

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ

Навчально-науковий інститут автоматичної, кібернетичної та
обчислювальної техніки

Кафедра комп'ютерних наук та прикладної математики

«До захисту допущений»

Завідувач кафедри комп'ютерних наук
та прикладної математики

_____ Турбал Ю.В.

«_____» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка чат-боту по доставці харчування»

Виконав: Міщенко Артем Сергійович

(прізвище, ім'я, по батькові)

_____ (підпис)

група КН-41

Керівник: старший викладач Остапчук О.П.

_____ (науковий ступінь, вчене звання, посада, прізвище та ініціали)

(підпис)

**Навчально-науковий інститут автоматики, 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 СТУДЕНТУ

Міщенку Артему Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка чат-боту по доставці харчування

керівник роботи Остапчук Оксана Петрівна, старший викладач кафедри
комп'ютерних наук та прикладної математики

затверджені наказом вищого навчального закладу від «_____» _____ 2023 року
С №-449

2. Термін здачі студентом закінченої роботи 10.06.2023

3. Вихідні дані до роботи: документація замовника

4. Зміст розрахунково-пояснювальної записки Розділ 1.

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>Остапчук О.П., ст. викладач</i>		
<i>Розділ 2</i>	<i>Остапчук О.П., ст. викладач</i>		
<i>Розділ 3</i>	<i>Остапчук О.П., ст. викладач</i>		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	<i>Опрацювання літератури та інтернет-джерел на задану тему.</i>		
2	<i>Узгодження із замовником.</i>		
3	<i>Проектування структури чат-боту.</i>		
4	<i>Програмна реалізація.</i>		
5	<i>Тестування та виправлення помилок.</i>		
6	<i>Оформлення теоретичної частини</i>		
7			
8			
9			

Студент _____ (Міщенко А.С)

Керівник кваліфікаційної роботи _____ (Остапчук О.П)

ЗМІСТ

РЕФЕРАТ	4
АНОТАЦІЯ	5
ВСТУП	7
РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1. Загальні відомості про чат-боти для доставки харчування	9
1.2. Ключові поняття	11
1.3. Формулювання задачі	13
РОЗДІЛ 2 ІНСТРУМЕНТИ РОЗРОБКИ	14
2.1. Вступ	14
2.2. Опис засобів розробки	Помилка! Закладку не визначено.
РОЗДІЛ 3 ЕТАПИ РОЗРОБКИ	17
3.1. Створення чат-боту	17
3.2. Розробка інтерфейсу	18
3.2.1. Обробка платежів	24
3.2.2. Профіль користувача	25
3.2.3. Кабінет адміністратора	257
РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ	28
4.1. Вступ	28
4.2. Головна сторінка	28
4.3. Створення замовлення	30
ВИСНОВОК	40
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	41
ДОДАТКИ	43
Додаток А. Код чат-бота	43
Додаток Б. Приклад коду оброблення платежів	43
Додаток В. Приклад коду користувацького інтерфейсу	47

РЕФЕРАТ

Кваліфікаційна робота: 41 лист, 32 малюнків, 3 додатки

Мета: Розробка чат-бота для доставки харчування по містах України.

Завдання: Розробити чат-бот для доставки харчування містами України, який буде зрозумілий для використання користувачем.

Для досягнення поставленої мети, потрібно виконати наступні пункти:

- Проаналізувати чат-боти у сфері доставки харчування та визначити їх проблеми
- Проаналізувати та визначити проблеми, які виникають під час співпраці клієнта та працівника
- Вивчити проблеми, з якими стикаються користувачі при замовленні їжі через чат-бот
- Створити план реалізації чат-боту (визначення технологій та функціоналу)
- Розробка функціоналу
- Розробка зручного та привабливого користувацького інтерфейсу
- Впровадження та підтримка

Результати: Розроблений чат-бот для доставки харчування по містах України, який надає зрозумілий та ефективний користувацький інтерфейс, швидкий у розумінні та функціонально зручний чат-бот для користувача.

Ключові слова: чат-бот, інтерфейс користувача, користувач, база даних, оптимізація алгоритмів доставки, інтеграція платежів.

АНОТАЦІЯ

Ця кваліфікаційна робота присвячена розробці чат-бота для доставки харчування по містах України. Головна мета є реалізація зрозумілого, зручного та ефективного чат-боту для користувачів, у яких є потреба скористатися послугами замовлення їжі та швидкого у розумінні інтерфейсу чат-боту для користувачів.

У роботі використовується мова програмування Python, одна з високорівневих мов програмування, яка використовує зрозумілу і читабельну синтаксичну структуру, що полегшує розробку програм і зрозуміння коду. За основу реалізації інтерфейсу чат-бота взято Python, який у поєднанні з бібліотеками надає необхідні інструменти для реалізації зручного інтерфейса, яка швидко та зручно взаємодіє з користувацькою частиною та має відповідний інструментарій для збереження інформації в базу даних.

Чат-бот забезпечує користувачів функціональним інтерфейсом, в якому можливо робити замовлення їжі, переглядати страви, вести комунікацію з персоналом та ставити оцінку і відгук для покращення якості чат-бота. Використання Python у розробці чат-бота дозволяє легко і швидко налагоджувати та розширювати функціонал програми. Python має багато бібліотек та фреймворків, що полегшують роботу з мережевими запитами, обробкою даних та роботою з базою даних. Чат-бот для доставки харчування розроблений з урахуванням потреб користувачів у зручному та швидкому способі замовлення їжі без необхідності використовувати телефон або веб-сайт. За допомогою бази даних можна зберігати інформацію про страви, користувачів та їх замовлення. Це дозволяє забезпечити швидкий доступ до даних та зручне управління замовленнями, а також зберігати історію замовлень для подальшого аналізу та покращення обслуговування. Окрім замовлення їжі, чат-бот може надавати додаткові функції, такі як розташування ресторанів на мапі, огляди та рейтинги страв, зручний пошук страв за категоріями або інгредієнтами, а також пропозиції та акції для користувачів. Чат-бот може використовувати технологію натуральної мови для забезпечення більш

зрозумілої та природної комунікації з користувачами. Він може розпізнавати запити, відповідати на питання та надавати рекомендації з урахуванням особистих вподобань користувача.

У кінцевому результаті був створений зрозумілий та ефективний чат-бот для доставки харчування, який доступний 24/7, дозволяє оформлювати замовлення страв та напитків через зручний інтерфейс для користувача, здійснювати безпечну оплату та отримувати безперервну підтримку.

ВСТУП

У сучасному суспільстві швидкість, зручність і доступність послуг є ключовими чинниками, що визначають наш спосіб життя. Розширення цифрових технологій та зростання популярності мобільних пристроїв приводять до змін у багатьох сферах нашого повсякденного життя, включаючи гастрономічну індустрію. Одним із інноваційних рішень, що виникають від цього, є розробка чат-ботів для доставки харчування. Чат-боти - це програми, засновані на штучному інтелекті, здатні спілкуватися з людьми у форматі текстових повідомлень. Вони стають дедалі більш популярними і знайшли своє застосування у різних галузях, включаючи гастрономічну сферу. Чат-боти для доставки харчування забезпечують зручний та ефективний спосіб замовлення їжі з великим вибором страв та ресторанів.

Але також є і проблеми, які є досить актуальними на цей час – проблема безпеки та конфіденційності користувача. Ця проблема стосується усіх галузей діяльності, які надають послуги, пов'язані з купівлею або замовленням. Однією з ключових проблем є зберігання та обробка особистих даних, які користувачі надають під час процесу замовлення через чат-бота. Ці дані можуть включати адреси, номери телефонів, фінансову інформацію тощо. Недостатня захищеність цих даних може призвести до серйозних проблем, включаючи крадіжку особистої інформації та фінансового маніпулювання. Розв'язання цих проблем вимагає впровадження ефективних заходів безпеки, таких як шифрування даних, використання безпечних протоколів передачі інформації та строгих політик доступу до даних. Компанії, що надають послуги доставки харчування через чат-бота, повинні приділяти особливу увагу захисту особистих даних своїх клієнтів та забезпечувати виконання відповідних правил та норм безпеки.

Актуальні проблеми, з якими можна стикнутися у сфері надання послуг:

- **Забезпечення високої якості обслуговування** – однією з ключових проблем є недоліки в якості надання послуг. Це може включати

нестачу навичок та знань персоналу, недосконалість процесів обслуговування або незадовільну якість використовуваних матеріалів та ресурсів.

- **Ефективне управління часом та ресурсами** – у сфері надання послуг часто виникає проблема з ефективним управлінням часом та ресурсами. Це може бути пов'язано з недостатнім плануванням, неефективним розподілом ресурсів або затримками в процесі надання послуг.

- **Задоволення потреб та очікувань клієнтів** – недостатня організація та управління часом можуть призвести до затримок у наданні послуг. Це може бути спричинено неоптимізованим середовищем для надання послуг. Важкий у розумінні працівника інструментарій та неефективно-функціональні шляхи взаємодії працівника з клієнтом та його замовленнями.

Тому все більше ресторанів та кафе, які мають ціль переманити на свою сторону побільше користувачів замовляють створити чат-ботів для доставки харчування, які можуть забезпечити автоматизовану систему прийому замовлень, що дозволяє клієнтам зручно та швидко замовляти їжу. Компанії, які надають послуги доставки харчування, можуть використовувати чат-ботів для прийому замовлень, стеження за статусом доставки та надання інформації про меню. Отже, чат-боти є потужним інструментом для компаній у сфері надання послуг, незалежно від того, чи це ресторан, туристичне агентство або фінансова установа. Вони забезпечують ефективну автоматизовану комунікацію з клієнтами, персоналізований підхід та полегшують процес надання послуг, що призводить до покращення задоволення клієнтів та зміцнення конкурентоспроможності компаній.

РОЗДІЛ 1

ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Загальні відомості про чат-боти по доставці харчування

У сучасному світі зростає популярність використання чат-ботів для замовлення та доставки харчування. Ці інтерактивні програми засновані на штучному інтелекті та машинному навчанні, які спрощують процес комунікації з клієнтами і забезпечують зручність та швидкість у замовленні та доставці їжі.

Чат-боти для доставки харчування - це програмні рішення, що дозволяють клієнтам здійснювати замовлення їжі через інтерфейс чату. Користувачі можуть спілкуватися з чат-ботом за допомогою текстових повідомлень, вказувати свої вподобання та замовляти страви з розширеного меню.

Основні функції чат-ботів для доставки харчування включають:

1. **Прийом замовлень:** Чат-боти приймають замовлення від клієнтів, аналізуючи текстові повідомлення та інструкції. Вони можуть запитувати додаткову інформацію, таку як адресу доставки, контактні дані або спеціальні побажання щодо страв.
2. **Рекомендації та персоналізоване обслуговування:** Завдяки аналізу даних та збереженню історії замовлень, чат-боти можуть надавати рекомендації щодо страв, враховуючи вподобання та дієтологічні обмеження користувача. Це дозволяє забезпечити персоналізоване обслуговування та покращити взаємодію з клієнтами.
3. **Інформаційна підтримка:** Чат-боти можуть надавати інформацію про доступні страви, акції та знижки, графік роботи та інші питання, що стосуються послуг доставки харчування. Вони можуть також надавати відповіді на часті запитання та допомагати вирішувати проблеми, пов'язані з доставкою.
4. **Оплата та фінансові операції:** Деякі чат-боти для доставки харчування підтримують онлайн-платежі та забезпечують безпеку фінансових транзакцій. Вони дозволяють користувачам здійснювати оплату

безпосередньо через інтерфейс чату, спрощуючи процес оплати та підвищуючи зручність для клієнтів.

5. Взаємодія зі службою підтримки: Чат-боти можуть надавати підтримку клієнтам, відповідаючи на запитання, розв'язуючи проблеми та направляючи клієнтів до живих операторів, якщо потрібно. Це сприяє вирішенню невеликих питань без затримок та полегшує комунікацію з компанією.

На сьогоднішній день чат-боти для доставки харчування знайшли широке застосування в ресторанному бізнесі, кафе, піцеріях та інших закладах громадського харчування. Компанії, які надають послуги доставки харчування, використовують чат-ботів для автоматизації процесу прийому замовлень, покращення комунікації з клієнтами та полегшення процесу доставки.

Окрім того, чат-боти для доставки харчування можуть бути інтегровані в месенджери, такі як Facebook Messenger, Telegram, WhatsApp, що дозволяє клієнтам здійснювати замовлення безпосередньо зі своїх обраного месенджера.

Застосування чат-ботів для доставки харчування приносить суттєві зміни в способи замовлення та отримання їжі. Вони забезпечують швидку, зручну та персоналізовану комунікацію з клієнтами, полегшуючи процес надання послуг та підвищуючи задоволення клієнтів. Розвиток цих технологій відкриває нові можливості для покращення гастрономічного бізнесу.

У зв'язку з цим, для ефективної реалізації потреби в чат-боті для доставки харчування, необхідно ретельно розглянути й оцінити інші варіанти чат-ботів, які вже використовуються у цій галузі. Це дозволить краще зрозуміти різні технології та підходи до виконання поставлених завдань. І для початку необхідно уточнити основні поняття, які спрямовані на спрощення розуміння процесу розробки чат-бота.

Чат-боти по доставці харчування є інноваційним рішенням, яке поєднує технології штучного інтелекту та гастрономічні задоволення. Вони дозволяють користувачам з легкістю замовляти їжу, ефективно вибирати страви та насолоджуватись комфортом онлайн-замовлення. Ці чат-боти пропонують

широкий вибір кухонь світу - від класичної піци та суші до екзотичних страв з інші країн. Вони допомагають знайти ідеальну страву для будь-якого настрою або захоплення.

Крім зручного замовлення, чат-боти також можуть надати корисні рекомендації, пропонувати акційні пропозиції та забезпечувати швидку інформаційну підтримку. Вони створюють інтерактивне середовище, де користувачі можуть отримати всю необхідну інформацію про меню, доставку та оплату без зайвих зусиль. Ці чат-боти також можуть надавати персоналізоване обслуговування, враховуючи попередні замовлення та уподобання користувачів. Вони зберігають історію замовлень, рекомендують популярні страви та пропонують індивідуальні акції та знижки. Чат-боти по доставці харчування дозволяють зекономити час та зусилля, оскільки вони працюють цілодобово і не потребують дзвінків або особистих візитів. Користувачі можуть замовляти їжу в будь-який час та з будь-якого місця, використовуючи простий інтерфейс чату.

Ці чат-боти також сприяють покращенню комунікації між ресторанами та клієнтами. Вони дозволяють точніше збирати замовлення, уникати недорозумінь та підвищувати рівень задоволення клієнтів. Чат-боти по доставці харчування постійно вдосконалюються та інтегруються з новими технологіями, такими як голосові команди або розпізнавання обличчя. Вони стають все більші інтуїтивними та зручними для користувачів, роблячи процес замовлення харчування ще більш приємним та ефективним.

1.2. Ключові поняття

Користувач – Особа, яка взаємодіє з чат-ботом для замовлення та отримання харчування, включаючи реєстрацію, авторизацію та надання необхідної інформації.

Чат-бот – це програмне рішення, засноване на штучному інтелекті, яке забезпечує автоматизовану комунікацію з користувачами через текстовий інтерфейс чату. В контексті доставки харчування, чат-бот може служити для прийому замовлень, надання інформації про страви та процес доставки.

Інтерфейс користувача – Це може бути текстовий інтерфейс, кнопки, меню, форми вводу, графічні елементи тощо, що дозволяє зручно та легко взаємодіяти з чат-ботом. Також включає у себе всі елементи та засоби взаємодії з користувачем.

База даних є централізованим місцем збереження інформації, необхідної для роботи чат-бота. Вона може містити дані про страви, меню, клієнтів, замовлення, адреси доставки та іншу важливу інформацію, яка використовується для обробки та виконання замовлень.

Оптимізація алгоритмів доставки використовується для поліпшення ефективності та швидкості процесу доставки харчування. Це може включати вибір оптимального маршруту доставки, розподіл замовлень між кур'єрами, врахування часу приготування страв та інші фактори, що допомагають забезпечити якісну та швидку доставку.

Відгук клієнта – це повідомлення або огляд, який клієнт залишає після замовлення та доставки харчування через чат-бота. Відгуки можуть містити коментарі щодо якості страв, швидкості доставки, обслуговування та загального задоволення замовленням. Відгуки клієнтів можуть допомогти компанії виявити сильні та слабкі сторони свого сервісу та здійснити відповідні корективи.

Інтеграція платежів включає підключення різних систем оплати до чат-бота, щоб забезпечити зручну та безпечну оплату замовлень. Це може включати інтеграцію з онлайн-платіжними системами, банківськими шлюзами або іншими електронними платіжними рішеннями, що дозволяють клієнтам зручно та безпечно оплачувати свої замовлення.

Редагування замовлення – це процес внесення змін до замовлення клієнта після його оформлення. Це важлива функція, яка дозволяє користувачам вносити зміни до своїх замовлень до моменту їх підтвердження або готування страв.

Оцінка клієнта – це система, яка дозволяє клієнтам виразити свою задоволеність чи незадоволеність якістю обслуговування. Оцінка може бути

представлена у вигляді зірок, числових значень чи інших символів. Клієнти можуть встановлювати оцінку відповідно до своєї задоволеності отриманим сервісом. Це дозволяє компанії визначити загальну якість обслуговування та вжити заходів для поліпшення.

1.3. Формулювання задачі

Метою кваліфікаційної роботи полягає у розробці та створенні чат-бота для доставки харчування з метою поліпшення процесу замовлення та обслуговування клієнтів.

Розроблена програмна система буде забезпечувати наступний набір функцій для:

Клієнта:

- Інтерфейс
- Реєстрація профілю користувача
- Редагування особистих даних
- Калькулятор ціни замовлення
- Оплата замовлення
- Відгуки та оцінки
- Перегляд статусу замовлення
- Спілкування зі службою підтримки

Адміністратора:

- Керування меню
- Обробка замовлень
- Редагування даних
- Управління користувачами
- Статистика та аналітика

РОЗДІЛ 2

ІНСТРУМЕНТИ РОЗРОБКИ

2.1. Вступ

Для початку реалізації чат-боту для доставки їжі, потрібно чітко розуміти, які інструменти для розробки чат-боту потрібно використовувати. Для створення даного чат-боту використовував мову програмування **Python**, а використовував для цього інтегроване середовище розробки **PyCharm** і базу даних **SQLite**.

Python є однією з найпопулярніших мов програмування, яка широко використовується у розробці чат-ботів. Вона відома своєю простотою, читабельністю та широким спектром бібліотек, що спрощують процес розробки. Мова Python має вбудовану підтримку роботи з текстовими даними, що є важливим для реалізації функціоналу чат-бота. Вона також пропонує різні фреймворки та бібліотеки, що дозволяють легко створювати та розширювати чат-ботів.

PyCharm є одним з найпопулярніших інтегрованих середовищ розробки (IDE) для роботи з мовою програмування Python. Він надає потужні інструменти для написання, налагодження та тестування коду Python. Завдяки своїм функціям автодоповнення, перевірки помилок та інтеграції з іншими інструментами, PyCharm допомагає розробникам покращити продуктивність та ефективність роботи над проектом чат-бота для доставки їжі.

SQLite є легким та ефективним вбудовуваним реляційним СУБД, яка може бути використана для зберігання даних про замовлення, користувачів та іншу інформацію, що використовується в чат-боті для доставки їжі. Вона забезпечує надійну та швидку роботу з базою даних, а також простоту інтеграції з мовою програмування Python. SQLite може бути зручним вибором для початкової розробки чат-бота, оскільки не вимагає окремого сервера баз даних та налаштування складних з'єднань.

2.2. Опис засобів розробки

Інтерфейс користувача в чат-боті для доставки їжі реалізований з використанням мови програмування Python. Python є однією з найпопулярніших мов програмування для розробки чат-ботів по доставці харчування. Він володіє простим і зрозумілим синтаксисом, що дозволяє розробникам швидко створювати функціональні та ефективні рішення. Використання Python дозволяє забезпечити зручну та ефективну взаємодію з користувачем через текстовий інтерфейс чат-бота. Python надає широкі можливості для створення інтуїтивно зрозумілого та зручного інтерфейсу, що сприяє покращенню досвіду користувача та забезпеченню зручного замовлення їжі. Замовлення користувачем у чат-боті для доставки їжі здійснюється за допомогою програмування шляхом взаємодії з інтерфейсом чат-бота.

У чат-боті для доставки їжі дані про страви та користувачів можуть бути збережені у базу даних SQLite, використовуючи програмування. SQLite - це легкий та ефективний реляційний база даних, який часто використовується для зберігання даних у чат-ботах по доставці харчування. Він надає зручні механізми для створення таблиць, вставки, оновлення та вибірки даних. SQLite підтримує мову SQL, що дозволяє легко взаємодіяти з базою даних. Перш за все, необхідно створити таблиці у базі даних для збереження інформації про страви та користувачів. Далі, при отриманні нового замовлення, дані про страви та користувача можна записати у відповідні таблиці, зберігаючи необхідну інформацію, таку як назва страви, ціна, адреса доставки, контактні дані тощо. Після записування даних, база даних SQLite може бути використана для подальшого доступу та обробки цих даних, забезпечуючи зручну та ефективну роботу з інформацією про страви та користувачів. За допомогою Python та SQLite, розробники можуть легко створювати бази даних для зберігання інформації про страви, замовлення та користувачів. Вони можуть створювати таблиці, виконувати запити до бази даних та здійснювати необхідні операції для забезпечення функціональності чат-бота.

PyCharm - це інтегроване середовище розробки (IDE) для Python, яке забезпечує розробникам зручні інструменти для програмування та налагодження чат-ботів по доставці харчування. Воно має багатий функціонал, такий як автодоповнення коду, візуальне налагодження, керування проектами та інтеграцію з системами контролю версій. PyCharm надає зручну розробку та налагодження коду Python для чат-ботів. Розробники можуть створювати модулі, функції та класи, реалізовувати логіку замовлень, обробляти вхідні дані користувачів та взаємодіяти з базою даних SQLite. PyCharm також підтримує розробку у сумісності з іншими фреймворками, такими як Flask або Django, що дозволяє розширювати можливості чат-бота та підключати додаткові функціональність.

З використанням Python, SQLite та PyCharm, розробники можуть ефективно створювати чат-боти по доставці харчування, забезпечуючи зручний інтерфейс, взаємодію з базою даних та ефективну обробку замовлень користувачів.

У чат-боті для доставки їжі здійснення способу оплати може бути реалізовано за допомогою Python і відповідних інструментів. Перш за все, можна створити функцію або модуль, що оброблює оплату, використовуючи зовнішні платіжні системи або API платіжних провайдерів. Наприклад, можна використати бібліотеку, яка забезпечує інтеграцію з платіжними шлюзами, такими як Stripe, PayPal або Braintree. Далі, після отримання даних оплати від користувача, ці дані можна обробити та перевірити їх правильність. Після успішної перевірки оплати, можна зберегти дані про платіж у базі даних, а також відправити підтвердження оплати користувачеві через чат-бот. В разі невдалої операції оплати, можна надіслати відповідне повідомлення та запропонувати інший спосіб оплати.

РОЗДІЛ 3

ЕТАПИ РОЗРОБКИ

3.1. Створення чат-бота

Метою є розробка та створення чат-боту для початку роботи із користувачами. Для цього використаємо батька усіх чат-ботів **BotFather**(це простий і швидкий спосіб розробки чат-бота на платформі Telegram). Щоб почати, необхідно створити бота в Telegram і отримати API-ключ від BotFather. За допомогою отриманого API-ключа, можна налаштувати взаємодію бота з користувачами.

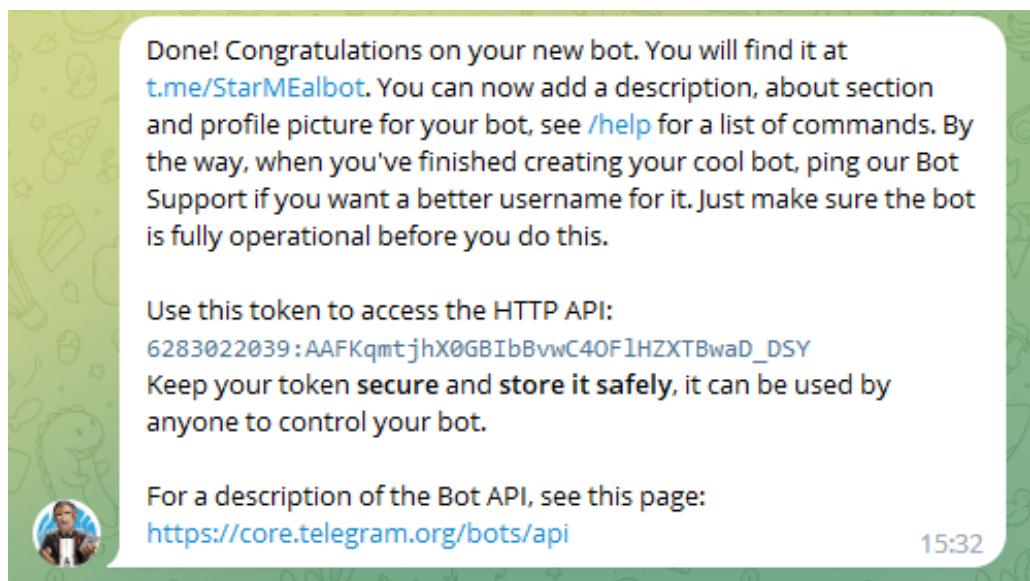


Рис. 3.1 Отримання токenu

Коли отримали токен від чат-бота, потрібно вставити в код у місце, де відбувається ініціалізація бота:

```
def main():
    TOKEN = '6283022039:AAFkqmtjhX0GBIbBvwC40F1HZXTBwaD_DSY'
```

Рис. 3.1 Вставлення токenu

Також за допомогою **BotFather** змінюємо зовнішній вигляд чат-бота для доставки харчування. А це можна зробити перейшовши до самого бота та у розділі “ ” змінити параметри і в кінцевому результаті отримуємо такий вигляд бота:

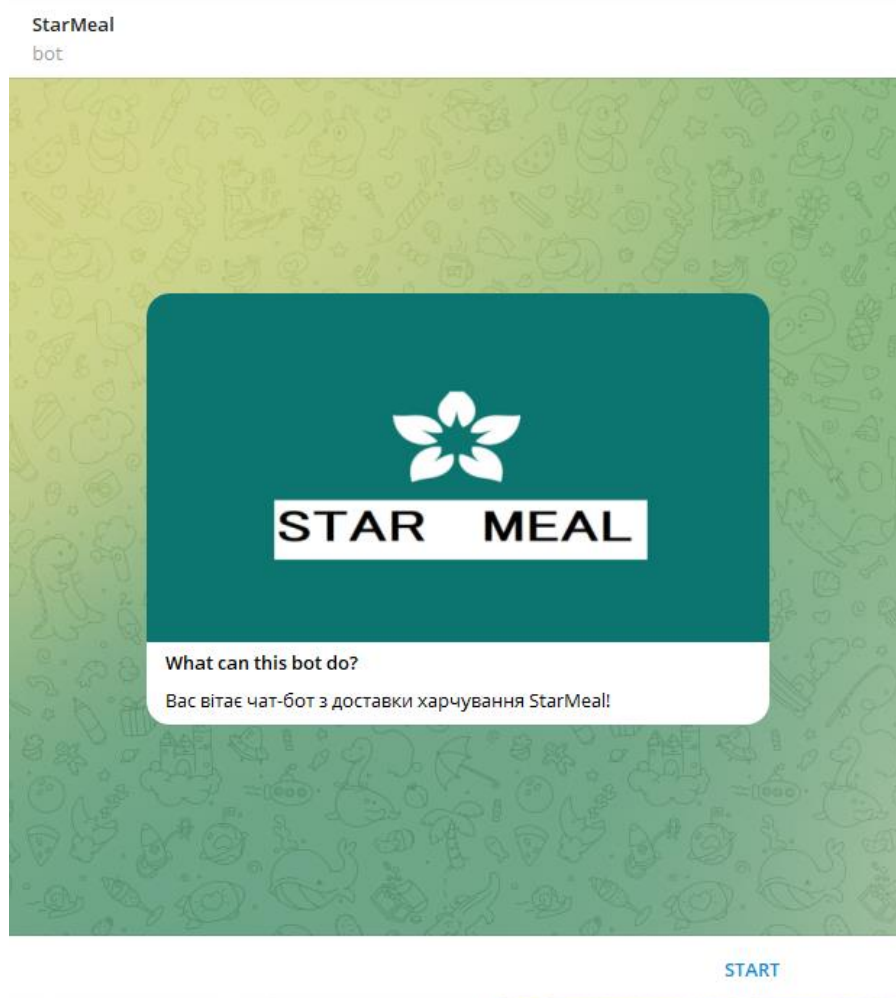


Рис. 3.3 Головне меню

3.2. Розробка інтерфейсу

Для кращого вигляду інтерфейсу спочатку створив 5 inline кнопок(Меню, Корзина, Профіль, Контакти, Про нас)

```
def start(update, context):

    keyboard = [
        [InlineKeyboardButton("Меню", callback_data='menu')],
        [InlineKeyboardButton("Корзина", callback_data='cart')],
        [InlineKeyboardButton("Профіль", callback_data='profile')],
        [InlineKeyboardButton("Контакти", callback_data='contacts')],
        [InlineKeyboardButton("Про нас", callback_data='about')],
    ]

    reply_markup = InlineKeyboardMarkup(keyboard)
```

Рис. 3.4 Inline кнопки

А ось так вони виглядають у самому боті:



Рис. 3.5 Меню користувача

Тепер розглянемо, які функції та дії відбудуться після натискання на кожну з цих п'яти кнопок:

1. Після натискання кнопки "Меню" можна побачити наступне:



Рис. 3.6 Розклад меню

А тепер розглянемо програмну реалізацію цієї частини:

```
def category_menu(update, context):
    query = update.callback_query
    query.answer()

    if query.data == 'pizza':
        query.edit_message_text(text="Ви обрали категорію Піца")
    elif query.data == 'sushi':
        query.edit_message_text(text="Ви обрали категорію Суші")
    elif query.data == 'pasta':
        query.edit_message_text(text="Ви обрали категорію Паста")
    elif query.data == 'burgers':
        query.edit_message_text(text="Ви обрали категорію Бургери")
    elif query.data == 'drinks':
        query.edit_message_text(text="Ви обрали категорію Напої")
```

- Після натискання кнопки "Мої замовлення" можна побачити дату та ціну, якщо користувач додав страву чи напій до замовлення, а також можна повернутись на попередню сторінку натиснувши кнопку "Назад":

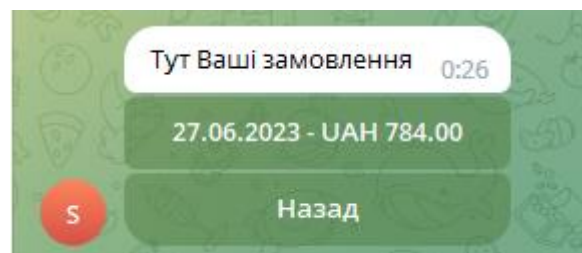


Рис. 3.7 Ваші замовлення

І також розглянемо програмну реалізацію:

```
class CartItem:
    def __init__(self, date, price):
        self.date = date
        self.price = price

class ShoppingCart:
```

```

def __init__(self):
    self.items = []

def add_item(self, item):
    self.items.append(item)

def remove_item(self, item):
    self.items.remove(item)

def calculate_total_price(self):
    total_price = 0
    for item in self.items:
        total_price += item.price
    return total_price

cart = ShoppingCart()

item1 = CartItem("2023-06-28", 50)
item2 = CartItem("2023-06-29", 30)
cart.add_item(item1)
cart.add_item(item2)

total_price = cart.calculate_total_price()
print("Total price in UAH:", total_price)

```

3. Після натискання кнопки "Контакти" можна дізнатись мобільний телефон персоналу, також пошту, на яку можна відправити повідомлення та ПІБ менеджера, що допоможе Вам у разі виникнення питань або проблем.

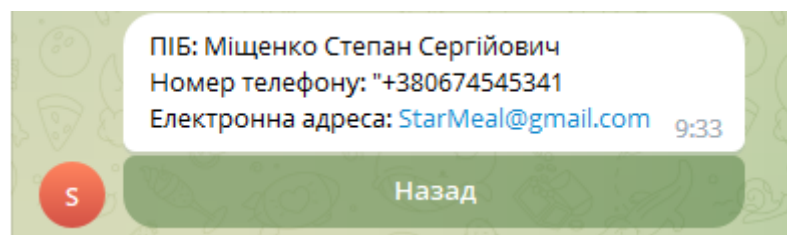


Рис. 3.8 Профіль користувача

А також подивимось, як виглядає це у коді:

```

def show_contact_info():
    pib = "Міщенко Артем Сергійович"
    phone_number = "+380674545341"
    email = "StarMeal@gmail.com"

    contact_info = f"ПІБ: {pib}\nНомер телефону: {phone_number}\nЕлектронна\nадреса: {email}"
    messagebox.showinfo("Контакти", contact_info)

```

4. Далі натискаємо на кнопку "Про нас" і бачимо загальну інформацію і також можемо повернутись на головне меню:

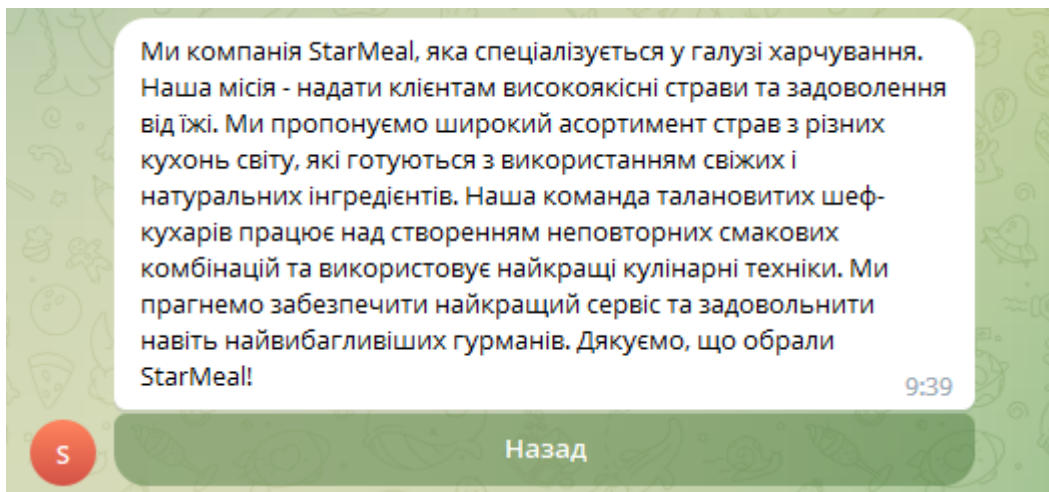


Рис. 3.9 Про нас

Програмна реалізація:

```
def show_about_us():
    about_text = "Ми компанія StarMeal, яка спеціалізується у галузі харчування.
    Наша місія - надати клієнтам високоякісні страви та задоволення від їжі. Ми
    пропонуємо широкий асортимент страв з різних кухонь світу, які готуються з
    використанням свіжих і натуральних інгредієнтів. Наша команда талановитих шеф-
    кухарів працює над створенням неповторних смакових комбінацій та використовує
    найкращі кулінарні техніки. Ми прагнемо забезпечити найкращий сервіс та
    задовольнити навіть найвибагливіших гурманів. Дякуємо, що обрали StarMeal!"

    messagebox.showinfo("Про нас", about_text)

about_button = Button(window, text="Про нас", command=show_about_us)
about_button.pack()
```

Далі підключення бази даних. Для роботи з базою SQLite застосовується бібліотека `sqlite3`, яка дає змогу підключити базу даних до чат бота та вносити дані у таблиці.

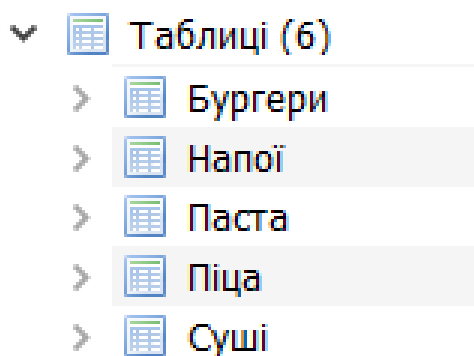


Рис. 3.10 База даних товару

Програмна реалізація бази даних:

```
import sqlite3

conn = sqlite3.connect('menu.db')
cursor = conn.cursor()

cursor.execute('''
    CREATE TABLE IF NOT EXISTS pizza (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT,
        ingredients TEXT,
        price REAL,
        quantity INTEGER,
        image BLOB
    )
''')

cursor.execute('''
    CREATE TABLE IF NOT EXISTS sushi (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT,
        ingredients TEXT,
        price REAL,
        quantity INTEGER,
        image BLOB
    )
''')

cursor.execute('''
    CREATE TABLE IF NOT EXISTS pasta (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT,
        ingredients TEXT,
        price REAL,
        quantity INTEGER,
        image BLOB
    )
''')

cursor.execute('''
    CREATE TABLE IF NOT EXISTS burgers (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT,
        ingredients TEXT,
        price REAL,
        quantity INTEGER,
        image BLOB
    )
''')

cursor.execute('''
    CREATE TABLE IF NOT EXISTS drinks (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT,
        ingredients TEXT,
        price REAL,
        quantity INTEGER,
        image BLOB
    )
''')
```



```
conn.commit()
conn.close()
```

Також можна знайти всі товари в режимі швидкого пошуку натиснувши кнопку "Швидкий пошук":

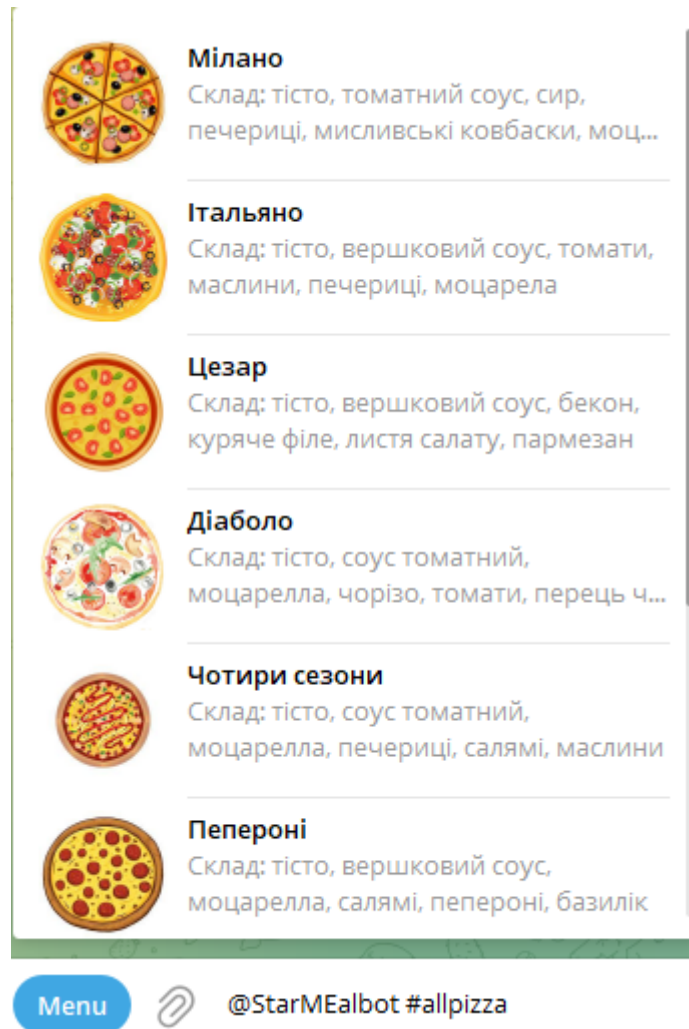


Рис. 3.11 Швидкий пошук страв

Програмна реалізація:

```
def search_product(message):
    search_query = message.text.lower()
    results = []

    for product in products:
        if search_query in product['name'].lower():
            results.append(product)
```

3.2.1. Обробка платежів

Описатиму загальний підхід до обробки платежів через PayPal та відправки PDF-файлу з даними замовлення користувачу. Будь ласка, зверніть

увагу, що реалізація цих функціональностей вимагає детальної інтеграції з сервісом PayPal та наявності системи для генерації PDF-файлів. Однак, я надам загальну концепцію і декілька кроків, які можна виконати для реалізації цих функцій:

Інтеграція з PayPal:

- Зареєструйтеся в сервісі PayPal та отримайте доступ до API-інтерфейсів.
- Налаштуйте ваші налаштування розрахунків та отримання платежів через PayPal.
- Використовуйте документацію PayPal API для створення функцій для обробки платежів, отримання підтверджень та зберігання інформації про транзакції.

Генерація PDF-файлу:

- Встановіть необхідну бібліотеку для генерації PDF-файлів, наприклад, ReportLab або PyFPDF.
- Створіть функцію, яка бере дані замовлення та генерує PDF-файл з необхідною інформацією. Ви можете додати дані про замовлення, включаючи назви продуктів, кількість, ціни, адресу доставки тощо.
- Збережіть згенерований PDF-файл на сервері або в обліковому сховищі.

3.2.2. Профіль користувача

Для того щоб користувач потрапив у свій профіль, потрібно натиснути на кнопку "Профіль" і можемо бачити, що після здійснення замовлення персональні дані такі як:

- Ім'я
- Номер телефона
- Адреса доставки

Зберігаються автоматично після натиснення кнопки "Профіль":

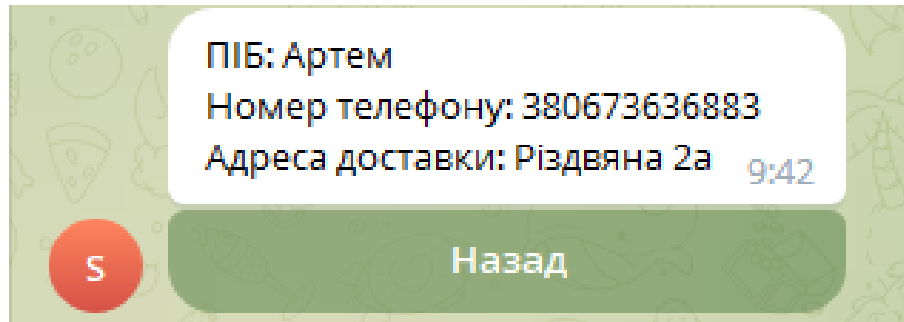


Рис. 3.12 Занесення даних користувача

І програмна реалізація цієї частини:

```
def show_profile():
    name = "Ім'я: Артем"
    phone_number = "Номер телефону: 380673636883"
    delivery_address = "Адреса доставки: вул. Різдяна 2а"

    profile_info = f"{name}\n{phone_number}\n{delivery_address}"
    messagebox.showinfo("Профіль", profile_info)
```

А також є ще база даних, куди записуються персональні дані користувачів:

Імя	Номер телефону	Адреса доставки
Фільтр	Фільтр	Фільтр
1	Артем	Різдяна 2а
2	Аня	Соборна 5
3	Степан	Степана Бандери 1
4	Олена	Проспект Миру 15

Рис. 3.13 Користувачі у базі даних

І програмна реалізація створення таблиці "Користувачі" та додавання їх до таблиці:

```
conn = sqlite3.connect('users.db')
cursor = conn.cursor()

cursor.execute('''
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT,
    phone_number TEXT,
    delivery_address TEXT
)
''')
```

```
name = "Артем "  
phone_number = "+380673636883"  
delivery_address = "Різдвяна 2а"  
  
# Додавання користувача до таблиці  
cursor.execute('''  
    INSERT INTO users (name, phone_number, delivery_address)  
    VALUES (?, ?, ?)  
''', (name, phone_number, delivery_address))  
  
conn.commit()  
  
conn.close()
```

3.2.3. Кабінет адміністратора

У власному кабінеті адміністратор має можливість побачити замовлення, яке зробив користувач та написати користувачу, для уточнення деталей, якщо це потрібно:

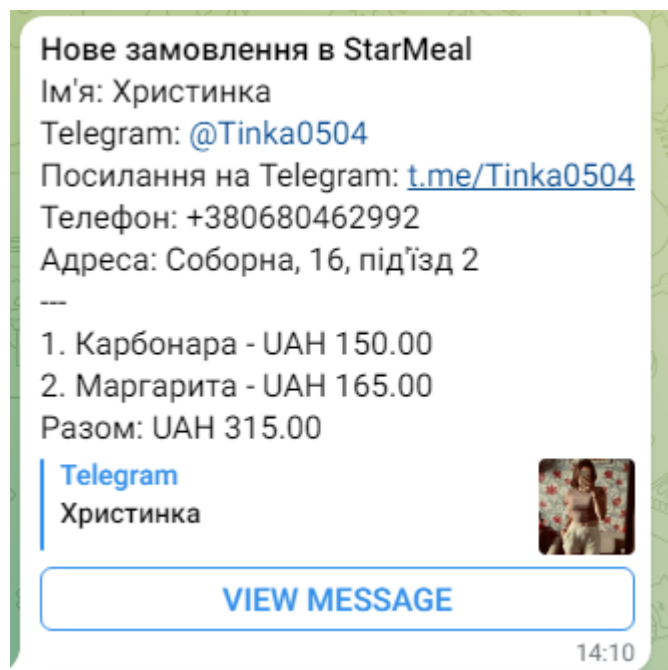


Рис. 3.14 Кабінет адміністратора

РОЗДІЛ 4

ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1. Вступ

Мета полягає в ознайомленні із головним функціоналом користувача у чат-боті, взаємодії його з інтерфейсом користувача та створення замовлення.

4.2. Головна сторінка

Коли користувач переходить до чат-боту з доставки харчування, він вітається головною сторінкою та навігаційною панеллю, які дозволяють йому легко переміщатися між основними елементами інтерфейсу та опціями, доступними для нього. (рис. 4.1 – 4.3)

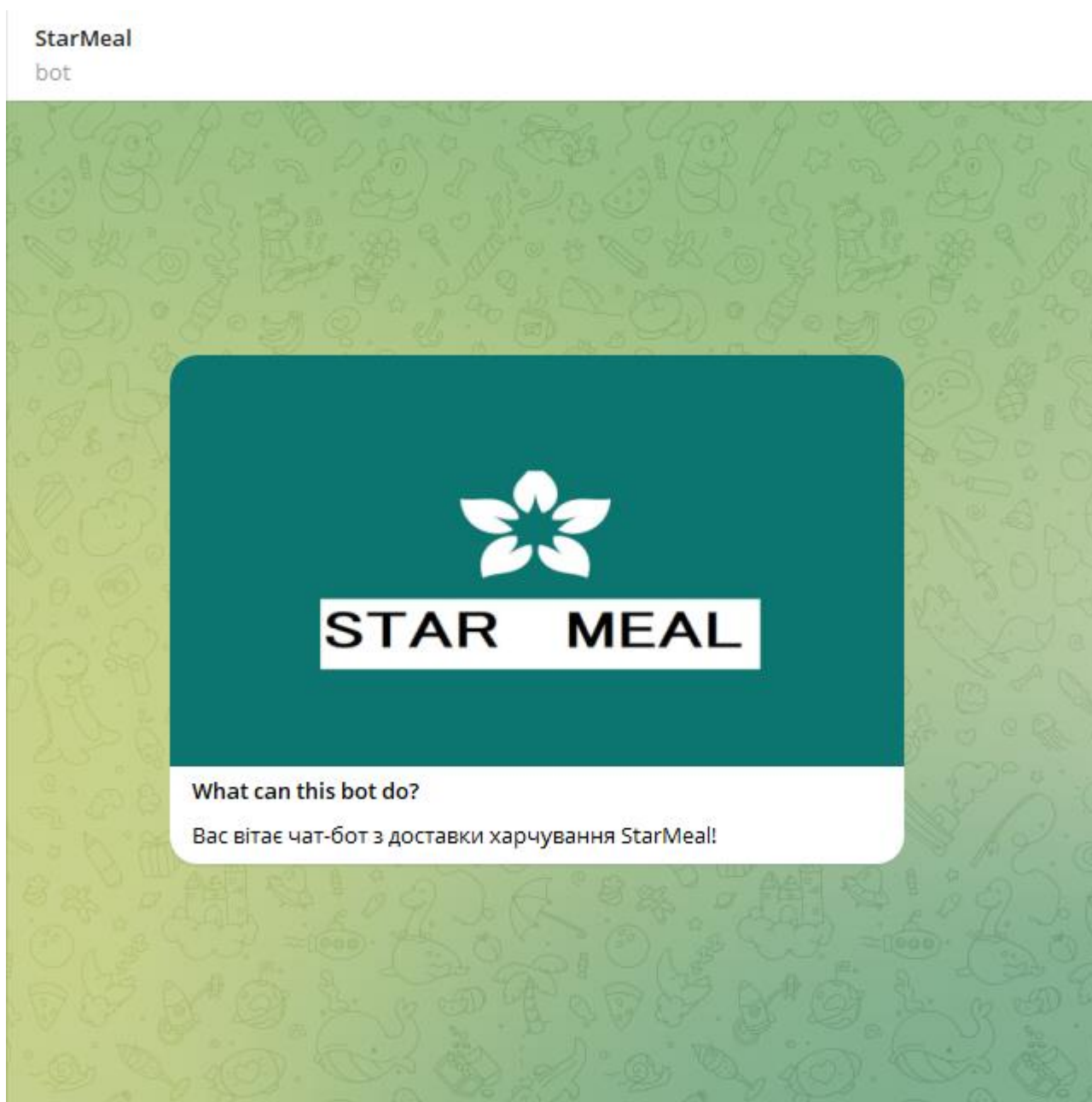


Рис. 4.1. Головна сторінка

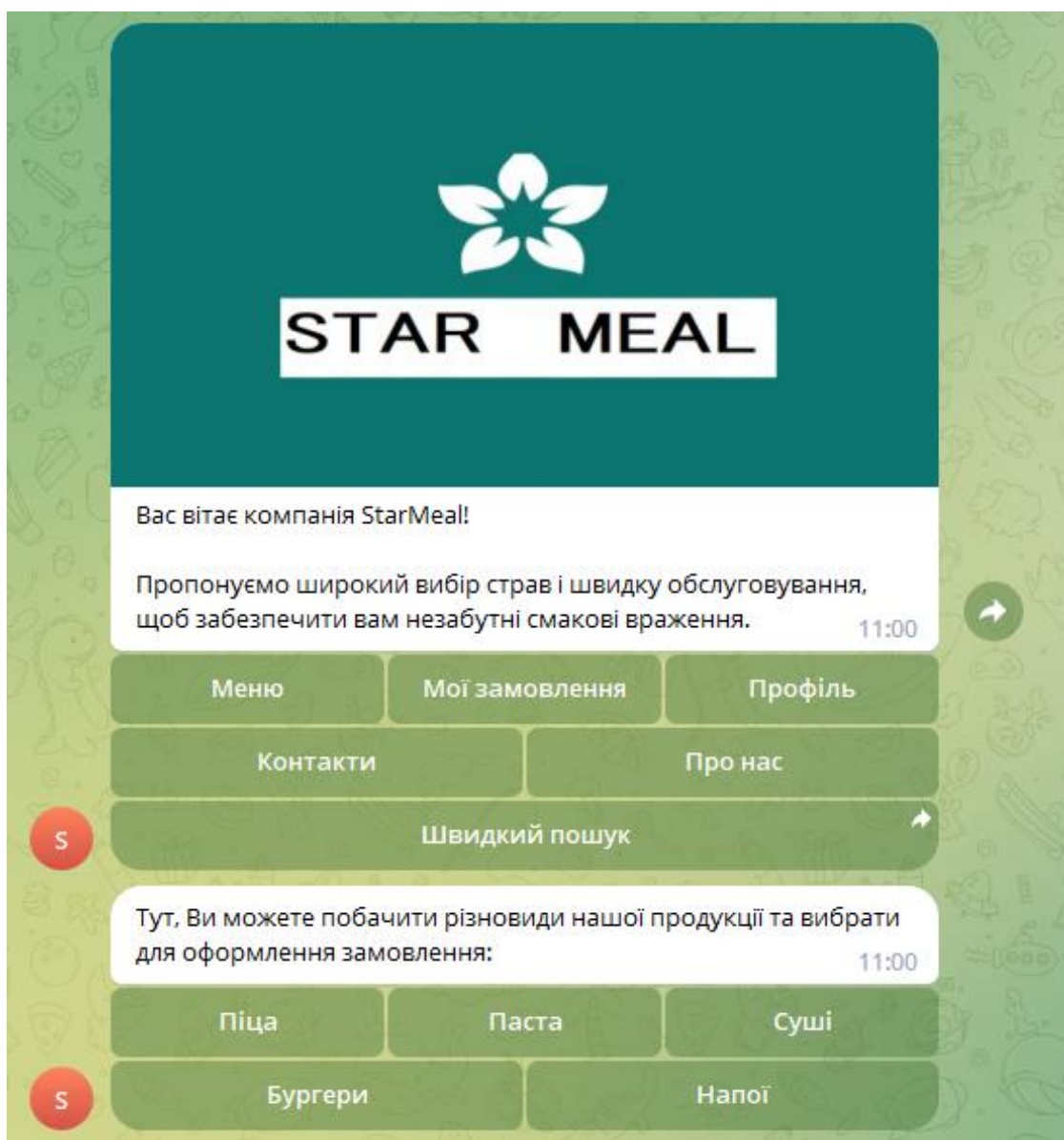


Рис. 4.2. Меню

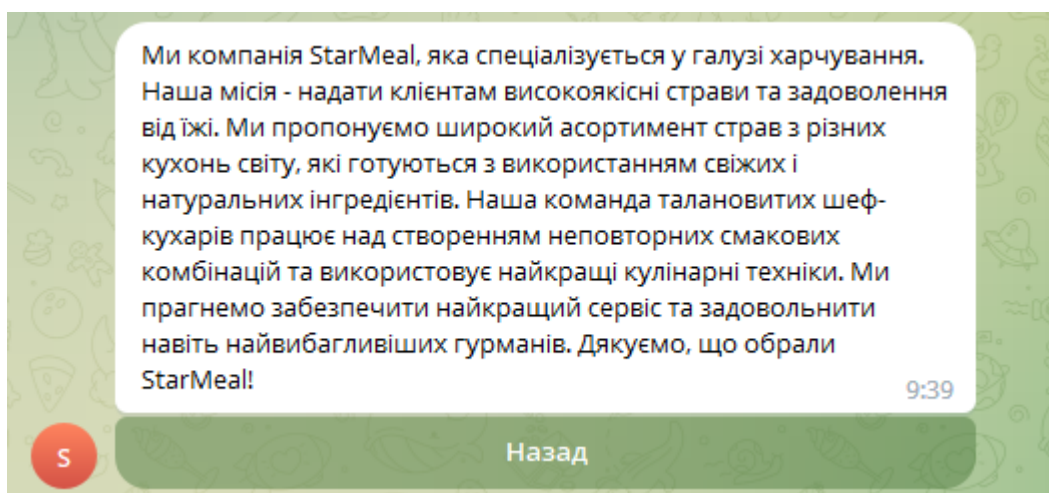


Рис. 4.3. Про нас

4.3. Створення замовлення

Створення замовлення відбувається у розділі меню. Для переходу в нього потрібно почати роботу з чат-ботом натиснувши кнопку “/start” та після успішного запуску з’явиться меню, де зможете вибрати відповідний товар та додати його до кошику. (рис. 4.8)



Рис. 4.4. Запуск чат-бота

Після чого користувач направляється в головне меню. (рис. 4.5)

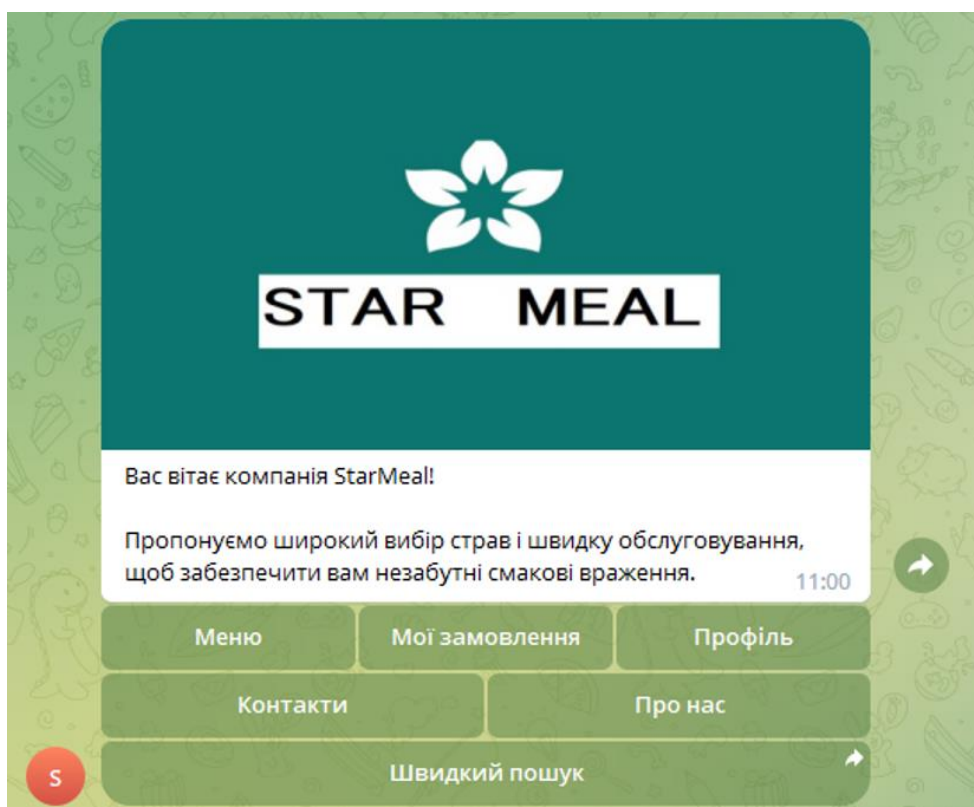


Рис. 4.5

Для створення замовлення потрібно вибрати товар після натискання одного із розділів на рис. 4.6. Для прикладу натиснемо на розділ «Піца», який містить 3 сторінки з товарами і виберемо відповідну страву 4.7.

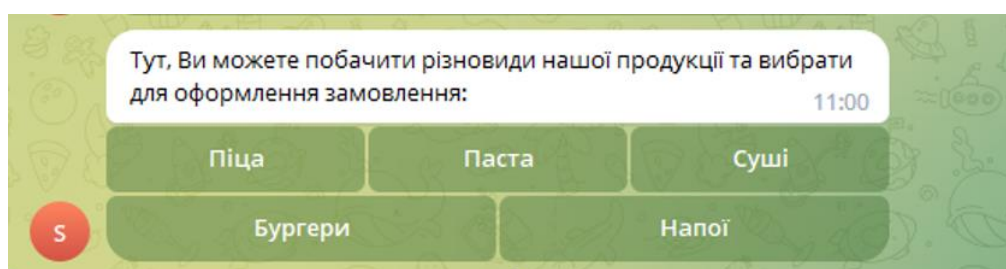


Рис. 4.6. Меню товару



Рис. 4.7. Меню товару Піца 1 сторінка

А щоб перейти на 2 сторінку або 3 потрібно написати у чаті цифру 2 або 3 і користувач має можливість перейти на відповідну сторінку(рис. 4.8-4.9).

Назва: Чотири сезони
Склад: тісто, соус томатний, моцарелла, печериці, селямі, маслини
ЦАН 200.00 13:07

Додати до кошика

Орен



13:07

Назва: Діаболо
Склад: тісто, соус томатний, моцарелла, чорізо, томати, перець чилі
ЦАН 167.00 13:07

Додати до кошика

Орен

Сторінка 2 з 3 13:07

Перейти до корзини

Рис. 4.8. Меню товару Піца 2 сторінка

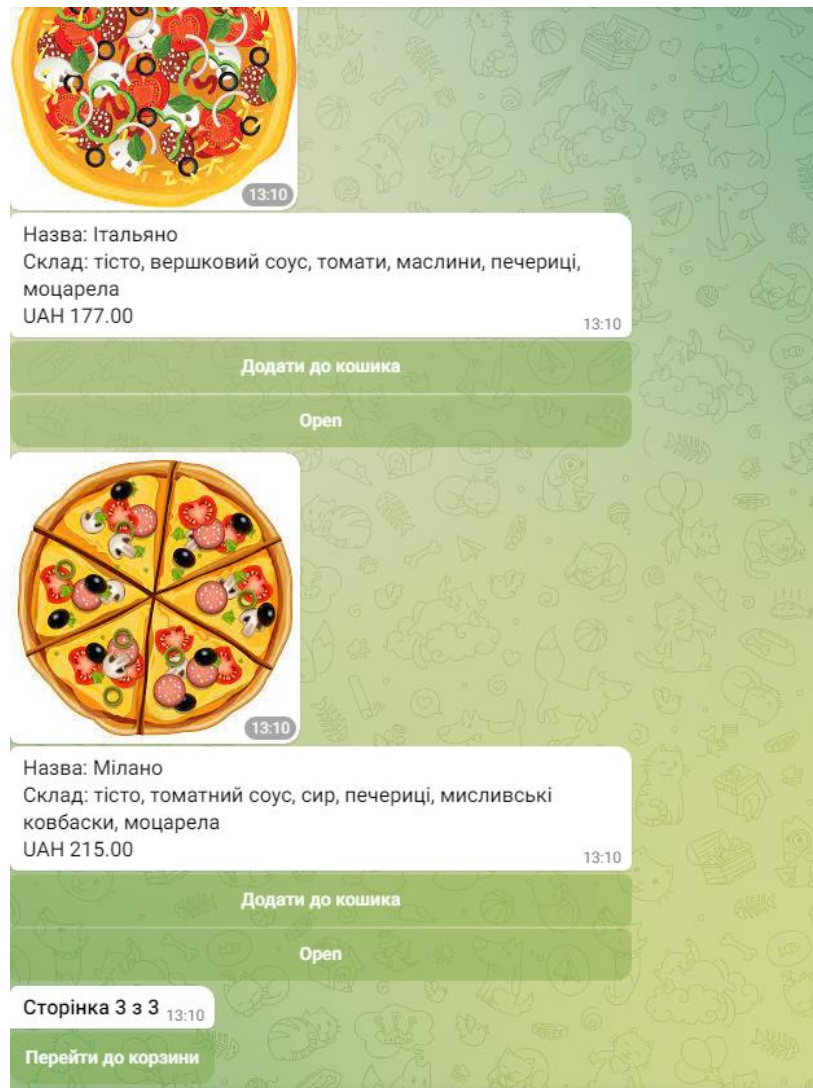


Рис. 4.9. Меню товару Піца 3 сторінка

Після чого вибираємо піцу, також можемо змінити кількість її та додаємо до корзини.(4.8, 4.9)

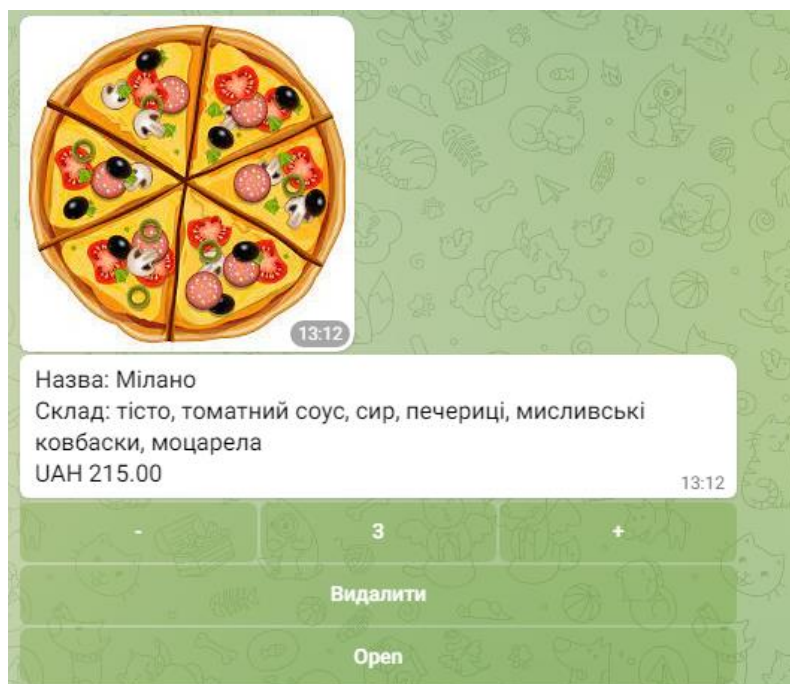


Рис. 4.8. Обрання товару з вказаною кількістю страви



Рис. 4.9. Додавання товару до корзини

При успішному додаванні товару, користувач повинен натиснути кнопку "Перейти до кошика, де буде вказано товар який вибрав користувач та ціну за замовлення"(рис. 4.10-4.12). Також ми можемо редагувати кількість і відразу оновлювати ціну замовлення, як показано на рис. 4.13, 4.14.

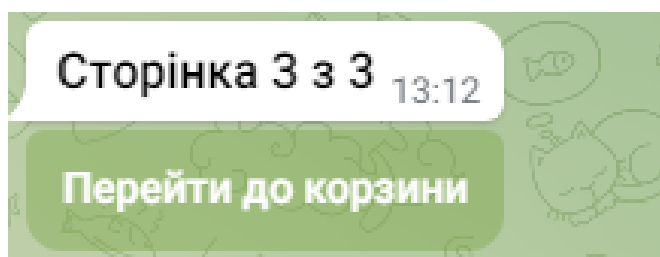


Рис. 4.10. Перехід до корзини



Рис. 4.11. Вибраний товар

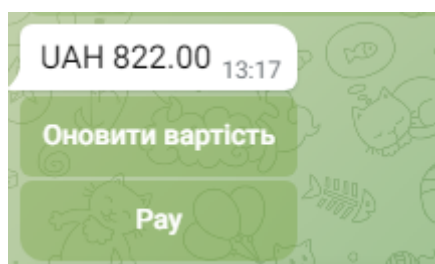


Рис. 4.12. Ціна замовлення



Рис. 4.13. Зміна кількості товару

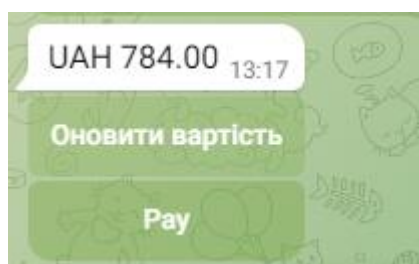


Рис. 4.14. Оновлення ціни замовлення

Після того як товар був доданий до кошика користувач має можливість перейти до оформлення замовлення натиснувши кнопку “Pay”. Для початку чат-бот запитає номер телефону та адресу доставки(4.15, 4.16), а після цього надішле замовлення користувача у pdf-файлі, як зображено на рис. 4.17:

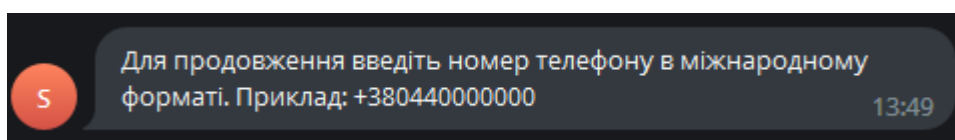


Рис. 4.15. Номер телефону

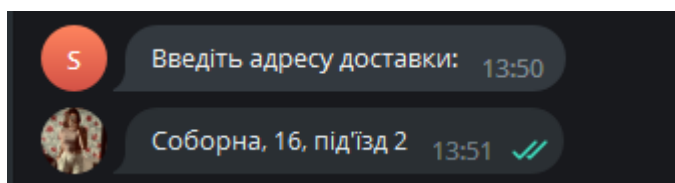


Рис. 4.16. Адреса доставки


				
From		For		
StarMeal		Христинка Соборна, 16, під'їзд 2 +380680462992		
Charges				
№	Продукт / Послуга	Вартість	Кількість	Сума
1	Діаболо	UAH 167.00	1.00	UAH 167.00
2	Чотири сезони	UAH 200.00	2.00	UAH 400.00
3	Пепероні	UAH 155.00	3.00	UAH 465.00
Subtotal:				UAH 1 032.00
Total:				UAH 1 032.00

Рис. 4.17. Замовлення у pdf-файлі

І після завершення замовлення користувача автоматично відправляє до головного меню чат-бота(рис. 4.18).

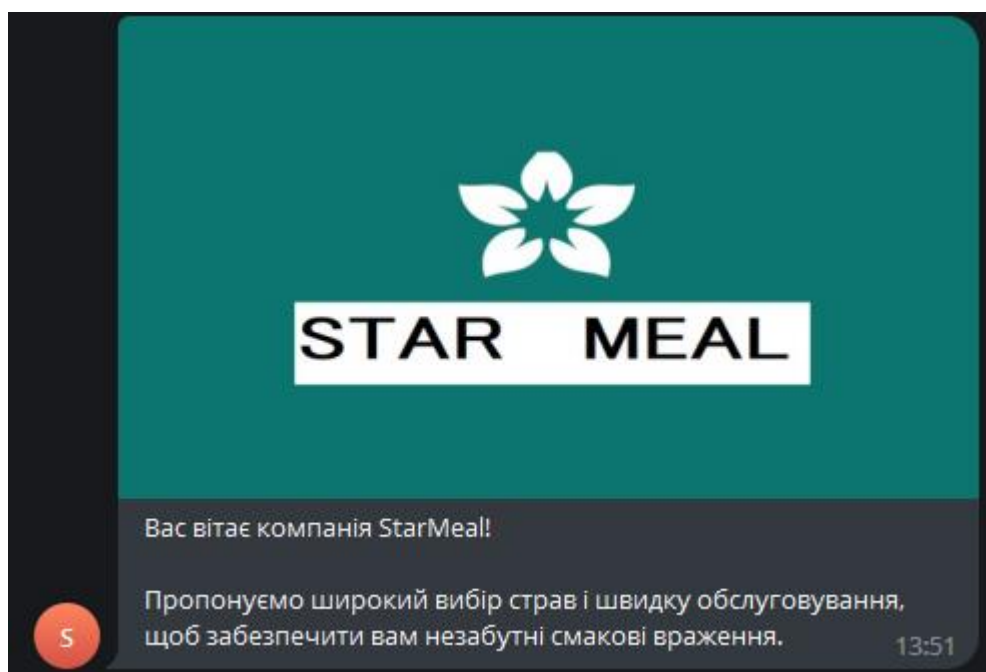


Рис. 4.18. Повернення до головного меню

ВИСНОВОК

У результаті виконання кваліфікаційної роботи було розроблено чат-бота з доставкою харчування по місту. Цей проект є інноваційним рішенням, яке сприяє зручності та швидкості замовлення їжі для мешканців міста. Застосування чат-бота у сфері харчування відкриває нові можливості для клієнтів, ресторанів і кур'єрських служб.

Зазначена розробка показала потенціал чат-ботів у сфері доставки харчування. Вони дозволяють забезпечити зручність, швидкість і персоналізацію обслуговування для клієнтів, а також оптимізувати процеси ресторанів та кур'єрських служб. Потенційний успіх таких рішень залежить від надійності технологій, ефективного управління даними та забезпечення безпеки в процесі обробки особистої інформації клієнтів. З метою подальшого розвитку і вдосконалення чат-ботів по доставці харчування, необхідно проводити додаткові дослідження та впроваджувати інноваційні технології.

У процесі реалізації та тестування чат-боту виявлено високий рівень професіоналізму та пристягнутості розробника. Він успішно виконав всі завдання, пов'язані з розробкою та втіленням функціональних та естетичних вимог. Розробник встановив високі стандарти якості, що дозволило реалізувати проект на високому рівні. Особливо важливим етапом був процес тестування, під час якого виявлено та усунуто функціональні та візуальні дефекти. Розробник зосередився на постійному вдосконаленні та оптимізації чат-боту з метою забезпечення максимально позитивного користувацького досвіду. Враховуючи отримані відгуки та рекомендації, було досягнуто високої функціональності та задоволення потреб користувачів. В цілому, успішна реалізація та тестування чат-боту підтверджують ефективність та компетентність розробника. Він продемонстрував свої навички у розробці програмного забезпечення, уважність до деталей та готовність до постійного вдосконалення. Результатом цих зусиль став високоякісний та функціональний чат-бот, який задовольняє потреби користувачів у зручному та ефективному замовленні доставки харчування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Офіційний сайт Python: [електронний ресурс] - <https://www.python.org/>
2. Python.org документація: [електронний ресурс] – <https://docs.python.org/3/>
3. Stack Overflow: [електронний ресурс] - <https://stackoverflow.com/questions/tagged/python>
4. Python for Beginners (Python для початківців): [електронний ресурс] - <https://www.python.org/about/gettingstarted/>
5. Real Python: [електронний ресурс] - <https://realpython.com/>
6. Офіційна документація SQLite: [електронний ресурс] - <https://www.sqlite.org/docs.html>
7. SQLite Tutorial - Tutorialspoint: [електронний ресурс] - <https://www.tutorialspoint.com/sqlite/index.htm>
8. SQLite на Stack Overflow: [електронний ресурс] - <https://stackoverflow.com/questions/tagged/sqlite>
9. SQLite відеоуроки на YouTube: [електронний ресурс] - https://www.youtube.com/results?search_query=sqlite+tutorial
10. SQLite веб-сайт: [електронний ресурс] - <https://www.sqlite.org/>
11. Telegram Bot API - API Telegram для створення та розробки чат-ботів: [електронний ресурс] - <https://core.telegram.org/bots/api>
12. "How Chatbots are Transforming the Food Delivery Industry" - [електронний ресурс] - <https://chatbotmagazine.com/how-chatbots-are-transforming-the-food-delivery-industry-6817d55dc37c>
13. "The Rise of Chatbots in Food Delivery Services" [електронний ресурс] - <https://dinarys.com/blog/the-rise-of-chatbots-in-food-delivery-services>
14. "Chatbot Technology and Its Impact on Food Delivery Services" [електронний ресурс] - <https://www.zendesk.com/blog/chatbot-technology-impact-food-delivery-services/>

15. "How Chatbots Are Revolutionizing the Food Delivery Business" [электронный ресурс] - <https://chatbotslife.com/how-chatbots-are-revolutionizing-the-food-delivery-business-3c976e9efb5a>
16. "Chatbots in Food Delivery: The Future is Here" [электронный ресурс] - <https://www.oneteam.us/chatbots-in-food-delivery-the-future-is-here/>
17. "The Benefits of Using Chatbots in Food Delivery" [электронный ресурс] - <https://clustaar.com/blog/benefits-of-using-chatbots-in-food-delivery/>
18. "Chatbots in the Food and Beverage Industry: Improving Customer Experience" [электронный ресурс] - <https://www.aiupnow.com/2021/01/chatbots-in-food-and-beverage-industry.html>

ДОДАТКИ

Додаток А. Код чат-бота

```

from flask import Flask, request
import sqlite3

app = Flask(__name__)

conn = sqlite3.connect('database.db')
cursor = conn.cursor()

def get_products(category):

    cursor.execute("SELECT * FROM products WHERE category=?", (category,))
    products = cursor.fetchall()
    return products

def get_user_info(user_id):

    cursor.execute("SELECT * FROM users WHERE user_id=?", (user_id,))
    user_info = cursor.fetchone()
    return user_info

def show_menu():

    return "Виберіть категорію товару:", ["Піца", "Суші", "Паста", "Бургери", "Напої"]

def handle_menu_choice(category):

    products = get_products(category)
    return "Товари у категорії " + category + ":", products

def show_my_orders(user_id):

    user_info = get_user_info(user_id)
    if user_info:
        orders = ...
        return "Ваші замовлення:", orders
    else:
        return "Ви не маєте жодного замовлення."

conn = sqlite3.connect('food_delivery.db')
cursor = conn.cursor()

cursor.execute("CREATE TABLE IF NOT EXISTS pizza
              (id INTEGER PRIMARY KEY AUTOINCREMENT,
               name TEXT,
               description TEXT,
               price REAL)")

pizza_data = [
    ("Маргарита", "Класична піца з томатним соусом, сиром Моцарелла та базиліком.", 10.99),

```

```

("Пепероні", "Піца з гострим салями Пепероні, томатним соусом та сиром Моцарелла.", 12.99),
("Гавайська", "Піца з куркою, ананасами, томатним соусом та сиром Моцарелла.", 11.99),
]

```

```

cursor.executemany("INSERT INTO pizza (name, description, price) VALUES (?, ?, ?)", pizza_data)
conn.commit()

```

```

cursor.execute("""CREATE TABLE IF NOT EXISTS sushi
                (id INTEGER PRIMARY KEY AUTOINCREMENT,
                 name TEXT,
                 description TEXT,
                 price REAL)""")

```

```

sushi_data = [
    ("Філадельфія", "Суші з лососем, огірком та сиром Філадельфія.", 9.99),
    ("Каліфорнія", "Суші з крабовим м'ясом, авокадо та огірком.", 8.99),
    ("Унагі", "Суші з вугрем, соєвим соусом та ікрою.", 11.99),
]

```

```

cursor.executemany("INSERT INTO sushi (name, description, price) VALUES (?, ?, ?)", sushi_data)
conn.commit()

```

```

cursor.execute("""CREATE TABLE IF NOT EXISTS burgers
                (id INTEGER PRIMARY KEY AUTOINCREMENT,
                 name TEXT,
                 description TEXT,
                 price REAL)""")

```

```

burger_data = [
    ("Чізбургер", "Соковитий бургер зі свинячим фаршем, сиром та соусом.", 9.99),
    ("Бекон-чеддер", "Бургер зі смаженим беконом, сиром Чеддер та соусом BBQ.", 11.99),
    ("Грильований курячий", "Бургер з грильованим курячим філе, свіжими овочами та майонезом.", 10.99),
]

```

```

cursor.executemany("INSERT INTO burgers (name, description, price) VALUES (?, ?, ?)", burger_data)
conn.commit()

```

```

cursor.execute("""CREATE TABLE IF NOT EXISTS pasta
                (id INTEGER PRIMARY KEY AUTOINCREMENT,
                 name TEXT,
                 description TEXT,
                 price REAL)""")

```

```

pasta_data = [
    ("Болоньезе", "Класична паста з м'ясним соусом Болоньезе.", 11.99),
    ("Карбонара", "Паста з млинцями бекону, яйцем та сиром Пармезан.", 10.99),
    ("Альфредо", "Паста з вершковим соусом Альфредо та курячим філе.", 12.99),
]

```

```

cursor.executemany("INSERT INTO pasta (name, description, price) VALUES (?, ?, ?)", pasta_data)
conn.commit()

```

```

cursor.execute("""CREATE TABLE IF NOT EXISTS drinks
                (id INTEGER PRIMARY KEY AUTOINCREMENT,
                 name TEXT,
                 description TEXT,
                 price REAL)""")

drinks_data = [
    ("Кока-Кола", "Газований безалкогольний напій з колою.", 2.99),

    ("Сік", "Свіжо вишатий сік з різних фруктів та ягід.", 3.99),

    ("Морс", "Натуральний морс з різних ягід та фруктів.", 3.99),

]

cursor.executemany("INSERT INTO drinks (name, description, price) VALUES (?, ?, ?)", drinks_data)
conn.commit()

conn.close()

def show_contacts():

    return "Контакти:\nПІБ: Міщенко Степан Сергійович\nНомер телефону: +380674545341\nЕлектронна адреса: StarMeal@gmail.com"

def show_profile(user_id):

    user_info = get_user_info(user_id)
    if user_info:

        return "ПІБ: " + user_info[1] + "\nНомер телефону: " + user_info[2] + "\nАдреса доставки: " + user_info[3]
    else:
        return "Профіль не знайдено."

def show_about_us():

    return "Ми компанія StarMeal, яка спеціалізується у галузі харчування. ..."

def show_all_products():
    products = ...
    return "Весь товар:", products

def process_message(user_id, message):
    if message == "Меню":
        return show_menu()
    elif message == "Мої замовлення":
        return show_my_orders(user_id)

```

```
elif message == "Контакти":
    return show_contacts()
elif message == "Профіль":
    return show_profile(user_id)
elif message == "Про нас":
    return show_about_us()
elif message == "Швидкий пошук":
    return show_all_products()
else:
    return "Невідома команда. Виберіть одну з доступних кнопок."

@app.route('/message', methods=['POST'])
def handle_message():
    data = request.get_json()
    user_id = data['user_id']
    message = data['message']
    response = process_message(user_id, message)
    return response

if __name__ == '__main__':
    app.run()
```

Додаток Б. Приклад коду здійснення платежу

```
import stripe

stripe.api_key = '6146691368:AAH69XRMfAp1KdtWBgdqcv5h_rpYXYn9O4'

def process_payment(amount, token):
    try:
        payment_intent = stripe.PaymentIntent.create(
            amount=amount,
            currency='usd',
            payment_method_types=['card'],
            payment_method=token,
            confirm=True
        )

        if payment_intent.status == 'succeeded':
            return "Платіж успішно здійснено."
        else:
            return "Платіж не вдалося здійснити."
    except stripe.error.StripeError as e:
        return str(e)

payment_amount = 1000
payment_token = '8765466765:AAH69XRMfAp1KdtWBgdqcv5h_rpYXYn9O4'
result = process_payment(payment_amount, payment_token)
print(result)
```

Додаток В. Приклад коду користувацького інтерфейсу

```
conn = sqlite3.connect('database.db')
cursor = conn.cursor()

def create_users_table():
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS users (
            user_id INTEGER PRIMARY KEY AUTOINCREMENT,
            username TEXT,
            phone_number TEXT,
            address TEXT
        )
    """)
```



```
    ")
    conn.commit()

def add_user(username, phone_number, address):
    cursor.execute("""
        INSERT INTO users (username, phone_number, address) VALUES (?, ?, ?)
    """, (username, phone_number, address))
    conn.commit()

def get_user_info(user_id):
    cursor.execute("""
        SELECT * FROM users WHERE user_id = ?
    """, (user_id,))
    user = cursor.fetchone()
    return user

def create_orders_table():
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS orders (
            order_id INTEGER PRIMARY KEY AUTOINCREMENT,
            user_id INTEGER,
            product_id INTEGER,
            FOREIGN KEY (user_id) REFERENCES users (user_id),
            FOREIGN KEY (product_id) REFERENCES products (product_id)
        )
    """)
    conn.commit()

def add_order(user_id, product_id):
    cursor.execute("""
        INSERT INTO orders (user_id, product_id) VALUES (?, ?)
    """, (user_id, product_id))
    conn.commit()
```

```
def get_user_orders(user_id):
```

```
    cursor.execute("""
        SELECT * FROM orders WHERE user_id = ?
    """, (user_id,))
    orders = cursor.fetchall()
    return orders
```

```
def get_all_orders():
```

```
    cursor.execute("""
        SELECT * FROM orders
    """)
    orders = cursor.fetchall()
    return orders
```

```
def create_admin_profile():
```

```
    cursor.execute("""
        INSERT INTO users (username, phone_number) VALUES (?, ?)
    """, ('admin', ""))
    conn.commit()
```

```
def update_phone_number(user_id, phone_number):
```

```
    cursor.execute("""
        UPDATE users SET phone_number = ? WHERE user_id = ?
    """, (phone_number, user_id))
    conn.commit()
```

```
create_users_table()
```

```
create_products_table()
```

```
create_orders_table()
```

```
def process_message(user_id, message):
```

```
    if message.startswith("+"):
```

```
        update_phone_number(user_id, message)
```

```
        return "Номер телефону оновлено."
```

```
    elif message == "Мої замовлення":
```

```
        orders = get_user_orders(user_id)
```

```
        if orders:
```

```
            order_list = "Ваші замовлення:\n"
```

```
            for order in orders:
```

```
                product = get_product_info(order[1])
```

```
                order_list += f"Товар: {product[1]}, Ціна: {product[2]} грн\n"
```

```
            return order_list
```

```
        else:
```

```
            return "У вас немає замовлень."
```

```
user_id = 1
```

```
message = "Мої замовлення"
```

```
result = process_message(user_id, message)
```

```
print(result)
```