

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ

«До захисту допущена»
Зав. кафедри комп'ютерних наук
та прикладної математики
д.т.н., професор Турбал Ю.В.
« ____ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
ПРОЄКТУВАННЯ ТА РОЗРОБКА ЧАТ-БОТУ ДЛЯ ПРОДАЖУ КНИГ

Виконала: Закотюк Ірина Володимирівна

Студентка навчально-наукового інституту автоматичної, кібернетичної та
обчислювальної техніки

група КН-41

підпис

Керівник: к.т.н., доцент Іванчук Наталія Віталіївна

підпис

Рівне-2023

Національний університет водного господарства та природокористування

**Навчально-науковий інститут автоматичної, кібернетики та
обчислювальної техніки**

Кафедра комп'ютерних наук та прикладної математики

Рівень вищої освіти бакалавр

Галузь знань 12 «Інформаційні технології»

Спеціальність 122 «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри
комп'ютерних наук та
прикладної математики
д.т.н., професор Турбал Ю.В.

«___» _____ 2023 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ**

Закотюк Ірині Володимирівні

(прізвище, ім'я, по батькові)

1. Тема роботи Проектування та розробка чат-боту для продажу книг

керівник роботи к.т.н., доцент Іванчук Наталія Віталіївна

затверджені наказом вищого навчального закладу від « 19 » квітня 2023 року

С №-449

2. Термін здачі студенткою закінченої роботи 31.05.2023

3. Вихідні дані до роботи: технології, що необхідні для створення чат-ботів

4. Зміст розрахунково-пояснювальної записки Розділ 1. Аналіз предметної області, Розділ 2. Проектування чат-бота, Розділ 3. Програмна реалізація чат-бота

5. Перелік графічного матеріалу мультимедійна презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Розділ 1</i>	<i>Доцент Іванчук Н.В.</i>		
<i>Розділ 2</i>	<i>Доцент Іванчук Н.В.</i>		
<i>Розділ 3</i>	<i>Доцент Іванчук Н.В.</i>		

7. Дата видачі завдання 01.02.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
<i>1</i>	<i>Ознайомлення з вимогами чат-бота.</i>	<i>17.02.2023</i>	<i>виконано</i>
<i>2</i>	<i>Аналіз технологій та алгоритмів штучного інтелекту.</i>	<i>10.03.2023</i>	<i>виконано</i>
<i>3</i>	<i>Вибір платформи та інструментів для розробки.</i>	<i>24.03.2023</i>	<i>виконано</i>
<i>4</i>	<i>Розробка структури чат-бота.</i>	<i>14.04.2023</i>	<i>виконано</i>
<i>5</i>	<i>Програмна реалізація.</i>	<i>05.05.2023</i>	<i>виконано</i>
<i>6</i>	<i>Оформлення звіту про роботу над проектом.</i>	<i>19.05.2023</i>	<i>виконано</i>
<i>7</i>	<i>Підготовка презентації про виконану роботу над проектом.</i>	<i>30.05.2023</i>	<i>виконано</i>

Студентка _____ (Закотюк І.В.)

Керівник кваліфікаційної роботи _____ (Іванчук Н.В.)

ЗМІСТ

РЕФЕРАТ	5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1	9
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1. Історія та розвиток чат-ботів: від початкових етапів до сучасних можливостей.....	9
1.2. Класифікація та особливості чат-ботів.....	11
1.3. Аналіз ботів у телеграм.....	17
1.4. Вплив телеграм-ботів на маркетингові стратегії: переваги та можливості.....	19
РОЗДІЛ 2.....	22
ПРОЄКТУВАННЯ ЧАТ-БОТА	22
2.1. Вибір технологій та інструментів для реалізації проєкту.....	22
2.1.1. JavaScript	22
2.1.2. Node.js	23
2.1.3. React	23
2.1.4. Telegram Bot API.....	24
2.1.5. Netlify	25
2.2. Визначення функціональних та нефункціональних вимог.....	25
2.3. Розробка структури бази даних.....	26
2.4. Вибір середовища для реалізації чат-бота	29
РОЗДІЛ 3.....	31
ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТА	31
ВИСНОВКИ	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	45
ДОДАТКИ.....	47
Додаток А. Компонент «Form»	47
Додаток Б. Компонент «ProductDetails»	50

РЕФЕРАТ

Кваліфікаційна робота: 51 с., 19 зображень, 5 таблиць, 2 додатків, 11 джерел.

Мета кваліфікаційної роботи: розробка та впровадження чат-боту для продажу книг з метою створення зручної та ефективної платформи для користувачів.

Об'єкт дослідження: чат-бот для продажу книг у Telegram для покращення користувацького досвіду при замовленні книг.

Предмет дослідження: використання сучасних засобів для проектування та реалізації чат-боту, взаємодія з користувачами, пошук та представлення книг.

Методи дослідження: використання JavaScript, React і Node.js для розробки чат-боту, а також аналіз Telegram API для взаємодії з платформою Telegram.

Результат роботи: розроблений та впроваджений чат-бот в Telegram для продажу книг.

Ключові слова: ЧАТ-БОТ, ПРОДАЖ КНИГ, ПЛАТФОРМА, ВЗАЄМОДІЯ, ІНТЕГРАЦІЯ, TELEGRAM.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ШІ – штучний інтелект

AIML (англ. Artificial Intelligence Markup Language) - мова розмітки штучного інтелекту

API (англ. Application Programming Interface) - інтерфейс програмування додатків

CSS (англ. Cascading Style Sheets) - каскадні таблиці стилів

DB (англ. Database)- база даних

DOM (Document Object Model) - модель об'єктів документа

JSON (англ. JavaScript Object Notation) - текстовий формат обміну даними між комп'ютерами

JS – мова програмування JavaScript

NLP (Natural Language Processing) - обробка природної мови

UI (англ. User Interface)- інтерфейс користувача

ВСТУП

У сучасному світі електронна комерція набуває популярності і приносить з собою нові способи продажу товарів та послуг. Завдяки своїй зручності та доступності, інтернет-магазини та онлайн-платформи стають все більш поширеними серед споживачів, що дозволяє їм насолоджуватися всіма перевагами, які пропонує інтернет.

Зі збільшенням кількості використання мобільних технологій та месенджерів, інтернет-магазини звернулися до чат-ботів, щоб автоматизувати процес продажу та надавати підтримку клієнтам. Чат-боти – це програмні агенти, що можуть автоматично спілкуватися з користувачами у вигляді текстових повідомлень. Вони мають потенціал розпізнавати та аналізувати запити користувачів та виконувати необхідні дії в межах своїх можливостей.

Чат-боти – це ефективний інструмент взаємодії зі споживачами, що забезпечує швидке обслуговування та зручний метод комунікації. Вони здатні відповідати на запитання про товари, надавати консультації, допомагати з оплатою та вирішувати інші проблеми, пов'язані з процесом купівлі. Чат-боти у змозі працювати в режимі 24/7 без перерви на відпочинок, що робить їх особливо корисними для клієнтів, які живуть у різних часових поясах.

Актуальність роботи зумовлена необхідністю розвитку та вдосконалення механізмів електронної комерції. Хоч і використання чат-ботів для продажу товарів популяризується у сучасному бізнес-середовищі, деякі аспекти цього процесу все ще потребують більш глибокого дослідження.

Мета дослідження полягає у проектуванні та розробці чат-боту для продажу книг з використанням сучасних технологій та алгоритмів штучного інтелекту. Після детального аналізу потреб та вимог користувачів, розроблений інноваційний інтерактивний інструмент, який забезпечить зручний та ефективний спосіб купівлі книг онлайн.

Завдання дослідження полягають у наступному:

- дослідження історії та розвитку чат-ботів;
- аналіз існуючих підходів до розробки чат-ботів та систем електронної комерції;
- аналіз потреб користувачів та написання вимог до чат-бота;
- розробка архітектури чат-бота для продажу книг та опис його функціоналу.
- огляд та вибір технологій та інструментів для реалізації бота;
- розробка та впровадження чат-бота з використанням відповідних програмних засобів;
- тестування розробленого чат-бота для продажу книг.

У даній кваліфікаційній роботі розглядаються аспекти, пов'язані з проектуванням та розробкою чат-бота для продажу книг. Буде здійснено аналіз, проведено практичну реалізацію та оцінку ефективності розробленого рішення. Результати дослідження дозволять оцінити потенціал використання чат-ботів у сфері електронної комерції та зробити висновки щодо їхнього подальшого розвитку і ролі.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Історія та розвиток чат-ботів: від початкових етапів до сучасних можливостей

Чат-бот можна визначити як комп'ютерну програму на основі штучного інтелекту, яка імітує людські розмови. Боти також відомі як цифрові асистенти. Вони інтерпретують наміри користувача, обробляють його запити і дають швидкі релевантні відповіді.

Історія та розвиток чат-ботів пройшла крізь кілька визначних етапів, які відображають поступове вдосконалення та поширення цих інтерактивних програм. Початкові стадії розвитку чат-ботів спостерігалися у 1960-х роках, коли були створені перші автоматизовані системи відповідей. Найвідомішим прикладом є чат-бот «ELIZA».

Бот «ELIZA» був створений Джозефом Вейценбаумом в 1966 році і використовував методику підбору і підстановки шаблонів. Програма була розроблена таким чином, що вона копіювала людську розмову. Чат-бот «ELIZA» працювала шляхом передачі слів, які користувачі вводили у комп'ютер, а потім зіставляла їх зі списком можливих відповідей, складеним за сценарієм. У ньому використовувався сценарій, що імітував психотерапевта. Даний скрипт справив значний вплив на опрацювання природної мови та неприродній інтелект, його копії та варіанти з'являлися в академіях по всій країні (рис. 1.1).

Однак Джозеф Вейценбаум був стурбований реакцією користувачів. Він задумував «ELIZA» як просту карикатуру на людську розмову, проте раптово користувачі стали довіряти ELIZA свої найглибші думки. Експерти заявляли, що через кілька років чат-боти будуть не відрізнятися від людей. Джозеф Вейценбаум відкидав ідею про те, що машини можуть замінити людський інтелект. Він стверджував, що такі пристрої є лише інструментами і продовженням людського розуму. Він також наголошував, що розуміння мови

комп'ютерами повністю залежить від контексту, в якому вони використовуються [1].

```
Welcome to
EEEEEE LL      IIII  ZZZZZZ  AAAAA
EE      LL      II   ZZ   AA  AA
EEEEEE LL      II   ZZ   AAAAAA
EE      LL      II   ZZ   AA  AA
EEEEEE LLLLLL  IIII  ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
```

Рис. 1.1. Розмова з «ELIZA»

У наступні десятиліття творці чат-ботів спиралися на модель Джозефа Вейценбаума, прагнучи до більш людиноподібної взаємодії. Проходження тесту Тюрінга стало спільною метою, у рамках якої нові боти перевіряли свої розмовні таланти перед колегією суддів із людей. Найскладніше в питанні тесту Тюрінга - це відсутність обмежень на те, що люди можуть обговорювати.

У 1990-х роках з'явилися чат-боти, які використовували скриптові мови, такі як AIML. Ці мови дозволяли програмістам створювати більш складні правила та логіку для чат-ботів, що давало їм більшу гнучкість у взаємодії з користувачами.

З появою штучного інтелекту та машинного навчання в останнє десятиліття розвиток чат-ботів значно прискорився. Машинне навчання дозволило чат-ботам навчатися на основі великого обсягу даних та покращувати свої навички взаємодії з користувачами. Алгоритми обробки природньої мови

забезпечують розуміння та генерацію текстових відповідей, надаючи чат-ботам здатність спілкуватися з людьми за допомогою природньої мови.

Сьогодні чат-боти широко використовуються в різних сферах, включаючи комерцію, клієнтське обслуговування, освіту та розваги. Вони можуть надавати розширені можливості, такі як персоналізовані рекомендації, обробка замовлень, проведення транзакцій та навіть виконання завдань штучного інтелекту. Чат-боти інтегруються з популярними месенджерами та соціальними мережами, забезпечуючи зручну та доступну взаємодію з користувачами у їхньому звичному середовищі.

Загалом, історія та розвиток чат-ботів свідчать про поступове вдосконалення їх функціональності та зростання їхнього впливу на спосіб, яким люди взаємодіють з технологіями та отримують інформацію.

1.2. Класифікація та особливості чат-ботів

Класифікація чат-ботів є важливим аспектом їхньої розробки та використання. Залежно від своїх функцій, можливостей та рівня інтелектуальності, чат-боти можуть мати різні характеристики. Основними критеріями, за якими можна класифікувати чат-ботів є : спосіб їхньої роботи, рівень інтелекту та принцип взаємодії з користувачами.

Таблиця 1.1

Типи чат-ботів за принципом роботи

Тип	Переваги	Недоліки
Кнопковий	- простота використання	обмежені робочі можливості

Продовження таблиці 1.1

На основі розпізнавання ключових слів	<ul style="list-style-type: none"> - точність у запитах клієнтів; - швидка відповідь 	не завжди може адаптуватись до непередбачуваних сценаріїв
На основі правил	<ul style="list-style-type: none"> - безкодовий редактор; - плавна крива навчання 	має бути ініційований клієнтом першим
Чат-бот зі штучним інтелектом	<ul style="list-style-type: none"> - більш персоналізований клієнтський досвід; - без участі людини 	потрібно постійно оновлювати та навчати
Гібридна модель	<ul style="list-style-type: none"> - простота ботів на основі правил; - можливості ШІ 	складність у налаштуванні
Голосовий	<ul style="list-style-type: none"> - здатність відповідати на голосові запити клієнтів; - найшвидший зв'язок 	може відходити від теми

1. Кнопковий бот

Кнопкові чат-боти - це найпростіший тип чат-ботів, який наразі доступний на ринку. Користувач просто обирає тему із заздалегідь визначених кнопок, щоб виконати бажану дію. Таким чином, клієнт не має змоги самостійно вводити текст.

Цілком можливо, що запит користувача не буде відображеним у списку існуючих тем й бот не буде корисним. Іншими словами, це "скриптовий чат-бот", що обмежений конкретними питаннями та відповідями.

2. Бот на основі розпізнавання ключових слів

Чат-боти на основі розпізнавання ключових слів є більш просунутими інструментами, ніж кнопкові чат-боти. Вони можуть розпізнавати релевантні ключові слова в запиті користувача та давати відповідні відповіді.

Наприклад, якщо потенційний клієнт запитує "скільки коштує товар X", бот може розпізнати ключові слова "ціна" та "товар X" й надати більше інформації про модель ціноутворення.

3. Бот на основі правил

Віртуальні помічники на основі правил можуть відповідати на запити на основі заздалегідь створених наборів засад. Складність правила може сильно варіюватися від простого набору до просунутого. Зазвичай це залежить від галузі, завдань, які компанія хоче вирішити за допомогою бота та ресурсів (гроші, люди).

Оскільки ці боти побудовані на умовах "if/then", їх дуже легко навчати. Тому для впровадження таких ботів не потрібно витратити багато грошей — достатньо лише написати скрипти. Проте важливо пам'ятати, що боти на основі правил не можуть навчатися самостійно. Вони лише надають ту інформацію, яка потрібна бізнесу.

4. Чат-бот зі штучним інтелектом

Боти зі штучним інтелектом здатні до самонавчання. Вони запрограмовані за допомогою обробки природної мови (NLP) та машинного навчання. На відміну від ботів на основі правил, створення та навчання таких ботів займає набагато більше часу.

Вони добре працюють у компаніях, які мають багато даних про користувачів, що дозволяє чат-ботам самонавчатися на них. Боти зі штучним інтелектом потенційно можуть розуміти і відповідати різними мовами, щоб забезпечити більш персоналізоване спілкування.

5. Бот із гібридною моделлю

Гібридна модель пропонує найкраще з трьох світів: простоту чат-бота, заснованого на правилах, розширені можливості бота зі штучним інтелектом та емпатію людської взаємодії. Результатом є універсальний підхід: з простими чат-ботами для повторюваних запитань, технологією штучного інтелекту для більш складних запитань і живими агентами для запитань, які потребують людської відповіді.

Хоча це може здатися складним, але це нічого не змінить у роботі агентів або в процесі отримання відповідей від клієнтів. Гібридний бот, який поєднує в собі функції, засновані на правилах, та штучний інтелект, виступатиме в ролі нового віртуального агента. Він відповідатиме на запити, з якими може впоратися самостійно, а складні багаторівневі питання передаватиме живим представникам служби підтримки.

6. Голосовий бот

Голосовий бот - це програма, що дозволяє користувачеві взаємодіяти з пристроєм при цьому розмовляючи з ним. Інструмент може розуміти вимовлене запитання, аналізувати його та структурувати відповідну аудіо-відповідь. Це не зовсім те саме, що чат-бот проте має багато спільного.

Голосові боти та чат-боти використовують схожу базову технологію. Вони можуть бути простими або ж просунутими («Siri», що працює на основі штучного інтелекту). По суті, вони мають однакове завдання - зрозуміти запитання клієнта та надати правильну відповідь [2].

Типи ботів залежно від того, як вони використовуються

Тип	Переваги	Недоліки
Бот підтримки	<ul style="list-style-type: none"> - багатофункціональне призначення; - проактивне охоплення; - обробка поширених запитань; - збір даних про клієнтів 	обмежені можливості для вирішення складних питань
Бот для маркетингу та продажів	<ul style="list-style-type: none"> - підвищення залученості клієнтів; - можливості для апселінгу та перехресних продажів 	надлишкові рекламні акції можуть перевантажити клієнтів
Бот із навичками	<ul style="list-style-type: none"> - багатозадачність; - виконує базові команди 	не розуміє складних питань
Розважальний бот	<ul style="list-style-type: none"> - покращене залучення клієнтів 	не спрямований на підтримку клієнтів

1. Бот підтримки

Цей тип чат-ботів використовується в клієнтському сервісі для виконання будь-яких завдань, пов'язаних з підтримкою користувачів. Зазвичай він відповідає на всі види поширених запитань і проводить відвідувача через весь

процес до етапу демо-версії або покупки. Якщо чат-бот не може надати відповідь, він перенаправляє клієнта до агентів служби підтримки.

Чат-боти підтримки можуть використовуватися у будь-яких галузях, включаючи банківську справу, роздрібну торгівлю, HR, цифровий маркетинг, охорону здоров'я тощо. Понад 67% клієнтів у всьому світі використовують чат-ботів для підтримки клієнтів.

Чат-боти підтримки можуть допомогти у таких аспектах:

- обробляти повторювані питання;
- зосередитися на більш важливих запитах клієнтів;
- збирати контактну інформацію;
- надавати проактивну підтримку;
- збирати відгуки клієнтів;
- заощаджувати операційні ресурси.

2. Бот для продажів та маркетингу

Ще одна популярна сфера використання чат-ботів — це продажі та маркетинг. Основна мета - допомогти клієнтам дізнатися про товари та послуги й придбати їх. Надсилаючи користувачам релевантну та цікаву інформацію, боти попередньо оцінюють потенційних клієнтів і просувають їх через воронку продажів. Повідомляється, що чат-боти збільшують продажі в середньому на 67%.

У сфері продажів і маркетингу чат-боти можуть допомогти:

- збирати інформацію про потенційних клієнтів та проводити попередню кваліфікацію;
- надсилати рекламні сповіщення;
- перепродавати продукти;
- персоналізувати клієнтський досвід;
- підвищувати залученість.

3. Бот із навичками

Це чат-боти, які можуть виконувати такі команди, як «ввімкнути пісню», «зателефонувати», «відкрити програму», «надіслати електронного листа» тощо. Для цього типу чат-ботів голосовий функціонал означає, що користувачеві не потрібно тримати пристрій або натискати кнопки.

При цьому команди можна подавати і в текстовому форматі. Програмне забезпечення здатне просто виконувати інструкції, що дозволяє користувачам працювати в режимі багатозадачності. У більшості випадків, щоб отримати базові навички чат-бота, не потрібно витратити багато часу і впроваджувати технологію штучного інтелекту. Достатньо буде просто навчитися, як саме давати команди боту, щоб пристрій знав, що робити.

Чат-боти з навичками можуть допомогти у таких аспектах:

- виконувати прості команди;
- дозволяти користувачам працювати в режимі багатозадачності;
- підвищувати залученість відвідувачів на веб-сайті або платформі;
- підвищувати задоволеність клієнтів.

1.3. Аналіз ботів у телеграм

Телеграм був запущений 14 серпня 2013 року як додаток для обміну повідомленнями без реклами, орієнтований на конфіденційність. Спочатку додаток був доступний для iOS, а альфа-версія для Android вийшла пізніше того ж року. У перший рік свого існування додаток залучив близько 100 000 щоденних активних користувачів.

Сьогодні він має понад 550 мільйонів активних користувачів щомісяця. За кількістю завантажень телеграм входить до топ-10 найпопулярніших соціальних мереж у світі [3].

Серед численних пропозицій, наданих телеграмом, досить популярними стали телеграм-боти. Боти в телеграмі - це, по суті, програми, які взаємодіють з користувачами за допомогою Telegram Bot API. Вони можуть бути розроблені сторонніми розробниками або ж самим додатком й пропонують широкий спектр функцій, які покращують досвід роботи в середовищі додатку. Ботів можна додавати до груп або отримувати до них доступ через прямі повідомлення, що робить їх універсальним інструментом як для окремих осіб, так і для команд.

Боти пропонують цілий ряд функцій для задоволення різних потреб та інтересів. Вони можуть надавати інформацію в режимі реального часу, наприклад, прогнози погоди, новини тощо. Ігрові боти дозволяють користувачам брати участь в інтерактивних іграх у самому додатку, заохочуючи змагання та розваги. Крім того, боти можуть сприяти автоматизації завдань, перекладу, плануванню та навіть електронній комерції. Широкий спектр функцій зробив телеграм-ботів незамінними для багатьох користувачів.

За лаштунками телеграм-боти покладаються на різні технології для ефективної роботи: Telegram Bot API являється основою для розробки будь-якого виду ботів і надає розробникам інструменти для створення та управління ними. Телеграм-ботів можна створювати за допомогою таких мов програмування, як Python, Node.js, JavaScript або Java, і розміщувати їх на різних платформах або серверах. Методи обробки природної мови (NLP), алгоритми машинного навчання та моделі штучного інтелекту дозволяють ботам розуміти користувацьке введення, розумно реагувати та постійно покращувати продуктивність.

Загалом, телеграм-боти класифікуються як чат-боти зі штучним інтелектом, оскільки вони використовують штучний інтелект і пов'язані з ним технології для ведення інтерактивних та інтелектуальних розмов у середовищі обміну повідомленнями.

1.4. Вплив телеграм-ботів на маркетингові стратегії: переваги та можливості

Телеграм став популярною платформою для взаємодії бізнесу зі своїми клієнтами, а з інтеграцією чат-ботів компанії можуть вивести свої маркетингові зусилля на новий рівень.

Маркетинг має вирішальне значення для охоплення цільової аудиторії та збільшення прибутку. Тому використання телеграм-ботів може змінити правила гри для компаній, які прагнуть випередити своїх конкурентів. Телеграм-боти - це універсальні інструменти, які можна використовувати для різних маркетингових цілей, таких як генерація потенційних клієнтів, підтримка клієнтів і запуск кампаній з підвищення обізнаності про бренд.

Переваги та можливості використання телеграм-ботів:

1. Пряме спілкування з клієнтами

Телеграм-бот дозволяє компаніям напряду спілкуватися зі своїми клієнтами, що може допомогти побудувати міцні відносини. За допомогою користувацького інтерфейсу чат-бота компанії можуть надсилати персоналізовані повідомлення своїм клієнтам, пропонувати підтримку та надавати всю необхідну інформацію.

2. Персоналізація та кастомізація

Гнучкий Bot API та користувацький інтерфейс дозволяють компаніям персоналізувати свої повідомлення і налаштовувати чат-ботів відповідно до індивідуальності їхнього бренду. Вони пропонують кілька варіантів налаштування, включаючи кнопки, швидкі відповіді та мультимедійні повідомлення, що дозволяє компаніям створювати унікальний досвід для своїх клієнтів.

3. Зворотній зв'язок та аналітика в режимі реального часу

API надає зворотній зв'язок та аналітику в режимі реального часу, що дозволяє компаніям відстежувати ефективність своїх чат-ботів. Інтерфейс пропонує такі показники: кількість надісланих, отриманих, прочитаних повідомлень тощо. Ця інформація може допомогти бізнесу оптимізувати своїх чат-ботів та покращити маркетингові зусилля.

4. Автоматизація завдань

Користувацький інтерфейс чат-бота дозволяє компаніям автоматизувати такі завдання, як надсилання сповіщень, планування зустрічей та обробка замовлень. Автоматизація може заощадити час та гроші, а також підвищити продуктивність бізнесу.

5. Рентабельний маркетинг

Користувацький інтерфейс чат-бота є економічно ефективним маркетинговим інструментом, оскільки компанії можуть охопити широку аудиторію, не витрачаючи багато грошей. Компанії можуть самостійно створювати чат-ботів і керувати ними без необхідності наймати додатковий персонал або звертатися до аутсорсингу.

6. Можливість охопити ширшу аудиторію

Telegram має величезну базу користувачів, а з інтеграцією чат-ботів бізнес може охопити ширшу аудиторію. Чат-боти можуть надавати клієнтам актуальну інформацію, пропонувати підтримку і навіть обробляти замовлення, полегшуючи бізнесу зв'язок зі своїми клієнтами.

7. Інтеграція з іншими платформами

Інтерфейс користувача чат-бота можна інтегрувати з іншими платформами, такими як фейсбук, інстаграм та твітер. Ця інтеграція дозволяє компаніям охоплювати своїх клієнтів на різних платформах і забезпечувати безперебійну

роботу. Телеграм-ботів можна також інтегрувати з іншими маркетинговими інструментами, такими як CRM-системи.

8. Підвищення залученості та взаємодії

Користувацький інтерфейс чат-бота може допомогти бізнесу підвищити рівень залученості та взаємодії зі своїми клієнтами. Чат-боти можуть пропонувати персоналізовані рекомендації, надавати релевантну інформацію та навіть ініціювати розмови, що може допомогти утримати клієнтів залученими та зацікавленими.

9. Покращене обслуговування клієнтів

Інтерфейс користувача чат-бота може допомогти бізнесу забезпечити краще обслуговування клієнтів, пропонуючи підтримку в режимі реального часу та оперативно вирішуючи проблеми клієнтів.

10. Побудова сильного іміджу бренду

Користувацький інтерфейс чат-бота може допомогти бізнесу створити сильний імідж бренду, забезпечуючи бездоганний клієнтський досвід. Пропонуючи персоналізовані повідомлення та підтримку, компанії можуть створити позитивне сприйняття свого бренду, що може призвести до лояльності клієнтів та збільшення доходів [4].

Отже, Telegram-боти можуть принести користь маркетинговим стратегіям, забезпечуючи пряму комунікацію, персоналізацію, зворотний зв'язок у режимі реального часу, автоматизацію, економічну ефективність, ширше охоплення аудиторії, інтеграцію з іншими платформами, підвищену залученість та взаємодію, покращений сервіс обслуговування клієнтів та сильний імідж бренду.

РОЗДІЛ 2.

ПРОЄКТУВАННЯ ЧАТ-БОТА

2.1. Вибір технологій та інструментів для реалізації проєкту

Вибір технологій та інструментів для реалізації проєкту - важливий крок у процесі розробки чат-бота. Вибір найбільш підходящої технології забезпечить ефективну роботу системи, зручний інтерфейс і масштабованість проєкту.

Для досягнення даних цілей розглянемо деякі з ключових інструментів, що можуть бути використані при розробці чат-бота для продажу книг, такі як JavaScript, Node.js, React, Telegram Bot API і Netlify. Кожна з цих технологій має свої переваги та специфічні особливості для забезпечення ефективної реалізації необхідного функціоналу та комфортного використання додатку користувачем.

2.1.1. JavaScript

JavaScript — це кросплатформна об'єктно-орієнтована мова сценаріїв, яка використовується для створення інтерактивних веб-сторінок. Існують також більш просунуті серверні версії JavaScript, такі як Node.js, які дозволяють додавати на веб-сайт більше функціональності, ніж просто завантаження файлів (наприклад, спільна робота в режимі реального часу між декількома комп'ютерами) [5].

JavaScript — одна з найпопулярніших та найпоширеніших мов програмування. Її широке використання забезпечує доступ до великої кількості розробників і спільноти, яка надає підтримку та різноманітні ресурси для розвитку проєктів. Завдяки простому синтаксису та високому рівню абстракції JavaScript спрощує процес розробки чат-ботів. Розробники можуть швидко створювати функціональні компоненти, обробляти події та взаємодіяти з Telegram Bot API для отримання та обробки повідомлень.

Дана мова програмування дозволяє легко інтегрувати чат-ботів з широким спектром систем і сервісів. Наприклад, його можна інтегрувати з базами даних

MySQL, що зберігають інформацію про книги та замовлення. JavaScript також пропонує можливість асинхронного програмування, особливо в контексті Node.js. Це дозволяє ефективно обробляти багатопотокові запити користувачів і дає можливість створеному чат-боту швидко реагувати на вхідні повідомлення.

2.1.2. Node.js

Node.js — платформа на основі JavaScript, що дозволяє розробникам створювати серверні додатки за допомогою цієї мови програмування. До переваги Node.js можна віднести асинхронну та подієво-керовану модель програмування. Дана модель базується на використанні неблокуючих операцій вводу/виводу, що оптимізують роботу мережеских запитів і дозволяють їм ефективно виконуватися без зайвих затримок.

Поєднання цієї моделі програмування з циклом обробки подій Node.js дозволяє обробляти багато запитів одночасно, вирішуючи проблему блокування виконання коду. Таким чином, Node.js може забезпечити високу продуктивність і низьку латентність при обробці запитів. Це дуже важливо для чат-ботів, що займаються комерцією, де швидка реакція на запити користувачів має вирішальне значення.

Варто також відзначити, що Node.js легко поєднується з іншими технологіями, що використовуються для розробки користувацьких інтерфейсів, такими як React.

2.1.3. React

React — це бібліотека для розробки користувацького інтерфейсу на основі JavaScript. Хоча й React являється бібліотекою, а не мовою проте вона широко використовується у веб-розробці. Дана бібліотека вперше з'явилася у травні 2013 року і зараз є однією з найбільш використовуваних фронтенд-бібліотек для веб-розробки [6].

З головних переваг React — є його віртуальний DOM та ефективна стратегія візуалізації. Віртуальний DOM — це внутрішнє представлення структури інтерфейсу в пам'яті. React порівнює віртуальний DOM із реальним, щоб визначити найкращі зміни, які потрібно внести в реальний DOM для оновлення UI. Такий підхід зменшує навантаження на браузер і дозволяє швидше створювати компоненти.

React дозволяє створювати компоненти з власною внутрішньою структурою, стилем та логікою, які можна повторно використовувати в різних частинах програми. Це полегшує розробку, підтримку та розширення інтерфейсу. React дозволяє також ефективно керувати станом компонентів і реактивно оновлювати їх, сприяючи інтерактивності інтерфейсу.

2.1.4. Telegram Bot API

Telegram Bot API — це набір інструментів та інтерфейсів, наданих телеграмом для розробки та взаємодії з ботами, що дозволяє розробникам створювати чат-боти, використовуючи при цьому різноманітні функції [7].

Telegram Bot API дозволяє отримувати та обробляти повідомлення від користувачів чат-бота. Це можуть бути текстові повідомлення, фотографії, відео, аудіофайли, документи або ж інші мультимедійні файли. Розробники можуть аналізувати та обробляти ці повідомлення для виконання різних функцій, наприклад, обробки замовлень на продаж книг.

Окрім отримання повідомлень, Telegram Bot API також дозволяє розробникам надсилати повідомлення від імені чат-бота. Повідомлення можуть надсилатися у відповідь на запити користувачів, сповіщення про статус замовлення, підтвердження реєстрації та інші повідомлення, необхідні для ефективного спілкування з користувачем.

До особливостей Telegram Bot API можна віднести підтримку клавіатур та інтерактивних елементів. Це дозволяє створювати такі елементи як кнопки та

меню, що спрощує навігацію додатком і взаємодію з користувачем. Клавіатура може містити: текстові кнопки, посилання, віджети та ін.

2.1.5. Netlify

Netlify — це платформа, спеціалізована на розгортанні та хостингу веб-додатків, включаючи чат-боти для продажу книг. Використання Netlify дозволяє швидко та зручно розмістити чат-бот в Інтернеті, забезпечуючи йому надійне та доступне онлайн-середовище [8].

Netlify привабливий своєю простотою використання. Платформа пропонує інтуїтивно зрозумілий інтерфейс і зручні інструменти для розгортання та керування чат-ботами. Крім того, Netlify пропонує систему масштабування, яка дозволяє легко масштабувати чат-бота зі збільшенням кількості користувачів і робочого навантаження. Платформа гнучко адаптується до мінливих вимог проекту, забезпечуючи стабільну та швидку роботу чат-бота навіть при високих навантаженнях. Netlify спрощує розробку та інтеграцію чат-ботів, дозволяючи взаємодіяти з іншими службами та інструментами.

2.2. Визначення функціональних та нефункціональних вимог

Для успішного впровадження чат-бота потрібні чітко визначені вимоги. На етапі аналізу системи та проектування ці вимоги мають значний вплив на процес розробки, адже вони визначають функції та можливості чат-ботів.

Функціональні вимоги визначають очікувану функціональність чат-бота. Їхня мета – створити зручний та функціональний інструмент для придбання книг. З іншого боку, нефункціональні вимоги встановлюють вимоги щодо якості чат-бота. Наприклад, ефективність вимагає, щоб система працювала на високій швидкості, а надійність — означає стабільність і постійну доступність послуг. Детальний опис функціональних і нефункціональних вимог дасть чітке уявлення про функції та якості, які очікуються від чат-бота.

Функціональні вимоги включають в себе:

- можливість заповнення даних користувача;
- відображення доступних книг разом з назвою, автором та ціною, а також обкладинкою;
- можливість перегляду детальної інформації про книгу;
- додавання книг у кошик.

Нефункціональні вимоги включають наступне:

- система повинна бути стабільною та надійною, з мінімальною кількістю збоїв;
- інтерфейс чат-бота повинен бути простим для розуміння і використання користувачами з різним рівнем технічної грамотності;
- чат-бот повинен бути сумісним з платформою телеграм;
- система повинна бути масштабованою, щоб відповідала зростанню кількості користувачів та трафіку;
- інформація про користувача повинна бути конфіденційною.

2.3. Розробка структури бази даних

У процесі проектування та впровадження чат-ботів для книжкових продажів значна увага приділяється розробці та оптимальному використанню баз даних для ефективного зберігання, управління та доступу до інформації про книги. Бази даних відіграють важливу роль у забезпеченні зручного пошуку книг, управлінні книжковими запасами та наданні користувачам актуальної інформації про товари.

Розробка бази даних передбачає визначення структури таблиць, що відображають різні аспекти інформації про книгу, такі як: назва, автор, опис, ціна та наявність. Кожен з цих аспектів визначається у відповідному полі таблиці, що дозволяє зберігати та організовувати дані у зручний спосіб. Крім того, при

розробці бази даних також необхідно враховувати вимоги до нормалізації та цілісності даних. Це означає, що дані можна зберігати в нормалізованому форматі, щоб уникнути дублювання та забезпечити невизначеність і узгодженість даних

Пропонується наступна структура бази даних для зберігання інформації про книги:

Таблиця 2.1

«Books»

Назва поля	Тип поля	Пояснення
book_id	INT	Унікальний ідентифікатор книги
title	TEXT(70)	Назва книги
author	TEXT(45)	Автор
price	FLOAT	Вартість книги
image	TEXT(200)	Зображення обкладинки
format	TEXT(45)	Формат книги
description	TEXT(400)	Опис книги
pages	INT	Кількість сторінок
publisher	TEXT(45)	Видавництво
year	INT	Рік видання

Таблиця 2.2

«Customers»

Назва поля	Тип поля	Пояснення
customer_id	INT	Унікальний ідентифікатор клієнта

Продовження таблиці 2.2

first_name	TEXT(20)	Ім'я клієнта
last_name	TEXT(30)	Прізвище клієнта
email	VARCHAR(45)	Електронна пошта
phone	VARCHAR(45)	Номер телефону
city	TEXT(30)	Місто
subject	TEXT(45)	Тип особи

Таблиця 2.3

«Orders»

Назва поля	Тип поля	Пояснення
order_id	INT	Унікальний ідентифікатор замовлення
customer_id	INT	ідентифікатор клієнта
book_id	INT	Ідентифікатор книги
total_price	FLOAT	Загальна вартість замовлення

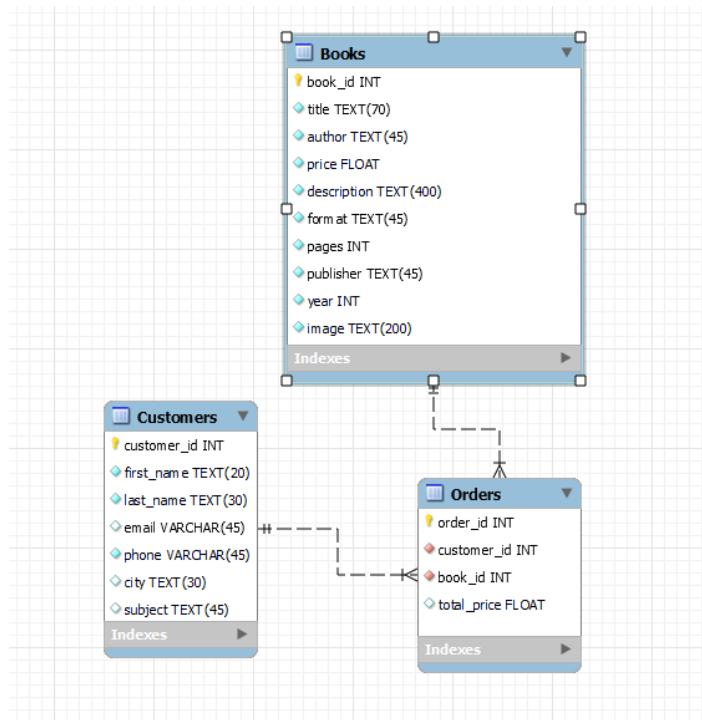


Рис. 2.1. Схема зв'язків між таблицями

1. Зв'язок між таблицею «Books» і таблицею «Orders»:

Поле «book_id» у таблиці «Orders» є зовнішнім ключем, який посилається на поле «book_id» у таблиці «Books». Цей зв'язок дозволяє встановити залежність між замовленнями та відповідними книгами.

2. Зв'язок між таблицею «Customers» і таблицею «Orders»:

Поле «customer_id» у таблиці «Orders» є зовнішнім ключем, який посилається на поле «customer_id» у таблиці «Customers». Цей зв'язок дозволяє встановити залежність між замовленнями та відповідними клієнтами.

Ці зв'язки дозволяють встановлювати залежності між різними таблицями, що допомагає забезпечити цілісність даних та зручність роботи з базою даних.

2.4. Вибір середовища для реалізації чат-бота

При виборі середовища розробки важливо ретельно проаналізувати різні фактори, що впливають на ефективність розробки та зручність використання.

Важливість даних критеріїв зумовила вибір Visual Studio Code як основного середовища для реалізації чат-бота.

Visual Studio Code поєднує в собі простоту редактора вихідного коду з потужним інструментарієм для розробника. Безпроблемний цикл редагування-збірки-налагодження означає: витрачання менше часу на налаштування середовища і більше часу на реалізацію своїх ідей.

Дане середовище розробки ідеально підходить для повсякденного використання. Підтримуючи сотні мов, VS Code допомагає миттєво підвищити продуктивність завдяки підсвічуванню синтаксису, узгодженню дужок, автоматичним відступам, виділенню блоків, фрагментів тощо. Інтуїтивно зрозумілі комбінації клавіш, легке налаштування та створені спільнотою відображення клавіатурних комбінацій дозволяють легко орієнтуватися в коді.

VS Code також підтримує Git, що дає можливість працювати з контролем вихідних кодів, не виходячи з редактора, зокрема переглядати відмінності змін, що очікують на розгляд [9].

РОЗДІЛ 3.

ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТА

BotFather є інструментом, який забезпечує можливість створювати та керувати бот-акаунтами в телеграмі . За допомогою BotFather можна також налаштовувати різні параметри бота, такі як аватар, опис та команди [10].

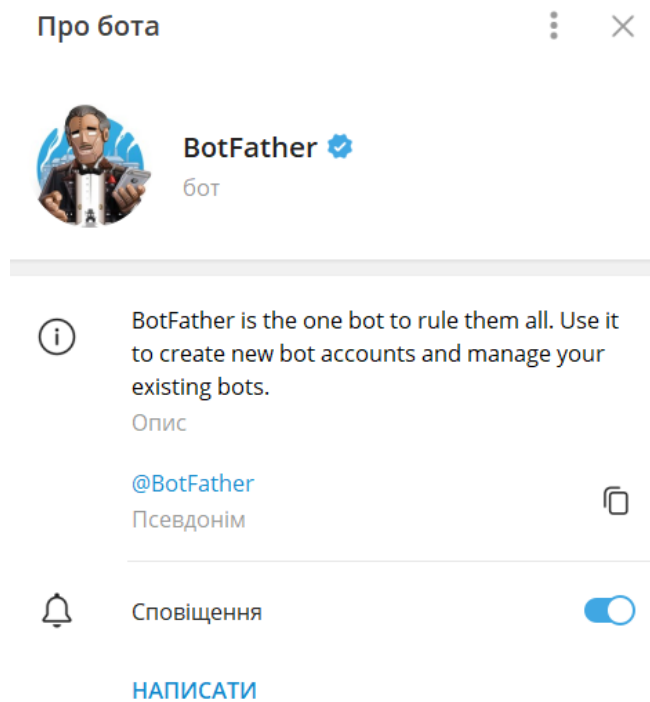


Рис. 3.1. Вигляд та короткий опис бота

Під час першого використання BotFather пропонує користувачам створити нового бота, вказавши його назву й після цього отримати його токен. Для створення чат-бота необхідно прописати команду */newbot*.

Назва створеного чат-бота: «BookWorm Express».

Токен: 6171441993:AAFtFkPhmON_iYNadSt-PKrT9Agoi4JbKI.

Надавати токен потрібно лише довіреним особам, які потребують прямого доступу до чат-бота. У разі необхідності можна скасувати старий токен та створити новий для забезпечення безпеки та контролю над чат-ботом.

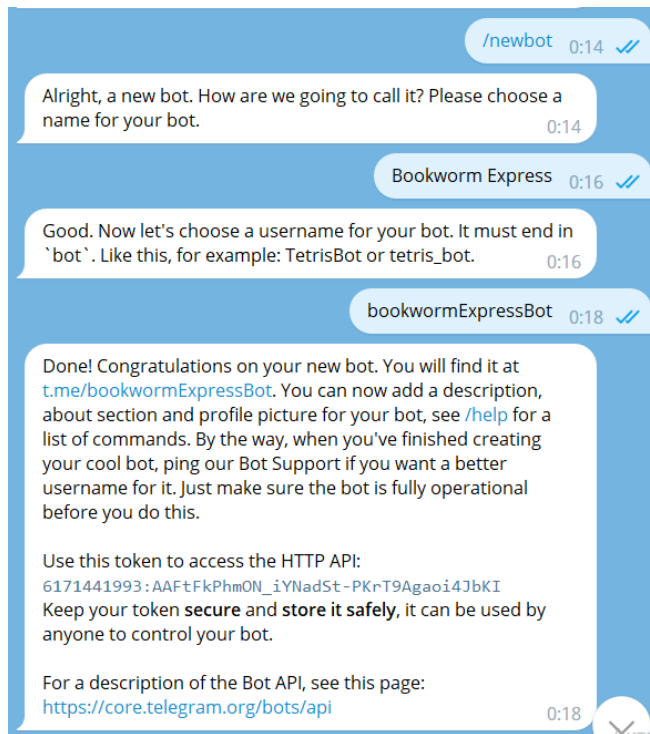


Рис. 3.2. Команди, необхідні для створення чат-бота

Для чат-бота необхідно створити два проєкти: перший проєкт слугуватиме для розробки на front-end, де використовуватиметься React, а другий – для back-end Node.js.

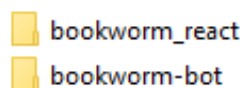


Рис. 3.3. Створені проєкти

В «bookworm_react» в терміналі необхідно прописати команду `npm create-react-app` для ініціалізації додатку.

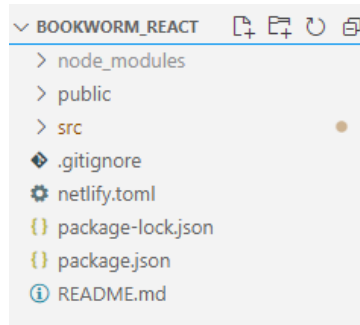


Рис. 3.4. Завантажені пакети

У свою чергу, «bookworm_bot» проєкт в термінал потрібно прописати наступні команди:

— `npm init -y`

Дана команда використовується для ініціалізації нового проєкту і створення файлу `package.json` зі стандартними значеннями. Опція `-y` вказує `npm` автоматично приймати значення за замовчуванням для всіх запитань, що зазвичай задаються під час створення `package.json`. Це означає, що команда буде запускатися без вимоги введення будь-яких додаткових даних [11].

— `npm i -D nodemon`

Команда встановлює пакет `nodemon` і додає його до `package.json` у розділ `devDependencies`, щоб використовувати його під час розробки проєкту.

— `npm i node-telegram-bot-api`

Команда, що використовується для встановлення пакету `node-telegram-bot-api` з використанням `npm`. Даний пакет призначений для розробки ботів Telegram з використанням Node.js.

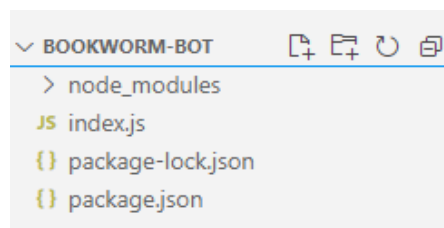


Рис. 3.5. Завантажені пакети та доданий вручну «index.js» файл

Спочатку розберемо ситуацію, коли після надісланої команди `/start` від користувача, необхідно відправити повідомлення щодо заповнення форми, а також кнопку, що міститиме посилання на дану форму.

```
bot.on('message', async(msg) => {
  const chatId = msg.chat.id;
  const text = msg.text;

  if (text === '/start') {
    await bot.sendMessage(chatId, "Привіт! Для початку необхідно заповнити форму, тому клікай на кнопку нижче ↓",{
      reply_markup: {
        keyboard: [
          [{text: "Заповнити форму", web_app:{url: formUrl }}]
        ]
      }
    });
  }
});
```

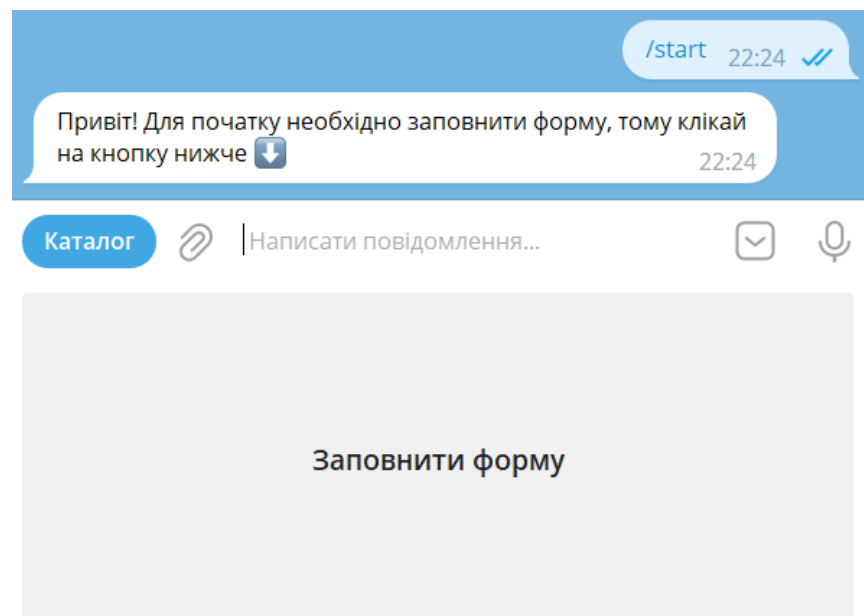


Рис. 3.6. Виконання команди `/start`

Наступною дією є створення веб-форми для заповнення персональних даних, в якій містяться такі поля: ім'я та прізвище користувача, електронна пошта, номер телефону, місто, а також тип особи.

Перш за все, створимо хук `useTelegram` для отримання доступу до функціональності «Telegram WebApp». Оголосимо константу `tg`, якій присвоємо

значення `window.Telegram.WebApp`. Передбачатиметься, що об'єкт буде доступний у глобальному контексті. Також визначимо функції `onClose()` та `onToggleButton()` :

```
const tg = window.Telegram.WebApp;
export function useTelegram() {
  const onClose = ()=> {
    tg.close()
  }
  const onToggleButton = ()=> {
    if(tg.MainButton.isVisible){
      tg.MainButton.hide();
    } else {
      tg.MainButton.show();
    }
  }
  return {
    onClose,
    onToggleButton,
    tg,
    user:tg.initDataUnsafe?.user,
  }
}
```

Компонент «Form» використовує хуки React для збереження та оновлення значень полів. Крім того, використовуються ефекти для взаємодії з «Telegram WebApp», зокрема для налаштування головної кнопки, контролю видимості кнопки в залежності від заповненості деяких полів форми, а також для реєстрації та відміни обробників подій (див. додаток А).

Умова, при якій кнопка «Надіслати дані» буде доступна лише після заповнення обов'язкових полів (ім'я, прізвище та номер телефону):

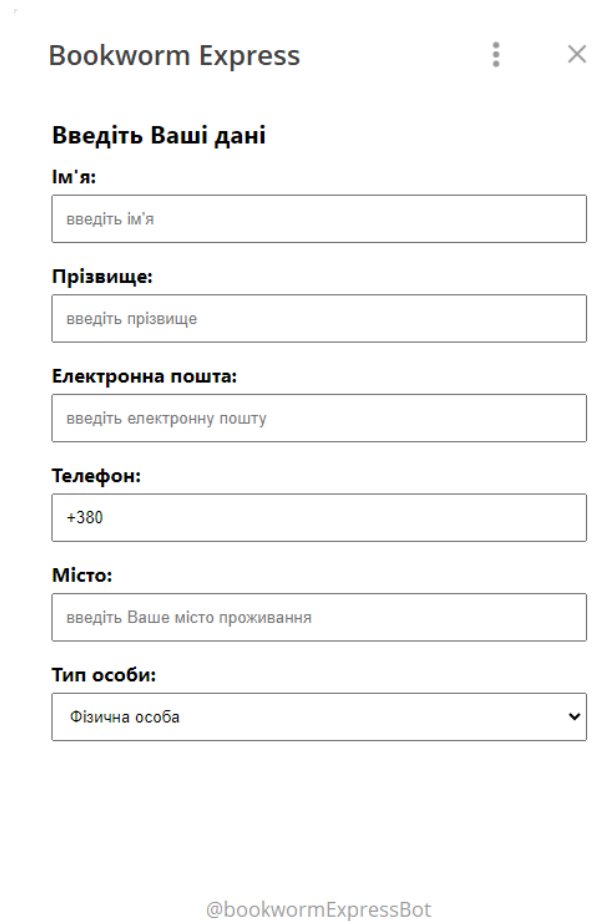
```
useEffect(() => {
  if (!phoneNumber || !name || !surname ) {
    tg.MainButton.hide();
  } else {
    tg.MainButton.show();
  }
}, [name, surname, phoneNumber]);
```

Для того, щоб веб-форми були адаптивними й змінювались в залежності від обраного режиму в телеграмі, варто в css файлі прописати такі значення:

```
body {  
  background: var(--tg-theme-bg-color);  
  color: var(--tg-theme-text-color);  
}
```

Щоб дебажити веб-форми із чат-ботом онлайн, доведеться створити репозиторій на gitHub та прив'язати його в Netlify. Отримавши посилання на застосунок, потрібно вставити його в код:

```
const webAppUrl = 'https://bucolic-concha-a1a711.netlify.app/';
```



Bookworm Express

Введіть Ваші дані

Ім'я:

Прізвище:

Електронна пошта:

Телефон:

Місто:

Тип особи:

@bookwormExpressBot

Рис. 3.7. Вигляд форми в світлому режимі

Bookworm Express

Введіть Ваші дані

Ім'я:

Прізвище:

Електронна пошта:

Телефон:

Місто:

Тип особи:

Рис. 3.8. Показ кнопки після заповнення полів, що є обов'язковими Дії, що відбуваються після натискання кнопки «Надіслати дані»:

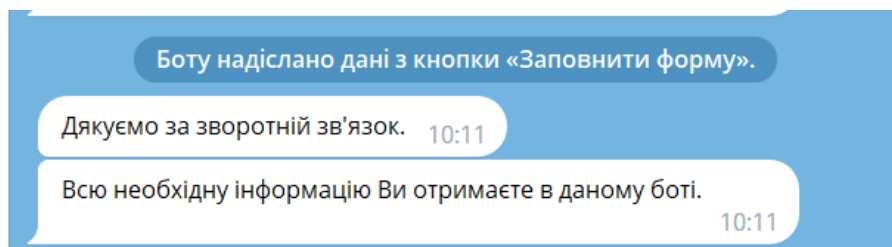


Рис. 3.9. Взаємодія з чот-ботом

Подивимось, який вигляд має форма для заповнення даних, якщо змінити режим телеграму в нічний:

Bookworm Express

Введіть Ваші дані

Ім'я:
введіть ім'я

Прізвище:
введіть прізвище

Електронна пошта:
введіть електронну пошту

Телефон:
+380

Місто:
введіть Ваше місто проживання

Тип особи:
Фізична особа

@bookwormExpressBot

Рис. 3.10. Вигляд форми під час нічного режиму

Наступний етап – створення каталогу. Для створення кнопки потрібно знову звернутись до BotFather та прописати команду `/setmenubutton`, а також дати відповідне URL, що посилається на сторінку каталогу.

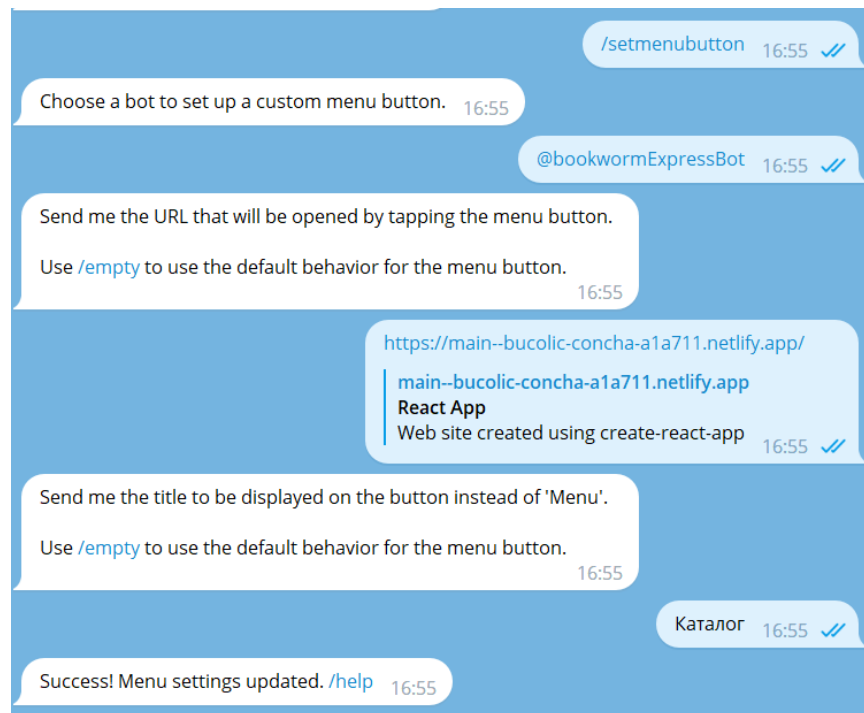


Рис. 3.11. Кастомізація кнопки «Каталог»

Для показу книг існує три компоненти – «`ProductItem`», «`ProductList`», та «`ProductDetails`».

«`ProductList`» використовує хук `useState`, щоб зберігати список доданих продуктів у стані `addedItems`. За допомогою функції `onAdd()` можна додати або видалити продукт зі списку `addedItems`.

Компонент також використовує хук `useTelegram` для отримання доступу до функціональності телеграму. Коли кнопка "Додати в корзину" натиснута, викликається функція `onSendData`, яка надсилає дані про додані продукти та загальну вартість через телеграм.

У «`ProductList`» також використовуються ефекти `useEffect()` для підписки на подію `mainButtonClicked` та налаштування відображення кнопки залежно від наявності доданих продуктів. Коли кнопка «Додати в корзину» стає видимою, вона також має відображати загальну вартість доданих продуктів.

Стосовно компоненту «`ProductItem`» – він відображає окремий елемент продукту та має властивості `product`, `className` і `onAdd()`, які передаються в

компонент зовні. Також компонент містить посилання на детальну сторінку продукту за допомогою *Link* з *react-router-dom*. При кліку на кнопку «Додати в корзину» викликається функція *onAddHandler()*, яка передає *product* в батьківський компонент.

Компонент «*ProductDetails*», у свою чергу, відображає деталі конкретного продукту. Він отримує дані про продукт через об'єкт *location.state* з використанням хука *useLocation* з пакету *react-router-dom* (див. додаток В).

У загальному компоненті «*App*», який створює основний каркас додатку варто відобразити необхідні компоненти за допомогою *Routes* і *Route* з пакету *react-router-dom* для того, щоб налаштувати маршрутизацію між цими компонентами:

```
const App = () => {
  const {onToggleButton,tg} =useTelegram();
  useEffect( () => {
    tg.ready();
  },[])
  return (
    <div className="App">
      <Header/>
      <Routes>
        <Route index element={<ProductList/>} />
        <Route path={'form'} element={<Form/>} />
        <Route path={'product/:id'} element={<ProductDetails/>} />
      </Routes>
    </div>
  );
};
```


Bookworm Express



 <p>"The Night They Vanished" Ванесса Севідж Вартість: 350 € Детальніше Додати в корзину</p>	 <p>"The hound of the baskervilles" Артур Конан Дойл Вартість: 150 € Детальніше Додати в корзину</p>
	


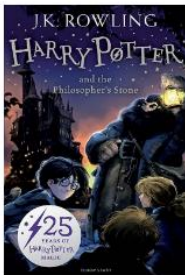


@bookwormExpressBot

Рис. 3.12. Перегляд каталогу книг в світлому режимі

Bookworm Express



zakotiukk

 <p>"Everything I Know About Love" Доллі Олдертон Вартість: 320 € Детальніше Додати в корзину</p>	 <p>"Harry potter and the philosopher's stone" Джоан Роулінг Вартість: 340 € Детальніше Додати в корзину</p>
	

Придбати 660

Рис. 3.13. Відображення кнопки «Придбати» та загальна вартість



Рис. 3.14. Детальна інформація стосовно книги



Рис. 3.15. Продовження сторінки



Рис. 3.16. Представлення каталогу книг в темному режимі



Рис. 3.17. Представлення детальної інформації в темному режимі

ВИСНОВКИ

У даній кваліфікаційній роботі було проведено детальне дослідження та розроблено чат-бот для продажу книг з використанням сучасних технологій та алгоритмів штучного інтелекту. Метою роботи було створення зручної платформи для клієнтів, яка дозволяє їм здійснювати покупки книг онлайн.

У розділі аналізу предметної області було вивчено історію та розвиток чат-ботів, класифікацію та особливості чат-ботів, а також вплив телеграм-ботів на маркетингові стратегії.

Розділ проектування чат-бота включав вибір необхідних технологій та інструментів, визначення функціональних та нефункціональних вимог, розробку структури бази даних та вибір середовища для реалізації чат-бота.

У розділі програмної реалізації чат-бота була проведена розробка та впровадження чат-бота з використанням відповідних програмних засобів. Було здійснено тестування розробленого чат-бота для продажу книг. Отримані результати свідчать про успішну реалізацію поставлених завдань дослідження.

Застосування чат-бота у сфері продажу книг має значний потенціал для подальшого розвитку. Можливі напрямки подальшого розвитку чат-бота включають розширення асортименту книг, впровадження додаткових функцій, таких як рекомендації книг на основі історії покупок користувача або ж можливість обміну відгуками та оцінками.

Розроблений чат-бот відповідає сучасним вимогам та стандартам і може бути успішно використаний в сфері продажу книг. Його функціонал та можливості можуть бути додатково розширені для вдосконалення користувацького досвіду та забезпечення більш ефективного та зручного процесу покупки книг онлайн.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The History Of Chatbots – From ELIZA to ChatGPT. [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://onlim.com/en/the-history-of-chatbots/> (дата звернення 10.03.2023)
2. 10 Types of Chatbots: Which One Will Your Business Pick? [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://helpcrunch.com/blog/types-of-chatbots/> (дата звернення 15.03.2023)
3. How Many People Use Telegram in 2023? 55 Telegram Stats [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://backlinko.com/telegram-users> (дата звернення 25.03.2023)
4. 10 Reasons Why Telegram Bot is a Powerful Marketing and Customer Engagement Tool [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://medevel.com/10-reasons-why-telegram-bot-is-a-powerful-marketing-tool/> (дата звернення 07.04.2023)
5. Introduction to JavaScript [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction> (дата звернення 07.04.2023)
6. The Best Guide to Know What Is React [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs> (дата звернення 10.04.2023)
7. Telegram Bot API [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://core.telegram.org/bots/api> (дата звернення 10.04.2023)
8. Netlify Security Overview [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://docs.netlify.com/> (дата звернення 15.04.2023)

9. Why Visual Studio Code? [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://code.visualstudio.com/docs/editor/whyvscode> (дата звернення 2.05.2023)
10. Bots: An introduction for developers [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://core.telegram.org/bots> (дата звернення 03.05.2023)
11. npm-init | npm Docs [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://docs.npmjs.com/cli/v9/commands/npm-init> (дата звернення 10.05.2023)

ДОДАТКИ

Додаток А. Компонент «Form»

```
import React, { useCallback, useEffect } from "react";
import "./Form.css";
import { useState } from "react";
import { useTelegram } from "../../hooks/useTelegram";
import Button from "../Button/Button";
import { useNavigate } from "react-router-dom";

const Form = () => {
  const [name, setName] = useState("");
  const [surname, setSurname] = useState("");
  const [email, setEmail] = useState("");
  const [city, setCity] = useState("");
  const [subject, setSubject] = useState("physical");
  const [phoneNumber, setPhoneNumber] = useState("+380");

  const { tg } = useTelegram();
  const navigate = useNavigate();

  const onSendData = useCallback(() => {
    const data = {
      name,
      surname,
      email,
      phoneNumber,
      city,
      subject,
    };
    tg.sendData(JSON.stringify(data));
  }, [name, surname, email, phoneNumber, city, subject]);

  useEffect(() => {
    tg.onEvent("mainButtonClicked", onSendData);
    return () => {
      tg.offEvent("mainButtonClicked", onSendData);
    };
  }, [onSendData]);

  useEffect(() => {
    tg.MainButton.setParams({
      text: "Надіслати дані",
    });
  }, []);
}
```

```

useEffect(() => {
  if (!phoneNumber || !name || !surname ) {
    tg.MainButton.hide();
  } else {
    tg.MainButton.show();
  }
}, [name, surname, phoneNumber]);

const onChangeName = (e) => {
  setName(e.target.value);
};

const onChangeSurname = (e) => {
  setSurname(e.target.value);
};

const onChangeEmail = (e) => {
  setEmail(e.target.value);
};

const onChangeCity = (e) => {
  setCity(e.target.value);
};

const onChangeSubject = (e) => {
  setSubject(e.target.value);
};

const onChangePhoneNumber = (e) => {
  setPhoneNumber(e.target.value);
};

const navigateToProductList = () => {
  navigate("/");
};

return (
  <div className={"form"}>
    <h3 className="heading">Введіть Ваши дані</h3>
    <label className="label" htmlFor="name">Ім'я:</label>
    <input
      className={"input"}
      id="name"
      type="text"
      placeholder={"введіть ім'я"}
    >
  </div>
)

```



```

    value={name}
    onChange={onChangeName}
  />
  <label className="label" htmlFor="surname">Прізвище:</label>
  <input
    className={"input"}
    id="surname"
    type="text"
    placeholder={"введіть прізвище"}
    value={surname}
    onChange={onChangeSurname}
  />
  <label className="label" htmlFor="email">Електронна пошта:</label>
  <input
    className={"input"}
    id="email"
    type="text"
    placeholder={"введіть електронну пошту"}
    value={email}
    onChange={onChangeEmail}
  />
  <label className="label" htmlFor="phoneNumber">Телефон:</label>
  <input
    className={"input"}
    id="phoneNumber"
    type="text"
    placeholder={"введіть номер телефону"}
    value={phoneNumber}
    onChange={onChangePhoneNumber}
  />
  <label className="label" htmlFor="city">Місто:</label>
  <input
    className={"input"}
    id="city"
    type="text"
    placeholder={"введіть Ваше місто проживання"}
    value={city}
    onChange={onChangeCity}
  />

  <label className="label" htmlFor="subject">Тун особи:</label>
  <select
    id="subject"
    value={subject}

```

```

    onChange={onChangeSubject}
    className={"select"}
  >
    <option value={"physical"}>Фізична особа</option>
    <option value={"legal"}>Юридична особа</option>
  </select>
</div>
);
};
export default Form;

```

Додаток Б. Компонент «ProductDetails»

```

import React from 'react';
import './ProductDetails.css';
import { useLocation } from 'react-router-dom';
import Button from "../Button/Button";
import imgSrc from "../ProductItem/ProductItem";
const ProductDetails = () => {
  const location = useLocation();
  const product = location.state.product;
  const onAddHandler = () => {
    onAdd(product);
  }
  const renderRating = () => {
    const stars = [];

    for (let i = 1; i <= 5; i++) {
      if (i <= product.rating) {
        stars.push(
          <span key={i} className="star filled-star" onClick={() => handleRatingClick(i)}>
            ☆
          </span>
        );
      } else {
        stars.push(
          <span key={i} className="star" onClick={() => handleRatingClick(i)}>

```

```

        ☆
    </span>
    );
    }}
    return stars;
};
const handleRatingClick = (rating) => {
    // Handle rating click event if needed
};
return (
    <div className={'product-details'}>
        <div className={'img'}>
            <img src={product.img} />
        </div>
        <div className={'det_title'}>{product.title}</div>
        <div className={'author'}>{product.author}</div>
        <div className={'description'}>{product.description}</div>
        <div className="format-pages-container">
            <div className={'format'}>Формат: {product.format}</div>
            <div className={'pages'}>Кількість сторінок: {product.pages}</div>
        </div>
        <div className="publisher-year-container">
            <div className={'publisher'}>Видавництво: {product.publisher}</div>
            <div className={'year'}>Рік видання: {product.year}</div>
        </div>
        <div className="rating" data-rating={product.rating}>
            {renderRating()}
        </div>
        <Button className="button_container" title="Додати в корзину" type="add"
    onClick={onAddHandler} /> </div>
    );}; export default ProductDetails;

```