

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

**Пояснювальна записка**

до бакалаврської роботи  
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОТРИМАННЯ  
ЗНИЖОК З ВИКОРИСТАННЯМ TELEGRAM API НА МОВІ ПРОГРАМИ  
PYTHON»**

Виконав: студент 4 курсу, групи ПД-41  
спеціальності

121 Інженерія програмного забезпечення  
(шифр і назва спеціальності/спеціалізації)

Станкевич Д.А.

(прізвище та ініціали)

Керівник Коба А.Б.  
(прізвище та ініціали)

Рецензент \_\_\_\_\_  
(прізвище та ініціали)

Київ –2021

# ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

## НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 року

### З А В Д А Н Н Я

#### НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

#### СТАНКЕВИЧУ ДЕНИСУ АНДРІЙОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення для отримання знижок з використанням telegram api на мові програми python»

Керівник роботи: Коба А.Б., старший викладач кафедри ІІЗ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «12» березня 2021 року №65.

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

3.2 Методи збору інформації;

3.3 Науково-технічна література з питань, пов'язаних з програмним забезпеченням щодо збору інформації та її групування.

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити):

4.2 Характеристика чат-ботів та аналіз існуючих додатків;

4.3 Вибір програмних засобів;

4.4 Проектування та реалізація додатку;

4.5 Інструкція користувача;

4.6 Висновки.

5. Перелік демонстраційного матеріалу
  - 5.2 Мета, об'єкт та предмет дослідження;
  - 5.3 Аналоги;
  - 5.4 Технічні завдання;
  - 5.5 Програмні засоби реалізації;
  - 5.6 Метод парсингу даних;
  - 5.7 Застосування чат-боту;
  - 5.8 Ініціалізація чат-боту;
  - 5.9 Висновки.

6. Дата видачі завдання 19.04.2021

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04-21.04	Виконано
2	Вимоги до системи	22.04-23.04	Виконано
3	Розробка алгоритму вилучення інформації	24.04-25.04	Виконано
4	Розробка алгоритму групування інформації	26.04-28.04	Виконано
5	Концепція та архітектура програмного забезпечення	29.04-30.05	Виконано
6	Вступ, висновки, реферат	01.05-08.05	Виконано
7	Розробка демонстраційних матеріалів	09.05	Виконано
8	Попередній захист роботи	11.05	Виконано
9	Здача роботи	01.06	Виконано

Студент \_\_\_\_\_

( підпис )

Станкевич Д.А.

(прізвище та ініціали)

Керівник роботи \_\_\_\_\_

( підпис )

Коба А.Б.

(прізвище та ініціали)





## РЕФЕРАТ

Текстова частина бакалаврської роботи 43с., 28 рис., 3 табл., 22 джерела.

*Об'єкт дослідження* – додаток для отримання знижок.

*Предмет дослідження* – додаток для пошуку та надання знижок на базі Telegram bot API.

*Мета роботи* – надання нових каналів отримання актуальних знижок з використанням методу парсингу даних.

*Методи дослідження* – методи пошуку, вилучення та групування інформації. В даній роботі було проаналізовано існуючі сервіси для отримання знижок, такі як «GoToShop», «Skidochnik», «UkraineSaleClub».

Загальним недоліком цих сервісів є наявність рекламних модулів, застаріла інформація про знижки та навантажений інтерфейс.

Описано залучені бібліотеки та основні принципи розробки. Основою програми є Telegram bot API та розроблена парсинг програма для вилучення даних та їх групування, в якості мови програмування було обрано мову Python, як середу розробки VSCode.

Отже, розроблено та описано додаток, задачею якого є швидке надання актуальних знижок на послуги чи товари.

*Галузь використання* – додаток може бути використано у будь-якій галузі життєдіяльності та не потребує професійних навичок для його використання.

ОТРИМАННЯ ЗНИЖОК, ДОДАТОК TELEGRAM, TELEGRAM BOT API, BOTFATHER, TELEBOT API, WEB PARSING, ЧАТБОТ, ПАРСИНГ

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....</b>	<b>9</b>
<b>ВСТУП.....</b>	<b>10</b>
<b>РОЗДІЛ 1. ХАРАКТЕРИСТИКА ЧАТ-БОТІВ ТА АНАЛІЗ ІСНУЮЧИХ ДОДАТКІВ.....</b>	<b>12</b>
1.1. Вступ.....	12
1.2. Загальні положення про штучний інтелект .....	12
1.3. Історія створення чат-ботів та їх характеристика.....	13
1.4. Принцип дії чат-бота.....	14
1.5. Види чат-ботів .....	14
1.6. Аналіз існуючих додатків для отримання знижок.....	15
<b>РОЗДІЛ 2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ РОЗРОБКИ.....</b>	<b>19</b>
2.1. Вибір середовища розробки.....	19
2.2. Огляд інтерфейсу та основних компонентів редактору коду VSCode .....	21
2.3. Вибір мови програмування .....	23
2.4. Застосування системи контролю версій Git та GitHub.....	25
<b>РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ДОДАТКУ .....</b>	<b>28</b>
3.1. Вимоги до програмного забезпечення .....	28
3.2. Діаграма діяльності.....	29
3.3. Діаграма варіантів використання .....	30
3.4. Використання додаткових бібліотек .....	31
3.4.1. Установка бібліотек .....	31
3.4.2. Бібліотека Beautiful Soup 4.....	31
3.4.3. Бібліотека Telebot.....	33
3.4.4. Бібліотека Requests.....	33
3.5. Метод збору та групування інформації .....	33
3.5.1. Алгоритм роботи парсеру .....	34

3.6. Огляд та аналіз програмного коду парсеру .....	35
3.7. Ініціалізація чат-бота використовуючи BotFather .....	39
3.8. Огляд та аналіз коду чат-бота .....	42
3.9. Тестування розробленого додатку .....	46
<b>РОЗДІЛ 4. ІНСТРУКЦІЯ КОРИСТУВАЧА .....</b>	<b>48</b>
<b>ВИСНОВКИ .....</b>	<b>50</b>



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – Програмне забезпечення

ШІ – Штучний інтелект

ОС – Операційна система

API – Application programming interface

URL – Universal resource locator

AI – Artificial intelligence

IDE – Integrated development environment

SMS – Short message service

HTML – HyperText markup language

pip – Python package manager

## ВСТУП

*Обґрунтування вибору теми та її актуальність.* Купляючи товари в інтернеті, часто виникає бажання заощадити гроші, тому люди звертаються до спеціалізованих сервісів, які надають знижкові промо-коди на товари чи послуги.

Для прискорення процесу пошуку та отримання знижок, необхідно розробити додаток, який буде шукати знижки в інтернеті та виводити згруповану інформацію про них.

*Ступінь вивчення проблеми.* На сьогоднішній день існує багато сервісів, які пропонують знижки на обрані користувачем категорії товарів чи послуг. Але багато з них, використовують рекламні модулі, які гальмують процес отримання знижок, додатки мають складний інтерфейс та не надають актуальну інформацію про знижки.

Використовуючи рекламні модулі в своїх додатках, веб-сайти та чат-канали роблять процес пошуку знижок менш зручним. Сайти втрачають швидкість користування їх продуктом, використовують більше інтернет-трафіка та деякі рекламні банери прибігають до використання трекінг програм, які спостерігають за діями користувача, збирають інформацію про їх активність та інтереси, та на основі отриманої інформації виводять більш релевантну рекламу.

*Об'єкт дослідження.* Сервіс для отримання знижок на товари чи послуги.

*Предмет дослідження.* Чат-бот система, для надання актуальних знижок на товари та послуги.

*Мета роботи.* Підвищення комфорту та безпеки отримання знижок, пришвидшення процесу їх отримання.

*Методика дослідження.* Насамперед, потрібно було проаналізувати існуючі проекти, виявити їх недоліки для кожного продукту окремо та звернути увагу на їх спільні проблеми.

На основі отриманих даних, потрібно розробити комплексний підхід для їх вирішення, обрати вірний набір інструментів, який би дозволив вирішити наявні проблеми застосовуючи мінімум ресурсів користувача.

Враховуючи всі аспекти до розробки програмного продукту, найкращим рішенням є чат-бот, який застосовує процес збору даних з подальшим їх групуванням та виводом.

**Наукова новизна** полягає в створенні чат-бота, який використовує інтуїтивно зрозумілий інтерфейс без рекламних модулів, та надає актуальну інформації про знижки.

**Практична значущість** результатів дослідження полягає у вирішенні важливої практичної задачі – розробки програмного забезпечення для надання знижок на товари чи послуги на мові програмування Python, що дозволяє прискорити процес отримання знижок, зробити процес більш безпечнішим та зменшити витрати інтернет трафіку.

Результати дослідження бакалаврської роботи апробовані на Всеукраїнській науково-технічній конференції: «Застосування програмного забезпечення в інфокомунікаційних технологіях».

## **РОЗДІЛ 1. ХАРАКТЕРИСТИКА ЧАТ-БОТІВ ТА АНАЛІЗ ІСНУЮЧИХ ДОДАТКІВ**

### **1.1. Вступ**

З настанням ери інформаційних технологій та глобальної комп'ютеризації, люди все частіше використовують штучний інтелект для досягнення своєї мети. З приходом штучного інтелекту в життя людей, багато підприємств суттєво зменшили свої витрати та збільшили свої статки, велика кількість компаній частково замінили працю людей на штучний інтелект тим самим пришвидшив своє виробництво.

### **1.2. Загальні положення про штучний інтелект**

Штучний інтелект – це властивість інтелектуальних систем виконувати обробку, здобуття та застосовувати знання та вміння. Часто термін «штучний інтелект» використовується для опису машини, котра імітує «когнитивні» функції, котрі люди пов'язують з людським розумом. ШІ має кілька основних напрямів:

- машинне мислення;
- машинне навчання;
- робототехніка.

Метою штучного інтелекту є:

- планування – під цим поняттям розуміється що робот буде в змозі виконувати прості дії та знаходити послідовність таких дій, для виконання більш важких завдань;
- машинне навчання – це поняття означає що машина зможе самостійно навчитися з введеного в неї масиву даних;

- обробка природної мови – іноді вважають що ця задача є основною для AI, бо ця задача потребує величезного масиву знань. Суть цієї задачі полягає в здатності машини розуміти та читати людську мову;
- машинний зір – суть цієї задачі полягає в наділенні комп'ютера зором який би міг бачити область свого інтересу.

Одним із видів комп'ютерних програм, які наділені штучним інтелектом та застосовуються в багатьох сферах людської діяльності, починаючи від великих підприємств та закінчуючи медичними закладами є чат-боти.

### **1.3. Історія створення чат-ботів та їх характеристика**

В перше термін «чатер-бот» в 1994 році, вжив інформатик Майкл Маулдін, для опису розмовних програм, які дозволять спілкуватися між людиною і чат-ботом, який використовував штучний інтелект.

В 1966 і 1972 роках були створені ELIZA та PARRY, які вважались першими історичними чат-ботами. Чат-бот PARRY був розроблений американським психіатром Кеннетом Колбі. Програма була створена для імітації реакції пацієнта, хворого на параноїдальну шизофренію.

Чат-бот – це комп'ютерна програма, яка була розроблена на основі технологій машинного навчання та нейронних мереж. Розробкою чат-ботів займаються люди, та наділяють їх різними здібностями для вирішення конкретних задач. Зазвичай чат-ботів розробляють для імітування розмови з людиною в інтернеті.

До переваг чат-ботів можна віднести:

- зручність їх використання;
- дешевизну розробки;
- легкість їх застосування та інтегрування в інші системи;
- швидкість роботи;

- не виникає потреби в їх обслуговуванні.

#### 1.4. Принцип дії чат-бота

Зазвичай інтелектуальні системи використовують базу знань, чат-боти не є виключенням. В цій базі містяться можливі запити користувача та відповідні на них відповіді.

До найбільш розповсюджених методів генерації відповіді відносяться:

- реакція бота на ключові слова: бот реагує на введене слово від користувача, якщо воно збігається з заздалегідь прописаним словом в базі даних бота та виконує подальший функціонал прописаний для цієї фрази в базі;
- реакція на фрази: принцип дії схожий на попередній метод, але боти можуть звертати увагу на порядок слів в фразі;
- реакція на контекст: бот аналізує попередньо введені фрази користувачем, та на основі них може згенерувати підходящу відповідь.

Також існують значно ефективніші чат-боти, ніж боти, які використовують заздалегідь готові відповіді на ключові слова та фрази. Такі боти розроблені на базі машинного навчання. Боти даного типу здатні самостійно генерувати відповіді, вони вивчають кожне слово в введеному запиті користувачем та надають відповідь.

#### 1.5. Види чат-ботів

Чат-боти сортуються на декілька груп:

- за навченістю;
- за платформою інтегрування;
- за функціоналом;

До першої групи відносять:

- чат-ботів, які використовують машинне навчання;
- боти які не піддаються навчанню або скриптовані боти, тобто ті боти, що мають заздалегідь прописані ключові фрази та відповіді на них.

До другої групи відносяться:

- які інтегровані в месенджерах;
- мобільних додатках;
- системах управління;
- пошукових системах;
- на веб-сайтах;
- SMS.

В третю групу входять:

- боти-консультанти;
- сервіс-боти;
- боти для збору даних.

## **1.6. Аналіз існуючих додатків для отримання знижок**

Розвиток комп'ютерних технологій та глобальна діджиталізація не минула і сферу продажу, люди все частіше почали робити онлайн покупки в інтернеті, витрачаючи менше часу та зусиль на традиційні походи в магазин. Але роблячи покупки в інтернеті або в магазині у людей часто виникає бажання зекономити якусь частину грошей, тому для таких потреб були розроблені сервіси, які надають список товарів чи послуг на які розповсюджуються акції та знижки. Розглянемо декілька з таких додатків, їх переваги та недоліки.

Веб-сайт «Skidochnik» - сервіс не потребує реєстрування власного акаунту на цій платформі, доступна українська та російська мова. Головна сторінка сайту (рис.

1.1), розподілена на різні категорії та підкатегорії в яких знаходяться товари на які діють знижки.

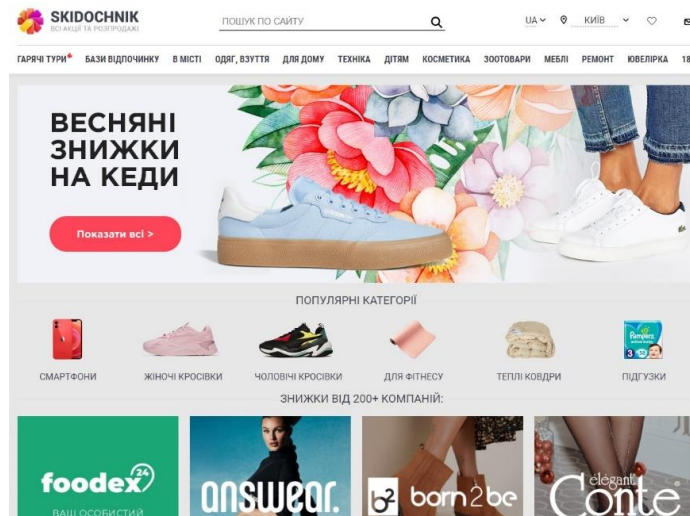


Рисунок 1.1 – Головна сторінка сайту

Сайт «GoToShop» - веб-сайт не потребує реєстрації, наявна українськомовна та російськомовна версія сайту, головна сторінка (рис. 1.2) відображає каталог магазинів в яких є знижки, каталог товарів з поточними та майбутніми знижками і акціями, поля для вибору міста та поле для пошуку за категоріями. Сайт спеціалізуються лише на харчових магазинах.

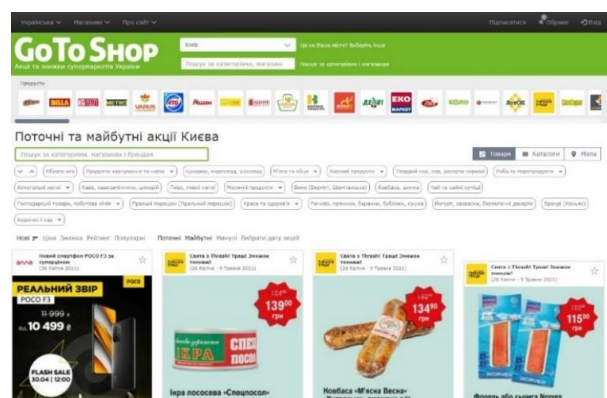


Рисунок 1.2 – Головна сторінка



Чат-канал «UkraineSaleClub» - канал який створений в мобільному і комп'ютерному додатку «Telegram», який надсилає посилання всім учасникам каналу, на акційні товари та товари зі знижками (рис 1.3). Для отримання сповіщень про нові знижки, потрібно натиснути кнопку «Приєднатися до каналу». Мова сповіщень – російська. Переваги та недоліки розглянутих додатків згруповано в табл. 1.1.

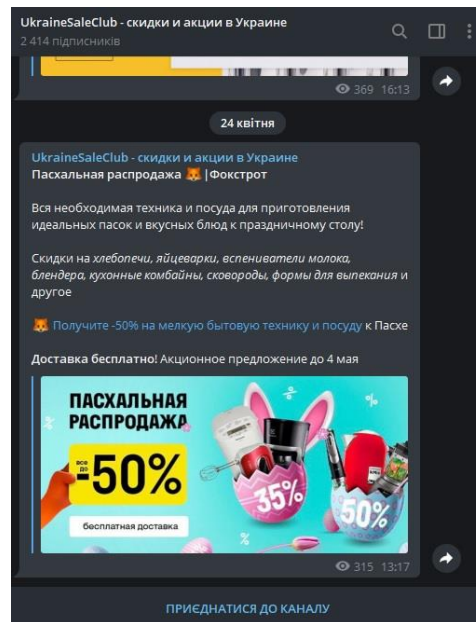


Рисунок 1.3 – Інтерфейс чат-каналу

Таблиця 1.1 – Переваги та недоліки існуючих сервісів

Назва сервісу	Переваги	Недоліки
Skidochnik	<ul style="list-style-type: none"> <li>- сучасний дизайн;</li> <li>- не потребує реєстрації;</li> <li>- зручний інтерфейс;</li> </ul>	<ul style="list-style-type: none"> <li>- погана локалізація сайту на українську мову;</li> <li>- повільна робота сайт;</li> <li>- присутні проблеми з відображенням контенту;</li> </ul>

## Продовження таблиці 1.1 – Переваги та недоліки існуючих сервісів

UkraineSaleClub	<ul style="list-style-type: none"> <li>- зручний та мінімалістичний дизайн;</li> <li>- функція підписки на сповіщення;</li> <li>- не треба реєструватися;</li> </ul>	<ul style="list-style-type: none"> <li>- рідке оновлення актуальної інформації про знижки;</li> <li>- складний та незручний функціонал;</li> <li>- присутні рекламні модулі;</li> </ul>
-----------------	--	---

Проаналізувавши різні за реалізацією додатки, можна прийти до висновку що у сайтів є проблеми зі швидкістю роботи, дизайном, рекламою та навантаженим інтерфейсом, а у сервісів розроблених на базі додатку «Telegram» є проблеми з відображенням актуальної інформації про знижки.

При розробці нового додатку було враховано всі недоліки існуючих програмних засобів. Розроблюваний додаток призначений для надання знижок та купонів на товари і послуги. Додаток може бути застосовано в будь-якій сфері життя та користувачем якого може бути, людина яка не володіє спеціальною освітою.

Додаток має мати зручний інтерфейс, зрозумілий для багатьох користувачів, швидку роботу його модулів, функціонал для отримання актуальної інформації про знижки та акції, відсутність рекламних модулів.

Метою є створення зручних умов для отримання знижок та полегшення їх знаходження в інтернеті.

Спираючись на данні отримані з сайту [gs.statcounter](http://gs.statcounter) за період з березня 2020 року по березень 2021 року, доля користувачів мобільних пристроїв становить – 52,25%, а доля користувачів комп'ютерами – 42,9%. Тому було прийнято рішення розробляти додаток на основі Telegram API для більшого покриття та підтримки усіма популярними девайсами.

## РОЗДІЛ 2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ РОЗРОБКИ

### 2.1. Вибір середовища розробки

Розглянемо декілька середовищ розробки ПЗ, для вибору найбільш підходящого під наші задачі. Для огляду було обрано наступні програми:

- IntelliJ IDEA;
- PyCharm;
- Visual Studio Code (далі VSCode).

IntelliJ IDEA – це інтегроване середовище розробки від компанії JetBrains, яке розповсюджується по умовно безкоштовній формі. Безкоштовна версія представляє урізану версію платної програми, без підтримки додаткових модулів. Програма в першу чергу орієнтована на розробників, які використовують мову програмування Java для своїх проєктів, але середовище розробки не обмежується підтримкою лише мови Java, в її підтримку також входить Python, Scala, PHP, Kotlin та інші мови програмування. Серед недоліків IntelliJ IDEA є навантажений інтерфейс, який може викликати труднощі при першій роботі з цією IDE.

PyCharm – середовище розробки, яке більш спеціалізоване на мові програмування Python. PyCharm має безкоштовну Community-версію з відкритим вихідним кодом, але з обмеженими функціями – в ній відсутні деякі модулі веб-розробки та немає підтримки баз даних. Серед переваг є підтримка Flask-фреймворка, інтеграція з системами контролю версій, UML діаграми класів та багато інших корисних модулів для розробників.

VSCode – редактор вихідного коду, який розповсюджується по безкоштовній формі. Це редактор коду для мультиплатформеної розробки, веб- та хмарних додатків. Редактор підтримує велику кількість мов програмування C++, C#, PHP, Python, JavaScript та інші. Програма розробляється з відкритим вихідним кодом, тому

підтримує велику кількість додаткових плагінів, розроблених користувачами цієї програми. VSCode має вбудований інструмент по роботі з GIT, GitHub, відладник коду. Програма має технологію IntelliSense, яка дописує назви функцій при воді початкових букв, значно прискорюючи процес розробки, також редактор має простий та зручний інтерфейс.

Розглянувши популярні середовища розробки, було обрано редактор VSCode. Дана програма надає всі необхідні можливості для швидкої, зручної та надійної розробки додатку. Головною особливістю є безкоштовна форма розповсюдження редактору та можливість легкого інтегрування додаткових плагінів і бібліотек для розширення функціоналу програми. Переваги та недоліки розглянутих середовищ розробки згруповано в табл. 2.1.

Таблиця 2.1 – Переваги та недоліки засобів розробки ПЗ

Назва засобу	Переваги	Недоліки
IntelliJ IDEA	<ul style="list-style-type: none"> <li>- інструменти для роботи з базами даних та SQL файлами;</li> <li>- інструменти для аналізу якості коду;</li> <li>- розширене форматування.</li> </ul>	навантажений інтерфейс, який може здаватися складним для першого користування.
PyCharm	<ul style="list-style-type: none"> <li>- підтримка Google App Engine, PyPy, PyGTK та інше;</li> <li>- підтримка Flask-фреймворка;</li> <li>- інтеграція з системами контролю версій;</li> <li>- редактор Javascript, HTML/CSS та інші.</li> </ul>	іноді зустрічаються баги при роботі з програмою.

## Продовження таблиці 2.1 – Переваги та недоліки засобів розробки ПЗ

VSCode	<ul style="list-style-type: none"> <li>- безкоштовна форма розповсюдження;</li> <li>- відкритий код;</li> <li>- вбудований інструмент для роботи з системами контролю версій;</li> <li>- технологія IntelliSense;</li> </ul>	іноді виникають проблеми з некоректною роботою нових функцій.
--------	--	---

## 2.2. Огляд інтерфейсу та основних компонентів редактору коду VSCode

Стартове вікно редактору (рис. 2.1) містить наступні модулі:

- модуль «Start» для створення нового файлу або відкриття папки з файлами;
- модуль «Recent», який відображає шлях до знаходження останнього відкритого файлу;
- модуль «Help» з допоміжними посиланнями на сторонні ресурси, які містять додаткові свідчення про програму та роботою з нею;
- та модулі для кастомізації програми (зміни мови програми, кольору теми та інше).

Вікно «Explorer», дозволяє відкрити папку з файлами або клонувати репозиторій з GitHub, також це вікно відображає ієрархію файлів та папок в відкритій папці (рис. 2.2).

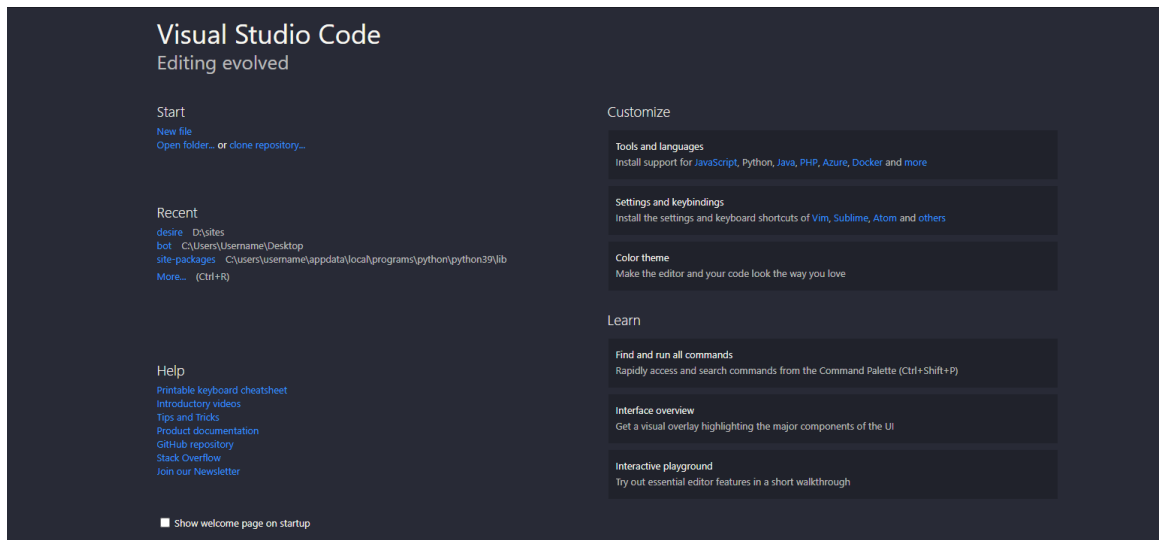


Рисунок 2.1 – Стартове вікно редактору VSCode

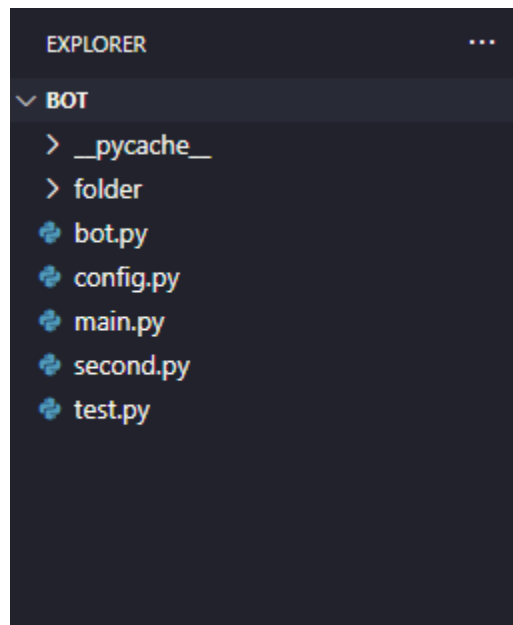


Рисунок 2.2 – Вікно «Explorer» з відображенням ієрархії файлів та папок в папці bot

При відкритті нового файлу чи папки з файлами, відкривається вікно, в якому відображається поле для вводу програмного коду, кнопка для запуску програми та

вікно яке відображає помилки в роботі програми, вихідні дані, термінал та консоль для відладки програми (рис. 2.3).

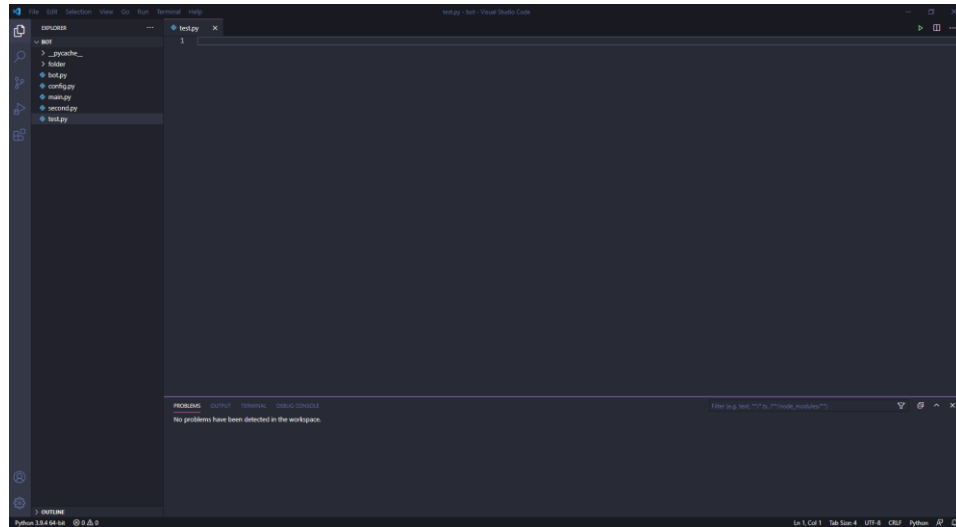


Рисунок 2.3 – Робоче поле вводу коду з вікном інформації про помилки

Робоча область програмування, може поділятися на декілька екрані, для паралельної роботи з двома файлами одночасно. Також, робоче вікно містить строку з вкладками, в яких міститься багато додаткових функцій з налаштування програми, редагування файлів, зміни зовнішнього вигляду програми та її компонентів, інформації про програму, а також меню для різних видів запуску програми.

### 2.3. Вибір мови програмування

Мовою програмування на якій буде написано додаток, було обрано Python. Обрана мова добре підходить для написання чат-ботів за рахунок використання додаткових бібліотек.

Python – це високорівнева мова програмування, орієнтована на підвищення продуктивності розробника. Дана мова програмування, використовує

мінімалістичний синтаксис, за рахунок чого підвищується швидкість написання програм, зменшується кількість написаного коду та підвищується його читаємість (рис. 2.4).

<pre>class HelloWorldApp {   public static void main(String[] args) {     System.out.println("Hello World!"); // Display the string.   } }</pre>	Java
<pre>#include&lt;stdio.h&gt; int main(int argc, char** argv) {   printf("Hello World"); }</pre>	C
<pre>print "Hi there"</pre>	Python

Рисунок 2.4 – Синтаксис python в порівнянні з іншими мовами програмування

Головною перевагою Python в написанні чат-ботів, є спеціально розроблені бібліотеки для їх написання.

Бібліотека в програмуванні – це збірник класів, компонентів чи модулів для вирішення різних задач. За рахунок їх використання, не виникає необхідності кожен раз прописувати стандартні елементи, наприклад, кнопки, запити та інше.

Бібліотека являє собою кимсь вже написані набори коду. Ці бібліотеки програмісти можуть залучати до своїх програм, вставляючи їх в свій програмний код за спеціальним алгоритмом, бібліотеки можна використовувати багаторазово.

Бібліотеки бувають вбудованими та додатковими. У кожній мові програмування є свої бібліотеки. Для прикладу Java використовує JHipster, C++ - Boost, а Python – Simplejson чи Requests та інші. Використання бібліотек є складовою функціонального програмування. За рахунок використання бібліотек зменшується час написання програм та зменшується кількість помилок при їх написанні.



## 2.4. Застосування системи контролю версій Git та GitHub

На сьогодні однією з популярних систем контролю версій (далі СКВ) є Git. Git – це розподілена СКВ або її ще називають децентралізованою, що записує будь-які зміни внесені програмістом у файл або групу файлів. Запис та збереження змін, дозволяє у разі, якщо було виявлено проблеми в роботі програмного забезпечення, повернутись на більш ранню та стабільну версію програми, побачити зміни в файлі, та хто останній вносив зміни в програму. Git дозволяє одночасну роботу декількох програмістів над одним проектом.

Існує три типи СКВ:

- локальна (рис. 2.5);
- централізована (рис. 2.6);
- розподілена (децентралізована) (рис. 2.7).

Локальна СКВ – має базу даних, яка зберігає всі зміни в файл під контролем версій. Всі зміни зберігаються у вигляді набору патчів, де кожен патч маркується часовою міткою. Дана система вирішила проблему появи помилок, при використанні звичайного копіювання файлів в окрему директорію.

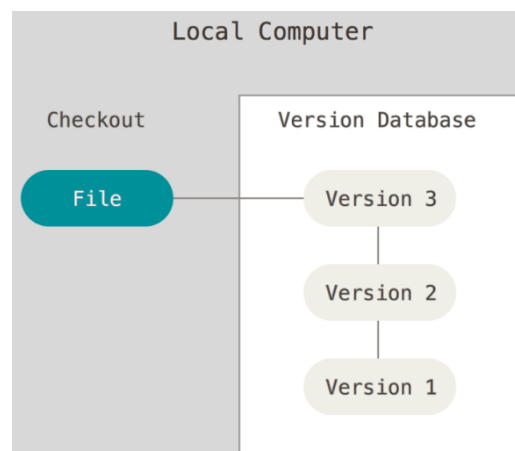


Рисунок 2.5 – Локальна СКВ

Централізована СКВ – така система представляю собою єдиний сервер, на якому містяться всі версії файлів. Доступ до серверу здійснюється через спеціальний клієнтський додаток. Розробка ЦСКВ, дозволила розробникам співпрацювати з іншими програмістами. До недоліків цієї системи можна віднести те, що всі файли зберігаються на одному сервері, і в разі його пошкодження, ви втрачаєте всі файли.

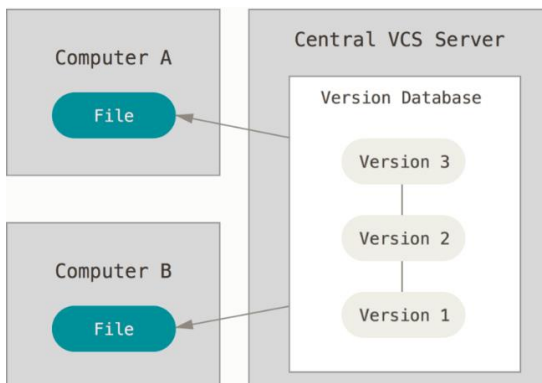


Рисунок 2.6 – Централізована система контролю версій

Розподілена (децентралізована) СКВ – ця система увібрала в себе всі позитивні характеристики попередніх систем та змогла вирішити проблему з втратою даних, в випадках коли сервер перестає працювати. В цій системі користувач отримує повну копію репозиторію, з повною історією його змін. І завдяки цьому, можна перенести всі дані проекту на інший сервер та продовжити роботу.

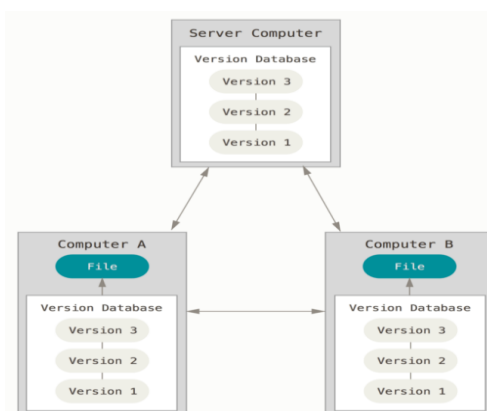


Рисунок 2.7 – Розподілена СКВ

Репозиторій – це каталог, в якому зберігаються файли вашого проекту (рис. 2.8). Він може бути розташований в сховищі GitHub або в локальному сховищі на вашому комп’ютері. Ви можете зберігати на ньому будь-які файли пов’язані з проектом.

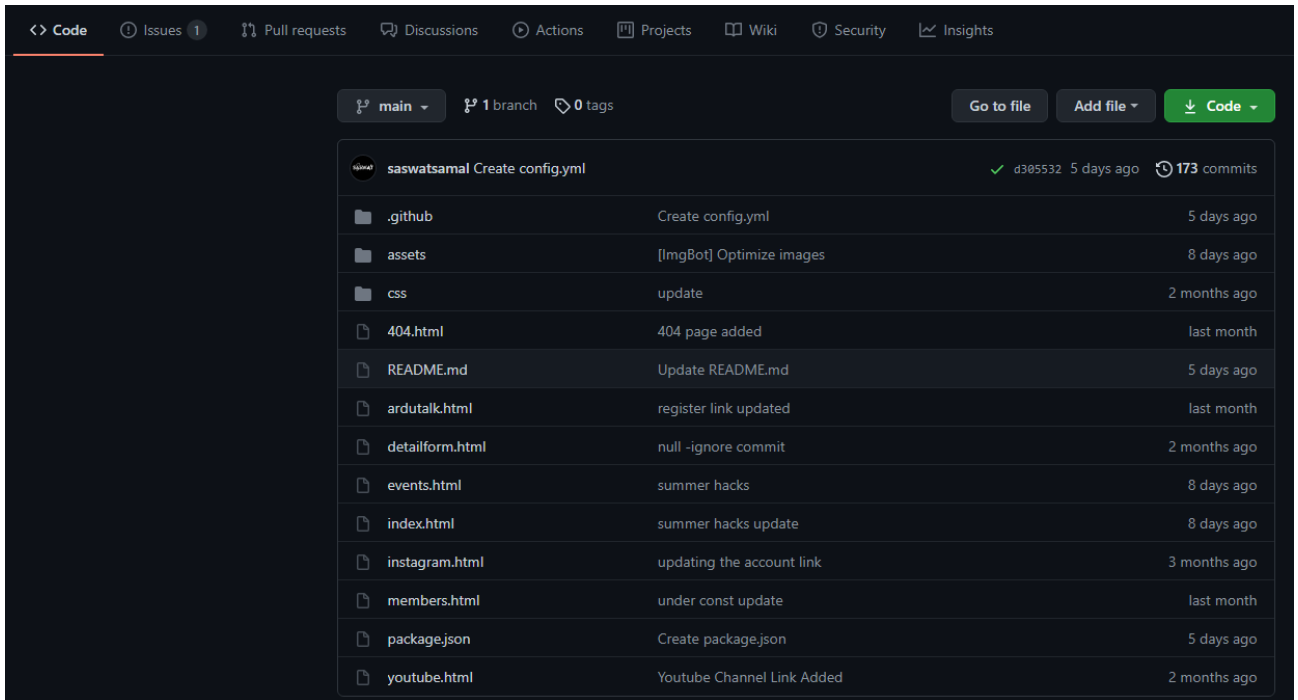


Рисунок 2.8 – Приклад інтерфейсу робочого репозиторію

GitHub – це система керування проектами та версіями коду, котра підтримує всі функції застосовані в Git, додає інші та є його невід’ємною складовою. Дана система забезпечує контроль доступу, а також функції спільної роботи, такі як відстеження помилок, запити функцій, управління задачами, історія комітів та інше. Завдяки відображенню історії комітів, можна дізнатися про зміни, які відбувались з файлом.

## РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ДОДАТКУ

### 3.1. Вимоги до програмного забезпечення

Використовуючи сервіси для отримання знижок, користувачі зустрічаються з проблемами повільної роботи додатку, з рекламними модулями, які заважають досягнути мети, використовують більше інтернет-трафіку та можуть включати в себе трекінг-програми для відстежування дій користувача та збирання інформації про нього. Користувач може потрапити на сайт, який заражений вірусами та поставити під загрозу цілісність свої даних. Також однією з проблем є застаріла інформація про товари на які діють знижки та навантажений інтерфейс.

Завдання, котре повинен вирішувати додаток – це надати можливість користувачеві, швидко, легко, безпечно, надати інформацію про актуальні знижки та акції на товари і послуги.

Важливою задачею є розробка простого та інтуїтивно зрозумілого інтерфейсу для зменшення витрат часу на знаходження актуальної інформації про знижки, та спрощення навігації по додатку. Також підтримку мультиплатформеності, для забезпечення роботи додатку на будь-якому електронному пристрої.

Основними можливостями даного ПЗ є: функціонал для вибору загальної категорії товару, функціонал для вибору конкретної категорії, текстові повідомлення про етап робочого процесу, групування важливої інформації та її вивід.

Розглянемо більш детально можливості ПЗ. Після запуску чат-боту, з'являється текстове повідомлення з подальшою інструкцією використання додатку, також з'являється навігаційне меню вибору категорії, після вибору однієї з запропонованих категорій, з'являється наступне повідомлення з інструкціями та клавіатура з вибором підкатегорії, обравши потрібну підкатегорію, з'являється інформаційне вікно, яке

відображає на якому етапі знаходиться програма, а після виводу інформаційного вікна стану, виводиться згрупована інформація про знижки та посилання на них.

### 3.2. Діаграма діяльності

Діаграма діяльності – відображає динамічні аспекти поведінки системи, дана діаграма стисло показує, як потік управління переходить від однієї діяльності до іншої.

Завдяки графічному зображенню процесів, можна легко зрозуміти як відбувається взаємодія з програмою. Діаграма діяльності розроблюваного додатку зображена на рис. 3.1

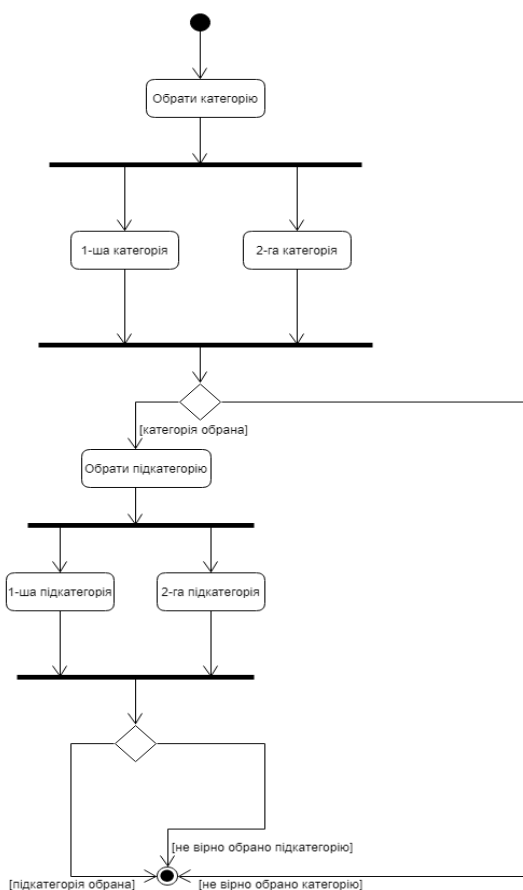


Рисунок 3.1 – Діаграма діяльності додатку

### 3.3. Діаграма варіантів використання

Діаграми варіантів використання, використовуються для відображення сценаріїв використання системи на якій зображено відношення між акторами та варіантами відношення.

Варіанти використання можуть бути пов'язані між собою трьома зв'язками:

- розширенням;
- включенням;
- узагальненням.

Відношення розширення визначає тип відношення, коли один варіант певних умов розширює інший.

Відношення включення визначає тип відношення, коли один варіант в певний момент виконання повністю включає інший.

Відношення узагальнення показує відношення між загальним і частковим. Також дане відношення використовують для поєднання акторів.

Спираючись на розроблені послідовності дій роботи додатку, було створено діаграму варіантів використання. На рис. 3.2 представлені основні дії, які можна виконати в додатку.

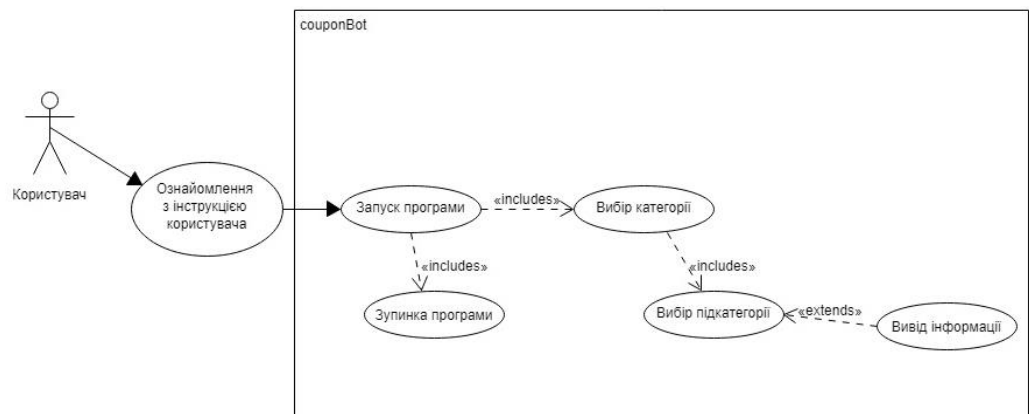


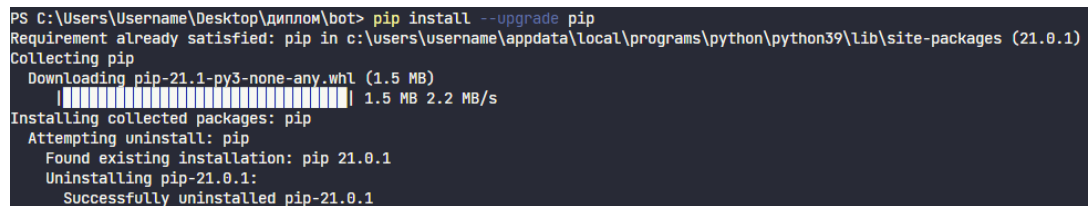
Рисунок 3.2 – UML діаграма варіантів використання

### 3.4. Використання додаткових бібліотек

#### 3.4.1. Установка бібліотек

Для установки додаткових бібліотек в Python, використаємо термінал в VSCode та менеджер пакетів pip. Pip – це система керування пакетами, яка використовується для установки та управління програмними пакетами, написаними на мові програмування Python.

В відкритому терміналі потрібно прописати `pip install` «назва потрібної бібліотеки» та натиснути Enter, після чого відбудеться інсталяція бібліотеки (рис. 3.3).



```
PS C:\Users\Username\Desktop\диплом\bot> pip install --upgrade pip
Requirement already satisfied: pip in c:\users\username\AppData\Local\Programs\Python\Python39\lib\site-packages (21.0.1)
Collecting pip
  Downloading pip-21.1-py3-none-any.whl (1.5 MB)
    |#####| 1.5 MB 2.2 MB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 21.0.1
    Uninstalling pip-21.0.1:
      Successfully uninstalled pip-21.0.1
```

Рисунок 3.3 – Інсталяція бібліотеки

#### 3.4.2. Бібліотека Beautiful Soup 4

Beautiful Soup 4 – це бібліотека Python, яка містить в собі функції та методи для вилучення даних з HTML та XML-файлів. Для більш детального огляду бібліотеки та розуміння її цінності використання, ми повинні познайомитися з HTML та структурою HTML-файлу.

HTML – це мова розмітки гіпертексту, засобами якої здійснюється розмітка веб-сторінок. HTML використовується для того, щоб браузер отримав інформацію про те як завантажена веб-сторінка повинна виглядати. Дана мова розмітки, надає інструменти для побудови структури веб-сайту використовуючи семантичні теги, наприклад, абзаци, заголовки, списки, посилання, форми вводу даних, медіа файли та інше.

Теги – це команди, які несуть в собі інформацію про об’єкт та поведінку об’єкту, який був відмічений тегом. Для прикладу, тег `<img>` надає браузеру інформацію, що цей тег містить в собі якесь зображення, а тег `<a>` повідомляє, що текст відмічений цим тегом – посилання. У деяких тегів є обов’язкові атрибути, без яких тег буде некоректно відображатися, як наприклад, тег `<img>` повинен містити в собі обов’язковий атрибут `src`, в який прописується шлях до потрібного зображення.

Загальна структура документу HTML (рис. 3.4) складається з трьох частин:

1. Декларація про тип документу (Doctype).
2. Голова документу (head).
3. Тіло документу (body).

В першу частину записується тип документу, для того, щоб браузер розумів, як потрібно інтерпретувати дану веб-сторінку, оскільки існує декілька версій HTML.

Друга частина призначена для зберігання інших елементів, ціль яких – допомогти браузеру в роботі з даними. Також в середині тегу `<head>` знаходяться метатеги, які використовуються для зберігання інформації призначеної для браузерів та пошукових систем. Пошукові системи звертаються до метотегів для отримання опису сайту, ключових слів та інших даних.

В третій частині міститься зміст веб-сторінки, який відображається в браузері.

```
1  <!DOCTYPE html>
2  <head>
3    <title>Назва сторінки</title>
4  </head>
5  <body>
6    <h1>Заголовок першого рівня</h1>
7  </body>
8  </html>
```

Рисунок 3.4 – Загальна структура HTML-файлу



Тому використовуючи бібліотеку Beautiful Soup, можна перетворити HTML розмітку сайту в синтаксичне дерево та пересуватись ним, робити пошук елементів по ньому та вносити в дерево зміни.

### 3.4.3. Бібліотека Telebot

Telebot – це спеціально розроблена Python бібліотека, яка спеціалізується на написанні чат-ботів. Дана бібліотека містить в собі різні модулі керування чат-ботами, які дозволяють налаштувати його під себе. Telebot є дуже ефективним інструментом при написанні ботів, за рахунок доступних в ній модулів, можна змінювати стандартну поведінку об'єктів, яка заздалегідь прописана в месенджері Telegram.

### 3.4.4. Бібліотека Requests

Requests – ця бібліотека дозволяє відправляти HTTP запити в Python.

HTTP запити лежать в основі інтернету. Коли відкривається веб-сторінка, браузер відправляє велику кількість запитів до серверу цієї веб-сторінки, і ваш браузер відображає її.

В складі запиту, клієнт відправляє дані за методом запиту. Клієнтом може виступати браузер чи написаний скрип на Python з використанням бібліотеки requests. Найбільш розповсюдженими методами запиту є GET, POST та PUT. Запити GET зазвичай призначені тільки для читання даних без їх зміни, а запити POST та PUT призначені для зміни даних на сервері.

## 3.5. Метод збору та групування інформації

В якості основного методу збору інформації та її подальшої обробки, було обрано метод парсингу.

Парсинг – це синтаксичний аналіз сайтів, який проводить спеціальна програма або скрипт. Зібрана інформація подається в визначеному виді, по визначених правилам, алгоритмам та виконується на одній із обраних мов програмування.

Об'єктом парсинга може бути інтернет-магазин, довідник, форум та інші інтернет-ресурси.

Парсинг є найкращим методом автоматизації процесу збору та збереження інформації для подальших дій над нею.

### 3.5.1. Алгоритм роботи парсеру

Не залежно від мови програмування на якій написано парсер, алгоритм його дії (рис. 3.5) залишається стандартним та однаковим для всіх.

Першим кроком парсеру є вихід в інтернет та отримання доступу к HTML-коду веб-сторінки.

Другим кроком є проходження програми по коду, як по звичайному тексту, та знаходження відповідностей з тими даними, які записані в її програмному коді. Парсер порівнює знайдені дані та зберігає тільки ті, які відповідають заздалегідь прописаним умовам збору даних.

Останній крок – збереження інформації в зручному для вас форматі даних. Це може бути TXT, SQL, XML, HTML та інші формати документів.

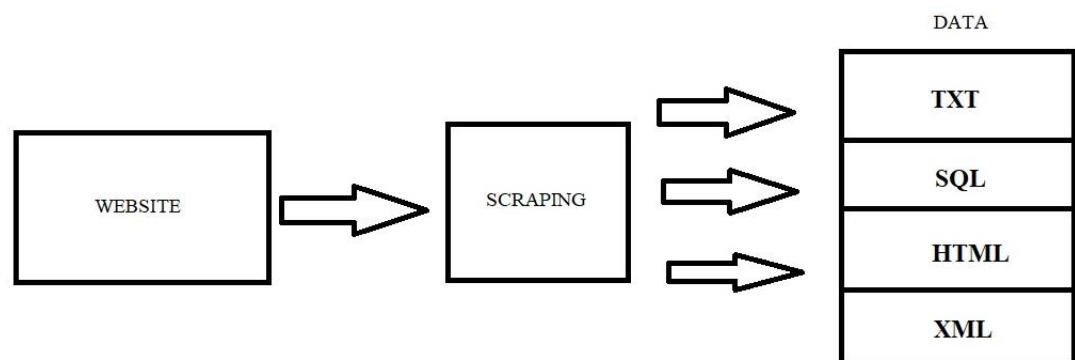


Рисунок 3.5 – Алгоритм роботи парсеру

Таким чином, використовуючи метод парсингу сайтів, можна швидко та ефективно зібрати масив інформації про наявні знижки в інтернеті, згрупувати їх, та передати актуальну інформацію про них в чат-бот і вивести користувачу.

### 3.6. Огляд та аналіз програмного коду парсеру

Для написання парсеру було підключено додаткові бібліотеки `requests` та `beautifulsoup` за допомогою наступних команд:

- `import requests`
- `from bs4 import BeautifulSoup as BS`

В змінну `promotions_url`, `companies_url` та `food_url` передаються URL адреси веб-сторінок на яких парсер буде шукати потрібну інформацію.

URL адреса – це форма унікальної адреси конкретної веб-сторінки в інтернеті. Вона може містити посилання на веб-сторінку, документ або зображення. URL адреса являє собою посилання на сервер, який являється сховищем для шуканого ресурсу.

URL адреса має визначену структуру (рис. 3.6), котра включає:

- метод доступу к ресурсу, який також має назву сітьовий протокол;
- авторизацію доступу;
- хост – DNS адреса, яка вказана як IP адреса;
- порт;
- трек – визначає інформацію про метод отримання доступу;
- параметр – внутрішні дані ресурсу про файл.

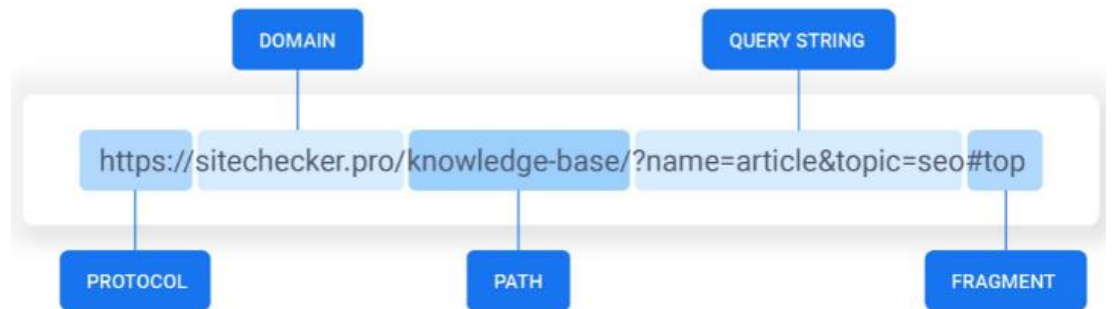


Рисунок 3.6 – Структура URL адреси

Словник під назвою *HEADERS* містить в собі ключі *user-agent* і *accept* та відповідні їм параметри.

Ключ *user-agent* є ідентифікаційним рядком, який клієнт посилає серверу разом з HTTP-запитом, даний рядок зазвичай містить інформацію про назву та версію клієнтського додатку, операційну систему комп'ютера та мову системи.

Ключ *accept* вказує, які типи контенту, виражені як MIME типи, клієнт може прийняти.

MIME типи – це, стандарт, який описує формат документа, файлу чи набору байтів.

Описана частина коду зображена на рис. 3.7

```

1 import requests
2 from bs4 import BeautifulSoup as BS
3
4
5 promotions_url = 'https://pokupon.ua/kyev/zdorove'
6 companies_url = 'https://pokupon.ua/kyev/merchant_pages'
7 food_url = 'https://pokupon.ua/kyev/eda_i_restorany'
8 HEADERS = {
9     'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.114 Safari/537.36',
10    'accept': '*/*'
11 }

```

Рисунок 3.7 – Структура коду з посиланнями на вхідні дані для парсеру

В функціях *promotions*, *companies* та *food* прописано алгоритм за яким парсер буде знаходити потрібні елементи в синтаксичному дереві та вилучати потрібні дані з них.

Дані функції приймають значення двох параметрів, перший параметр – *message* та другий параметр – *url*.

Параметр *message*, приймає значення категорії, обране користувачем в чат-боті та в залежності від обраної категорії, запускає потрібну функцію вилучення інформації.

Параметр *url*, приймає відповідну змінну до назви функції. Для прикладу змінна *url*, в функції *promotions*, приймає значення змінної *promotions\_url*. Во всіх інших функціях, змінні *url* працюють аналогічно до наведеного вище прикладу.

В змінній *session* використано метод із бібліотеки *requests* під назвою *Session()*, даний метод маскує нашу програму під звичайного користувача і сайт бачить нас як звичайну людину.

Змінна *response*, застосовує метод *get()* з бібліотеки *requests* та приймає *url* та *HEADERS* параметри. Даний метод призначений для отримання HTML-коду зі сторінки та подальшого його перетворення в синтаксичне дерево.

Наступна змінна *soup* залучає бібліотеку *BeautifulSoup* для парсингу отриманих даних зі змінної *response*, використовуючи метод парсингу *html.parser*. Таким чином ми отримали синтаксичне дерево елементів, над якими ми можемо провидити різні дії.

*Items* – це змінна, яка використовує метод *find()* з бібліотеки *beautifulsoup* і шукає в дереві елементів, всі *div*-елементи з класом *campaigns-list*.

Змінна *promotions\_list* є пустим списком, який надалі буде заповнюватись знайденими даними через цикл *for*.

Перший цикл *for* виконує пошук елементів, застосовуючи вкладений цикл. Пошук відбувається в кожному знайденому *div*-елементі в змінній *items*.

Вкладений цикл *for* застосовує метод *findAll()*, для пошуку всіх посилань з класом *small*.

Зі знайдених посилань ми отримуємо URL адресу на знижку, застосовуючи метод `get()`, котрий шукає атрибут `href`, який є обов'язковим для HTML тегу `a`.

В змінній `title`, ми шукаємо `div`-елементи з класом `name` та приміняємо метод `.text` для вилучення зі знайденого елемента текстової частини. Таким чином ми отримали назву товару чи послуги на які розповсюджується знижки.

Змінна `discount`, шукає `div`-елемент з класом `discount` за допомогою методу `find()`, в знайденому об'єкті, відбувається додатковий пошук `span`-елемента та вилучення з нього тексту. Виконавши всі дії над елементами, в змінну `discount` записується значення знижки для послуг і товарів.

`Events_str` виконує конкатенацію строк. Тобто склеює строки, використовуючи службовий символ `\n` для переводу строки на нову строку.

В змінну `promotions_list`, яка є порожнім списком, почергово записуються знайденні дані в циклі, тим самим заповнюючи список, для подальшого виведення його на екран користувачу. Інструкція `return` вертає записані в циклі значення в список.

Розглянута частина коду зображена на рис. 3.8

```

14 def promotions(message, url=promotions_url):
15     session = requests.Session()
16     response = session.get(url, headers=HEADERS)
17     soup = BS(response.content, 'html.parser')
18     items = soup.find('div', class_="campaigns-list")
19
20     promotions_list = ''
21
22     for name in items:
23         for name in soup.findAll('a', class_='small'):
24             link = name.get('href')
25             title = name.find('div', class_='name').text
26             discount = name.find('div', class_='discount').find('span').text
27             events_str = title + '\n' + discount + '\n' + link + '\n'
28             promotions_list += events_str
29     return promotions_list

```

Рисунок 3.8 – Цикл для пошуку елементів та вилучення даних з них

Таким чином, застосовуючи описані вище операції, відбувається парсинг обраного сайту, для вилучення з нього актуальної інформації про знижки, і у разі оновлення інформації про наявні знижки, користувач додатком, також отримає оновлені дані. Застосувавши метод парсингу сайту, ми вирішили проблему з відображенням неактуальної інформації про знижки та пришвидшили швидкість пошуку знижок.

З повним програмним кодом парсеру, можна ознайомитись у Додатку А.

### **3.7. Ініціалізація чат-бота використовуючи BotFather**

Для створення чат-бота та подальшого його програмування для виводу отриманих даних з парсеру та налаштування користувацького інтерфейсу, було обрано BotFather.

BothFather – це спеціалізований сервіс для реєстрації, налаштування та управління, створеними вами чат-ботами.

Взаємодія з BothFather відбувається шляхом вводу в текстове поле, спеціально зарезервованих команд, зображених на рис. 3.9. Команди мають свою назву та короткий опис до них. Відкрити вікно для перегляду команд, можна ввівши в текстове символ «/».

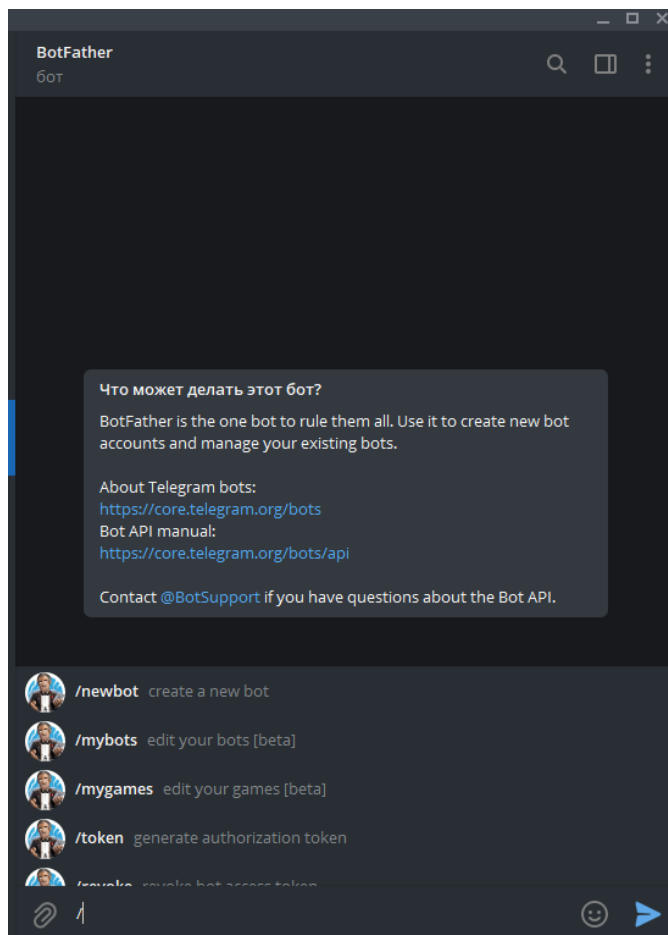


Рисунок 3.9 – Інтерфейс BotFather з частиною зарезервованих команд

За допомогою команди */newbot*, можна створити ного чат-бота, основним критерієм для його створення, є унікальне ім'я, яке не було використано іншими користувачами, BotFather, а також створення додаткового ім'я, яке має обов'язково містити слово *bot* наприкінці додаткового ім'я.

Процес створення тестового боту зображено на рис. 3.10.



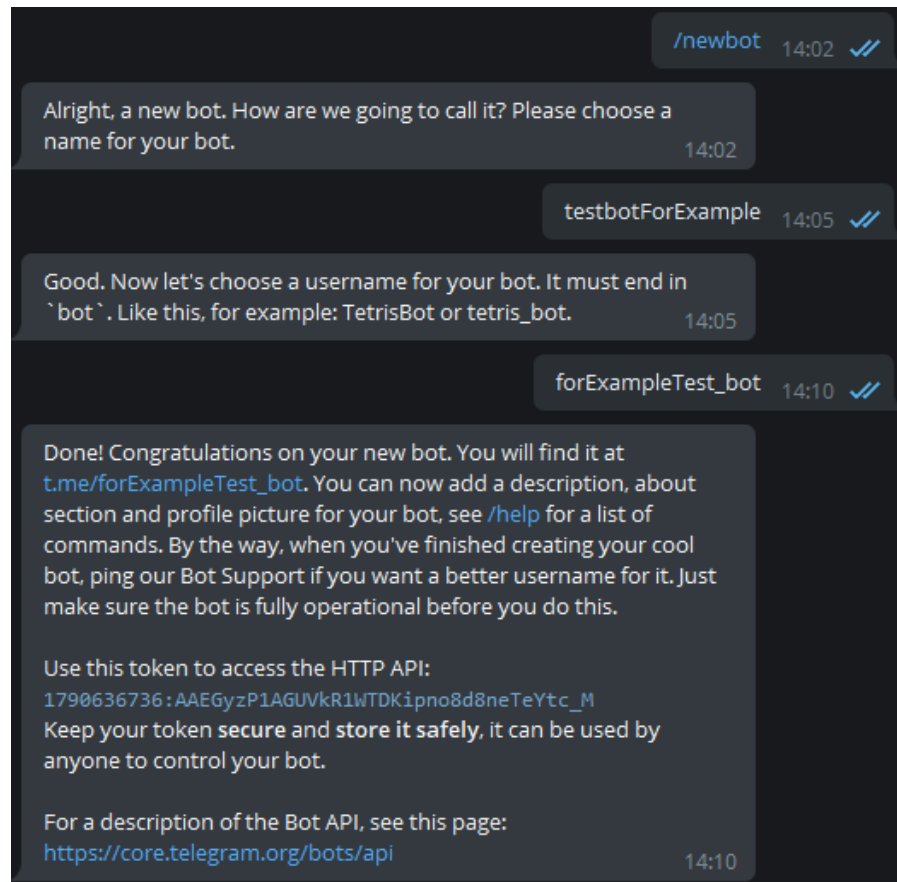


Рисунок 3.10 – Процес створення тестового чат-боту

Після успішного створення чат-бота, йому призначається унікальний токен, який використовується для отримання доступу до функцій бота і подальшого його програмування з додаванням нового функціоналу та редагуванням стандартних налаштувань бота. Також створюється сам інтерфейс чат-бота для взаємодії з його функціоналом та отримання більш розгорнутої інформації про його можливості.

Переглянути всіх створених вами ботів, можна за допомогою команди */mybots* введеної в BotFather, після цього відобразиться список доступних вам ботів.

Обравши один з них, ви можете переглянути токен вашого боту, подивитися налаштування створеного боту, редагувати його опис, додавати нові команди, змінювати зображення бота, ім'я та видалити обраного чат-бота.

Переглянути інтерфейс додаткових функції бота, можна на рис. 3.11.

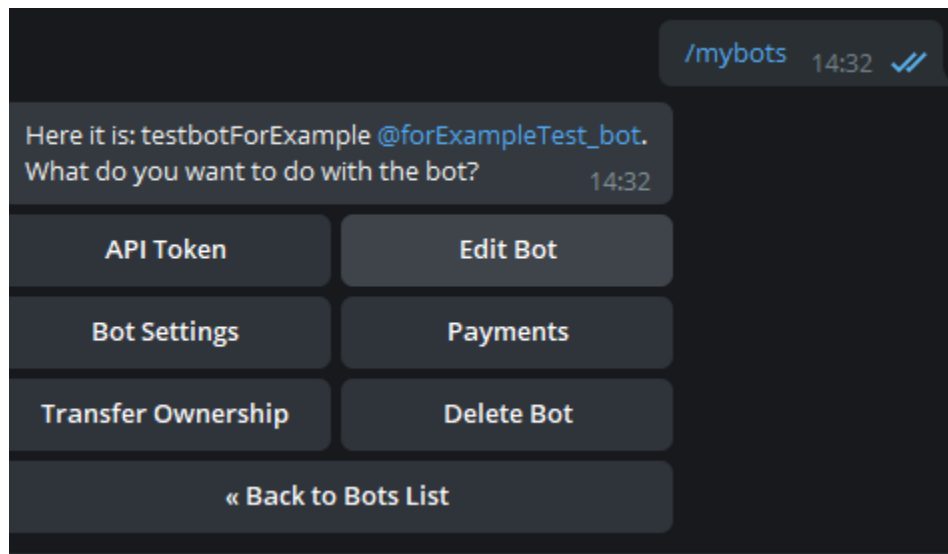


Рисунок 3.11 – Інтерфейс додаткових функцій для управління ботом

BotFather є найкращим рішенням для створення чат-ботів та редагування його функціоналу. Обравши цей сервіс, ми створили чат-бота, який має зручний функціонал, швидку роботу та результат його роботи, однаково відображається на різних електронних пристроях, зробивши розроблюваний додаток доступним для більшої кількості людей.

### 3.8. Огляд та аналіз коду чат-бота

Для початку створення додаткового функціоналу для чат-бота, потрібно імпортувати додаткові бібліотеки та функції з файлу, який займається парсингом даних з сайту за допомогою наступних строк коду:

- *import config;*
- *import telebot;*
- *from main import promotions, companies, food;*
- *from telebot import types.*

Змінна `bot` використовує функцію `telebot()`, яка приймає значення токена з файлу `config.py`. Передавши значення токена, ми отримуємо доступ до нашого чат-боту, і можемо керувати його функціоналом та додавати новий.

Після передачі токена, потрібно створити обробник повідомлень за допомогою команди `@bot.message_handler`, який приймає параметр `commands` в якому міститься назва команди, на яку бот буде реагувати та відпрацьовувати прописаний в боті функціонал. Даний параметр, може містити декілька ключових слів для запуску бота, основною вимогою є те, що вони мають бути написані латинськими літерами. Окрім запуску бота при введенні ключових слів, бот, може виконувати інший функціонал при їх введенні. Наприклад, при введенні слова `help`, чат-бот може виводити додаткову інформацію про його команди та призначення боту.

Описаний вище функціонал зображений на рис. 3.12.

```
1 import config
2 import telebot
3 from main import promotions, companies, food
4 from telebot import types
5
6 bot = telebot.TeleBot(config.TELEGRAM_TOKEN)
7
8
9 @bot.message_handler(commands=['start'])
```

Рисунок 3.12 – Підключення додаткових бібліотек та функцій для керування ботом.

Функція `send_welcome`, відповідає за відображення навігаційної клавіатури на привітальному екрані та виводу привітального тексту, після запуску чат-бота.

Змінна `markup` задає тип кнопок та приймає два параметри, `resize_keyboard` та `row_width`.

Перший параметр, який відповідає за автоматичну зміну розміру клавіатури, у підходящих для цього випадках. Наприклад, зменшити розмір клавіатуру, якщо використовуються тільки два ряди кнопок. Даний параметр, робить інтерфейс більш компактним та зручним для користування.

Другий параметр визначає, яка кількість кнопок вміститься в кожному рядку, перш ніж продовжити наступний рядок. За замовчування даний параметр має значення в три рядки.

Змінні *itembtn1* та *itembtn2* використовують функцію *KeyboardButton()*, в яку передається назва для кнопок.

Використовуючи функцію *add()* для змінної *markup*, ми додаємо обрані кнопки в масив.

Змінна *msg*, виконує відправку привітального тексту після запуску бота, використовуючи функцію *send\_message()*, в яку передається декілька параметрів:

- Параметр *message.chat.id*, передає id адресу чата, в який потрібно відправити привітальний текст;
- Другий, передає будь-який текст, який потрібно відобразити після запуску програми;
- Наступний параметр, відключає звук сповіщення про нове повідомлення. Цей параметр є опціональним та застосований для підвищення комфорту користування додатком;
- Останній параметр, передає масив з кнопок для того, щоб користувач міг обрати потрібну йому категорію товарі чи послуг.

Наступною функцією програми є *register\_next\_step\_handler()*, дана функція відповідає за зчитування дій користувача. Тобто, коли користувач виконав перший крок, ця функція переправляє його до наступного кроку.

Описаний програмний код, зображено на рис. 3.13

```

10 def send_welcome(message):
11     markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
12     itembtn1 = types.KeyboardButton('👉 Акції')
13     itembtn2 = types.KeyboardButton('👉 Знижки компаній')
14
15     markup.add(itembtn1, itembtn2)
16     msg = bot.send_message(
17         message.chat.id, 'Привіт, обери потрібну категорію', disable_notification=True, reply_markup=markup)
18     bot.register_next_step_handler(msg, selectCategories)

```

Рисунок 3.13 – Функція відправки привітального тексту та відображення клавіатури.

Функція `selectCategories`, відповідає за функціонал вибору категорії. Дана функція використовує конструкцію розгалуження `if`.

В функцію `if()`, ми прописуємо умову, при істинності якої, будуть виконуватись подальші дії. В даній конструкції ми перевіряємо, якщо була натиснута кнопка «Акції», то ми передаємо змінну `message`, у функцію `select`, яка відповідає за вікно вибору підкатегорії та діє аналогічно до функції `send_welcome`.

Якщо умова функції `if()`, не є істиною, тоді ми переходимо до наступної функції `elif()`, в якій ми також прописуємо умову її виконання.

У разі, якщо користувач натиснув на кнопку «Знижки компаній», тоді змінна `markup`, застосовує функцію `ReplyKeyboardRemove()`, яка убирає попередню клавіатуру, яка є відповідною за вибір категорії. Наступним кроком, в чат відправляється текстове повідомлення «Шукаю знижки», після цього, функція `send_message()`, відправляє змінну `message`, в функцію `companies()`, яка парсить потрібну нам веб-сторінку, та виводить отримані дані користувачу. Функція `send_message()`, також приймає додатковий, але не обов'язковий параметр `disable_web_page_preview`, який відповідає за відображення прев'ю веб-сторінок, в даному випадку, прев'ю сторінок не будуть відображатися, для зручності перегляду списку знижок.

Розглянутий вище програмний код, зображено на рис. 3.14.

```

21 def selectCategories(message):
22     if (message.text == '📌 Акції'):
23         select(message)
24     elif (message.text == '📌 Знижки компаній'):
25         markup = types.ReplyKeyboardRemove()
26         bot.send_message(message.chat.id, '🔍 Шукаю знижки...',
27                           disable_notification=True,)
28         bot.send_message(message.chat.id, companies(
29             message), disable_web_page_preview=True, disable_notification=True, reply_markup=markup)

```

Рисунок 3.14 – Функція вибору категорії

Функція *subCategoriesSelect()*, працює по аналогії з функцією *selectCategories()*, однак змінна *message*, відправляється до відповідної функції, яка відповідає за відповідну їй підкатегорію.

З повним програмним кодом, функціоналу чат-бота, можна ознайомитись у Додатку Б.

### 3.9. Тестування розробленого додатку

Тестування є невід’ємною частиною при розробці програмного забезпечення. Завдяки тестуванню, можна виявити проблеми при роботі з ПЗ, тим самим підвищуючи його якість та комфорт користування додатком. Для тестування коректності роботи функціоналу в розробленому ПЗ, було проведено декілька ключових тест-кейсів.

Тест-кейс – це описаний алгоритм тестування програми, спеціально створений для перевірки визначеного функціоналу в розроблюваному програмному забезпеченні. У табл. 3.1 наведено приклад тест-кейсів додатку.

Таблиця 3.1 – Тест-кейс для тестування функціоналу вибору категорій та підкатегорій

Функціонал вибору категорій та підкатегорій
Передумова: Запустити чат-бот

Продовження таблиці 3.1 – Тест-кейс для тестування функціоналу вибору категорій та підкатегорій

Крок тесту	Дані тесту	Очікуваний результат
Введення назви відсутньої категорії або підкатегорії	<ul style="list-style-type: none"> <li>- Ввести в текстове поле назву відсутньої категорії/підкатегорії;</li> <li>- Натиснути клавішу Enter для вводу даних.</li> </ul>	Відображення попередження про відсутність категорії/підкатегорії
Вибір одної з запропонованих категорій	Вибрати одну із запропонованих категорій, шляхом натискання на графічну клавіатуру або ввести назву категорії в текстове поле	Відображення знайдених даних про знижки або перехід до кроку вибору підкатегорії, при її наявності
Вибір одної з запропонованих підкатегорій	Вибрати одну з запропонованих категорій, шляхом натискання на графічну клавіатуру або ввести назву категорії в текстове поле	Відображення знайдених даних про знижки

## РОЗДІЛ 4. ІНСТРУКЦІЯ КОРИСТУВАЧА

Для початку роботи з програмним забезпеченням потрібно завантажити месенджер Telegram. Даний месенджер доступний на всіх електронних пристроях. Для завантаження месенджера на телефонах під управлінням операційної системи Android, потрібно використати Play Market, для телефонів на базі ОС iOS, використати Apple Store.

Після завантаження Telegram, потрібно зареєструвати аккаунт, якщо його нема. Після створення аккаунту, потрібно у вікно пошуку ввести назву чат-бота «couponBot», та натиснути на нього лівою кнопкою миші (далі ЛКМ) для його вибору зі списку. Результат пошуку зображено на рис 4.1.

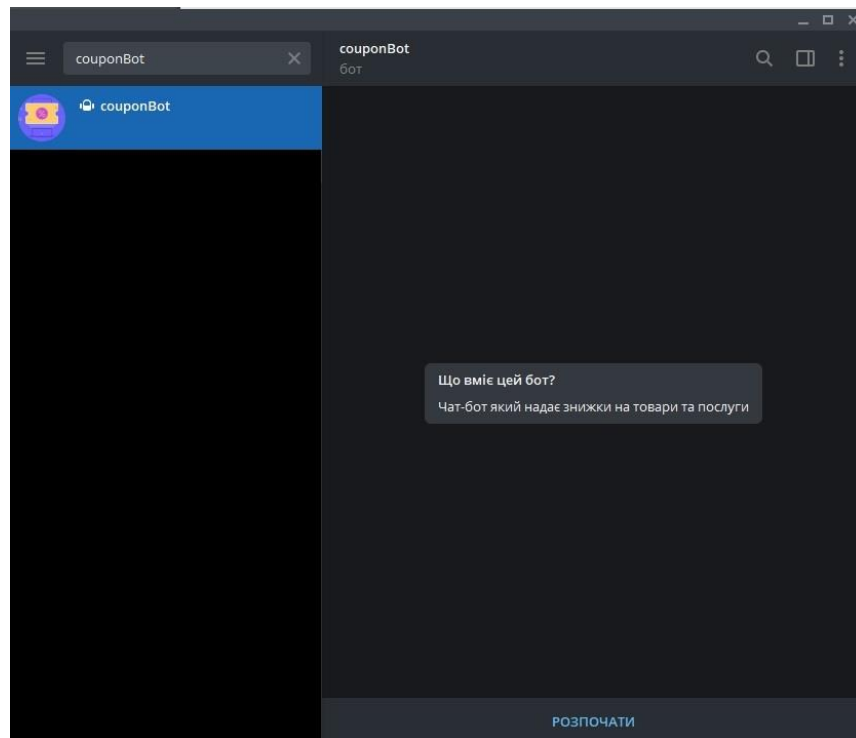


Рисунок 4.1 – Результат пошуку додатку

У відкритому вікні потрібно натиснути кнопку «Розпочати» для запуску боту.



Після запуску боту, в чаті з'явиться команда запуску «/start» та привітальне повідомлення з описом наступного кроку, також з'явиться клавіатура та текстове поле, в яке можна ввести назву потрібної категорії (рис. 4.2).

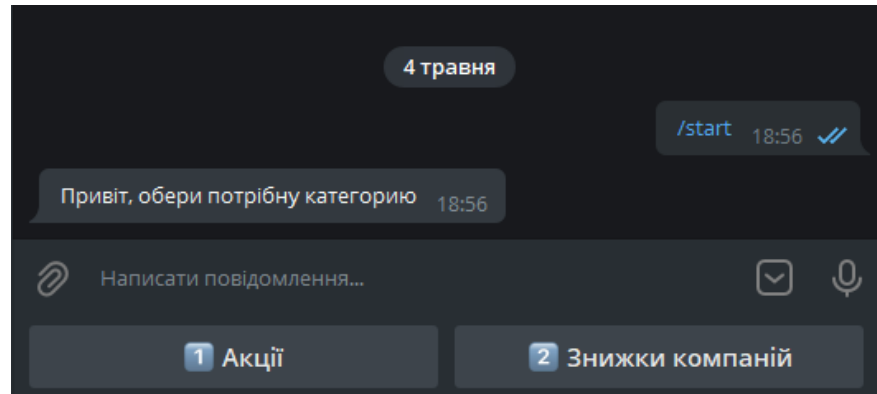


Рисунок 4.2 – Привітальне вікно з вибором категорії

Після натискання ЛКМ по одній з категорій, відбудеться перехід до вибору підкатегорії, де вам буде запропоновано декілька варіантів, після вибору якого, відбудеться вивід даних про знижки для користувача (рис 4.3). Даний перехід відбудеться, якщо категорія містить в собі підкатегорії, якщо ні, то відбудеться автоматичний пошук інформації про знижки та вивід інформації користувачу.

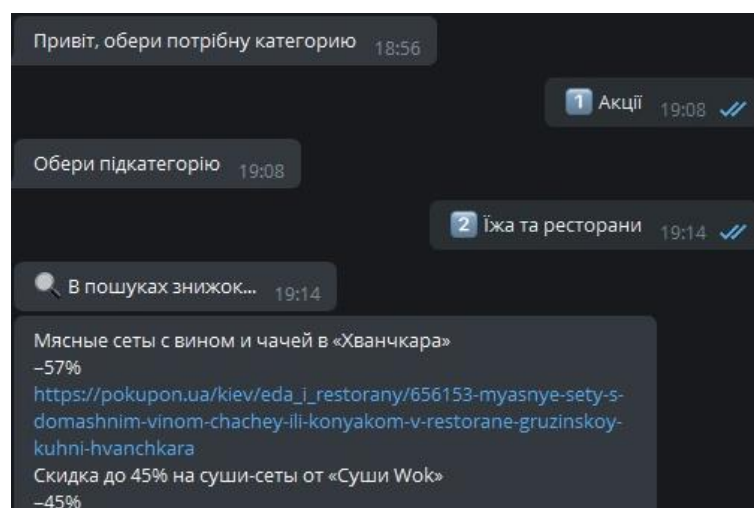


Рисунок 4.3 – Вибір підкатегорії та вивід даних про знижки

## ВИСНОВКИ

У даній роботі було розроблено програмне забезпечення для отримання знижок на товари та послуги, використовуючи Telegram Bot API та застосовуючи алгоритм парсингу даних.

В першому розділі було розглянуто характеристику чат-ботів, принцип їх дії та види, а також історію їх створення. Було розглянуто та проаналізовано існуючі додатки для отримання знижок. В ході аналізу було знайдено їх проблеми, в загальному, проблеми були пов'язані з наданням користувачу неактуальної інформації, навантажений інтерфейс та використання реклами в своїх додатках.

В другому розділі, в ході аналізу існуючих середовищ розробки, було обрано найбільш підходяще програмне забезпечення, яке в повній мірі підходило для вирішення поставлених задач. Також було розглянуто інтерфейс обраного редактору коду. В якості мови програмування було обрано Python, завдяки наявності в ній спеціалізованих додаткових бібліотек для створення чат-ботів, простого синтаксису та легкої читаємості коду. Було розглянуто принципи дії та варіанти використання системи контролю версій Git та GitHub.

В третьому розділі було визначені вимоги для розроблюваного додатку, розроблено діаграму діяльності та варіантів використання. Було більш детально розглянуто використані бібліотеки для написання чат-ботів. Також було розглянуто метод парсингу даних для збору та групування інформації, принцип його дії та варіанти використання. Був проведений детальний огляд програмного коду парсеру та чат-бота, з описом використаних функцій та методів при написанні програми. Також було описано алгоритм ініціалізацію чат-бота, використовуючи спеціальний сервіс для їх створення та керування BotFather. Було проведено тестування основних функцій чат-бота.

В останньому розділі була створена інструкція з користування розробленим додатком.

Отже, при виконаній даної роботи, було проведено всі ключові етапи при розробці програмного забезпечення. Починаючи від аналізу існуючих додатків та з'ясування їх проблем до вибору підходящих засобів та методів для їх вирішення, розробки додатку та тестування.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке чат-боти та кому вони потрібні [Електронний ресурс] – Режим доступу: <https://creativesmm.com.ua/shho-take-chatbot-ta-komu-vonu-potribni/>
2. Chatbots.org, Parry [Електронний ресурс] – Режим доступу: <https://www.chatbots.org/chatbot/parry/>
3. Глибовець М. М., Олецький О.В. «Штучний інтелект» — Київ : «Києво-Могилянська академія», 2002. — 364 с.
4. Документація мови програмування Python [Електронний ресурс] – Режим доступу: <https://docs.python.org/3/index.html>
5. Visual Studio Code on GitHub [Електронний ресурс] – Режим доступу: <https://github.com/Microsoft/vscode>
6. Офіційна сторінка Visual Studio Code [Електронний ресурс] – Режим доступу: <https://code.visualstudio.com/>
7. Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon and John Riedl, 1997, “GroupLens: applying collaborative filtering to Usenet news,” Communications of the ACM, 40(3), pp. 77-87.
8. PyCharm Homepage [Електронний ресурс] – Режим доступу: <https://www.jetbrains.com/pycharm/>
9. PyCharm Community Edition [Електронний ресурс] – Режим доступу: <https://github.com/JetBrains/intellij-community/tree/master/python>
10. Scott Chacon, Ben Straub «ProGit», 2021 – 10-13 с.
11. Яковенко, А. (2018). «Основи програмування. Python. Частина 1». Київ: КПІ ім. Ігоря Сікорського. с. 195.
12. Guido van Rossum, Python Reference Manual, release 2.4.4, 18 October 2006.
13. Офіційна веб-сторінка IntelliJ IDEA [Електронний ресурс] – Режим доступу: <https://www.jetbrains.com/idea/>

14. Eduo Shaun, Difference Between Git and GitHub 2017 [Електронний ресурс] – Режим доступу: <https://medium.com/@eduoshaun/difference-between-git-and-github-807f1a57d438>
15. James Rumbaugh, Ivar Jacobson, Grady Booch (1999). «The unified modeling language reference manual».
16. Електронний HTML і CSS довідник українською мовою [Електронний ресурс] – Режим доступу: <https://html-css.co.ua/>
17. Machine Learning in Python [Електронний ресурс] – Режим доступу: <http://scikit-learn.org/stable/>
18. pyTelegramBotAPI github [Електронний ресурс] – Режим доступу: <https://github.com/eternnoir/pyTelegramBotAPI>
19. BeautifulSoup Documentation [Електронний ресурс] – Режим доступу: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
20. Web technology for developers [Електронний ресурс] – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web>
21. Chapman, Nigel P., LR Parsing: Theory and Practice, Cambridge University Press, 1987.
22. Grune, Dick; Jacobs, Criel J.H., Parsing Techniques - A Practical Guide.

## Додаток А

### Програмний код парсеру

```
import requests

from bs4 import BeautifulSoup as BS

promotions_url = 'https://pokupon.ua/kiev/zdorove'
companies_url = 'https://pokupon.ua/kiev/merchant_pages'
food_url = 'https://pokupon.ua/kiev/eda_i_restorany'
HEADERS = {
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.114 Safari/537.36',
    'accept': '*/*'
}

def promotions(message, url=promotions_url):
    session = requests.Session()
    response = session.get(url, headers=HEADERS)
    soup = BS(response.content, 'html.parser')
    items = soup.find('div', class_="campaigns-list")

    promotions_list = ""

    for name in items:
        for name in soup.findAll('a', class_='small'):
            link = name.get('href')
            title = name.find('div', class_='name').text
```

```
discount = name.find('div', class_='discount').find('span').text
events_str = title + '\n' + discount + '\n' + link + '\n'
promotions_list += events_str
return promotions_list
```

```
def companies(message, url=companies_url):
    session = requests.Session()
    response = session.get(url, headers=HEADERS)
    soup = BS(response.content, 'html.parser')
    items = soup.find('div', class_="campaigns-list")

    companies_list = ""

    for name in items:
        for name in soup.findAll('a', class_='small'):
            link = name.get('href')
            title = name.find('div', class_='name').text
            discount = name.find('div', class_='discount').find('span').text
            events_str = title + '\n' + discount + '\n' + link + '\n'
            companies_list += events_str
    return companies_list
```

```
def food(message, url=food_url):
    session = requests.Session()
    response = session.get(url, headers=HEADERS)
    soup = BS(response.content, 'html.parser')
    items = soup.find('div', class_="campaigns-list")

    food_list = ""
```

```
for name in items:
    for name in soup.findAll('a', class_='small'):
        link = name.get('href')
        title = name.find('div', class_='name').text
        discount = name.find('div', class_='discount').find('span').text
        events_str = title + '\n' + discount + '\n' + link + '\n'
        food_list += events_str
return food_list
```



## Додаток Б

### Програмний код чат-боту

```
import config
import telebot
from main import promotions, companies, food
from telebot import types

bot = telebot.TeleBot(config.TELEGRAM_TOKEN)

@bot.message_handler(commands=['start'])
def send_welcome(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    itembtn1 = types.KeyboardButton('Акції')
    itembtn2 = types.KeyboardButton('Знижки компаній')

    markup.add(itembtn1, itembtn2)
    msg = bot.send_message(
        message.chat.id, 'Привіт, обері потрібну категорію', disable_notification=True,
        reply_markup=markup)
    bot.register_next_step_handler(msg, selectCategories)

def selectCategories(message):
    if (message.text == 'Акції'):
        select(message)
    elif (message.text == 'Знижки компаній'):
        markup = types.ReplyKeyboardRemove()
        bot.send_message(message.chat.id, 'Шукаю знижки...',
            disable_notification=True,)
        bot.send_message(message.chat.id, companies(
```

```
message), disable_web_page_preview=True, disable_notification=True, reply_markup=markup)
```

```
def select(message):
```

```
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
```

```
    itembtn1 = types.KeyboardButton('Медицина')
```

```
    itembtn2 = types.KeyboardButton('Їжа та ресторани')
```

```
    markup.add(itembtn1, itembtn2)
```

```
    msg = bot.send_message(
        message.chat.id, 'Обери підкатегорію', disable_notification=True, reply_markup=markup)
```

```
    bot.register_next_step_handler(msg, subCategoriesSelect)
```

```
def subCategoriesSelect(message):
```

```
    if (message.text == 'Медицина'):
```

```
        markup = types.ReplyKeyboardRemove()
```

```
        bot.send_message(message.chat.id, 'Шукаю акції...',
                          disable_notification=True,)
```

```
        bot.send_message(message.chat.id, promotions(
            message), disable_web_page_preview=True, disable_notification=True, reply_markup=markup)
```

```
    elif (message.text == 'Їжа та ресторани'):
```

```
        markup = types.ReplyKeyboardRemove()
```

```
        bot.send_message(message.chat.id, 'В пошуках знижок...',
                          disable_notification=True,)
```

```
        bot.send_message(message.chat.id, food(
            message), disable_web_page_preview=True, disable_notification=True, reply_markup=markup)
```

```
if __name__ == '__main__':
```

```
    bot.polling(none_stop=True)
```



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
 ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



## РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОТРИМАННЯ ЗНИЖОК З ВИКОРИСТАННЯМ TELEGRAM API НА МОВІ ПРОГРАМИ PYTHON

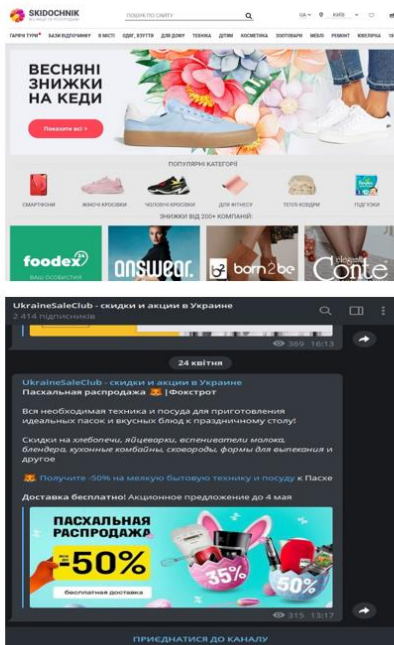
Виконав студент 4 курсу  
 Групи ПД-41  
 Станкевич Д.А.  
 Керівник роботи  
 Коба А.Б.

Київ – 2021

### МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета** даної роботи - дослідження та розробка програмного забезпечення для надання знижок на товари чи послуги використовуючи парсинг даних для збору та групування інформації.
- **Об'єктом дослідження** є додатки для надання знижок на товари та послуги.
- **Предметом дослідження** даної роботи є алгоритм побудови програмного забезпечення для надання знижок.

## АНАЛОГИ



- + Сучасний дизайн
- + Не потребує реєстрації
- + Зручний інтерфейс

- Погана локалізація
- Повільна робота
- Проблеми з відображенням контенту

- + Швидка робота
- + Мінімалістичний інтерфейс

- Рідке оновлення інформації про знижки
- Відсутність категоризації

3

## ТЕХНІЧНІ ЗАВДАННЯ

- Додаток повинен бути мультиплатформений;
- Наявність навігаційної клавіатури;
- Інтуїтивно зрозумілий інтерфейс;
- Текстовий супровід подальших дій;
- Програма повинна містити актуальну інформацію про знижки.

4

## ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



5

## МЕТОД ПАРСИНГУ ДАНИХ



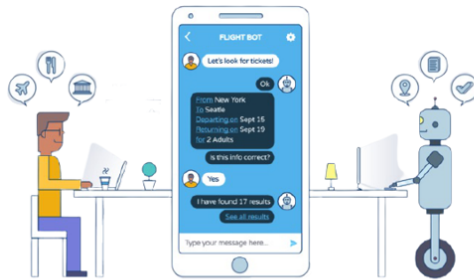
Завантаження  
контенту  
веб-сторінки

Парсинг та  
вилучення  
структурован  
их даних

Збереження  
даних в  
JSON, БД, XML  
та інші

6

## Застосування чат-боту



Чат-бот — комп'ютерна програма, розроблена на основі нейромереж та технологій машинного навчання, за допомогою якої можливо здійснювати комунікацію в аудіо- або текстовому форматі.

7

## Ініціалізація чат-боту

BotFathet – сервіс для реєстрації, управління та налаштування створених чат-ботів.



8

## ВИСНОВКИ

В даній роботі було проаналізовано існуючі додатки та виявлено їх переваги і недоліки.

На основі отриманих даних було розроблено додаток для отримання знижок використовуючи метод парсингу даних.

**Перспективою подальших досліджень** є можливість розширення категорій товарів, розробка функції підписки на обрані категорії товарів.

Результати дослідження бакалаврської роботи апробовані на Всеукраїнській науково-технічній конференції: «Застосування програмного забезпечення в інфокомунікаційних технологіях».

ДЯКУЮ ЗА УВАГУ!