Визначення функціональних та нефункціональних

вимог до програми, яка буде розроблятися.

Функціональні вимоги програми:

1. Програмний інтерфейс – буде надана висока увага швидкій взаємодії різних API веб сайт, аби користувачу була доступна можливість швидко та легко отримати дані. Усі частини bac-end будуть протестовані та мати належну документацію , аби усі бажаючи могли отримати доступ до того, чи іншого коду у своїх цілях, а також , аби було можливе легке розширення порталу
2. Функції продукту – Продукт буде містити наступні функції: Можливість отримати усю потрібну інформацію, аби проаналізувати певний продукт, а саме буде надано коментарі взяті із великої кількості веб-ресурсів таких як Розетка, Ebay, hotline, aliexpress, та інших. Також буде надано можливість переглянути різні відео ревю на певний продукт , якщо такий є у наявності, аби полегшити життя користувача , ми також даму йому можливість обрати за котрими саме додатковими ключами він хоче побачити ці ревю ( наприклад шукаючи телефон користувач можливо захоче подивитись краштест цього продукти, він матиме можливість ввести одне слово у стрічку та отримати додаткову , потрібну йому інформацію. Вся маніпуляцію із відео-роликами буде здійсненна із-за допомогою You-Tube API , котрий надає можливість отримати потрібні відео кліпи надавши ключові слова. Також буде надано аналіз цін продукту, а саме котра є зараз середня ціна на продукт та де його можна придбати.
3. Безпека користувача – Ми приділимо також велику увагу безпеці даного веб додатку , аби при реєстрації були запобіжно усіх можливих непорозумінь та у наслідок цього втрати даних із однієї зі сторін(клієнта чи веб-додатку)
4. Керування помилками – веб додаток буде розроблений на бек енд фреймофорку Flask на python в свою чергу це дає можливість максимально ефективно , а головне не резикуючи керувати усіма процесами на нашому ресурсі , аби користувач мав ідеальний user Friendly experience. Буде створено максимально ефективну структуру Error handling котра забезпечить усі вище переліченні можливості , що в свою чергу буде надавати цей надзвичайний досвід користування

Нефункціональні вимоги програми:

1. Доступність – Цей ресурс буде повність безкоштовний , а це в свою черго означає , що будь який користувач незалежно від віку чи статті , зможе отримати потрібну йому інформацію про будь який продукт(Якщо він є доступний на нашому веб- ресурсі, адже є обмеження у певних продуктах та типах продуктів)
2. Інтерфейс користувача – Це буде приємний та зрозумілий інтерфейс , котрий дозволить клієнту легко доступитись до потрібного йому продукти не потребуючи детального аналізу сторінки.
3. Комунікаційні інтерфейси – зважаючи на те, що це є в першу чергу веб ресур, він буде підтримуватись усіма можливими браузерами, та дасть клієнту, не залежно від його пристрою перегляду (комп’ютора, телефону чи планшету - отримати усі можливості нашого ресурсу).

У цій частині другого етапу курсової роботи ми будемо говорити про такі важливі інструменти у мові програмування Python як json( Java Script Object Notation ), xml ( Extensible Markup Language ) та Html ( HyperText Markup language ).

І так, json як ми вже сказали – це певний спосіб представлення інформації, більше того вона подається у вигляді тексту, однак надає можливість чітко структурувати об’єкти у тексті. Це надає можливість легко передавати цей тип даних у просторах інтернету, так як по суті він являє собою звичайний текст. Безумовно, що безліч мов програмування надають відповідні інструменти, для умовного розпакування цього json об’єкта з метою доступатися до полів та просто працювати з ним як зі звичайною структурою даних у цій мові програмування, яку можна змінювати, зберігати, копіювати, пересилати та виконувати безліч інших потрібних для розробника речей. Важливо сказати, що json містить в собі або масив певних значень, або якийсь об’єкт. Значеннями у такому масиві можуть бути як і примітивні типи даних так і найрізноманітніші об’єкти, полями яких можуть бути також різноманітні об’єкти.

Один з вивід використання json є обмін даними між браузером та сервером. У такому випадку запит надсилається з JavaScript ( мова програмування реалізація стандарту ECMAScript ), одна з функціональних можливостей, якого саме відповідає за виконання Get та Post ( і не тільки ) методів до певного серверу. JavaScript це може робити через Ajax запити, вихідні та вхідні дані, яких представляють собою json об’єкти. Це дозволяє розробляти якусь логіку на html сторінці без її перезавантаження та запобігти передаванню більш громіздких html файлів зі сторони сервера. Якщо говорити про веб технології, то json є альтернативою більш довгому способу представлення інформації, як xml.

Важливо зауважити, що json також використовується для передавання інформації з сервера до іншої програми. Це відбувається через використання API того чи іншого сервісу. До прикладу, на лекції з програмування ми використовуючи бібліотеку requests, робили запит з вказаною URL адресою та отримували, дані, які за допомогою методу json конвертувалися у словник ( dict ).

Json пакет надає функціонал, який дозволяє конвертувати дані до json та навпаки.

Для того, щоб перетворити Python об’єкт у стрічку ( json ) потрібно скористатися функцією json.dumps, яка приймає об’єкт мови програмування Python та повертає об'єкт класу str. У сигнатурі цієї функції є іменний параметр sort\_keys, який за замовчуванням дорівнює False, однак може бути поміняний на True. Якщо значення цього параметра дорівнює True, то ключі будуть посортовані у порідку зростання, якщо ключі є типу str, то вони будуть розміщення у алфавітному порядку, стрічку також порівнювальні у Python. Пакт json надає можливість зберігати конвертований об’єкт не тільки у комірках оперативної пам’яті після чого посилання на нього буде присвоєно певній змінній, але й у інші структури даних, наприклад об’єкт класу StringIO, цей клас знаходиться у Python модулі під назвою io. Це можу бути надзвичайно корисними коли потрібно певний об’єкт конвертувати та записати у файл. Так як конвертований об’єкт є представником класу str, то він через метод write() може бути записаний у файл від якого викликається цей метод. Для кращої читабельності конвертованого об’єкту було б доцільно використовувати, ще один іменований параметр у сигнатурі функції dumps, а саме indent, який може приймати значення деякого натурального числа. В результаті, при виведенні на екран, чи записі у файл стрічка буде приймати більш читабельного вигляду, через наявність відступів перед кожним полем.

Зараз ми наведемо кілька прикладів використання dumps, після цього розповімо про так би мовити про обернену до функції dumps, а саме функцію loads.

import json  
  
  
dictionary = {"b": 1, "a": 2}  
first\_json = json.dumps(dictionary)  
print(first\_json, "\n")  
second\_json = json.dumps(dictionary, sort\_keys=True, indent=3)  
print(second\_json)

Даний кусок коду виведе на екран наступні значення:

{"b": 1, "a": 2}

{

“a”: 1,

“b”: 1

}

Можна побачити, що в результаті використання sort\_keys=True, indent=3 представлення стрінга стає більш елегантним.

Цей кусок коду показує, як можна конвертувати словник у json стрічку та записати її у файл для збереження чи пересилання.

import json  
  
  
dictionary = {"Andrii": 17, "Vova": 21, "Vlad": 19}  
print(json.dump(dictionary, open("file.json", "w")))

При запуску даного куска коду на екран буде виведено None, так як функція dump при такому наборі вхідних параметрів нічого не повертає, однак у поточній директорії буде створено файл з розширенням .json в якому буде записано наступне:

{"Andrii": 17, "Vova": 21, "Vlad": 19}

Якщо говорити про функцію json.loads(), то вона, як ми вже згадували вище, є оберненою до функції json.dumps. Однак зважаючи на те, що Python програма може приймати json з різних джерел, де типи даних відрізняються від тих, що є у Python, можна припустити, що десь у пакеті json відбувається перетворення значень до таких, які є об’єктами класів у Python.

Зараз ми наведемо приклад використання функції loads:

import json  
  
  
dictionary = {"a": 1, "b": 2}  
new\_dictionary = json.dumps(dictionary)  
new\_dictionary = json.loads(new\_dictionary)  
print(new\_dictionary == dictionary)

В результаті виконання цієї програми буде виведено на екран значення True, що є яскравим прикладом того, що ці дві функції є оберненими між собою. Це є надзвичайно потужним інструментом, адже можливість приймати дані з різних джерел форматом json після чого конвертувати їх у об’єкт програми з, яким можна працювати через весь інтерфейс класу представником, якого він являється.

Наступний формат, який ми коротко опишемо це xml. Так як і json використовується для обміну інформацією між певними джерелами чи застосунками ( і не тільки ). Цей формат зберігає ієрархічно структуровані дані. Це можна розуміти так, що дані можна представити у вигляді дерева. Пакет xml у Python має функціонал, який позволяє видобувати дані з файлу з розширенням .xml та проводити з ними деякі маніпуляції. Наприклад, швидкий доступ до певних атрибутів та тексту. Клас ElementTree представляє весь xml файл як структуру даних, а клас Element лише певну гілку. Серед функціоналу є можливість рекурсивно проходитися по тегам дерева та можливість зміни тексту в тезі чи навіть додавання нового тегу. Щоб створити об’єкт класу ElementTree, який би репрезентував певний .xml файл потрібно присвоїти певній змінній значення ElementTree.parse(“filename.xml”). Щоб отримати представника класу Element з коренем у першому тезі xml файла потрібно присвоїти змінній значення ElementTree.parse(“filename.xml”).getroot(). Цей об’єкт є ітерований, проходячись по ньому ми отримуємо об’єкти класу Element (тільки дітей). Якщо хочемо отримати всіх нащадків потрібно пройтися по ElementTree.parse(“filename.xml”).getroot().iter().

Зараз ми також коротко опишемо пакет html у Python. Для роботи з html сторінками потрібно створити об’єкт класу, який буде наслідувати клас html.parser.HTMLParser та перевизначити у ньому методи handle\_starttag, handle\_endtag, handle\_data ( можливо й перевизначити й інші методи для обробки коментарів до прикладу ). Все це робиться для того, щоб після створення об’єкта цього класу до нього можна було б застосувати метод .feed(line), де line об’єкт типу str. Після виконання цього методу на екран буде виведено перелік початкових тегів, значень всередині них та завершальні теги ( відповідно до такого перевизначення методів ). Цей інструмент дає змогу аналізувати html сторінки, для подальшої праці з ними.

Завдання нашої курсової, більшої мірою полягає у видобуванні коментарів стосовно конкретного товару для подальшого застосування. Очевидно, що для сервісів, котрі не надають свого власно API або саме API не містить достатньо вичерпної інформації про продукт, нам прийдуться використовувати такі пакети мови програмування Python, які надають можливість робити запит на сторінку-характеристику потрібного продукту та працювати вже з цією інформацією з метою відшукання коментарів. Тип даних котрий повертає бібліотека urllib.requests це клас http.client.HTTPResponse. З отриманим об’єктом важко проводити будь-які маніпуляції, отож його без сумніву потрібно перетворювати у більш сприятливий для аналізу тип даних. Насправді відповідь тут очевидна, об’єкт даного потрібно конвертувати у об’єкт типу стрінг ( str ). Це можна легко зробити за допомогою методу класу http.client.HTTPResponse під назвою read, котрий повертає повністю всю структуру html файла у вигляді байтів.

Зараз, ми наведемо дуже короткий приклад вище-поданої інформації, а після напишемо про те як ми б хотіли безпосередньо працювати з цією стрічкою.

import urllib.request  
  
url = "https://helpx.adobe.com/ua/acrobat/using/converting-web-pages-pdf.html"  
with urllib.request.urlopen(url) as response:  
 print(type(response).\_\_name\_\_)  
 response = response.read()  
 print(type(response).\_\_name\_\_)  
 response = response.decode(" utf-8 ")  
 print(type(response).\_\_name\_\_)

Отож, після виконання ця програма виведе на екран наступне:

HTTPResponse

bytes

str

Це означає, що для перетворення байтів, у бажану html сторінку ми повинні розкодувати її. Це робиться за допомогою метода об’єкта класу bytes, а саме decode(“ utf-8 ”).

Отже, отримавши готову html сторінку у вигляді стрічки, ми повинні відшукати відгуки користувачів стосовно даного продукту. Ми вирішили використовувати такий інструмент Python, як bs4. А якщо бути точнішим, то клас, який знаходиться у модулі bs4, під назвою BeautifulSoup. Цей інструмент дозволяє легко працювати з html сторінкою, яка представлена стрінгом, а саме надає можливість знаходити всі теги, які відповідають певним вимогам ( наприклад клас чи то айді ). Таким чином можна поступово знаходити теги в, яких зберігається потрібна нам інформація та розробити алгоритм, який міг би швидко знаходити відгуки на html сторінці.

Наведемо приклад використання BeautifulSoup:

import urllib.request  
from bs4 import BeautifulSoup  
  
url = "https://helpx.adobe.com/ua/acrobat/using/converting-web-pages-pdf.html"  
with urllib.request.urlopen(url) as response:  
 response = response.read()  
 response = response.decode(" utf-8 ")  
 html = BeautifulSoup(response, "html.parser")  
 print(type(html).\_\_name\_\_)  
 html = html.find("div",  
 class\_="image-caption")  
 print(html.text.strip())

В результаті виконання даної програми, буде виведено на екран наступне:

BeautifulSoup

Меню на панелі інструментів PDF пропонує можливості для швидкого перетворення та друку.

You-Tube Api надає дані на основі отриманих, а саме назві певного відео кліпу, ключових слів, котрі відповідають його змісту та зможуть як умова краще надати корисну інформацію користувачеві Формат даних отриманий із за допомогою You-Tube API: отримується дані у форматі json. Там можна отримати різну інформацію таку, як кількість лайків/дизлайків , кількість переглядів, посилання на саме відео, а саме ключ до посилання, ютуб канал на котрому розміщено дане відео, також ми отримуємо опис відео кліпу, його назву . Підібравши різні клчові слова на назви даних, котрі ми хочемо отримати, можна отримати різні json файли, адже усі відео-кліпи мають свою певну специфіку.

Така велика кількість даних дозволяє здійснити детальний аналіз та підібрати сам те відео, котрі дадуть найкраще розуміння товару користувачеві.

from apiclient.discovery import build  
  
DEVELOPER\_KEY = "AIzaSyArhRiaMcsLQIyhfH2\_c32OE3N9YjtDelA"  
YOUTUBE\_API\_SERVICE\_NAME = "youtube"  
YOUTUBE\_API\_VERSION = "v3"  
  
  
def youtube\_search(options):  
 youtube = build(YOUTUBE\_API\_SERVICE\_NAME, YOUTUBE\_API\_VERSION,  
 developerKey="AIzaSyArhRiaMcsLQIyhfH2\_c32OE3N9YjtDelA")  
  
 search\_response = youtube.search().list(  
 # the key on which it finds the video   
 q=options  
 part="id,snippet",  
 maxResults=5  
 ).execute()  
 # outputs the json file  
 return search\_response.get("items", [])