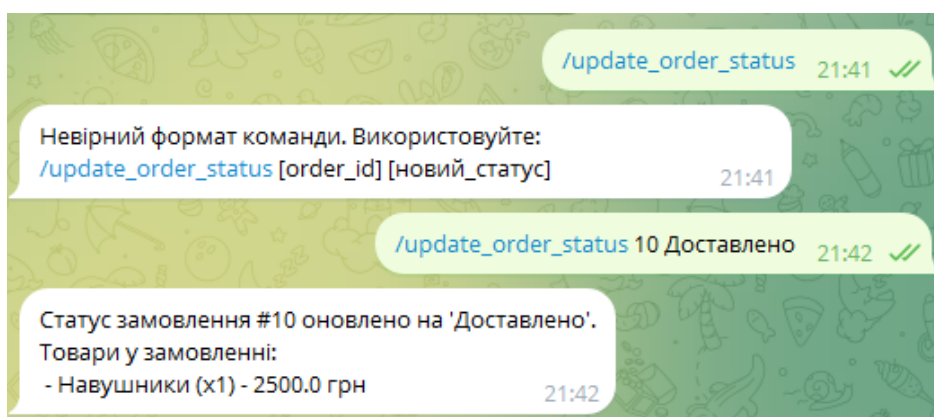


Рисунок 3.16 – Оновлення статусу замовлення

Висновки до розділу 3

У третьому розділі було здійснено практичну реалізацію чат-бота для автоматизації обслуговування клієнтів у сфері електронної комерції, розроблено структуру бази даних, проведено тестування та налагодження інформаційно-комп'ютерної системи, а також сформовано інсталяційний пакет для розгортання програмного забезпечення. В процесі розробки особливу увагу приділено вибору оптимальних інструментів і технологій, що забезпечило ефективну інтеграцію Python, Node.js та Telegram API для реалізації функціоналу чат-бота.

Проведене тестування підтвердило відповідність розробленої системи функціональним вимогам, її стабільність та надійність у роботі під реальними навантаженнями. Розроблений чат-бот дозволяє забезпечити коректну експлуатацію на різних апаратних платформах. Налагодження та усунення виявлених недоліків сприяли підвищенню продуктивності та зручності використання чат-бота.

Таким чином, результати, отримані в даному розділі, підтверджують доцільність і ефективність впровадження чат-бота для автоматизації обслуговування клієнтів у сфері електронної комерції, а також для подальшого розвитку та масштабування системи відповідно до потреб бізнесу.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досягнуто поставлену мету – розроблено функціональний чат-бот для автоматизованого обслуговування клієнтів у сфері електронної комерції з використанням Python, Node.js. Реалізований програмний продукт забезпечує ефективну взаємодію між користувачем і торговою платформою, дозволяючи переглядати каталог товарів, додавати їх до кошика, оформлювати замовлення, переглядати історію покупок, а також дає можливість адміністраторам керувати категоріями, товарами та статусами замовлень.

Виконанні всі завдання, які були поставлені на початку роботи:

- проаналізовані існуючі рішення та технології для розробки чат-ботів, зокрема сценарних та інтелектуальних підходів, а також розглянуто можливості інтеграції з популярними платформами електронної комерції і визначено переваги та недоліки чат-ботів;
- обрано оптимальні інструменти та технології для реалізації, а саме мову програмування Python для розробки логіки чат-бота та Node.js для організації серверної частини й асинхронної обробки повідомлень поставленого завдання з урахуванням специфіки бізнес-процесів у сфері електронної комерції;
- спроектована архітектура чат-бота, яка включає серверну частину на Node.js, бізнес-логіку на Python, базу даних для зберігання інформації про користувачів, товари й замовлення, а також використання хмарних технологій для забезпечення масштабованості та надійності системи, яка забезпечує масштабованість, надійність та зручність використання;
- реалізовано функціонал чат-бота, що включає каталог товарів, кошик покупок, оформлення замовлень, відстеження статусу замовлення, підтримку клієнтів та сповіщення;
- протестована та оптимізована робота чат-бота, забезпечення його інтеграція з існуючими системами підприємства через API-інтерфейси,

налаштовано взаємодію з CRM-системою для відстеження історії замовлень, впроваджено механізми синхронізації з базою даних товарів та реалізовано автоматичне оновлення каталогу при зміні асортименту, що дозволило створити єдину екосистему обслуговування клієнтів.

Якісні показники реалізації засвідчують зручність інтерфейсу, стабільність роботи програми, відповідність функціональності очікуванням користувачів і легкість масштабування системи. Кількісними результатами є реалізація понад десяти основних команд для взаємодії з ботом, інтеграція з базою даних, реалізація підсистеми періодичних завдань для автоматизації рутинних процесів.

Якісні показники:

1) зручність інтерфейсу – реалізовано інтуїтивне меню з інтерактивними кнопками, що дозволяє користувачу виконувати дії без введення складних команд;

2) стабільність роботи – бот стабільно функціонує в середовищі Telegram, обробляє запити без помилок навіть при паралельній взаємодії кількох користувачів;

3) відповідність функціональності очікуванням – усі ключові процеси онлайн-магазину (перегляд товарів, оформлення замовлень, управління кошиком) реалізовані згідно зі стандартними вимогами до e-commerce сервісу;

4) можливість масштабування – структура бази даних дозволяє додавати нові категорії, функції, користувачів та автоматизовані завдання без необхідності суттєвої перебудови системи.

Виконана робота має безпосередній зв'язок з напрямками науково-дослідної діяльності кафедри інженерії програмного забезпечення, зокрема в галузі інтелектуальних інформаційних систем, автоматизації бізнес-процесів та використання хмарних технологій. Результати розробки можуть бути використані для подальших наукових досліджень у сфері застосування чат-ботів у різних галузях – від електронної комерції до медичних і освітніх сервісів.

У процесі роботи було отримано нові результати, що стосуються побудови архітектури масштабованої системи з підтримкою Telegram API, застосування засобів асинхронної обробки подій у Python та інтеграції планувальника фонових завдань з основною логікою бота. Отриманий досвід може бути узагальнений у наукових публікаціях, а програмне рішення – використане як приклад у навчальному процесі або доопрацьоване з метою комерційного впровадження. Зокрема, вже наявний функціонал може бути доповнений модулями онлайн-оплати, аналітичними панелями для адміністратора, адаптацією під інші платформи месенджерів або інтеграцією з CRM-системами.

Практична цінність розробки полягає у можливості її реального застосування у малому та середньому бізнесі, який потребує автоматизації процесів обслуговування без залучення великої кількості персоналу. Універсальність архітектури та відкритість реалізованого рішення дають змогу адаптувати систему під потреби конкретного підприємства, змінюючи лише вміст бази даних та логіку взаємодії без необхідності повної переробки проекту.

Подальші шляхи вдосконалення розробки включають розширення можливостей чат-бота за рахунок штучного інтелекту – зокрема, впровадження модулів обробки природної мови (NLP) для покращення взаємодії з користувачем, а також реалізацію машинного навчання для персоналізації рекомендацій товарів. Крім того, перспективним є впровадження механізмів авторизації користувачів, формування звітності про активність клієнтів, створення візуальної статистики у вигляді графіків, а також інтеграція зі сторонніми сервісами доставки та оплати.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чат-боти. Переваги та принцип роботи. URL: <https://sendpulse.ua/support/glossary/chatbot> (дата звернення: 05.02.2025).
2. Що таке чат-бот? Як його використовувати та які плюси є для бізнесів? URL: <https://helpcrunch.com/blog/uk/shcho-take-chat-bot/> (дата звернення: 10.02.2025).
3. Що таке чат-бот: простими словами. URL: <https://jedidesk.com/ua/blog/chto-takoe-chat-bot-prostimy-slovami/> (дата звернення: 22.02.2025).
4. Все про чат-боти: переваги, типи та схема роботи. URL: <https://interkassa.com/blog/vse-pro-chat-boti-perevagi-tipi-ta-shema-roboti> (дата звернення: 28.02.2025).
5. Що таке Chatbot (чат-боти) та кому вони потрібні? URL: <https://creativesmm.com.ua/shho-take-chatbot-ta-komu-vonu-potribni/> (дата звернення: 10.03.2025).
6. Все про чат-боти: типи і приклади. URL: <https://web-promo.ua/ua/blog/vse-o-chat-botah-tipy-i-primery-kakomu-biznesu-podojdet-sписок-konstruktorov-dlya-sozdaniya/> (дата звернення: 20.03.2025).
7. Безсонов С. О. Модель інтелектуальної системи чат-ботів. Харків: Харківський національний університет радіоелектроніки, 2022. 85 с.
8. Арешин О. В. Розробка чат-бота з використанням машинного навчання. Рівне: Рівненський державний гуманітарний університет, 2024. 75 с.
9. Веревоchкін О. С. Розробка чат-бота для Telegram на мові Python. Київ: КПІ ім. Ігоря Сікорського, 2020. 60 с.
10. Як створити Telegram-чат-бота. URL: <https://sendpulse.ua/knowledge-base/chatbot/telegram/create-telegram-chatbot> (дата звернення: 01.04.2025).

11. Як створити чат-бот для Telegram-каналу: інструкція для адміністраторів. URL: <https://netpeak.net/uk/blog/yak-stvoriti-chat-bot-dlya-telegram-kanalu-instruktsiya-dlya-administratoriv/> (дата звернення: 07.04.2025).
12. Telegram-бот: що це таке, як створити та використовувати. URL: <https://hostiq.ua/blog/ukr/chatbot-telegram/> (дата звернення: 15.04.2025).
13. Чат-бот для Telegram: що це, як створити і навіщо потрібен бізнесу. URL: <https://helpcrunch.com/blog/uk/chat-bot-dla-telegram/> (дата звернення: 25.04.2025).
14. Як створити Telegram-бота: покрокова інструкція. URL: <https://artjoker.ua/ru/blog/kak-sozdat-telegram-bota/> (дата звернення: 30.04.2025).
15. Чат-боти на Python: що потрібно знати. URL: <https://foxminded.ua/chat-boty-na-python/> (дата звернення: 03.05.2025).
16. Найкращих бібліотек Python для обробки природної мови. URL: <https://www.unite.ai/uk/10-best-python-libraries-for-natural-language-processing/> (дата звернення: 06.05.2025).
17. Як створити Telegram-бота на Python. URL: <https://spacelab.ua/articles/yak-stvoriti-telegram-bota-na-python/> (дата звернення: 07.05.2025).