

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт  
на здобуття ступеня бакалавра  
за освітньо-професійною програмою «Інтегровані інформаційні системи»  
спеціальності 126 «Інформаційні системи та технології»  
на тему: «Вебзастосунок StoriedFood на базі Telegram»**

Виконав:

студент ІV курсу, групи ІА-94  
Снігур Андрій Валерійович

\_\_\_\_\_

Керівник:

Старший викладач кафедри ІСТ  
Хмелюк Марина Сергіївна

\_\_\_\_\_

Рецензент:

Доц. каф. ІІІ, к.т.н., доц.  
Лісовиченко Олег Іванович

\_\_\_\_\_

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_

Київ – 2023 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Снігуру Андрію Валерійовичу**

1. Тема проєкту «Вебзастосунок StoriedFood на базі Telegram», керівник проєкту Хмелюк Марина Сергіївна, старший викладач, затверджені наказом по університету від «31» травня 2023 р. №2101-с.
2. Термін подання студентом проєкту: 12 червня 2023 року
3. Вихідні дані до проєкту: мова програмування JavaScript, вебзастосунок на платформі Telegram, бібліотеки React та Node.js, система управління базами даних MongoDB.
4. Зміст пояснювальної записки: вступ, опис предметної області, аналіз існуючих рішень, вебзастосунки на базі Telegram, проєктування вебзастосунку, реалізація вебзастосунку, керівництво з використання, висновки, перелік використаних джерел.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): діаграма діяльності, діаграма варіантів використання, діаграма послідовностей, діаграма станів.
6. Дата видачі завдання: 1 березня 2023 року

## Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз предметної області	17.04.2023	
2	Огляд існуючих рішень	24.04.2023	
3	Розробка структурної та функціональної схеми веб-додатку	29.04.2023	
4	Вибір засобів та методів реалізації	03.05.2023	
5	Розробка бази даних та проєктування системи	15.05.2023	
6	Реалізація вебзастосунку	22.05.2023	
7	Розробка діаграм	28.05.2023	
8	Оформлення документації	01.06.2023	
9	Подання готового проєкту	12.06.2023	

Студент

Андрій СНІГУР

Керівник

Марина ХМЕЛЮК

## АНОТАЦІЯ

Снігур Андрій Валерійович StoriedFood Вебзастосунок на базі Telegram. КІП імені Ігоря Сікорського, Київ, 2023.

Пояснювальна записка дипломного проєкту складається з 5 розділів, містить 69 сторінок тексту, 38 рисунків, 1 таблицю, 1 додаток, 20 інформаційних посилань та 4 кресленики.

Ключові слова: telegram web app bot, вебзастосунок, Telegram, React, Node.js, MongoDB, купівля їжі, ресторан.

Об'єктом розроблення є вебзастосунок для замовлення продукції.

Предметом розроблення є спосіб реалізації вебзастосунка у месенджері Telegram.

Метою проєкту є впровадження нових рішень у сфері проєктування вебзастосунків для продажу товарів.

У дипломному проєкті реалізована можливість замовлення продукції через вебзастосунок, який інтегрований у Telegram; оновлення та редагування продукції через панель адміністратора; оплата через платіжну систему; перегляд замовлень.

Отримані результати можуть бути корисними як для користувачів, які прагнуть зручного та швидкого замовлення їжі, так і для підприємців які хочуть запровадити найновіші рішення у сфері замовлення продукції у Інтернеті.

## SUMMARY

Snihur Andrii Valeriyovych StoriedFood Web application based on Telegram.  
Igor Sikorsky KPI, Kyiv, 2023.

The explanatory note of the diploma project consists of 5 chapters, contains 69 pages of text, 38 figures, 1 table, 1 appendix, 19 informational references and 4 drawings.

Keywords: telegram web app bot, web application, Telegram, React, Node.js, MonogDB, food purchase, restaurant.

The object of development is a web application for ordering products.

The subject of development is a method of implementing a web application in the Telegram messenger.

The goal of the project is to implement new solutions in the field of designing web applications for the sale of goods.

The diploma project implemented the possibility of ordering products through a web application integrated into Telegram; updating and editing products through the administrator panel; payment through a payment system; viewing orders.

The obtained results can be useful both for users who seek convenient and fast food ordering, and for entrepreneurs who want to introduce the latest solutions in the field of ordering products on the Internet.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IA94.180БАК.004 ПЗ	Пояснювальна записка	69		
6	A3	IA94.180БАК.004 Д1	Вебзастосунок StoriedFood	1		
7			на базі Telegram. Діаграма			
8			діяльності			
9	A3	IA94.180БАК.004 Д2	Вебзастосунок StoriedFood	1		
10			на базі Telegram. Діаграма			
11			варіантів використання			
12	A3	IA94.180БАК.004 Д3	Вебзастосунок StoriedFood	1		
13			на базі Telegram.			
14			Діаграма послідовностей			
15	A3	IA94.180БАК.004 Д4	Вебзастосунок StoriedFood	1		
16			на базі Telegram.			
17			Діаграма станів			
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						

**IA94.180БАК.004 ТП**

Зм.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		Снігур А.В.			Вебзастосунок StoriedFood на базі Telegram. Відомість проекту	Літ.	Аркуш	Аркушів
Керівн.		Хмельюк М.С.				Т	1	1
						КПІ ім. Ігоря Сікорського Група ІА-94		
Затв.								

**Пояснювальна записка  
до дипломного проекту  
на тему: «Вебзастосунок StoriedFood на базі  
Telegram»**

Київ – 2023



3.6 База даних .....	36
3.7 Вибір хостингів.....	38
3.7.1 Netlify.....	39
3.7.2 Render .....	40
3.8 Платіжна система. Stripe.....	41
Висновок до розділу.....	42
4 РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ .....	43
4.1 Загальний опис реалізації .....	43
4.2 Реєстрація чат-бота та отримання токенів.....	44
4.3 Розробка та реалізація back-end частини .....	46
4.3.1 Налаштунок Stripe .....	48
4.3.2 Налаштунок MongoDB .....	49
4.4 Завершення налаштування серверної частини .....	50
4.5 Розробка та реалізація front-end частини.....	51
4.6 Інтеграція застосунку в Telegram .....	53
Висновок до розділу.....	54
5 КЕРІВНИЦТВО З ВИКОРИСТАННЯ .....	56
5.1 Інструкція адміністратора .....	56
5.1.1 Запуск хостингів.....	56
5.1.2 Додавання нових позицій .....	58
5.1.3 Перегляд історії .....	60
5.2 Інструкція користувача.....	63
ВИСНОВКИ.....	67
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
ДОДАТОК А .....	70

## ВСТУП

Сьогоднішній світ залежить від технологій більше, ніж будь-коли раніше. У сучасному цифровому світі все більше сфер діяльності прагнуть до цифрової трансформації та автоматизації. Тому не дивно, що сфера замовлення їжі з плином часом також спрощується. Інтернет-ресурси, мобільні додатки, чат-боти та інші інструменти, які допомагають замовляти їжу та доповідати про її склад та корисність, стають все популярнішими серед споживачів.

В рамках цієї тенденції, представлено проєкт на базі чат-бота в Telegram, який має назву StoriedFood. Даний проєкт розроблений, як вебзастосунок, усередині додатку месенджера, який дозволяє користувачам замовляти їжу безпосередньо в Telegram, замість використання звичайного сайту чи мобільного додатку. StoriedFood пропонує асортимент пропозицій, з можливістю замовлення та зручної оплати.

Однією з переваг StoriedFood в порівнянні з подібними сервісами, є відсутність у необхідності завантажувати окремий мобільний додаток, це означає, що користувачі можуть замовляти їжу безпосередньо в Telegram, що є зручно та ефективно. Замість того, щоб завантажувати та оновлювати окремий додаток на своєму смартфоні, користувачі можуть комунікувати з чат-ботом, що значно зменшує час та зусилля для здійснення замовлення.

Телеграм є однією з найпопулярніших платформ для обміну повідомленнями, яка відрізняється високою стабільністю та безпекою. Більш того, Telegram має відкритий API, що дозволяє розробникам створювати чат-боти та інші вебзастосунки на його базі. Саме тому було обрано Telegram як платформу для чат-бота.

StoriedFood буде мати досить простий та зрозумілий інтерфейс. Після короткого спілкування з ботом користувач зможе відкрити веб-інтерфейс всередині месенджера та серед відповідних позицій обрати необхідну. Після обрання позицій, користувач може зробити замовлення, вказавши необхідну кількість страв, після чого зможе оплатити своє замовлення.

						Лист
					ІА94.180БАК.004 ПЗ	4
Зм.	Лист	№ докум.	Підпис	Дата		

Крім того, чат-бот пропонуватиме користувачам зручний та швидкий спосіб замовляти їжу за допомогою Telegram, не використовуючи сторонні додатки, вебресурси.

Актуальність застосунку для замовлення їжі через Інтернет є очевидною, оскільки як клієнти, так і бізнеси прагнуть до полегшення процесу замовлення продукції. Чат-бот StoriedFood відповідає цій вимозі, забезпечуючи зручну платформу для взаємодії та швидку обробку замовлень.

Отже, вважаю обрану тему дипломного проєкту доцільною для аналізу, розробки та подальшої підтримки.

Об'єктом розроблення є вебзастосунок для замовлення продукції.

Предметом розроблення є спосіб реалізації вебзастосунка у месенджері Telegram.

Метою проєкту є впровадження нових рішень у сфері проєктування вебзастосунків для продажу товарів.

Практичне значення розроблюваного ресурсу полягає в тому, що готова програмна реалізація надасть користувачу можливість замовляти страви за допомогою додатку месенджера Telegram.

Програма розроблена з використанням мови програмування JavaScript, бібліотек React та Node.js та СУБД MongoDB.

Під час роботи над дипломним проєктом були поглиблені знання у зазначених вище технологіях, а також був розроблений інтернет магазин, що дозволяє замовляти продукцію через Telegram.

Дипломний проєкт складається з наступних розділів: вступ, основні розділи, висновки, список використаних джерел та один додаток. Графічна частина включає в себе чотири кресленики. Загальний обсяг пояснювальної записки дипломного проєкту 69 сторінки.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Опис предметної області

Замовлення їжі онлайн стає все більш популярнішим серед користувачів в усьому світі. Це обумовлено багатьма факторами, зокрема зручністю та швидкістю процесу замовлення, можливістю порівняти різні варіанти та акції в одному місці, а також збереженням часу та енергії, які можуть бути витрачені на самостійне приготування їжі.

Найбільшу користь це несе для людей, які ведуть активний та зайнятий спосіб життя, які мають мало часу на приготування їжі або просто хочуть економити на цьому свій вільний час. Замовлення їжі онлайн також популярне серед студентів та молоді, які часто перебувають в дорозі, у навчальних закладах або на роботі, і хочуть швидко та легко перекусити без відволікання на пошук місця для їжі та очікування замовлення.

Більше того, замовлення харчів онлайн може бути корисним для людей з обмеженими можливостями, які можуть мати складнощі з приготуванням їжі та пересуванням до ресторану або кафе.

Отже, вебзастосунок Telegram для замовлення їжі може бути дуже корисним для людей, які шукають зручний та швидкий спосіб замовлення та отримання їжі. Це дозволяє ресторанам та кафе залучати нових клієнтів та збільшувати свій обіг товарів, а також забезпечувати високу якість обслуговування та швидкість доставки.

## 1.2 Огляд існуючих рішень

### 1.2.1 Критерії для аналізу аналогів

Перед написанням дипломного проєкту необхідно було розглянути та проаналізувати аналоги застосунку для наочного порівняння та конструювання загальних вимог перед розробкою. В ході пошуку програм, що надають схожий

функціонал, було виділено декілька сайтів. Аналізуючи ресурси, особлива увага приділялась наступним критеріям:

- зручний користувацький інтерфейсу;
- надійність застосунку;
- зрозумілий механізм оплати;
- перегляд вибраних позицій при замовленні;
- необхідність у інсталяції додатку;

Перед розробкою власного проєкту було важливо продивитися, які схожі сервіси вже представлені на ринку. Для аналізу були обрані ресурси: сайт «contrabanda.kiev.ua», Телеграм бот «UNO Pizza&Grill», мобільний застосунок «Юджин Фуд» та Інстаграм сторінка «nisha.bistro».

### 1.2.2 Сайт «contrabanda.kiev.ua»

Заклад «contrabanda.kiev.ua» [1], спеціалізується не тільки на звичному фаст-фуді, а й пропонує цікаві позиції в меню: лапша, рис, а також супи різних видів.

На рисунку 1.1 зображений інтерфейс головного вікна сайту.

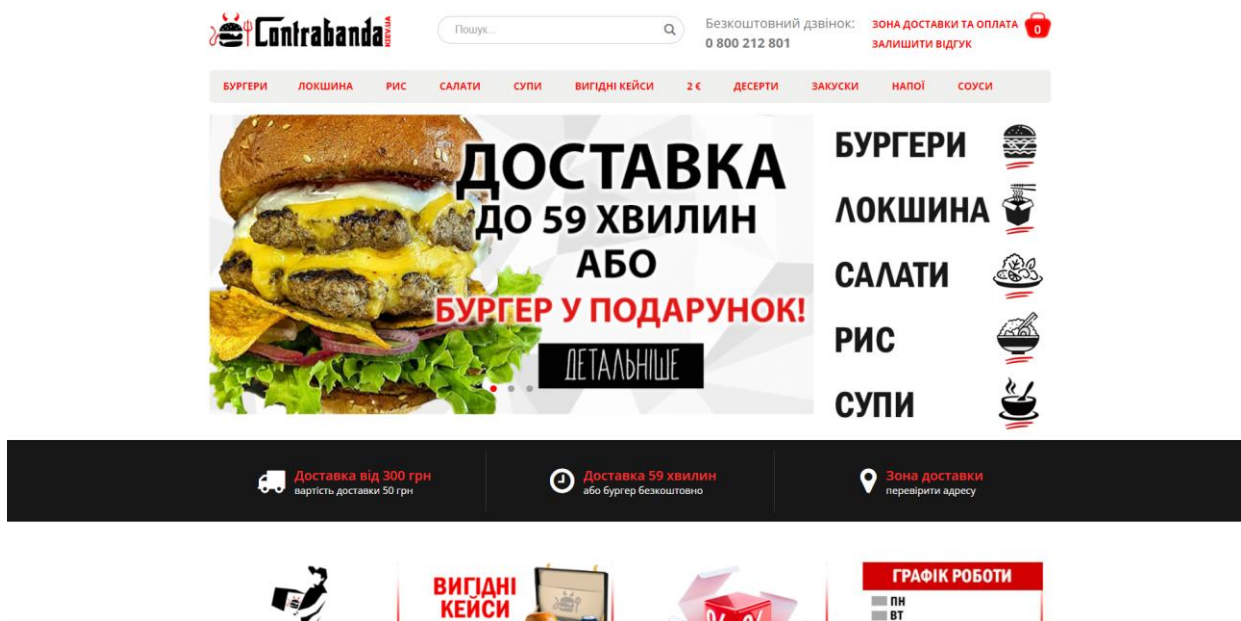


Рисунок 1.1 – Сайт contrabanda.kiev.ua

Перше, що зустрічає нас - це інтерфейс сайту, який доволі звичайний, для сайтів такого типу. Також, хочеться зауважити, що інтерфейс перенасичений інформацією.

Явним мінусом сервісу – є відсутність чату на сайті. Інформація щодо контакту з представниками сервісу зображено на рисунку 1.2. Якщо у користувача виникне якесь термінове питання або доповнення до замовлення то у нього, є можливість або подзвонити на гарячу лінію або ж писати на email, що є не практично для більшості клієнтів.

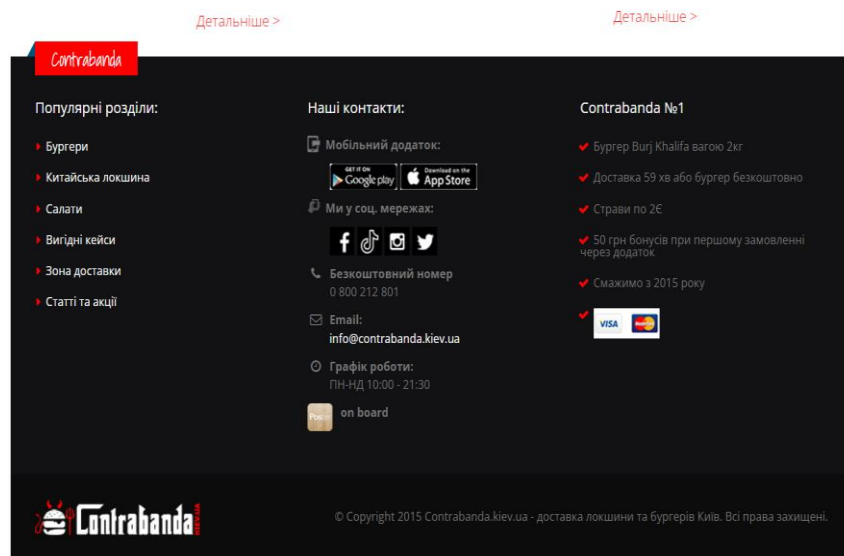


Рисунок 1.2 – Панель контакту

Сформовано усі переваги та недоліки сайту «contrabanda.kiev.ua».

Переваги:

- великий вибір продукції;
- адаптивність сайту;
- є мобільний додаток.

Недоліки:

- перевантажений інтерфейс;
- відсутність чату з працівником закладу;

Зм.	Лист	№ докум.	Підпис	Дата

- немає можливості зареєструватися, що веде за собою недолік, наприклад неможливість перегляду минулих замовлень;
- необхідність у завантаженні додатку, для розкриття повного потенціалу сервісу.

### 1.2.3 Телеграм бот «UNO Pizza&Grill»

Додаток який реалізований на основі боту Telegram, ресторану «UNO Pizza&Grill» [2].

Даний бот має тільки лінійну взаємодію з клієнтом, тобто бот постійно пропонує зі списку варіантів вибрати один єдиний, приклад взаємодії з ботом представлений на рисунку 1.3. Мінусів цього підходу відразу декілька. По-перше, щоб після вибору потрібної позиції з меню, оформити замовлення - потрібно пару разів натиснути кнопку назад та повернутися на головне меню і тільки після цього ви зможете сформулювати замовлення. По-друге, користувач не може спостерігати за динамічним поповненням кошика, це можна побачити лише наприкінці при оформленні замовлення.

Також боти такого типу можуть бути важкі в розумінні для недосвідчених клієнтів, хто стикається з таким підходом вперше. Через відсутність хоч якогось графічного інтерфейсу деяким користувачам може бути незручно користуватися цим застосунком.

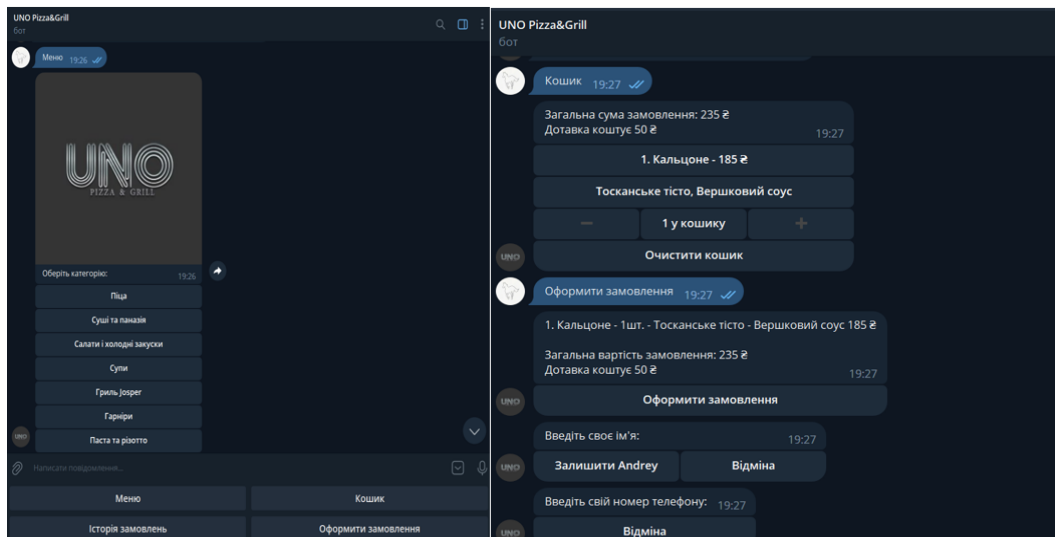


Рисунок 1.3 – Взаємодія з ботом

Нижче перераховано переваги та недоліки телеграм боту «UNO Pizza&Grill».

Переваги:

- немає потреби завантажувати додаток;
- великий вибір страв.

Недоліки:

- непродумана взаємодія з клієнтом;
- незрозумілий механізм замовлення.

#### 1.2.4 Мобільний застосунок «Юджин Фуд»

Сітка фаст-фуд їжі Юджин Бургер [3], яка представлена мобільним додатком, інтерфейс якого представлений на рисунку 1.4.

Додаток доступний як у AppStore так і у PlayMarket. Інтерфейс доволі приємний та зрозумілий, єдине що через великий асортимент продукції тяжко знайти деякі пропозиції. Кількість категорій та продукції нерівномірно зосереджено в додатку, тому клієнти, які ще не знайомі з закладом, можуть просто не знайти деякі позиції в меню.

Зм.	Лист	№ докум.	Підпис	Дата

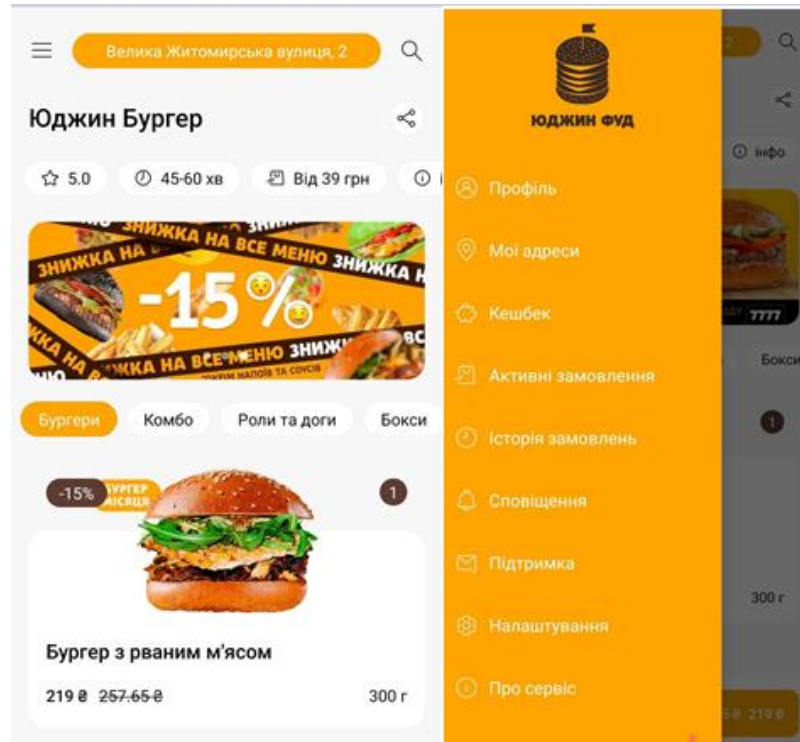


Рисунок 1.4 - Інтерфейс «Юджин Фуд»

Нижче перелічені основні переваги та недоліки мобільного застосунку «Юджин Фуд».

Переваги:

- швидка робота додатку;
- великий вибір їжі;
- багатofункціональність додатку.

Недоліки:

- потреба завантажувати додаток;
- не найкращий вибір позиціонування продукції.

### 1.2.5 Інстаграм сторінка бургерної «nisha.bistro»

Заклад під назвою «Бургерна Ніша» [4], який спеціалізується на різних видах бургерів. Даний ресторан обрав саме Інстаграм сторінку для реалізації та просування своїх товарів.

Зм.	Лист	№ докум.	Підпис	Дата

Спершу потрібно зрозуміти різницю між платформами Інстаграм та Телеграм. Інстаграм - це соціальна мережа, яка зазвичай використовується для ділових цілей та продажу товарів. Вона має більше 1 мільярда користувачів по всьому світу та має можливість відкрити бізнес-аккаунт, що дозволяє продавцям створювати публікації з фотографіями та описами страв, вказувати ціни та посилати користувачів на сайт для замовлення або ж писати напряму співробітникам закладу у месенджер «Direct». Однак, продавці мають обмежений контроль над своїми публікаціями та доступним функціоналом.

Телеграм - це месенджер з можливістю створення каналів та ботів. Він охоплює велику аудиторію користувачів та є популярним серед бізнесу, оскільки має більш гнучкий функціонал для продажу. Зокрема, на Телеграмі можна створювати канали з описом страв, фотографіями, цінами та можливістю приймати замовлення через ботів. Також, у Телеграмі можна підключати платіжні системи для прийому оплати, що робить процес продажу більш простим та зручним.

У порівнянні з Інстаграмом, Телеграм має кращий та ширший функціонал для продажу їжі, оскільки дає можливість інтегрувати вебзастосунки та ботів для прийому замовлень. Крім того, на Телеграмі можна підключати платіжні системи, що теж являється плюсом. Таким чином, Телеграм може бути кращим вибором для відкриття онлайн магазину.

Найбільший недолік Інстаграму – це те, що не можливо одразу після вибору потрібних товарів, відразу сплатити замовлення, приклад представлений на рисунку 1.5.

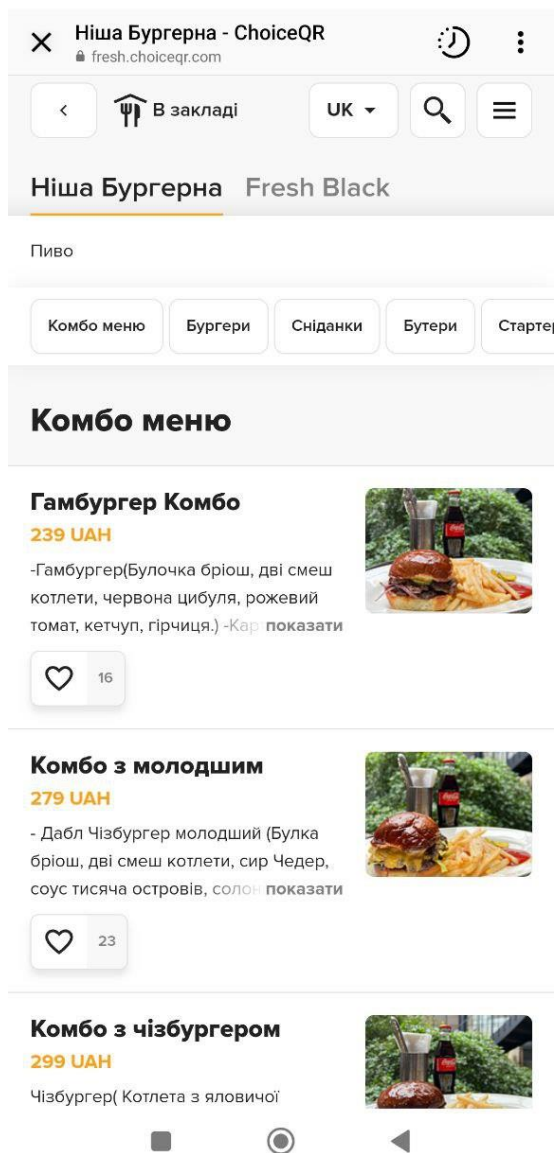


Рисунок 1.5 - Інтерфейс меню «Бургерна Ніша»

Можливий лише варіант надання меню всередині соціальної мережі, а замовлення вже відбувається через месенджер. Через що, губиться зайвий час клієнта, а в доповнення потрібні додаткові витрати з боку власників, щоб оплачувати спеціаліста, який відповідатиме клієнтам, приклад взаємодії з клієнтами представлений на рисунку 1.6.

Зм.	Лист	№ докум.	Підпис	Дата

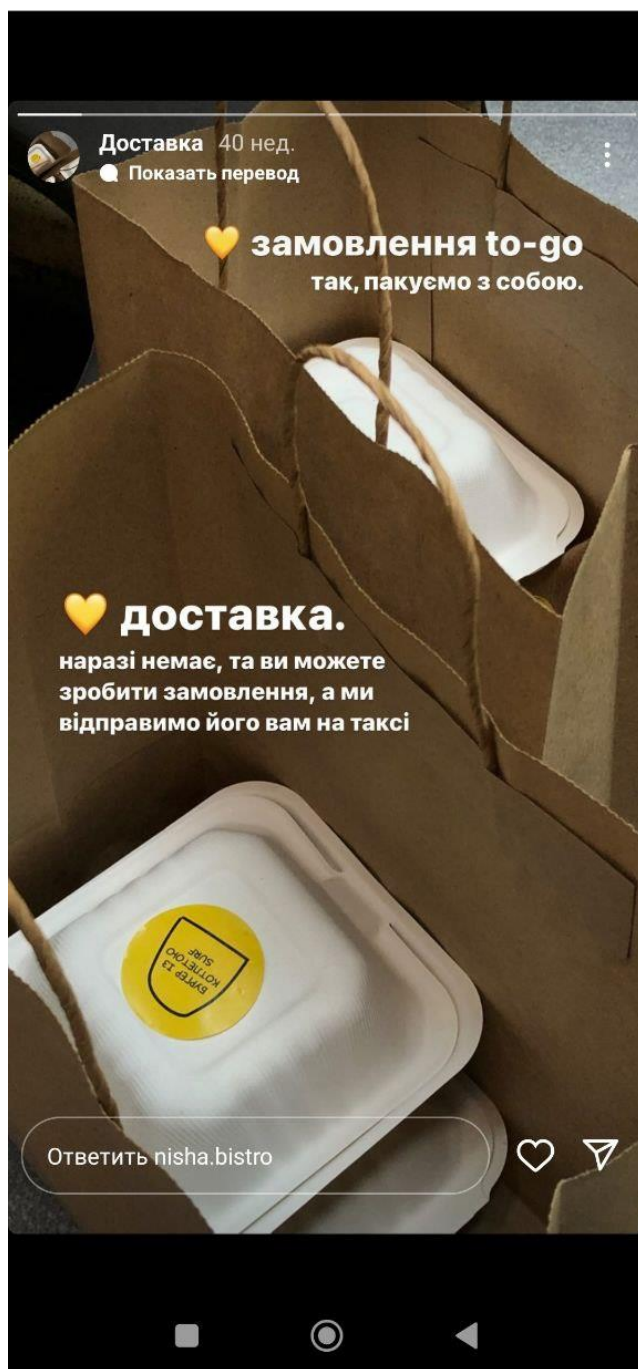


Рисунок 1.6 - Доставка «Бургерна Ніша»

Переваги та недоліки інстаграм сторінки бургерної «Бургерна Ніша».

Переваги:

- легкість створення;
- можливості для піару бренду;
- клієнтська підтримка.

Зм.	Лист	№ докум.	Підпис	Дата

ІА94.180БАК.004 ПЗ

Лист

14

Недоліки:

- функціональність;
- немає змоги одразу оплатити товар;
- необхідність у співробітнику, хто буде приймати замовлення.

### 1.3 Підсумковий аналіз розглянутих застосунків

При детальному аналізі вище зазначених сервісів вдалося виявити позитивні та негативні якості кожного з них. Порівняльна таблиця 1.1 буде містити у собі раніше визначені критерії якості.

Для оцінки якості буде застосовано технологію з п'яти-бальної оцінки оцінювання. Де 1 самий поганий показник, а 5 найбільш позитивний.

Таблиця 1.1 – Порівняння з існуючими рішеннями

Критерій оцінки	StoriedFood	contrabanda	UNO Pizza&Grill	Юджин Фуд	Бургерн а Ніша
зручний користувацький інтерфейсу	4	3	2	5	3
надійність застосунку	4	4	4	3	4
зрозумілий механізм оплати	4	4	2	4	1
перегляд вибраних позицій при замовленні	3	5	2	5	2

Зм.	Лист	№ докум.	Підпис	Дата

Необхідність у інсталяції додатку	5	4	5	3	5
---	---	---	---	---	---

#### 1.4 Постановка задачі

Метою проекту є розробка вебзастосунку, який надасть користувачам можливість зручно замовляти продукцію через платформу Telegram та оплачувати її використовуючи сучасну, міжнародну платіжну систему. Застосунок також повинен надати адміністратору зручний спосіб перегляду замовлень через панель адміністратора, а також можливість редагування продукції та додавання нових позицій.

Основні завдання проекту включають:

- розробка фронтенду: Створення зручного інтерфейсу для користувачів, що дозволить замовляти продукцію;
- реалізація серверної частини: Розробка серверного коду, який оброблятиме запити користувачів та зберігатиме інформацію про замовлення;
- інтеграція з платіжним шлюзом: Забезпечення безпечної оплати замовлень через платіжний шлюз;
- розробка адміністративної панелі: Створення інтерфейсу для адміністратора, що дозволить переглядати та керувати замовленнями та продукцією;
- деплоймент: Розгортання вебзастосунку на відповідних хостингах для забезпечення доступу користувачів.

Для досягнення поставленої мети і забезпечення зручного та безперебійного функціонування додатку мають бути застосовані сучасні технології.

					ІА94.180БАК.004 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		16

## Висновок до розділу

Проаналізувавши найбільш популярні веб-сервіси, можна зробити висновок, що кожен із них має свої переваги і недоліки.

Були розглянуті не лише окремі заклади, а й різні технології та типи ресурсів для продажу їжі, такі як вебзастосунок, мобільний додаток, чат-бот Telegram, Instagram бізнес сторінка та вебзастосунок на базі Telegram. Кожен з яких, має свої сильні та слабкі сторони. Хтось користується вже перевіреними технологіями для продажу своїх товарів, хтось довіряється більш новітнім засобам. Для ефективного просування свого продукту ідеальним буде варіант, комбінувати ці рішення.

Остаточний вибір технологій та підходів для розробки веб-додатку повинен базуватися на конкретних вимогах проєкту, цільовій аудиторії та бізнес-стратегії. Варто балансувати між перевагами та недоліками кожного підходу, а також врахувати тенденції ринку та очікування користувачів. Завдяки правильно обраному комплексному підходу можна створити високоякісний та конкурентоздатний вебзастосунок, який забезпечить зручну та ефективну взаємодію з користувачами та сприятиме успіху бізнесу.

По аналізу вище, видно що така технологія, як вебзастосунок у Telegram має право складати конкуренцію більш популярним рішенням на ринку. Також була постановлена задача, в рамках якої було визначено основні функції та вимоги до вебзастосунку. Враховуючи потреби користувачів та бізнесу, було встановлено, що необхідно створити зручний інтерфейс для замовлення продукції через Telegram.

					ІА94.180БАК.004 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		17

## 2 ВЕБЗАСТОСУНКИ НА БАЗІ TELEGRAM БОТІВ

### 2.1 Відомості про чат-ботів

У наш час все частіше можна почути наступну думку, що цифровізація перетворює наше суспільство на “мобільне населення”. І з цим твердженням неможливо сперечатися, з кожним роком число користувачів новеньких гаджетів стає все більше і більше.

Найбільш популярними додатками для обміну повідомленнями – є чат-боти. Чат-боти - це програми, що використовують штучний інтелект для імітації розмови з користувачем через різні канали комунікації, такі як додатки для обміну повідомленнями, веб-сайти та мобільні додатки. Вони розроблені для автоматизації конкретних завдань і можуть бути підключені через API до різних джерел даних, щоб надавати користувачам потрібну інформацію або послуги.

Інтелектуальні розмовні чат-боти часто є інтерфейсами для мобільних додатків і змінюють спосіб взаємодії компаній і клієнтів. Чат-боти дозволяють компаніям спілкуватися з клієнтами особисто без допомоги представників. Наприклад, багато питань або проблем, які виникають у клієнтів, є звичайними і на них легко відповісти.

Однією з ключових переваг чат-ботів є їх зручність у користуванні - користувачі можуть спілкуватися з ними без потреби завантажувати додатки або оновлювати їх. Це зменшує використання пам'яті на пристрої та сприяє швидшому доступу до необхідної інформації.

Інша важлива перевага полягає у можливості інтегрувати кілька чат-ботів в один чат. Це забезпечує зручний та простий інтерфейс, що дає можливість взаємодіяти з декількома сервісами та отримувати необхідну інформацію в одному місці. Також, зі збільшенням кількості чат-ботів, що підтримують різні сервіси, користувачам стає доступною більш широка база інформації та можливостей.

Також чат-боти стали популярними як засіб економії часу та грошей для бізнесу, через те що не потрібні додаткові витрати на найм співробітників для спілкування з клієнтами. Чат-боти з кожним новим роком набувають нові функції

									Лист
									18
Зм.	Лист	№ докум.	Підпис	Дата					

та опції для більш ефективного впровадження їх у бізнес системи. Наприклад, у даній роботі буде представлений бот з вбудованим вебзастосунком для замовлення їжі через відомий месенджер Telegram.

У 2018 році кількість користувачів, що використовують відомі месенджери обігнали по популярності відомі соціальні мережі. З чого можна зробити висновок, що месенджери більш цікаві для щоденного використання. Причини цього очевидні, новітні месенджери мають набагато кращу приватність, аніж соціальні мережі. Також, є інші фактори – такі як, потреба меншої кількості пам'яті, менша кількість недоречної інформації та звичайно менше або повна відсутність реклами.

Тому, очевидно що інтеграція чат-ботів у месенджери буде найбільш доцільною та доречною. При правильному налаштуванні всіх компонентів бот зможе економити час, а головне - гроші для бізнесу. При цьому, це не псує відношення клієнтів до кінцевого продукту. Такий чат-бот буде слугувати добрим всебічним інструментом.

## 2.2 Telegram боти

Telegram боти - це програмні застосунки, які працюють всередині месенджера Telegram та забезпечують взаємодію з користувачами через чат-інтерфейс. Боти можуть бути розроблені на будь-якій мові програмування, що підтримує HTTP-запити та JSON-протокол. Логіка ботів повністю контролюється за допомогою запитів користувача.

Telegram боти стали доступні ще у 2015 році після того, як Telegram випустив своє API для розробників. З того часу Telegram боти стали дуже популярними серед бізнесу та розробників, які використовують їх для автоматизації процесів та покращення взаємодії з користувачами.

Основна перевага Telegram ботів полягає в їх здатності автоматизувати рутинні процеси та забезпечити швидку взаємодію з користувачами через чат-інтерфейс. Боти можуть виконувати різноманітні завдання, такі як обробка

замовлень, відповідь на запитання, надання інформації про товари та послуги, розсилка повідомлень та багато іншого.

Telegram боти також підтримують інтеграцію з іншими сервісами та додатками, що дозволяє їм отримувати доступ до зовнішніх джерел даних та забезпечувати більш широкий спектр послуг для користувачів.

### 2.3 Створення Telegram чат-бота

Для створення бота в Telegram потрібно насамперед зареєструватися в месенджері та отримати API ключ від BotFather, який дозволить взаємодіяти з API Telegram, продемонстровано на рисунку 2.1.

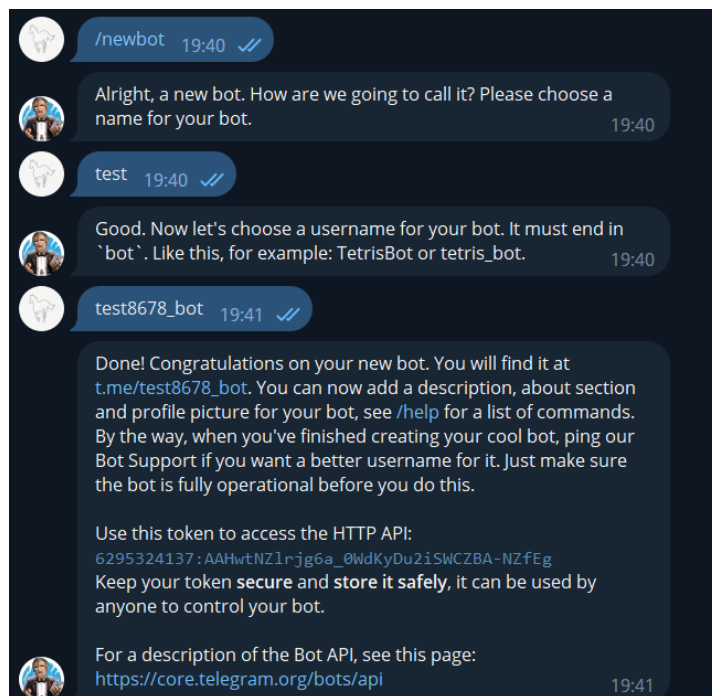


Рисунок 2.1 – Реєстрація боту

Після реєстрації необхідно створити нового бота та налаштувати його параметри, такі як ім'я, коротке ім'я користувача, опис та профільну картинку.

Telegram боти мають змогу виконувати різноманітні завдання, такі як обробка замовлень, відповідь на різноманітні запитання, надання інформації про



використані для різних цілей, включаючи замовлення товарів, бронювання послуг, отримання необхідної інформації, збереження та подальшою роботою з інформацією та багато іншого.

Нарешті, Telegram боти дають можливість як і малому бізнесу, який тільки планує масштабуватися так і вже досвідченим підприємцям розгорнути власний бізнес в месенджері. Завдяки чат-ботам, бізнеси можуть забезпечити зручний та ефективний сервіс для своїх клієнтів, а також збільшити свій дохід і вплив на своєму ринку.

## 2.4 Відомості про вебзастосунки

Вебзастосунки (Web Application) - це програмне забезпечення, яке використовується для доступу до веб-сайту або веб-сторінки через інтернет за допомогою браузера. Вебзастосунки можуть бути створені за допомогою різних технологій, таких як HTML, CSS, JavaScript, PHP, Python, Ruby та інших мов програмування.

Вебзастосунки є невід'ємною частиною сучасного Інтернету та мають різноманітні застосування, від простих блогів до складних систем електронної комерції та банківських систем.

Робота типового вебзастосунку продемонстровано на рисунку 2.2 [5].

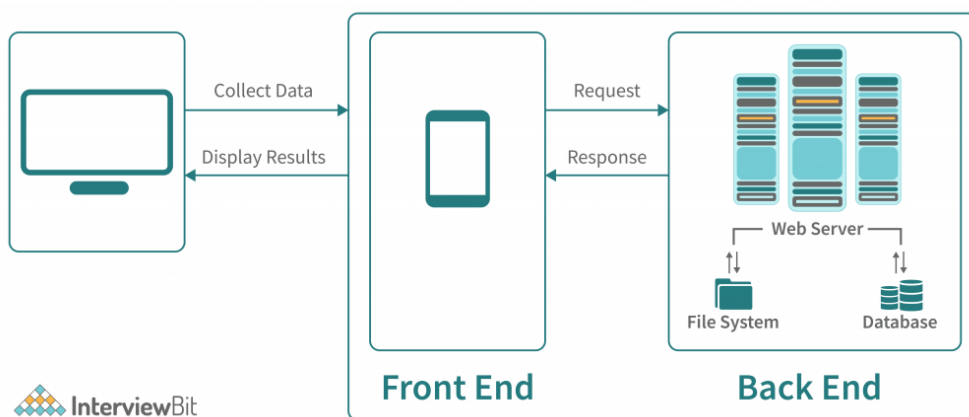


Рисунок 2.2 - Робота типового вебзастосунку

Зм.	Лист	№ докум.	Підпис	Дата



– оновлення: Одна з великих переваг вебзастосунків полягає в тому, що можна оновлювати додаток на сервері без необхідності оновлювати програмне забезпечення на кожному окремому пристрої, що використовує додаток. Це дає можливість розробникам бути впевненими, що користувачі використовують останню версію додатку, забезпечуючи їм оптимальне функціонування;

– швидкий розгортання: Вебзастосунки можуть бути швидко розгорнуті на сервері та бути доступними користувачам. Це означає, що користувачі можуть швидко почати використовувати додаток після його створення.

Як можна побачити вебзастосунки мають велику кількість дійсно значних плюсів.

#### 2.4.2 Недоліки вебзастосунків

Звичайно вебзастосунки не є бездоганними і мають низку своїх недоліків. Однак, на сьогоднішній день вони є популярним та ефективним засобом для багатьох бізнес-задач.

Розглянемо основні недоліки вебзастосунків:

– залежність від Інтернету: Незважаючи на те, що ми живемо в епоху Інтернету, втрата з'єднання з Інтернетом є досить поширеним явищем. І жоден Інтернет безпосередньо не призведе до втрати можливості запускати веб-програму. Таким чином, надійне підключення до Інтернету є обов'язковим у будь-який час для перегляду веб-сайту та запуску програми;

– знижена швидкість: Часто веб-програми працюють повільніше, ніж програми, які фізично розміщені на локальному сервері, і тому вони не можуть повністю замінити мобільні додатки. Вебзастосунки також залежать від браузера, в якому вони використовуються, що призводить до збільшення їхнього розміру програми. Вебзастосунок також безпосередньо пов'язана з браузером, через що його розмір програми має тенденцію до збільшення. Таким чином, велика програма значно повільніша за нативну настільну. Крім того, оскільки веб-

програма повністю працює в Інтернеті, вона часто може працювати повільніше через якість інтернет-з'єднання, що впливає на їх продуктивність.;

- загрози безпеки: Вебзастосунки можуть бути більш вразливі до кібератак, зокрема до атак на перехоплення даних, SQL-ін'єкцій, XSS-атак, DDoS-атак та інших видів хакерських атак;

- обмежені можливості без Інтернету: Деякі вебзастосунки можуть працювати офлайн, але їх можливості будуть значно обмежені в порівнянні з онлайн-версією;

- обмежена функціональність: Вебзастосунки працюють у браузері, що означає, що вони не можуть виконувати деякі операції, які можуть бути доступні на операційній системі;

- проблеми з сумісністю: Якщо вебзастосунок розроблений для певного браузера, то він може не коректно працювати на всіх інших браузерах або версіях браузерів;

- перенасиченість функцій: Часто вебзастосунки мають багато функцій, що можуть зробити їх складними та перенасиченими для користувачів. Недостатня увага до дизайну може призвести до складнощів в навігації та пошуку необхідних функцій.

Хоча вебзастосунки мають свої недоліки, вони все ж мають багато значних переваг, які роблять їх популярними серед користувачів та розробників.

## 2.5 Вебзастосунки на базі Telegram

Боти Telegram мають можливість розміщувати повноцінні веб-програми, реалізовані за допомогою JavaScript. Це дозволяє створювати дуже гнучкі інтерфейси, які можуть управляти різноманітними функціями, від онлайн-магазинів до аркадних ігор. У порівнянні з веб-сайтами, боти в Telegram підтримують авторизацію та сповіщення з коробки, що забезпечує безперебійну комунікацію з користувачами [6].

Наразі Telegram підтримує шість різних способів запуску веб-програм:

									Лист
									25
Зм.	Лист	№ докум.	Підпис	Дата					

- за допомогою кнопки на клавіатурі;
- за допомогою вбудованої кнопки;
- за допомогою кнопки меню бота;
- за допомогою вбудованого режиму;
- за прямим посиланням;
- навіть за допомогою меню вкладень, приклад вказано на рисунку 2.3.

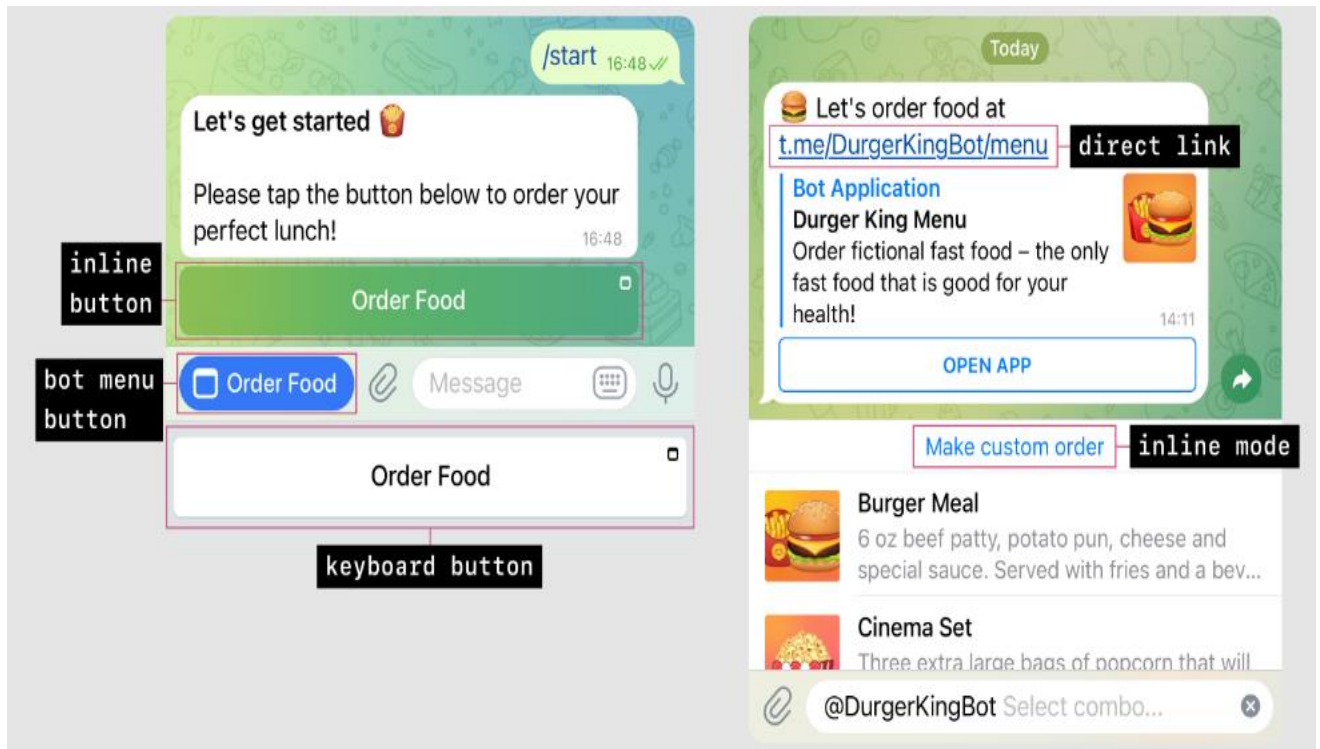


Рисунок 2.3 – Запуск вебзастосунку в Telegram

Боти вебзастосунків можуть бути додані безпосередньо до меню вкладень користувача, що дозволяє швидко запускати їх із будь-якого типу чату. Є можливість налаштувати, у яких типах чатів можна запускати вебзастосунок з меню вкладень (приватні, групи, супергрупи чи канали).

Боти можуть значно полегшити інтеграцію з іншими службами, приклад зображено на рисунку 2.4. На сьогоднішній день багато популярних платформ вже мають свої офіційні боти в Telegram. Це дозволяє зручно спілкуватися з різноманітними сервісами та отримувати необхідну інформацію у зручному форматі.

Зм.	Лист	№ докум.	Підпис	Дата

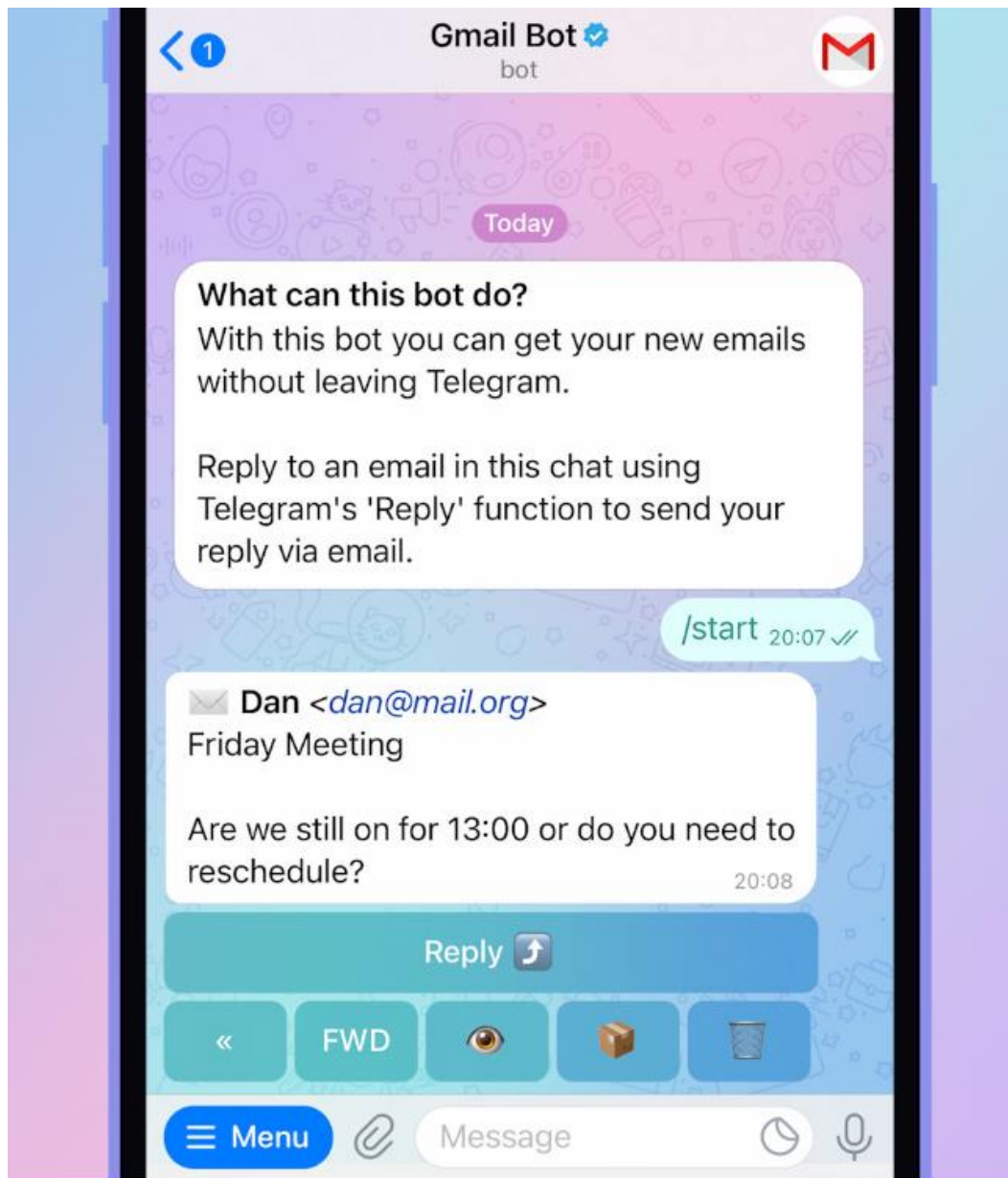


Рисунок 2.4 – Інтеграція популярних рішень у Telegram

Також, великим плюсом є те, що з наявністю знань в області побудови вебзастосунків, існує велика ймовірність успішної інтеграції проєкту на платформу Telegram.

Можливості ботів у Telegram майже безмежні - від простих сценаріїв до складних веб-програм. Якщо мета полягає в масштабуванні бізнесу або просуванні продукту чи бренду, використання Telegram може принести значну користь.

## Висновок до розділу

У даному розділі дипломного проєкту було проведено вивчення сутності Telegram як у якості функціонального інструменту, так і в якості інструменту для бізнесу. Була проведена детальна аналітика основних функцій та можливостей, що пропонує дана платформа.

Зокрема, була проведена детальна робота щодо Telegram ботів. В аналізі були виділені основні переваги та недоліки цієї можливості. Особливий акцент був зроблений на вебзастосунках на базі Telegram, які є досить популярними серед користувачів та бізнесу.

Дослідження показало, що Telegram боти є ефективним інструментом для автоматизації бізнес-процесів та для підтримки зв'язку з клієнтами. Завдяки цій можливості, бізнес може забезпечити швидкий та якісний сервіс для своїх клієнтів. Крім того, Telegram боти дозволяють зменшити витрати на персонал, що може бути важливим для бізнесу.

Після детального опису можливостей Telegram ботів, було встановлено, що Telegram є кращим варіантом для продажу їжі. Це пояснюється тим, що Telegram поєднує в собі широкі можливості та зручність використання як для продавців, так і для користувачів. Крім того, Telegram має велику базу користувачів, що може забезпечити більшу кількість клієнтів для бізнесу.

Отже, можна стверджувати, що боти на основі даної платформи є одним з кращих варіантів для бізнесу, який дозволяє ефективно просувати товари і послуги.

					ІА94.180БАК.004 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		28

## 3 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ

### 3.1 Вибір платформи

Telegram - це безкоштовний месенджер, який був запущений в 2013 році братами Павлом та Ніколаєм Дуровими. Він став дуже популярним і швидко зарекомендував себе як надійний та безпечний месенджер з великою кількістю користувачів у багатьох країнах світу.

Сильною стороною Telegram вважають великий ступінь безпеки та приватність. У ньому використовується потужне шифрування, яке забезпечує гарний захист користувачів від перехоплення даних третіми особами. Крім того, Telegram має низку інших корисних функцій, таких як можливість створення закритих груп, голосові та відео дзвінки, вбудований редактор фотографій та відео.

Telegram має 55,2 мільйона активних користувачів щодня та 700 мільйонів активних користувачів на місяць, що приблизно на 230% більше, ніж у 2019 році. База користувачів складається з 58,6% чоловіків та 41,4% жінок, а середній користувач витрачає 2,9 години на Telegram щомісяця. Месенджер підтримує багато мов та дозволяє користувачам взаємодіяти один з одним, використовуючи багато різних форматів повідомлень.

Щодо роботи з чат-ботами, Telegram має дуже потужну платформу для розробки чат-ботів. Це дає можливість бізнесу створювати різноманітні чат-боти, які можуть забезпечити клієнтам різні послуги. Чат-боти можуть допомогти відповідати на запитання клієнтів, здійснювати продажі, надавати інформацію про товари та послуги та навіть забезпечувати підтримку клієнтів. Отже, можна зробити висновок що, Telegram ідеальний для відкриття у ньому магазину, використовуючи чат-бот.

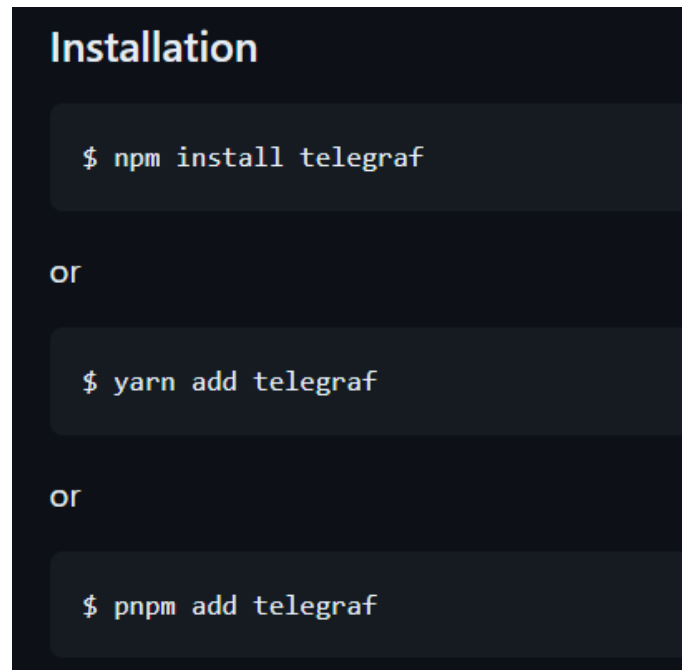
### 3.2 Бібліотека Telegraf

Telegraf - це бібліотека для побудови веб-додатків у чат-ботах Telegram. Вона є повністю асинхронною бібліотекою для Telegram Bot API, написаною на

					ІА94.180БАК.004 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		29

asuncio та aiohttp. Вона дозволяє легко створювати чат-ботів з функціями, такими як обробка команд, обробка подій та інші.

Для успішного використання цієї бібліотеки її знадобиться встановити одним із доступних методів, які зображені на рисунку 3.1 [7].



```
Installation

$ npm install telegraf

or

$ yarn add telegraf

or

$ pnpm add telegraf
```

Рисунок 3.1 – Інсталяція бібліотеки Telegraf

Основні особливості бібліотеки Telegraf:

- відповіді на повідомлення: Telegraf дозволяє обробляти вхідні повідомлення від користувачів і відправляти їм відповіді. Бібліотека дозволяє створювати текстові повідомлення, відправляти фотографії, відео, аудіо, документи та інші типи медіафайлів;
- простота використання: Telegraf має зрозумілий API, що дозволяє швидко розпочати роботу;
- обробка клавіатур: Telegraf дозволяє створювати клавіатури для взаємодії з користувачами. Можете створювати прості кнопки: вибір зі списку, кастомні клавіатури та інші елементи;

- відстежування подій: Є можливість слідкувати за подіями, такими як вступ до групи, вихід з групи, зміна назви чи зображення групи тощо. Можливо налаштувати бота для реагування на ці події і виконувати необхідні дії;
- middleware: Telegraf підтримує middleware - це функції, які виконуються перед обробкою кожного вхідного повідомлення або події. Middleware дозволяє здійснювати обробку, логування, аутентифікацію та інші операції до або після обробки основної функціональності бота;
- розширені можливості: Є багато розширених можливостей, таких як робота з файлами, аудіо- та відеозаписами, розсилка повідомлень, використання клавіш у сповіщеннях та багато іншого.

Загалом, Telegraf - це потужна бібліотека для створення ботів у Telegram, яка дозволяє легко взаємодіяти з Telegram Bot API і створювати різноманітну функціональність для бота. Вона підходить як для початківців, так і для досвідчених розробників.

### 3.3 Вибір мови програмування

Для реалізації проєкту була обрана мова програмування JavaScript, та її популярна бібліотека React. Ця бібліотека слугує для більш легкого створення користувацьких інтерфейсів. Вона була розроблена компанією Facebook і вперше була випущена в 2013 році. React став одним з найпопулярніших інструментів для розробки фронтенду, завдяки своїй ефективності, гнучкості та широким можливостям [8].

Основні переваги бібліотеки React:

- JSX синтаксис: React використовує JSX синтаксис, який дозволяє змішувати HTML та JavaScript код. Це робить розробку інтерфейсу більш зрозумілою та зручною [9];
- віртуальний DOM: React використовує віртуальний DOM, що дозволяє ефективно оновлювати лише необхідні елементи на сторінці, забезпечуючи швидкість рендерингу та покращення продуктивності додатку;

– компонентна архітектура: У React інтерфейс розбивається на невеликі, незалежні компоненти. Кожен компонент відповідає за власну логіку та представлення і може бути повторно використаний в різних частинах додатку. Це дозволяє створювати комплексний інтерфейс шляхом комбінування простих компонентів;

– односторонній потік даних: React працює за принципом одностороннього потоку даних, що сприяє передбачуваності та легкості управління станом додатку.

Отже, React є потужним і популярним інструментом для побудови вебзастосунків, який також може бути успішно інтегрований на платформу Telegram.

### 3.4 Node.js

Для розробки вебзастосунка у Telegram був обраний потужний та гнучкий стек технологій, який поєднує в собі Node.js та React. Даний вибір зумовлений їхніми унікальними можливостями та сприятливими умовами для розробки сучасних та ефективних вебзастосунків.

Node.js відкриває нові горизонти у розробці серверних додатків. Він дозволяє виконувати події JavaScript на стороні сервера, що забезпечує єдину мову програмування для як фронтенду, так і бекенду. Це робить розробку більш зручною та ефективною, оскільки розробники можуть легко переходити від однієї частини додатка до іншої без необхідності вивчати нові мови або інструменти.

Node.js, подібно до систем, таких як Ruby's Event Machine і Python's Twisted, має схожий дизайн. Однак, Node.js відзначається передовим використанням та розвитком моделі подій. Він презентує цикл подій як конструкцію часу виконання, а не просто як бібліотеку. В інших системах часто присутні блокуючі виклики, які використовуються для запуску циклу подій, тоді як у Node.js даний підхід не потрібний [10].



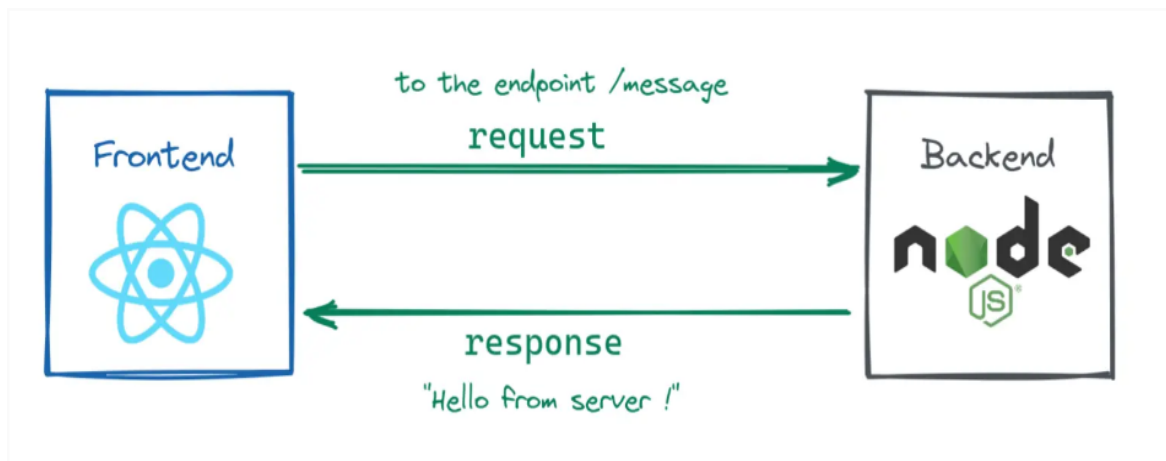


Рисунок 3.2 – Приклад взаємодії React та Node.js

Для підключення Node.js до проєкту необхідно виконати наступні кроки:

- встановити Node.js: Завантажити та встановити останню версію Node.js з офіційного веб-сайту [11]. Інсталятор встановить Node.js і одразу включить в себе пакетний менеджер NPM (Node Package Manager), який є частиною Node.js;
- ініціалізувати проєкт: Відкрити командний рядок або термінал у папці проєкту та виконайте команду `npm init`. Ця команда попросить ввести деякі основні дані про проєкт і створить `package.json` файл, який міститиме інформацію про проєкт;
- встановити залежності: Використовуючи NPM, можливо легко встановити залежності для проєкту. Наприклад, якщо є потреба у використанні Express.js, можна виконати команду `npm install express`. Це завантажить та встановить Express.js та всі його залежності в папку `node_modules` проєкту. Також, є можливість встановити всі необхідні залежності, використовуючи подібний підхід;
- створення файлу сервера: Проєкт буде потребувати файл сервера, який буде відповідати за обробку запитів. Можна створити файл, наприклад, з назвою `server.js` і використовувати його для налаштування сервера з використанням Node.js;

– налаштування сервера: У файлі сервера (server.js або іншому файлі) можна використовувати Node.js для налаштування серверного середовища, визначення маршрутів та обробки запитів. Можна використовувати різні фреймворки або модулі, такі як Express.js, Hapi.js, Koa.js тощо, для створення серверної логіки та обробки маршрутів;

– запуск сервера: Запустіть сервер, виконавши команду node server.js або інший файл, в якому було налаштовано серверну логіку. Сервер почне прослуховувати вказаний порт (наприклад, 3000) і оброблятиме запити, які надходять на вказаний порт.

Це лише загальний опис процесу підключення Node.js до проєкту. Залежно від потреб проєкту і використовуваних фреймворків або модулів, можуть бути додаткові кроки або особливості у налаштуванні середовища розробки.

Таким чином, Node.js - це потужна технологія, яка дозволяє виконувати JavaScript на стороні сервера. Вона забезпечує розширені можливості для розробки серверної логіки, обробки запитів, підключення до баз даних та інтеграції з іншими службами. За допомогою Node.js, розробники можуть створювати потужні та ефективні серверні додатки, що дозволяють взаємодіяти з клієнтами та забезпечувати надійну роботу веб-проєктів.

### 3.5 Express.js

Express.js є найпопулярнішим веб-фреймворком для Node.js. Він призначений для створення веб-додатків і API і був названий де-факто стандартною серверною структурою для Node.js.

Створення бекенду з нуля для програми на Node.js може бути важким і трудомістким. Від налаштування портів до обробників маршрутів, написання всього шаблонного коду позбавляє того, що справді має значення, а саме написання бізнес-логіки для програми. Використовуючи такі веб-фреймворки, як Express.js, розробники можуть заощадити час і зосередитися на інших важливих завданнях [12].

Один з ключових елементів Express.js - це його концепція маршрутів (routes), яка дозволяє логічно організувати обробку різних URL-шляхів. За допомогою Express.js можна легко визначити маршрути та пов'язані з ними обробники, які відповідають на запити та виконують певні дії, такі як отримання даних з бази даних або відправлення відповіді клієнту.

Express.js також дозволяє легко підключати і використовувати додаткові модулі та бібліотеки для розширення його функціональності. Це дає можливість використовувати різноманітні модулі для роботи з базами даних, шаблонізації HTML та інших аспектів розробки веб-додатків.

### 3.6 База даних

Для створення застосунку обрано документо-орієнтовану базу даних MongoDB, адже це система, призначена для швидкої розробки веб-додатків та інтернет-інфраструктури. Модель збереження даних і стратегії, створеної для високої пропускної здатності читання та запису та здатності легкого масштабування за допомогою автоматичного відновлення після відмови. Незалежно від того, чи потрібен програмі лише один вузол бази даних або їх десятки, MongoDB може забезпечити напруження високої продуктивності [13].

MongoDB є базою даних типу NoSQL (Not Only SQL). Це означає, що MongoDB не використовує традиційну реляційну модель даних, яку використовують у SQL базах даних, таких як MySQL або PostgreSQL. Замість цього, MongoDB використовує документоорієнтований підхід до зберігання та опрацювання даних.

В реляційних базах даних, дані організовані у вигляді таблиць, що містять рядки та стовпці зі структурою, заданою заздалегідь. У MongoDB, дані зберігаються у вигляді колекцій, що містять документи формату BSON (Binary JSON). Документи можуть мати різну структуру, і не потребують заданої схеми, що дозволяє гнучко працювати з даними і змінювати їх структуру під час розвитку проекту.

База даних MongoDB є чудовим вибором для вебзастосунків, насамперед, коли програма обслуговує багато користувачів, які не взаємодіють. Наприклад, програми фінансових послуг вимагають від користувачів доступу лише до їхніх даних.

Щоб отримати доступ до всієї інформації про користувача, реляційній базі даних потрібно буде зробити кілька запитів або складних запитів JOIN. Безсхемна база даних документів MongoDB дозволяє зберігати всю інформацію про користувача в одному місці [14].

Одна з головних переваг MongoDB полягає в тому, що вона зберігає дані у вигляді документів формату JSON, відомих як BSON (Binary JSON). Це дозволяє легко і просто зберігати та опрацьовувати дані без потреби реляційних схем, що вимагають жорстко визначених структур даних. Це особливо корисно під час розробки динамічних веб-додатків, де структура даних може змінюватися з часом.

MongoDB також має вбудовану підтримку реплікації та горизонтального масштабування, що робить її підходящою для розробки масштабованих та надійних додатків. Вона може легко інтегруватися з середовищем Node.js та бібліотекою React, оскільки має офіційні бібліотеки та драйвери для роботи з цими технологіями.

Завдяки цим перевагам, MongoDB є популярним вибором для розробки вебзастосунків разом з React та Node.js, дозволяючи розробникам швидко та зручно зберігати, опрацьовувати та отримувати доступ до даних у своїх проектах.

Коли мова йде про стек розробки, що об'єднує React, Node.js, Express.js та MongoDB, це надзвичайно потужний та популярний стек. React використовується для розробки фронтенду, Node.js та Express.js - для розробки серверної частини, а MongoDB - як база даних.

Даний стек надає зручну та ефективну платформу для розробки повнофункціональних веб-додатків. React забезпечує потужні можливості управління станом та компонентною моделлю розробки, а Node.js з Express.js дозволяють побудувати швидкий та масштабований сервер. MongoDB, як база

даних типу NoSQL, пропонує гнучку схему даних та швидкий доступ до інформації.

### 3.7 Вибір хостингів

Розділення програми на фронтенд та бекенд є стратегічно важливою практикою при розробці програмного забезпечення. Даний підхід дозволяє розділити відповідальності між клієнтською та серверною частинами, забезпечуючи кращу організацію та масштабованість проекту. Фронтенд відповідає за інтерфейс користувача та взаємодію з ним, тоді як бекенд обробляє бізнес-логіку, зберігає та обробляє дані. Даний підхід також дозволяє використовувати різні технології та мови програмування для кожної компоненти, що забезпечує гнучкість та незалежність від технологій.

Основні переваги розподіленої системи:

- розділення відповідальностей: Фронтенд та бекенд виконують різні функції і мають різні відповідальності. Фронтенд відповідає за взаємодію з користувачем, представлення даних та інтерфейс користувача. Бекенд відповідає за обробку бізнес-логіки, збереження даних, аутентифікацію, та взаємодію зі зовнішніми службами. Розділення цих функцій дозволяє краще управляти розробкою;
- масштабованість: Розділення на фронтенд та бекенд спрощує масштабування системи. Наприклад, можна масштабувати бекенд-сервери, не змінюючи фронтенд-код. Це дає можливість гнучко реагувати на зростання навантаження та забезпечувати стабільну роботу системи;
- підтримка: Розділення спрощує підтримку та розробку системи. Команди розробників можуть працювати паралельно над фронтендом та бекендом, не заважаючи одне одному;
- технологічна незалежність: Розділення на фронтенд та бекенд дає можливість використовувати різні технології та мови програмування для кожної частини системи;

– безкоштовне розгортання проекту: Безкоштовні плани можуть мати обмеження по ресурсах, таким як обсяг трафіку, потужність сервера тощо. Але для запуску та початку тестування продукту Netlify та Render дають унікальну змогу захостити проект безкоштовно.

Отже, розділення програми на фронтенд та бекенд є стратегічно важливим рішенням, яке забезпечує кращу організацію, масштабованість та незалежність від технологій. Даний підхід дозволяє ефективно розробляти та підтримувати програмні проекти, забезпечуючи зручну взаємодію з користувачами та надійну обробку бізнес-логіки на серверній стороні.

### 3.7.1 Netlify

Для хостингу фронтенду вебзастосунку в стеку розробки з React та Node.js була обрана платформа Netlify. Даний вибір зумовлений його простотою використання, потужними інструментами автоматизації та підтримкою для статичного та серверного рендерингу React-додатків. Завдяки Netlify є змога легко та ефективно розгортати додатки, забезпечуючи швидкість, надійність та зручність для наших користувачів.

Одна з головних переваг Netlify полягає в його простоті використання та налаштуванні. Розгортання вебзастосунків на Netlify є швидким та простим процесом завдяки його інтуїтивному інтерфейсу та потужному інструментарію автоматизації. Можна зв'язати репозиторій з Netlify та автоматично розгортати зміни з кожним комітом до репозиторію, що значно спрощує процес розробки та публікації вебзастосунків. Netlify виявляє зміни, коли надсилаються зміни до Git, і запускає автоматичне розгортання, переміщуючи глобальні активи, логіку, функції тощо через всю інфраструктуру [15].

Крім того, Netlify має вбудовану підтримку для статичного рендерингу та серверного рендерингу React-додатків, що дає можливість зберігати та обслуговувати сторінки React як статичні файли або виконувати серверний

рендеринг за допомогою Node.js. Це дозволяє покращити швидкість завантаження додатку та покращити його показники продуктивності.

Також варто зазначити, що Netlify має вбудовану підтримку для інших функціональних можливостей, таких як контроль версій, автоматичні SSL-сертифікати, форми зворотного зв'язку та багато іншого. Всі ці можливості забезпечують швидку та зручну розробку та розгортання веб-додатків на цій платформі.

У підсумку, Netlify надає зручний та простий у використанні інтерфейс, який спрощує розгортання та керування додатками. Його можливості статичного та серверного рендерингу дозволяють покращити продуктивність та швидкість завантаження веб-додатків. Завдяки розширеним функціональним можливостям та інтеграції з сучасним стеком розробки, Netlify сприяє швидкій та зручній розробці, забезпечуючи надійний хостинг та підтримку для нашого проєкту.

### 3.7.2 Render

Для хостингу бекенду був обраний Render, це уніфікований хмарний сервіс для створення та запуску програм і веб-сайтів із безкоштовними сертифікатами TLS, глобальним CDN, захистом від DDoS, приватними мережами та автоматичним розгортанням від Git [16].

Render забезпечує простоту у використанні, оскільки він пропонує інтуїтивний інтерфейс та простий процес розгортання бекенд-додатків. Можна легко підключити свій репозиторій з кодом та автоматично розгорнути бекенд на Render. Крім того, Render підтримує багато популярних технологій, таких як Node.js, Python, Ruby, Go та багато інших, що дає вибір і гнучкість при розробці бекенд часатини [17].

Одне із ключових переваг Render є його висока продуктивність. Використовуючи високопродуктивні сервери та оптимізовані конфігурації, Render забезпечує швидку та ефективну обробку запитів до нашого бекенду. Це особливо

важливо, коли додаток залежить від швидкості та масштабованості для задоволення потреб аудиторії.

Render є надійною, простою у використанні та продуктивною платформою хостингу для розгортання бекенду. Його підтримка різноманітних технологій, висока продуктивність та надійність роблять Render оптимальним рішенням для розробників, що шукають ефективний спосіб розгортання та управління бекенд-додатками.

### 3.8 Платіжна система. Stripe

Для вебзастосунку потрібно підключити платіжну систему, Stripe для цього повністю підходить. Stripe є потужною платіжною міжнародною системою, яка чудово підходить для інтеграції в бекенд-додатки. Він надає зручні та безпечні механізми оплати, дозволяючи здійснювати транзакції в реальному часі та підтримувати різні способи оплати.

Що підтверджує безпечність Stripe [18]?

- захист SSL: Усі транзакції захищені протоколом SSL (Secure Socket Layer). Наша інформація та інформація клієнтів безпечно передається під час обробки всіх платежів;
- сумісність з PCI: Stripe є постачальником послуг PCI рівня 1, який забезпечує найвищий рівень безпеки обробки платежів;
- зашифрованість: Усі номери кредитних карток зашифровані та безпечно зберігаються в сучасному центрі обробки даних Stripe. Це гарантує як безпеку, так і цілісність інформації про клієнтів.

Крім того, Stripe надає багато корисних інструментів для аналізу платежів, відстеження стану транзакцій та керування платіжними профілями. Можна легко отримувати звіти про прибуток, збори та іншу статистику, що дозволяє краще розуміти фінансовий стан бізнесу.

Stripe є потужною та безпечною платіжною системою, яка ідеально підходить для інтеграції у вебзастосунки. Використання Stripe дозволяє

забезпечити зручну та безпечну оплату, підтримувати різноманітні способи оплати та використовувати різні моделі платежів. Stripe також надає багато корисних інструментів для аналізу та керування платежами, що допомагає краще управляти фінансовою складовою проєкту.

### Висновок до розділу

У даному розділі дипломного проєкту були розглянуті та обрані інструменти та технології для реалізації поставлених задач, щодо побудови Telegram Web App Bot. Використання платформи Telegram та бібліотеки Telegraf надають потужні можливості для взаємодії з користувачами, створення вебзастосунків та обробки повідомлень.

Для побудови фронтенду була обрана бібліотека React, яка дозволяє створювати динамічні та ефективні інтерфейси користувача. На стороні сервера були використані Node.js та Express.js для створення бекенду, який забезпечує обробку запитів та взаємодію з базою даних.

MongoDB була обрана як база даних для зберігання історії покупок та іншої інформації. Вона пропонує гнучку та масштабовану структуру для зберігання нереляційних даних (NoSQL), що дозволяє зручно та швидко здійснювати пошук та фільтрацію даних.

Для забезпечення платіжних можливостей у застосунку була обрана та інтегрована платіжна система Stripe. Вона надає безпечні та зручні інструменти для обробки платежів, що дозволяє користувачам легко та безпечно здійснювати оплату за продукти.

Комбінація цих технологій та інструментів дозволяє створити потужний та функціональний Telegram Web App Bot зі зручним інтерфейсом, можливостями платіжних операцій та збереженням даних. Вибір цих технологій забезпечує швидку розробку, гнучкість та надійність системи, що є ключовими аспектами для успішної реалізації та використання бота у різних сценаріях та середовищах.

## 4 РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

### 4.1 Загальний опис реалізації

У цьому розділі розглянуто реалізацію вебзастосунку для продажу товарів через Telegram Web App Bot. Даний додаток складається з двох основних компонентів: фронтенду і бекенду. Фронтенд відповідає за візуальну частину додатку, взаємодію з користувачем та відображення інформації про доступні продукти. Бекенд відповідає за обробку запитів, зв'язок з базою даних та виконання бізнес-логіки. Загальну роботу системи можна переглянути на кресленику ІА94.180БАК.004 ДЗ.

Основна взаємодія між фронтендом і бекендом полягає в обміні даними. Фронтенд здійснює запити до бекенду, передаючи необхідні параметри. Бекенд, отримавши запит, оброблює його, виконує необхідні операції, такі як отримання списку доступних продуктів з бази даних та підготовку даних для відображення на фронтенді.

Окремо варто згадати про інтеграцію міжнародної платіжної системи Stripe. При створенні нового продукту в системі, до його опису додається ціна, яка вказується в системі Stripe. Коли користувач вибирає продукт і бажає здійснити оплату, відбувається перенаправлення на сторінку Stripe, де користувач може внести необхідні дані для оплати.

У разі невдалої оплати, коли клієнт спробує здійснити оплату, але виникають помилки, система автоматично перенаправляє його на сторінку, яка повідомляє про невдачу операції. Це може бути пов'язано з помилкою при оплаті, недостатнім балансом на картці, відказом клієнта або іншими причинами. Клієнту буде відображено відповідне повідомлення про невдачу при оплаті, яке дозволить йому усвідомити, що оплата не відбулася успішно, та він зможе спробувати інший спосіб платежу.

У випадку успішної оплати, система автоматично створює замовлення для адміністратора. Це замовлення містить всю необхідну інформацію про здійснене покупцем замовлення, таку як ідентифікатор замовлення, список придбаних

									Лист
Зм.	Лист	№ докум.	Підпис	Дата					43

продуктів, деталі клієнта та інші додаткові дані. Ця інформація передається адміністратору, щоб він міг вжити відповідних заходів для обробки замовлення. Детально роботу системи можна переглянути на кресленику ІА94.180БАК.004 Д4.

Таким чином, система передбачає обробку невдалої оплати та відображення відповідного повідомлення про невдачу, а також автоматичне створення замовлення для адміністратора при успішній оплаті.

#### 4.2 Реєстрація чат-бота та отримання токенів

Зв'язок з Telegram API відбувається за допомогою спеціальних токенів, які виділяє BotFather - офіційний бот для створення і налаштування інших ботів в Telegram. Щоб отримати токен для бота, потрібно зареєструватись у BotFather, надіслати йому команду /newbot та слідувати інструкціям, приклад представлений на рисунку 4.1.

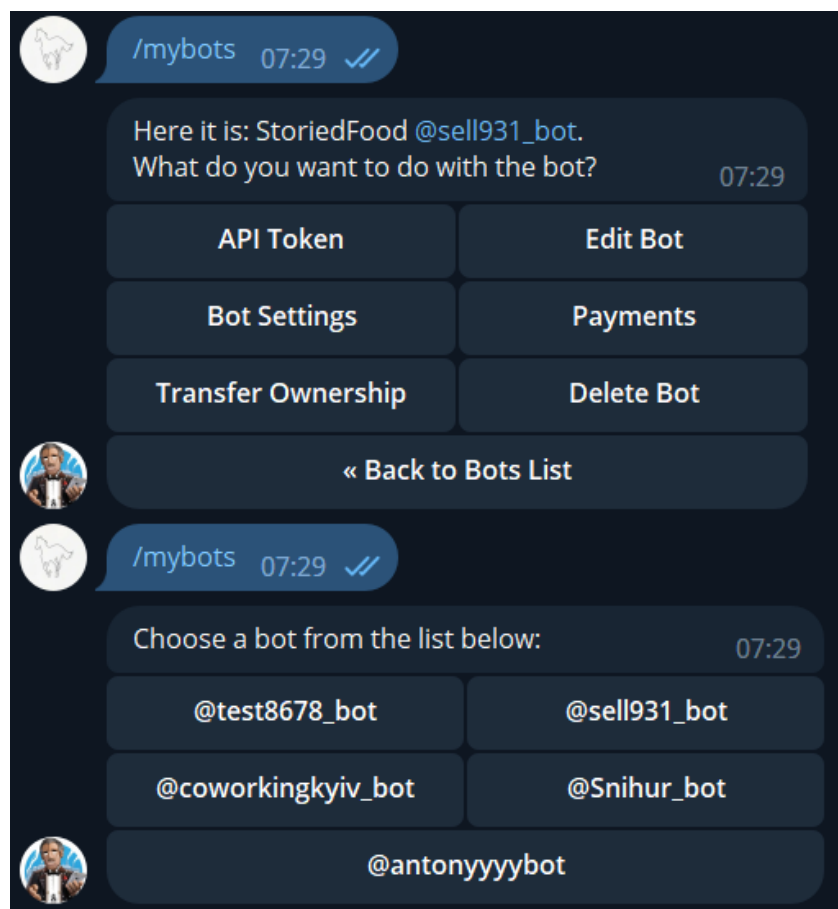


Рисунок 4.1 – Взаємодія з BotFather

Через BotFather проходить загальне налаштування бота. Потрібно вказати його ім'я, коротке ім'я для сповіщень, та надаючи токен, який потрібен для зв'язку з API Telegram.

Отримавши токен, можна використовувати API Telegram для розробки бота. Є декілька видів API, які можна використовувати, залежно від мети бота та його функціональності.

Основні види API, які можна використовувати для розробки чат-бота, це Bot API та Telegram API. Bot API дозволяє отримувати повідомлення від користувачів, відповідати на них та виконувати різні дії, такі як надсилання повідомлень, створення кнопок та інше. Telegram API дозволяє забезпечувати зв'язок між користувачами бота та виконанням різних операцій, таких як зміна інформації про користувачат та інше.

Для доступу до деяких функцій API, які пропонує Telegram, потрібні спеціальні права доступу. Наприклад, для використання кнопок в повідомленнях необхідно відкрити доступ до об'єкта InlineKeyboardMarkup.

Крім того, Telegram надає API для роботи зі статистикою та аналітикою, яка дозволяє збирати дані про використання бота. Це можна використовувати для аналізу роботи бота та його ефективності в різних сферах.

Також, для повноцінної роботи з вебзастосунком необхідно отримати API для підключення оплати. Для цього потрібно написати команду /mybots у BotFather та обрати пункт Payments. Після цього, відкривається перелік доступних платіжних систем, з яких потрібно обрати підходящий, зображено на рисунку 4.2.

Необхідно виконати реєстрацію у вибраній платіжній системі, пройти автентифікацію та подальше налаштування, після чого буде відправлений токен для оплати. Вже з цим токеном можна вільно налаштовувати банківські операції у боті.

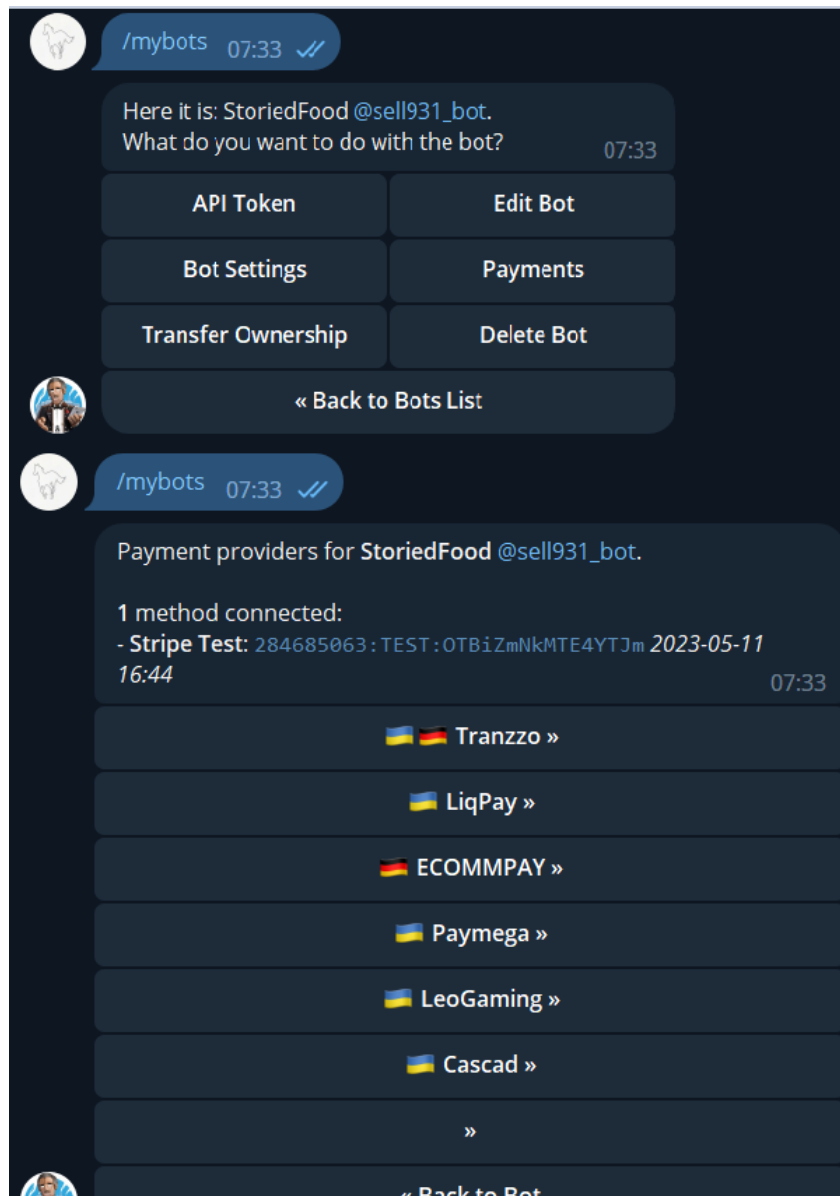


Рисунок 4.2 – Підключення оплати у BotFather

Отже, BotFather - це важливий інструмент для розробки чат-ботів на платформі Telegram, він допомагає налаштувати та отримати токен для зв'язку з API, а також надає доступ до різних видів API для реалізації функціоналу бота.

#### 4.3 Розробка та реалізація back-end частини

Даний проєкт розроблявся у досить відомому редакторі коду - Visual Studio Code, який був створений компанією Microsoft для таких операційних систем, як

Windows, Linux та macOS. На рисунку 4.3 можна побачити лістинг коду для бекенд частини [19].

```

  ✓ models
    JS Cart.js
    JS Order.js
    JS Product.js
    JS User.js
  > node_modules
  ✓ routes
    JS auth.js
    JS cart.js
    JS order.js
    JS product.js
    JS stripe.js
    JS user.js
    JS verifyToken.js
  > tgbot_rep
  > tgbot_StoriedFood
  ◆ .gitignore
  JS bot.js
  JS index.js
  {} package-lock.json
  {} package.json
  ⓘ README.md
```

Рисунок 4.3 – Лістинг бекенд частини

У папці models зберігаються файли, для створення колекцій в базі даних MongoDB. У папці routes, файли для поліпшення структури проекту. В routes продуктів винесені всі запити по продуктах і підключені в індекс файлі. У файлі bot.js налаштована взаємодія з ботом Telegram.

Для деплою серверної частини проекту був обраний хостинг Render. Для коректної роботи додатка потрібно додати секретний файл, який зображений на рисунку 4.4.

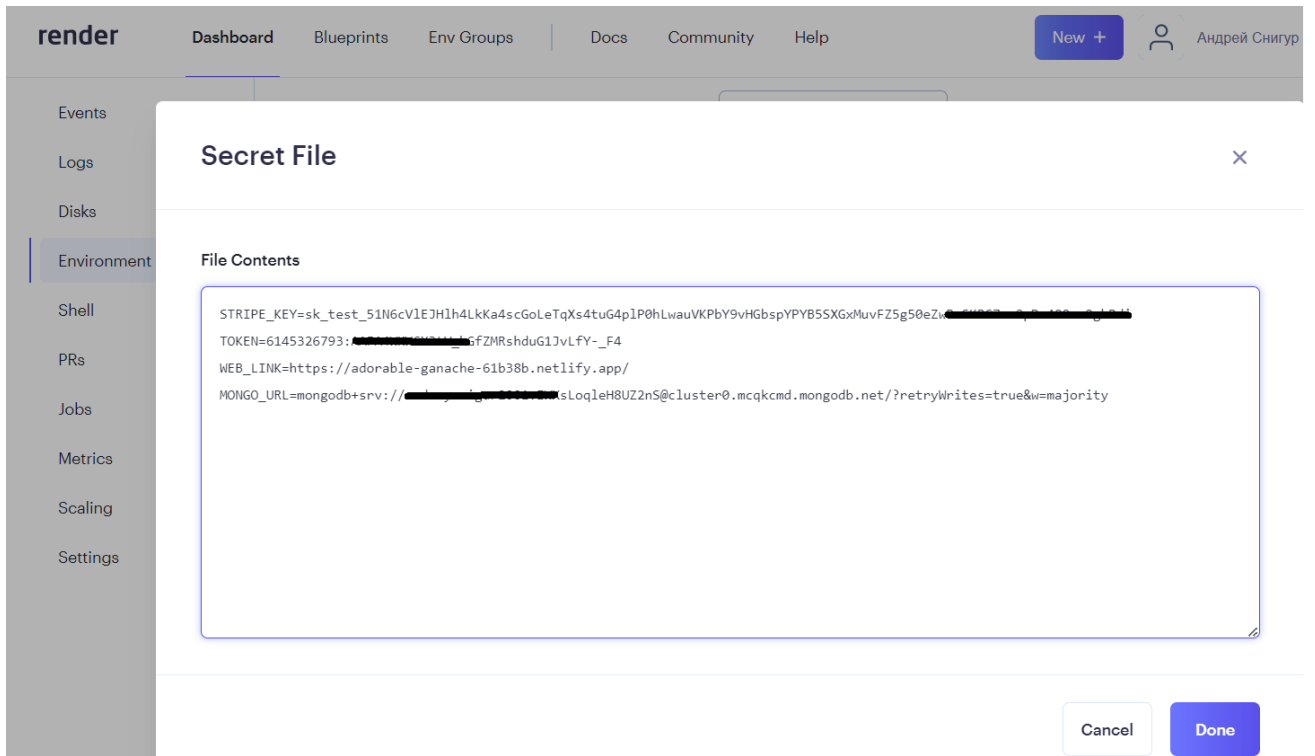


Рисунок 4.4 – Секретний файл Render

Елементи які було використано:

- STRIPE\_KEY – ключ для страйпу;
- TOKEN – токен для нашого телеграм боту;
- WEB\_LINK - посилання на фронт для редіректу зі страйпа;
- MONGO\_URL – посилання на mongodb для логіну.

#### 4.3.1 Налаштунок Stripe

Після успішного створення кабінету у страйпі та налаштуванні платіжного облікового запису, переходимо у кабінет Developers, де знаходиться вікно API Keys, там відображається згенерований особистий ключ для роботи з платіжною системою, який зображений на рисунку 4.5.

Рисунок 4.5 – Генерування ключа страйп

Ключ Stripe (STRIPE\_KEY) - це конфіденційний API-ключ, який використовується для інтеграції платіжної системи Stripe з нашим додатком. Він необхідний для забезпечення безпеки та автентифікації при здійсненні оплат через Stripe.

### 4.3.2 Налаштунок MongoDB

Наступним кроком у розгортанні системи є реєстрація та налаштування бази даних MongoDB, яка є потужною системою керування базами даних, що дозволяє зберігати інформацію в гнучкому форматі документів. Це дозволяє ефективно зберігати, опрацьовувати та отримувати доступ до даних вебзастосунку.

Для початку потрібно створити обліковий запис на MongoDB Atlas, який є хмарним сервісом бази даних MongoDB. Після реєстрації потрібно налаштувати базу даних, включаючи вибір регіону розташування серверів, конфігурацію кластеру та налаштування безпеки.

Після успішного налаштування отримаємо MongoURL, який представляє собою URL-адресу підключення до щойно створеної бази даних MongoDB. Цей

MongoURL буде використовуватись бекендом для підключення до бази даних та взаємодії з нею, він зображений на рисунку 4.6.

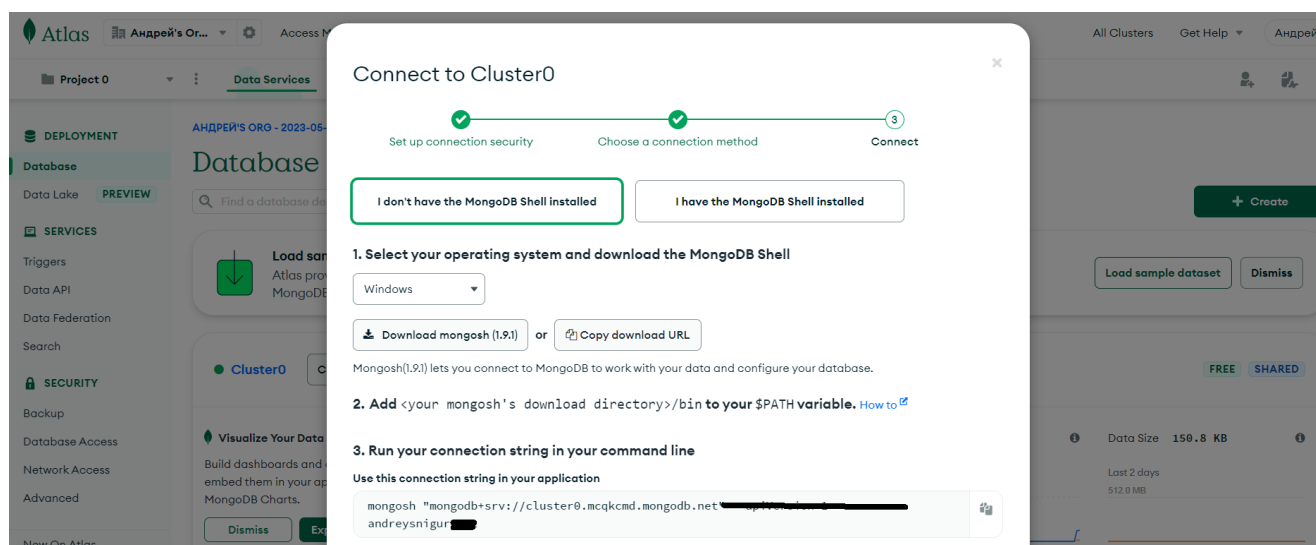


Рисунок 4.6– Отримання MongoURL

Завдяки базі даних MongoDB наш вебзастосунок зможе ефективно зберігати, отримувати та оновлювати дані, що дозволяє забезпечити потрібну функціональність та продуктивність для застосунку.

#### 4.4 Завершення налаштування серверної частини

Після налаштування цих елементів в середовищі Render, додаток буде готовий до використання - STRIPE\_KEY для платежів, токена телеграму для сповіщень, веб-посилання фронтенду для відображення інтерфейсу користувача та MongoURL для збереження та отримання даних у базі даних MongoDB.

Наступним етапом створюємо 2 веб-служби для серверу та для бота відповідно, це можна легко зробити завдяки підтримці GitHub з боку Render, створення вказано на рисунку 4.7.

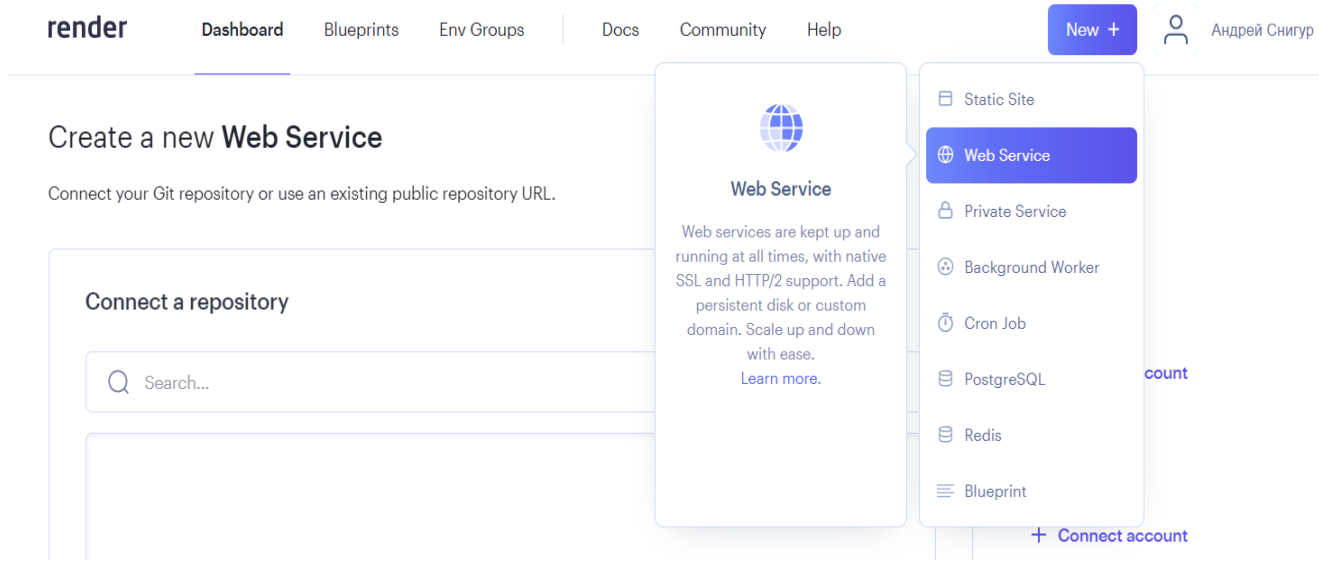


Рисунок 4.7 – Створення веб служб для подальшого деплою

Після успішного налаштування та прив'язки до GitHub отримаємо 2 готових задеплоєних веб-сервіси – server та bot, які зображені на рисунку 4.8.

Overview

Q Search services						
NAME	STATUS	TYPE	RUNTIME	REGION	▼ LAST DEPLOYED	
bot	Deploy in progress	Web Service	Node	Frankfurt	2 minutes ago	
server	Deploy succeeded	Web Service	Node	Frankfurt	7 hours ago	

Рисунок 4.8 – Налаштований хостинг для бекенду

4.5 Розробка та реалізація front-end частини

В підрозділі продемонстровано роботу фронтенд частини проекту. На рисунку 4.9 можна побачити лістинг коду для клієнтської частини [20].

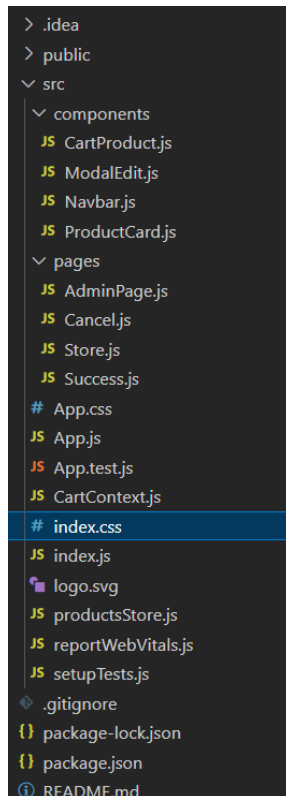


Рисунок 4.9 – Лістинг фронтенд частини

Директорія src відповідає за візуальну частину та логіку програми. Наприклад, у вкладці pages створені сторінки, які відповідають за перенаправлення користувача на відповідні сторінки після успішної або невдалої покупки, цю взаємодію можна переглянути у файлі App.js. На рисунку 4.10 зображено фрагмент коду для перенаправлення на сторінки, повний код файлу міститься у додатку А.

```
12 function App() {
13   return (
14     <CartProvider>
15       <Container>
16         <NavbarComponent></NavbarComponent>
17         <BrowserRouter>
18           <Routes>
19             <Route path='store_admin' element={<AdminPage />} />
20             <Route path='store' element={<Store />} />
21             <Route path='success' element={<Success />} />
22             <Route path='cancel' element={<Cancel />} />
23             <Route path='*' element={<Navigate replace to='store' />} />
24           </Routes>
25         </BrowserRouter>
26       </Container>
27     </CartProvider>
28   );
29 }
```

Рисунок 4.10 – Перенаправлення на сторінки

Для хостингу фронтенд частини проекту у Netlify, знадобиться привязати його до другого репозиторія. Після успішної синхронізації, ми отримаємо посилання на вебзастосунок, яке зображене на рисунку 4.11.

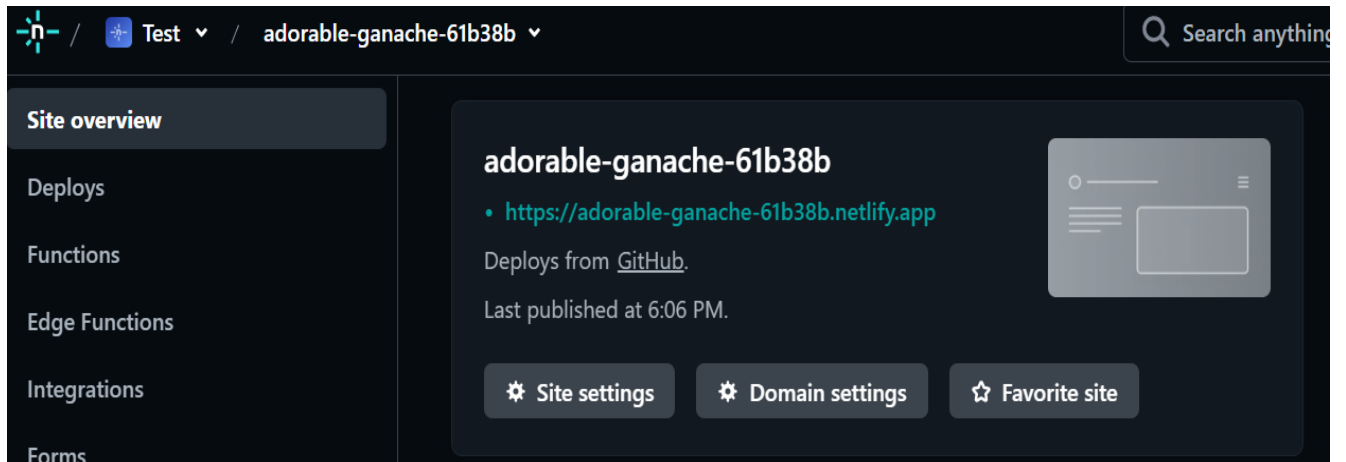


Рисунок 4.11 – Посилання на проект у Netlify

Це посилання також додаємо у секретний файл у Render. Після цього, серверна та клієнтська частина буде інтегрована одна з одною.

#### 4.6 Інтеграція застосунку в Telegram

Переходимо у BotFather у Telegram. Для початку потрібно написати команду /newapp яка допоможе інтегрувати вебзастосунок у месенджер. Після цієї команди потрібно вибрати серед ботів, потрібний. Далі потрібно надіслати йому раніше згенероване посилання з Netlify, після чого слідує налаштування елементів, таких як: невеликий опис бота, обрання картинки або GIF та інші гнучкі налаштування, які вказані на рисунку 4.12.

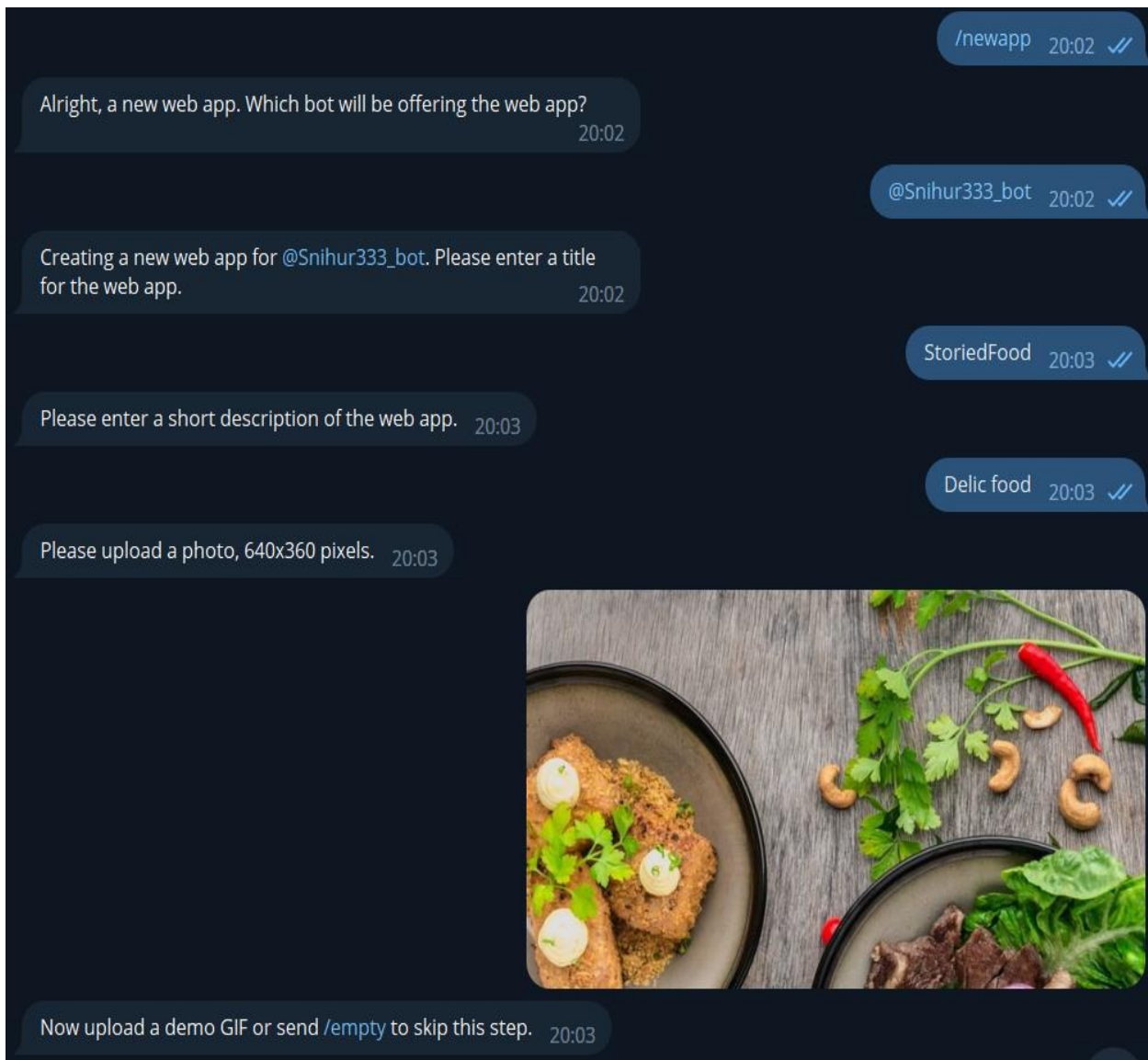


Рисунок 4.12 – Інтеграція вебзастосунку у Telegram

При потребі задані налаштування бота завжди можна змінити завдяки таким командам як `/myapps` (дозволяє редагувати налаштування веб-програми) та `/mybots` (дозволяє редагувати кожний аспект бота). BotFather має багато корисних можливостей, таких як регенерування токенів, встановлення приватності у ботах та інше.

#### Висновок до розділу

У даному розділі дипломного проекту були описані основні складові реалізації Telegram Web App Bot з використанням різноманітних технологій та

інструментів. Були розроблені клієнтська та серверна частина додатка, а також налаштовані хостинги та платіжна система.

Починаючи з клієнтської частини, яка була побудована з використанням React, було створено інтуїтивний та зручний інтерфейс для користувачів. Реалізація серверної частини була заснована на Node.js та Express.js, що забезпечило обробку запитів та взаємодію з базою даних MongoDB.

У процесі розробки веб-застосунку було використано два хостингу для розгортання фронтенду та бекенду. Хостинг Netlify був обраний для фронтенду, забезпечуючи його автоматичне оновлення через GitHub. Render, у свою чергу, виявився оптимальним варіантом для хостингу бекенду, забезпечуючи швидке та надійне розгортання додатку.

Для забезпечення функціональності оплати була інтегрована платіжна система Stripe, яка дозволяє користувачам здійснювати безпечні та зручні платежі за продукти. Завдяки цьому, клієнти можуть легко оплачувати свої замовлення і отримувати відповідне підтвердження.

В результаті роботи у даному розділі було успішно реалізовано повноцінний вебзастосунок, який забезпечує користувачам зручну та зрозумілу взаємодію з Telegram Bot. Застосування сучасних технологій та інструментів дозволило створити потужний та ефективний додаток, який готовий до використання в комерційних цілях.

## 5 КЕРІВНИЦТВО З ВИКОРИСТАННЯ

### 5.1 Інструкція адміністратора

У цьому розділі наведено детальну інструкцію для адміністратора вебзастосунку на базі Telegram. Інструкція має на меті допомогти розробникам оволодіти основними функціями та завданнями, які вони можуть виконувати у системі. За допомогою інструкції, адміністратори зможуть ефективно керувати вебзастосунком, додавати нові позиції, переглядати історію замовлень та забезпечувати безперебійну роботу системи.

Для зручності користувачів було розроблено інтуїтивно зрозумілий і простий у використанні інтерфейс адміністратора. Це дозволить адміністраторам швидко та ефективно здійснювати необхідні операції у системі.

У цьому розділі будуть наведені детальні кроки та пояснення щодо запуску хостингів, додавання нових позицій у меню та перегляду історії замовлень. Крім того, будуть надані необхідні пояснення щодо роботи з платіжною системою та забезпечення безперебійної роботи системи.

#### 5.1.1 Запуск хостингів

Перш ніж розпочати використання вебзастосунку у Telegram, потрібно налаштувати та запуснути хостинги, що забезпечать доступність додатка в Інтернеті та забезпечать його безперебійну роботу. У цьому розділі наведені кроки, які допоможуть розпочати роботу з хостинговими платформами та налаштуванням вебзастосунку.

Запуск хостингів є важливим етапом, оскільки він дозволяє забезпечити доступ користувачів до нашого застосунка з будь-якого місця та в будь-який час. Крім того, він забезпечує масштабованість та надійність системи, дозволяючи обробляти більше користувачів та забезпечувати швидку реакцію на їх запити.

Для запуску або перезапуску бекенд частини проєкту, потрібно зайти у особистий кабінет користувача, перейти до вкладки «Dashboard» обрати потрібний веб-сервіс та натиснути «Deploy latest commit», зображено на рисунку

5.1). Після невеликого очікування система обробить останні зміни у проєкті та запусить сервіс.

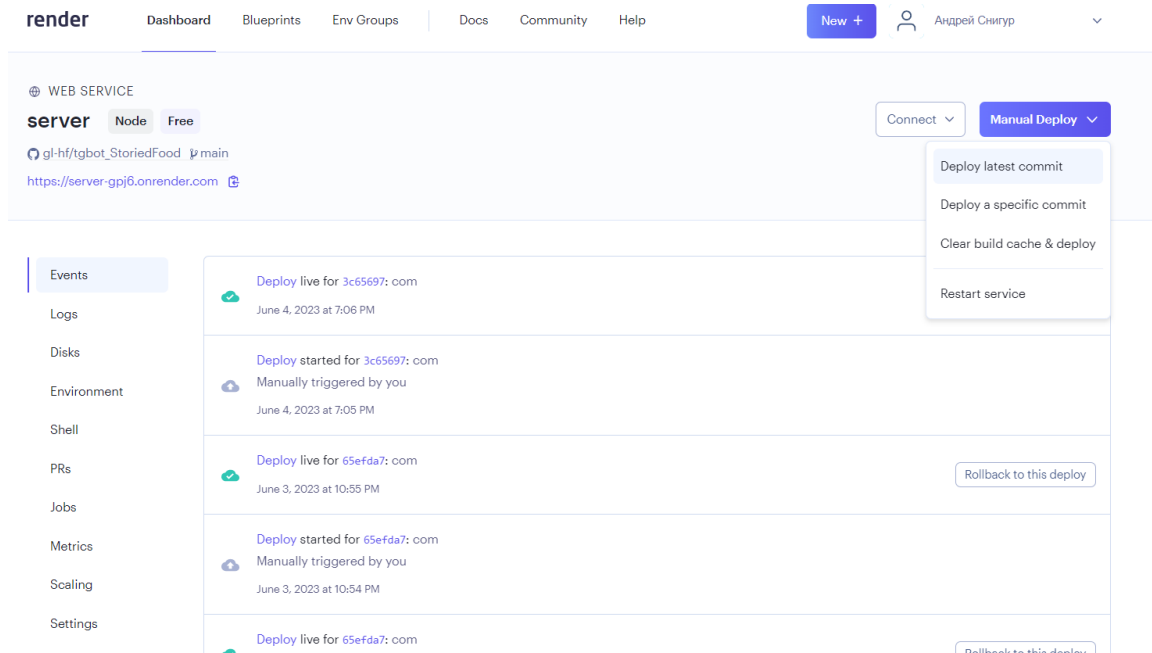


Рисунок 5.1 – Запуск хостингу Render

Подібна процедура є і у Netlify, якщо з якоїсь причини хостинг Netlify не працює коректно можна його перезапустити вручну. Для цього у особистому кабінеті потрібно зайти у вкладку «Deploys» та натиснути на кнопку розгортання «Trigger deploy» та обрати пункт «Deploy site», зображено на рисунку 5.2.

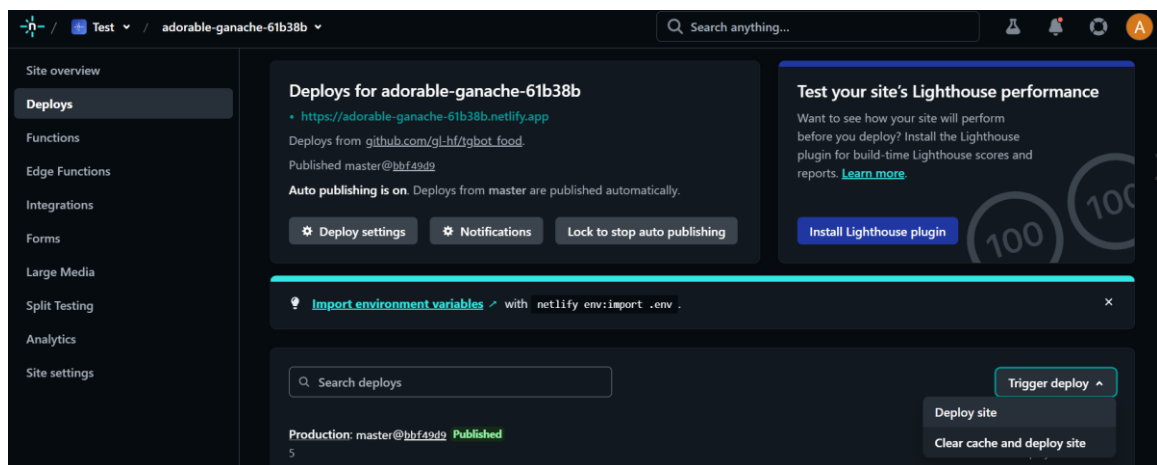


Рисунок 5.2 – Запуск хостингу Netlify

Зм.	Лист	№ докум.	Підпис	Дата

## 5.1.2 Додавання нових позицій

Один із важливих аспектів управління вебзастосунку - це здатність додавати нові позиції для подальшого продажу. Даний підрозділ надасть чіткі кроки та інструкції, як додавати нові позиції та забезпечувати актуальність асортименту в додатку.

Додавання нових позицій є процесом, який дозволяє адміністраторам оновлювати список товарів або послуг, доступних для замовлення. Це важливо для забезпечення різноманітності та задоволення потреб користувачів. Надаючи адміністраторам доступ до інструментів додавання позицій, це дає їм можливість легко оновлювати і розширювати асортимент продуктів чи послуг у вебзастосунку.

Для початку переходимо до платіжної системи Stripe у вже зареєстрований та прив'язаний особистий кабінет, та переходимо у вкладку Products, де натискаємо на Add product. Після чого можемо додати нові позиції у магазин, детально зображено на рисунку 5.3.

The screenshot displays the Stripe 'Add product' interface. It is divided into two main sections: 'Product details' and 'Price information'.  
**Product details:** Includes a 'Name' field with the text 'Premium Plan, sunglasses, etc.', a 'Description' field with a placeholder and a 'Description' label, and an 'Image' section with an 'Image' label and an 'Upload' button.  
**Price information:** Includes a 'Pricing model' dropdown set to 'Standard pricing', a 'Price' field set to '€ 0.00' with a currency dropdown set to 'EUR', and radio buttons for 'Recurring' (selected) and 'One time'. Below this is a 'Billing period' dropdown set to 'Monthly' and a checkbox for 'Usage is metered'. A green checkmark indicates 'This is the default price'. There are also expandable sections for 'Additional options' in both parts.

Рисунок 5.3 – Додавання нових позицій через Stripe

Тут потрібно заповнити усі необхідні поля. Після чого, отримуємо товарний API ID для щойно створеного продукту, зображено на рисунку 5.4, який знадобиться для додавання його до панелі адміністратора.

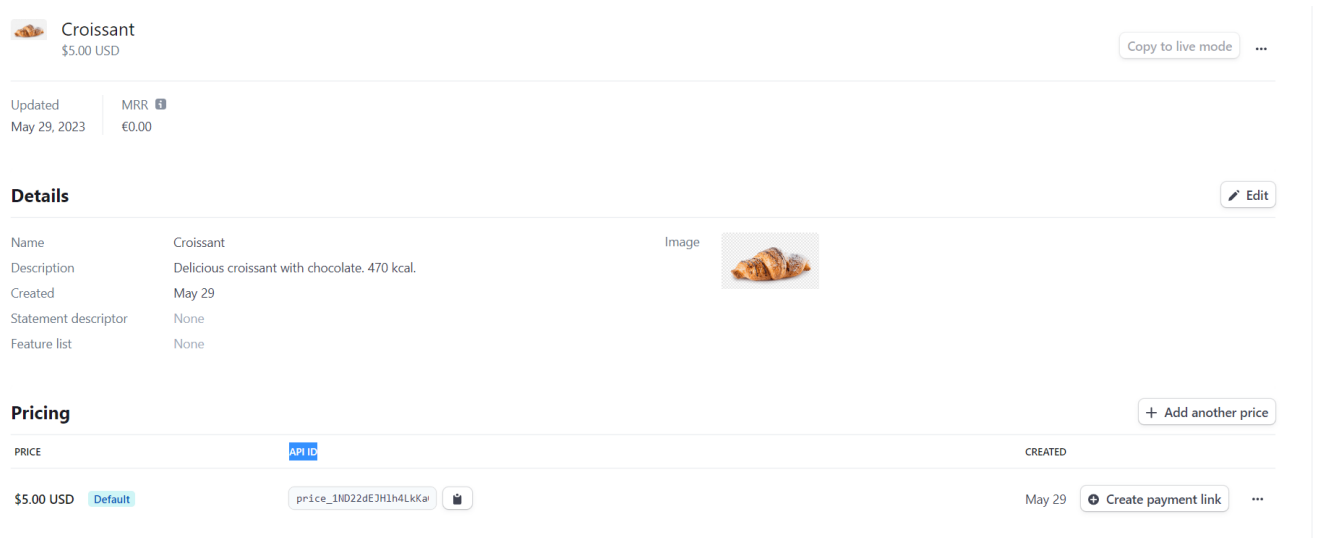


Рисунок 5.4 – Отримання API ID продукту у Stripe

API ID продукту в Stripe використовується для ідентифікації конкретного продукту. Кожен продукт в Stripe має унікальний ідентифікатор, який використовується для взаємодії з цим продуктом за допомогою програмного інтерфейсу (API).

Переходимо до панелі адміністратора, вебзастосунок налаштований так що у власника є змога додавати/видаляти/редагувати потрібні позиції, вказано на рисунок 5.5.

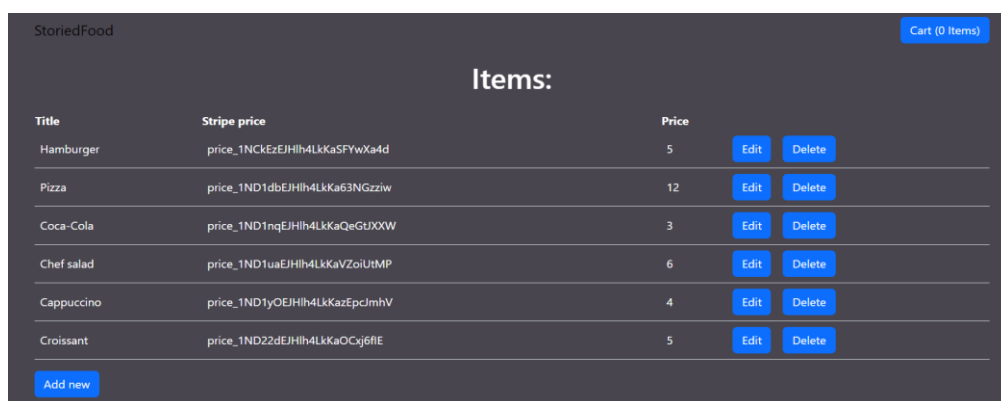


Рисунок 5.5 – Редагування позицій в меню у панелі адміністратора

При додаванні нових позицій, у панелі адміністратора потрібно заповнити 3 поля, зображено на рисунку 5.6:

- Title: назва позиції;
- Stripe price: API ID продукту в Stripe;
- Price: ціна (повинна бути ідентичною до ціни у Stripe).

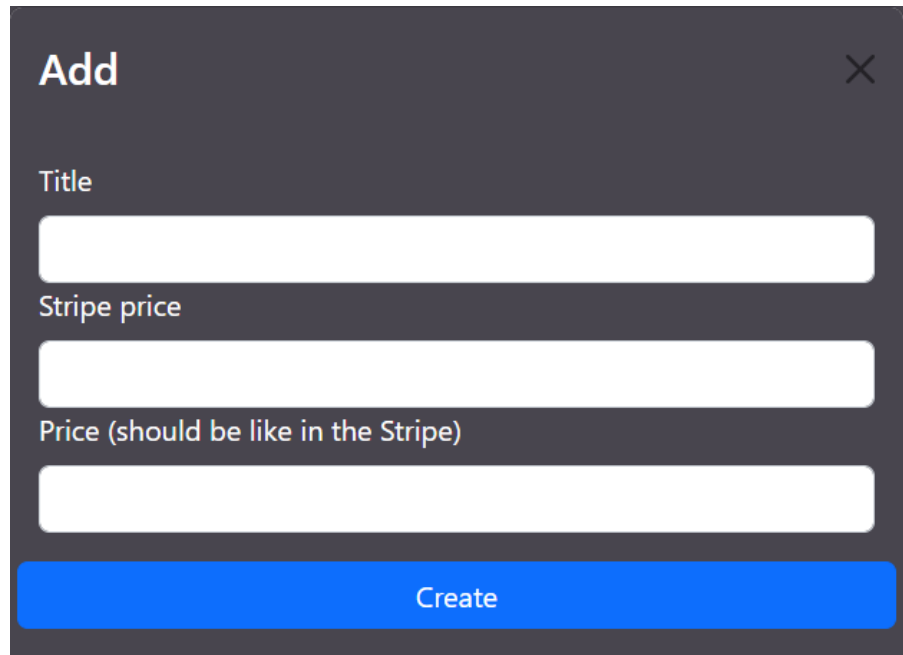


Рисунок 5.6 – Додавання нової позиції у панелі адміністратора

Після коректного налаштування у страйпі та панелі адміністратора нова позиція з'явиться на головній сторінці вебзастосунку.

### 5.1.3 Перегляд історії

Одним з ключових аспектів користування вебзастосунком у Telegram - це можливість адміністратора переглядати історію замовлень. У цьому підрозділі надано необхідну інформацію та кроки, які дозволять зручно відстежувати та аналізувати замовлення, що здійснюються у додатку.

Перегляд історії замовлень є важливим функціональним елементом для ефективного управління бізнесом у вебзастосунку. Він дозволяє адміністраторам

отримати доступ до повної інформації про кожне замовлення, включаючи деталі клієнта, товар чи послугу, дату замовлення та інші важливі дані. Це сприяє кращому розумінню потреб клієнтів, плануванню запасів і аналізу ефективності бізнесу.

Розглянемо як зберігаються замовлення у нашій базі даних, у страйпі та панелі адміністратора.

Так як MongoDB є нереляційною базою даних, то кожен запис про покупку може бути представлений у вигляді документа JSON, який містить різні поля, що описують деталі покупки, зображено на рисунку 5.7.

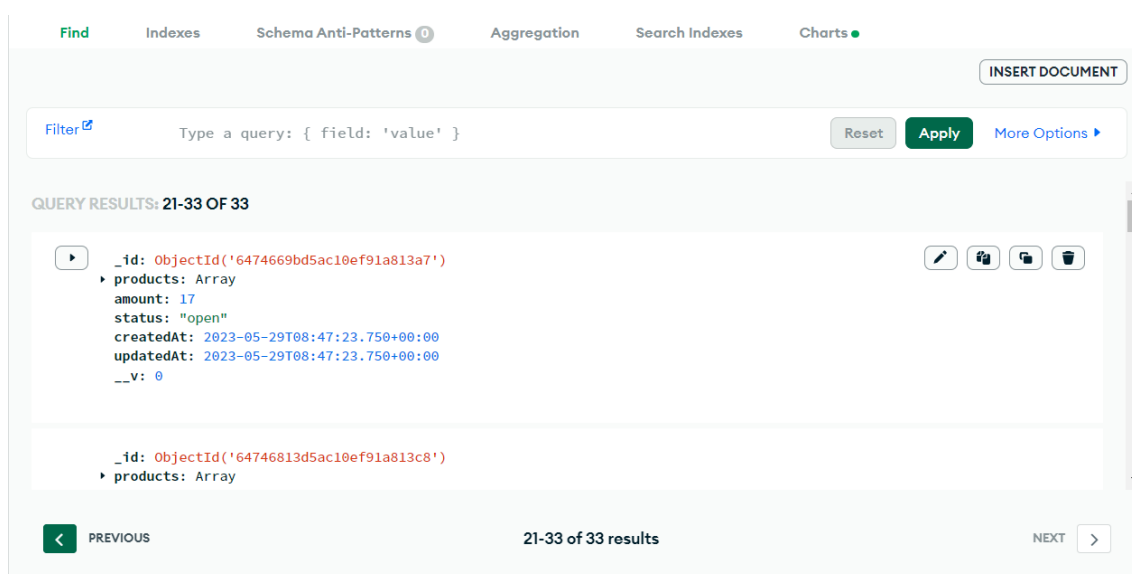


Рисунок 5.7 – Збережені записи у БД

На рисунку 5.8 зображена панель адміністратора де реалізована змога перегляду усіх замовлень, вони підтягуються на головну сторінку адміністратора з бази даних.

Orders:		
Time	Products	Amount
5/28/2023, 4:07:27 PM	Hamburger Quantity - 1	5
5/28/2023, 4:07:27 PM	Hamburger Quantity - 1	5
5/28/2023, 7:54:37 PM	Hamburger Quantity - 1	5
5/28/2023, 7:54:39 PM	Hamburger Quantity - 1	5
5/28/2023, 7:55:31 PM	Hamburger Quantity - 8	40

Рисунок 5.8 – Історія заказів у панелі адміністратора

Платіжна система Stripe надає розгорнуту інформацію про історію покупців. У системі є можливість, як подивитися повну історію, так і перейти і подивитися детально кожне замовлення окремо, детально зображено на рисунку 5.9.

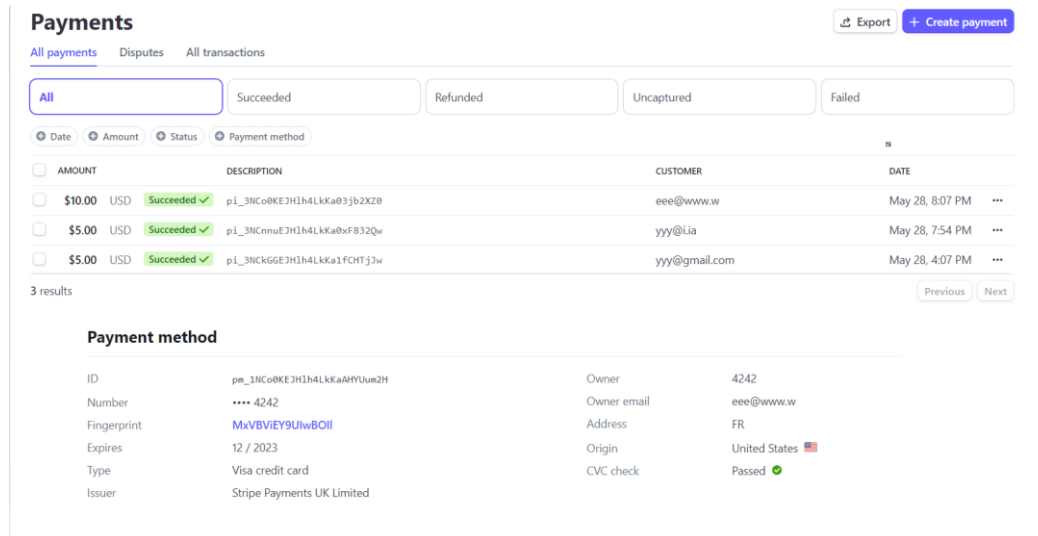


Рисунок 5.9 – Перегляд замовлень у Stripe

Є змога подивитися всю інформацію про користувача:

- ID;
- Інформацію банківської картки;
- Fingerprint (це унікальний ідентифікатор, який використовується для виявлення та запобігання шахрайській діяльності);
- ПІБ;
- Email;
- Час замовлення;
- Адреса.

Також, у Stripe є фільтрація та сортування замовлень. Це дозволяє фільтрувати та сортувати замовлення відповідно до поточних потреб. Можна застосовувати різні фільтри, наприклад, за датою, статусом або іншими атрибутами, щоб знайти конкретні замовлення або здійснити пошук за певними критеріями. Не мало важливо що можна завантажувати звіти, Stripe надає

можливість створювати звіти про замовлення, які містять аналітичну інформацію про операції у системі.

Отже, було детально розглянути процес перегляду історії замовлень у вебзастосунку StoriedFood. А саме, у декількох джерелах, де можна отримати інформацію про замовлення: база даних, адміністративна панель та платіжна система Stripe.

## 5.2 Інструкція користувача

У цьому розділі буде розглянуті основні кроки і дії, які потрібно виконати для ефективного використання нашого вебзастосунку. Буде написана інструкція для ознайомлення користувачів з основними функціями додатку, включаючи створення замовлень, взаємодію з ботом, оплата через платіжну систему та інше.

Зайшовши у бота є варіанти надіслати команди /help та /start. При відправці першої команди, бот привітається та попросить надіслати команди /start для повноцінної роботи додатку. Після того як команда буде надіслана, користувачу приходить повідомлення у якому він дає згоду на відправлення інформації про IP адресу, приклад зображено на рисунку 5.10. Слід зазначити, що таке повідомлення користувач отримує тільки перший раз, при відкритті застосунка.



Рисунок 5.10 – Повідомлення при першому запуску застосунка

Після повідомлення у боті з'являється кнопка, при натисканні на яку, користувач потрапляє у вебзастосунок, який зображено на рисунку 5.11.

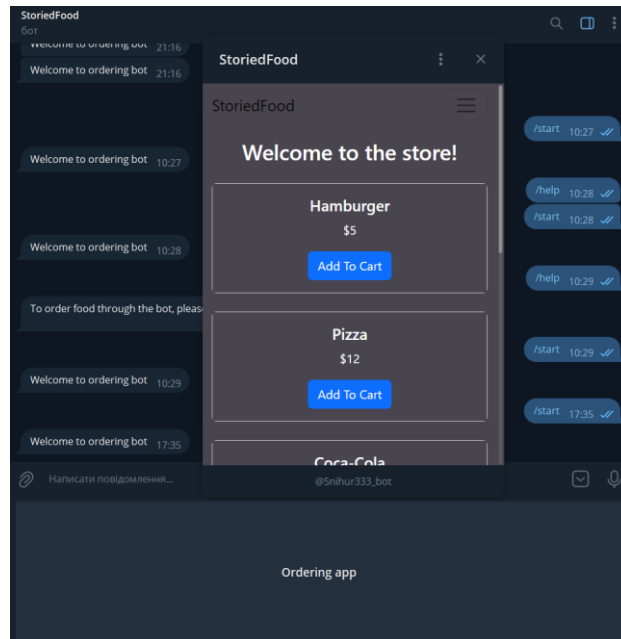


Рисунок 5.11 – Демонстрація вебзастосунку у боті з ПК

На рисунку 5.12 продемонстровано інтерфейс застосунку через мобільний девайс.

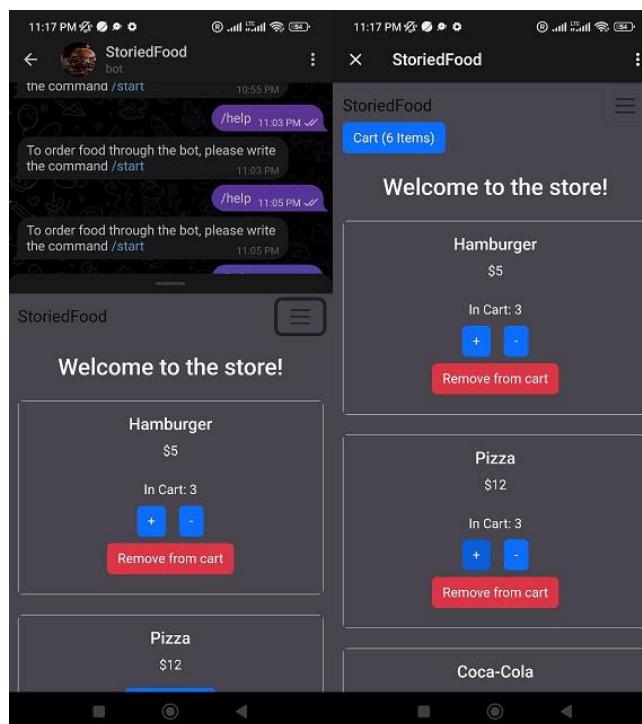


Рисунок 5.12 – Демонстрація роботи застосунку зі смартфона

Зм.	Лист	№ докум.	Підпис	Дата

При роботі з додатком у Telegram доступні усі функції, що і у звичайному веб-сайті. Після вибору конкретних позицій, в меню клієнт може безпосередньо перейти до оплати, яка продемонстрована на рисунку 5.13.

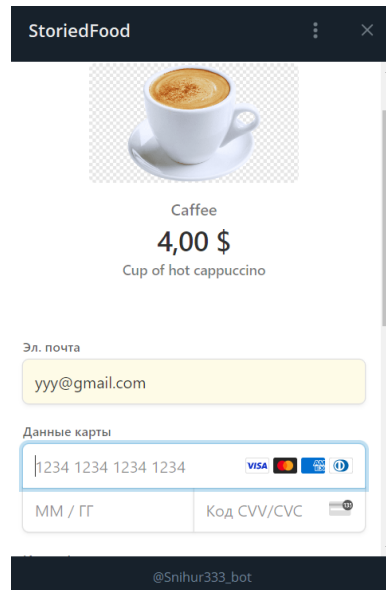


Рисунок 5.13 – Механізм оплати у застосунку

Після успішної оплати, клієнт перенаправляє на сторінку успіху, яка зображена на рисунку 5.14. Якщо у процесі оплати буде якась помилка, то клієнту висвітиться відповідна сторінка з повідомленням про невдалу оплату. Детальна робота системи представлена на кресленику ІА94.180БАК.004 Д1.

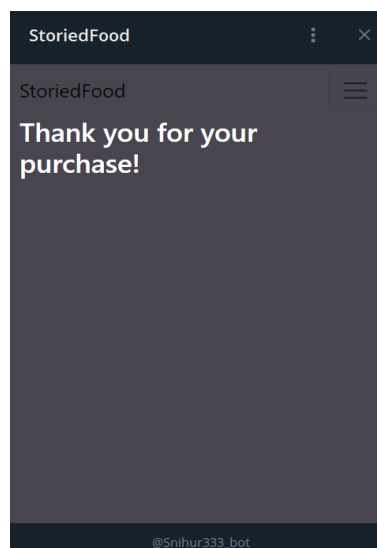


Рисунок 5.14 – Повідомлення після вдалої оплати

## Висновок до розділу

В даному розділі дипломного проєкту наведені керівництва, як для адміністратора так і для користувача. В першому підрозділі, «Інструкція адміністратора», було детально розглянуто процес додавання нових позицій, перегляду історії через базу даних, адміністративну панель та інтеграцію з платіжною системою Stripe. Даний розділ надає адміністраторам всю необхідну інформацію для успішної роботи з чат-ботом та підтримки його функціональності.

У другому підрозділі, «Інструкція користувача» розписано, як користувачі можуть використовувати чат-бот застосунку StoriedFood. В інструкції детально вказано процеси відкриття, додавання та оплати товарів через чат-бот. Крім того, також представлено роботу з чат-ботом як з персонального комп'ютера, так і зі смартфона, що робить платформу StoriedFood зручною і доступною для широкого кола користувачів. Детальна інструкція для адміністратора та користувача зображена на кресленику ІА94.180БАК.004 Д2.

					ІА94.180БАК.004 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		66

## ВИСНОВКИ

Об'єктом розробки даного дипломного проєкту є вебзастосунок на базі Telegram, який призначений для швидкого та зручного замовлення їжі онлайн. Також було поставлено завдання під час виконання даного проєкту отримати практичні навички у розробці вебзастосунків використовуючи новітні технології розробки.

Метою проєкту є впровадження нових рішень у сфері проєктування вебзастосунків для продажу товарів. Поставлену мету було успішно досягнуто, було спроектовано та розроблено вебзастосунок, який вирішує усі поставлені задачі.

Проведений аналіз та вибір необхідних технологій дозволив побудувати ефективну архітектуру додатку. В результаті цього розроблений вебзастосунок вносить значний вклад у сферу розробки веб-ресурсів для онлайн-продажів, вдосконалюючи їх та пропонуючи нові можливості.

В процесі реалізації проєкту було вирішено складні завдання, пов'язані з розробкою клієнтської та серверної частини, інтеграцією з платіжною системою та забезпеченням зручної взаємодії з користувачами.

Також, додаток має на меті подальший розвиток та вдосконалення, можливі покращення включають розширення функціональності, вдосконалення інтерфейсу користувача, оптимізацію продуктивності, підвищення рівня безпеки та аналіз фідбеку користувачів. Ці заходи сприятимуть поліпшенню досвіду користувачів та забезпеченню більш ефективної роботи системи в цілому.

Підсумовуючи, даний дипломний проєкт успішно реалізував задуману ціль і надав значний внесок у розробку веб-додатків для Telegram. Використання сучасних технологій, відповідно підібраних інструментів та професійний підхід до всіх етапів проєктування та розробки дозволили створити функціональний, стабільний та зручний користувацький продукт.

					ІА94.180БАК.004 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		67

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сайт ресторану contrabanda. URL: <https://contrabanda.kiev.ua/> (дата звернення 24.04.2023)
2. Телеграм бот ресторану UNOPizzaGrill. URL: [https://t.me/UNOPizzaGrill\\_Bot](https://t.me/UNOPizzaGrill_Bot) (дата звернення 25.04.2023).
3. Мобільний додаток ресторану Юджин фуд. URL: <https://youdgin.menu/> (дата звернення 26.04.2023).
4. Інстаграм сторінка ресторану Бургерна Ніша. URL: <https://www.instagram.com/nisha.bistro/> (дата звернення 28.04.2023).
5. WEB. URL: <https://www.interviewbit.com/blog/web-application-architecture> (дата звернення 30.04.2023).
6. Telegram. URL: <https://core.telegram.org/bots/webapps> (дата звернення 02.05.2023).
7. Бібліотека Telegraf. URL: <https://github.com/telegraf/telegraf> (дата звернення 02.05.2023).
8. Бібліотека React. URL: <https://react.dev/> (дата звернення 05.05.2023)
9. JSX JavaScript XML. URL: <https://uk.legacyjs.org/docs/introducing-jsx.html> (дата звернення 09.05.2023)
10. Node.js. URL: <https://nodejs.org/en/about> (дата звернення 09.05.2023)
11. Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App (дата звернення 12.05.2023)
12. Express.js. URL: <https://www.codecademy.com/article/what-is-express-js> (дата звернення 15.05.2023)
13. Banker, K., Garrett, D., Bakum, P. and Verch, S., 2016. MongoDB In Action: Covers MongoDB version 3.0. Simon and Schuster. (дата звернення 17.05.2023)
14. MongoDB. URL: <https://medium.com/arc10/usage-of-react-js-node-js-and-mongodb-in-a-project-160248220914> (дата звернення 17.05.2023)
15. Netlify. URL : <https://www.netlify.com/products/> (дата звернення 19.05.2023)

						Лист
						68
Зм.	Лист	№ докум.	Підпис	Дата	ІА94.180БАК.004 ПЗ	

16. Render. URL: <https://render.com/> (дата звернення 19.05.2023)
17. Host on Render. URL: <https://gohugo.io/hosting-and-deployment/hosting-on-render/> (дата звернення 19.05.2023)
18. Stripe. URL: <https://www.donorhut.com/stripe> (дата звернення 23.05.2023)
19. Посилання на репозиторій. URL: [https://github.com/gl-hf/tgbot\\_StoriedFood](https://github.com/gl-hf/tgbot_StoriedFood) (дата звернення 05.06.2023)
20. Посилання на репозиторій. URL: [https://github.com/gl-hf/tgbot\\_food](https://github.com/gl-hf/tgbot_food) (дата звернення 05.06.2023)

ДОДАТОК А  
Код з файлу App.js

```
import './App.css';
import 'bootstrap/dist/css/bootstrap.min.css';
import NavbarComponent from './components/Navbar';
import { Container } from 'react-bootstrap';
import { BrowserRouter, Routes, Route, Navigate } from 'react-router-dom';
import Cancel from './pages/Cancel';
import AdminPage from './pages/AdminPage';
import Store from './pages/Store';
import Success from './pages/Success';
import CartProvider from './CartContext';

function App() {
  return (
    <CartProvider>
      <Container>
        <NavbarComponent></NavbarComponent>
        <BrowserRouter>
          <Routes>
            <Route path='store_admin' element={<AdminPage />} />
            <Route path='store' element={<Store />} />
            <Route path='success' element={<Success />} />
            <Route path='cancel' element={<Cancel />} />
            <Route path='*' element={<Navigate replace to='store' />} />
          </Routes>
        </BrowserRouter>
      </Container>
    </CartProvider>
  );
}
```

```
);  
}
```

```
export default App;
```