



Internship Finder Chatbot

Programmed by **Malena Ballon**
Azra Cagsawa & Jhane Hermano

ITMGT 25.03 J | July 20

Internship Finder Chatbot

A chatbot that returns job or internship offers based on the user's interests, schedule, desired pay, and other preferences.

But why?



To make internship or job
search easier and faster



Overview

★ SCRAPING
FB POSTS

★ CLEANING
THE DATASET

★ PROGRAMMING
THE CHATBOT

Scraping FB Posts

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.webdriver.support.wait import WebDriverWait
import time
from time import sleep

# Create an instance of the Safari webdriver
driver = webdriver.Safari()

# Open the Facebook homepage
driver.get("http://www.facebook.com/groups/1215947428519908/")

# Iterate over the posts
captions_links = set()
```

Importing Selenium for web scraping

Importing time to allow posts to load

Importing Safari Webdriver to allow the program to access Safari

Accessing ADMU Jobs & Internships FB group

Creating a placeholder to store captions & links

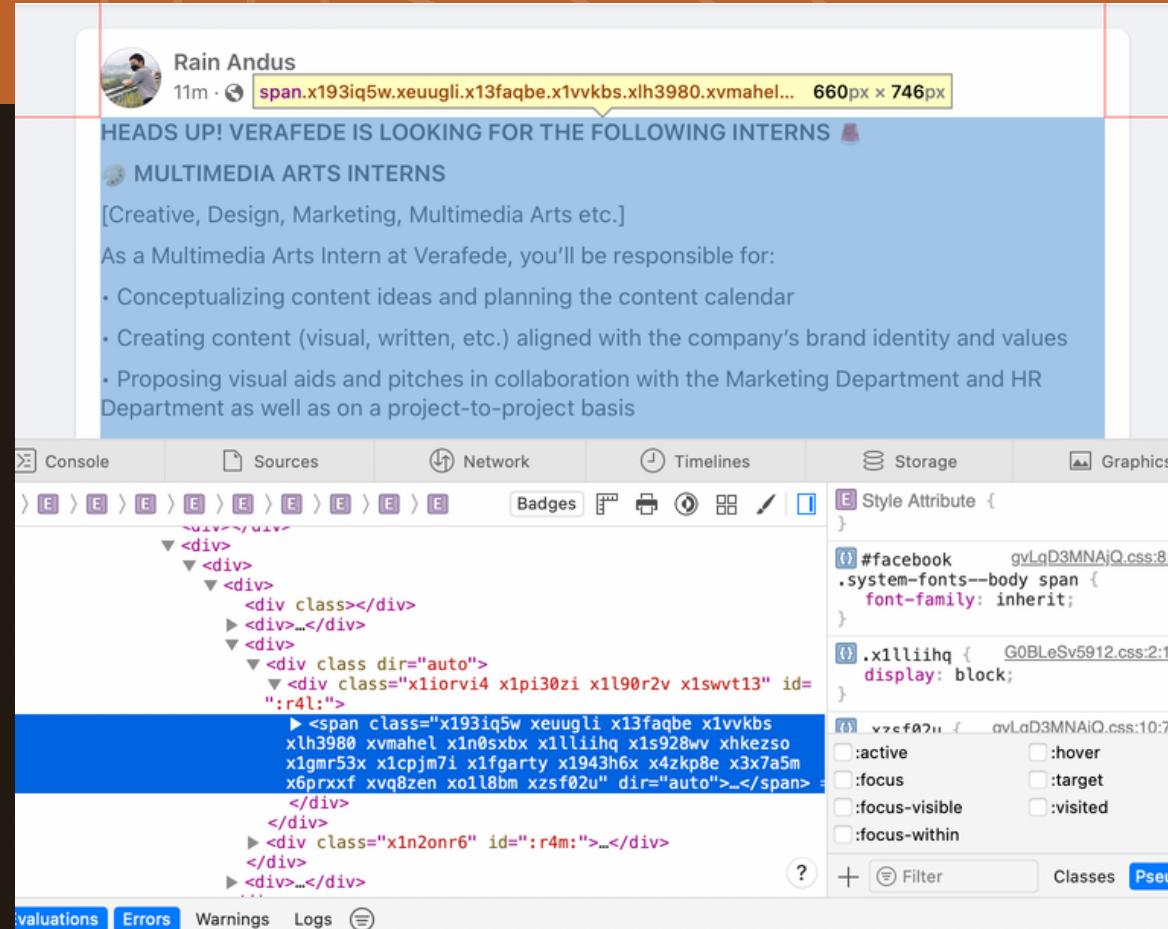
Scraping FB Posts

```
for i in ["posts"]:  
    # ****  
    # !! Change the group URL to the desired Facebook group URL !!  
    # ****  
  
    # Scroll down (increase the range to scroll more)  
    for j in range(0, 3):  
        driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")  
        time.sleep(2)  
  
    # Target all the link elements on the page using JavaScript  
    anchors = driver.execute_script("return document.getElementsByTagName('a');")  
    anchors = [a.get_attribute('href') for a in anchors]  
  
    # Narrow down all links to caption links only  
    links = [a for a in anchors if str(a).startswith("https://www.facebook.com/groups/1215947428519908/posts/")  
            and not a.endswith("#")]
```

Scrolling through the FB group to allow more posts to load

Storing the links of every post visible after scrolling

Filtering the stored links and not a.endswith("#")



Scraping FB Posts

```
for caption_link in links:
    driver.get(caption_link)
    sleep(2) # Add a delay of 2 seconds here
```

Accessing each link stored in 'links'

```
# Find all spans with the relevant class names within the caption element
caption_spans = driver.find_elements(By.XPATH, "//span[contains(@class, 'x193iq5w')][contains(@class, 'xeuu')]"
                                         "[contains(@class, 'xvmahel')][contains(@class, 'x1n0sxbx')][contains(@class, 'x6prxxf')][contains(@class, 'xvq8zen')]
                                         ][contains(@class, 'xo1l8bm')][contains(@class, 'xzsf02u')]"")
```

```
# Concatenate the text from all the spans to form the caption
caption_text = ' '.join(span.text for span in caption_spans)

# Associate the caption with its corresponding link
captions_links.add((caption_link, caption_text))

driver.quit()
```

Extracting the caption of every post

Adding the caption and its link to the placeholder

Scraping FB Posts

```
print('Found ' + str(len(captions_links)) + ' links and captions')
print('I scraped ' + str(len(captions_links)) + ' links and captions!')
```

Found 488 links and captions
I scraped 488 links and captions!

Counting scraped posts

Scraping FB Posts

```
# Specify the file path where you want to save the links and captions
file_path = "/Users/azrasamanta/Desktop/demo_scraping.txt"

# Save the links and captions to a text file
with open(file_path, "w", encoding="utf-8") as file:
    for link, caption in captions_links:
        file.write("Link: " + link + "\n")
        file.write("Caption: " + caption + "\n\n")

print(f"The links and captions have been saved to {file_path}.")
```

The links and captions have been saved to /Users/azrasamanta/Desktop/demo_scraping.txt.

Saving the scraped data as a txt file

Problems encountered

Scraping incomplete / unreachable links

Link: <https://www.facebook.com/groups/1215947428519908/posts/#>

Caption: Forgot Account?

Link: <https://www.facebook.com/groups/1215947428519908/posts/#>

Caption: [REMOTE] [SALES] [COMMISSION] [HIGH-TICKET] [WFH] [FLEXIBLE] [PART-TIME]

Scraping one post multiple times

Link: <https://www.facebook.com/groups/1215947428519908/posts/#>

Caption: Forgot Account?

Link: [https://www.facebook.com/groups/1215947428519908/posts/6494791530635445/?__cft__\[0\]=AZUnsPszLsZFJCTRDEqYDEacCm1AK_5GvGSNWFNeCAR8eMR5sVtm-GJ0oy0FuN9CWQRrqMzyuaA3SH81IE7iTecWHhe7hyPMMZ4vkltiKejA9CpyKUFNND2bGHQK-n6VE7_CBAKN8g0r6w-s3sMdZGeZQRSwtywjax8lBlVXLd384SdDcwMpjXccsF0p7u03XFU_HPU9dSz9l9o9YME7btG&__tn__=%2C0%2CP-R](https://www.facebook.com/groups/1215947428519908/posts/6494791530635445/?__cft__[0]=AZUnsPszLsZFJCTRDEqYDEacCm1AK_5GvGSNWFNeCAR8eMR5sVtm-GJ0oy0FuN9CWQRrqMzyuaA3SH81IE7iTecWHhe7hyPMMZ4vkltiKejA9CpyKUFNND2bGHQK-n6VE7_CBAKN8g0r6w-s3sMdZGeZQRSwtywjax8lBlVXLd384SdDcwMpjXccsF0p7u03XFU_HPU9dSz9l9o9YME7btG&__tn__=%2C0%2CP-R)

Caption: [REMOTE] [SALES] [COMMISSION] [HIGH-TICKET] [WFH] [FLEXIBLE] [PART-TIME]*Hiring Remote Appointment Setters*Hi everyone! Sharing this opportunity for part-time work with a flexible schedule.Role:Outreach to prospective clients and booking an appointment with them. Once the appointment is booked, the job is done.See more

Link: <https://www.facebook.com/groups/1215947428519908/posts/#>

Caption: [REMOTE] [SALES] [COMMISSION] [HIGH-TICKET] [WFH] [FLEXIBLE] [PART-TIME]

Link: [https://www.facebook.com/groups/1215947428519908/posts/6494791530635445/?comment_id=6496438153804116&__cft__\[0\]=AZUnsPszLsZFJCTRDEqYDEacCm1AK_5GvGSNWFNeCAR8eMR5sVtm-GJ0oy0FuN9CWQRrqMzyuaA3SH81IE7iTecWHhe7hyPMMZ4vkltiKejA9CpyKUFNND2bGHQK-n6VE7_CBAKN8g0r6w-s3sMdZGeZQRSwtywjax8lBlVXLd384SdDcwMpjXccsF0p7u03XFU_HPU9dSz9l9o9YME7btG&__tn__=R](https://www.facebook.com/groups/1215947428519908/posts/6494791530635445/?comment_id=6496438153804116&__cft__[0]=AZUnsPszLsZFJCTRDEqYDEacCm1AK_5GvGSNWFNeCAR8eMR5sVtm-GJ0oy0FuN9CWQRrqMzyuaA3SH81IE7iTecWHhe7hyPMMZ4vkltiKejA9CpyKUFNND2bGHQK-n6VE7_CBAKN8g0r6w-s3sMdZGeZQRSwtywjax8lBlVXLd384SdDcwMpjXccsF0p7u03XFU_HPU9dSz9l9o9YME7btG&__tn__=R)

Caption: *Hiring Remote Appointment Setters*

localhost

Facebook Twitter Desktop/Python/ JSON and Mat...er Notebook data

jupyter [Demo] Facebook Web Scraper Last Checkpoint: 4 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel) O

In []:

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.webdriver.support.wait import WebDriverWait
import os
import wget
import time
from time import sleep

# Create an instance of the Safari webdriver
driver = webdriver.Safari()

# Open the Facebook homepage
driver.get("http://www.facebook.com/groups/1215947428519908/")

# Iterate over the posts
captions_links = set()

for i in ["posts"]:
    # *****
    # !! Change the group URL to the desired Facebook group URL !!
    # *****

    # Scroll down (increase the range to scroll more)
    for j in range(0, 3):
        driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
        time.sleep(2)

    # Target all the link elements on the page using JavaScript
    anchors = driver.execute_script("return document.getElementsByTagName('a');")
    anchors = [a.get_attribute('href') for a in anchors]

    # Narrow down all links to caption links only
    links = [a for a in anchors if str(a).startswith("https://www.facebook.com/groups/1215947428519908/posts/")]

    for caption_link in links:
        driver.get(caption_link)
        sleep(2) # Add a delay of 2 seconds here
```

Cleaning the dataset

```
# IMPORTING THE TXT FILE
from pprint import pprint
import pandas as pd
import re

with open('raw_links_and_captions_dataset.txt', encoding='utf-8') as file:
    data = file.read()

links = re.findall(r'Link: (.*)\n', data)
captions = re.findall(r'Caption: (.*)\n', data)
```

IMPORT TXT FILE

DISTINGUISH LINK AND CAPTIONS

SAMPLE PRINT

Cleaning the dataset

```
# SAMPLE PRINT TO CHECK
print(links[0])
print("\n")
print(captions[0])
```

[https://www.facebook.com/groups/1215947428519908/posts/6514220408692557/?comment_id=6517780705003194&_cft_\[0\]=AZXmZLrcz_bAipK6VGTYCIPxko-RhG1kDDSJxUlkx1kFCDft8TJbVfw0q0l3nId2_xPPBid9JfJ2ruiu8cmjzrMOrrhHONAc5-e7-KAblhwpZH90awSIhZwq-b0Prlqe6MPvX4NBV7cAMD1n0--n923g9f4AUEQPV_BP8cDW9xF2Gsnyutb-P-TNXk5LV7StGVJ7pI2J8hxWfqkpVom24pc&_tn_=R](https://www.facebook.com/groups/1215947428519908/posts/6514220408692557/?comment_id=6517780705003194&_cft_[0]=AZXmZLrcz_bAipK6VGTYCIPxko-RhG1kDDSJxUlkx1kFCDft8TJbVfw0q0l3nId2_xPPBid9JfJ2ruiu8cmjzrMOrrhHONAc5-e7-KAblhwpZH90awSIhZwq-b0Prlqe6MPvX4NBV7cAMD1n0--n923g9f4AUEQPV_BP8cDW9xF2Gsnyutb-P-TNXk5LV7StGVJ7pI2J8hxWfqkpVom24pc&_tn_=R)

Forgot Account? Interested! Info po interested po! PELAJAR INTERN DIPERLUKAN SEGERA! Pekerja yang diperlukan bagi jurusan berikut : Digital Communication Sales Accounting & finance Journalism Creative Content Creator Multimedia Nama Syarikat: AGENSI PEKERJAAN MLMS SDN BHD Lokasi: LOT 73620-3-2 PERKEDAIAN PAK SABAH IV, 23000 DUNGUN, TERENGGANU. Jika berminat untuk intern: Boleh tekan link ws di bawah untuk hantar resume anda. <https://mahuintern.wasap.my> Rathina property consultant WE ARE HIRING INTERNSHIP IN MALACCA OPEN POSITION Located at JLN KOTA LAKSAMAMA 3/15, TAMAN MAHAJAYA, 75200 MELAKA. ADMIN HUMAN RESOURCES SECRETARY MARKETING PERSONAL ASSISTANT BENEFITS: ALLOWANCE ADDITIONAL COMMISSION COMFORTABLE WORK ENVIRONMENT TRAINING PROVIDED INTERESTED DO PM THE NUMBER BELOW 011-33320483 | RATHINA017-2379064 | UNES FULL-TIME] [REAL ESTATE] [PROPERTY SPECIALIST] [AYALA LAND PREMIER] Hello everyone! The No. 1 Real Estate Company in the PH is hiring! Premier Property Specialist If you are looking for: * An opportunity to grow and be successful in the field of real estate sales * A young, vibrant, and open-minded work environment that allows you to think out of the box and pursue your passions * A top-caliber team that handles an impressive portfolio. * earning millions in just months what you can earn in years You can reach me at 0917 633 1890 or send your CV at lusterio.emmanuel@ayalalandpremier.co "You don't have to be great to start, but you have to start to be great" Emmanuel Chua Lusterio 吳五金 Sales Manager Team New Manila Ayala Land Premier

Cleaning the dataset

GENERATE KEYWORDS

- **MANUALLY DETERMINED CATEGORIES**
- **CHATGPT PRODUCED KEYWORDS**

```
pprint(keywords)
'GRAPHIC DESIGN': ['GRAPHIC DESIGN',
'VISUAL DESIGN',
'ART DIRECTION',
'ILLUSTRATION',
'LOGO DESIGN',
'TYPOGRAPHY',
'LAYOUT',
'BRANDING',
'USER INTERFACE DESIGN',
'USER EXPERIENCE DESIGN',
'PRINT DESIGN',
'DIGITAL DESIGN',
'PACKAGING DESIGN',
'MOTION GRAPHICS',
'INFOGRAPHICS',
'WEB DESIGN',
'MARKETING DESIGN',
'ADVERTISING DESIGN',
'PUBLICATION DESIGN',
'CREATIVE DIRECTION']
```

```
pprint(keywords)
{'FINANCE': ['FINANCE',
'ACCOUNTING',
'INVESTMENT',
'BANKING',
'FINANCIAL ANALYSIS',
'RISK MANAGEMENT',
'ASSET MANAGEMENT',
'PORTFOLIO MANAGEMENT',
'FINANCIAL PLANNING',
'BUDGETING',
'TAXATION',
'AUDITING',
'FINANCIAL REPORTING',
'CORPORATE FINANCE',
'FINANCIAL MODELING',
'MERGERS AND ACQUISITIONS',
'HEDGE FUNDS',
'EQUITY RESEARCH',
'TREASURY',
```

Cleaning the dataset

WRITE CSV FILE

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Post	Link	FINANCE	GRAPHIC	INTERNSH	MANAGEM	MARKETIN	ONLINE	SE	TECHNOLO	PAID	HYBRID	ONSITE
2	Post #1	https://wv	YES	NO	YES	YES	YES	NO	YES	YES	YES	YES	YES
3	Post #2	https://wv	YES	NO	NO	YES	NO	NO	YES	NO	YES	YES	YES
4	Post #3	https://wv	NO	NO	NO	NO	YES	NO	YES	NO	YES	YES	NO
5	Post #4	https://wv	YES	NO	YES	YES	YES	NO	YES	YES	YES	YES	YES
6	Post #5	https://wv	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES
7	Post #6	https://wv	YES	NO	YES	YES	YES	NO	YES	NO	YES	NO	YES
8	Post #7	https://wv	YES	NO	NO	YES	YES	NO	YES	NO	YES	YES	NO
9	Post #8	https://wv	YES	NO	NO	YES	YES	NO	YES	YES	YES	YES	YES
10	Post #9	https://wv	YES	NO	NO	YES	YES	NO	YES	YES	YES	YES	YES
11	Post #10	https://wv	YES	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES
12	Post #11	https://wv	NO	NO	YES	YES	NO	NO	YES	YES	NO	NO	YES
13	Post #12	https://wv	NO	YES	YES	YES	YES	NO	YES	NO	YES	YES	YES
14	Post #13	https://wv	YES	NO	YES	YES	NO	NO	YES	YES	YES	YES	YES
15	Post #14	https://wv	NO	NO	YES	NO	NO	NO	YES	YES	NO	NO	YES
16	Post #15	https://wv	YES	NO	NO	YES	NO	NO	YES	YES	YES	YES	YES
17	Post #16	https://wv	YES	NO	NO	NO	YES	NO	YES	NO	YES	YES	YES
18	Post #17	https://wv	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES
19	Post #18	https://wv	NO	NO	NO	NO	YES	NO	YES	NO	NO	NO	YES
20	Post #19	https://wv	NO	NO	NO	NO	YES	NO	YES	YES	YES	YES	NO
21	Post #20	https://wv	YES	NO	YES	YES	YES	NO	YES	YES	YES	YES	YES
22	Post #21	https://wv	YES	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES
23	Post #22	https://wv	YES	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES
24	Post #23	https://wv	NO	NO	NO	NO	NO	YES	YES	YES	NO	NO	YES
25	Post #24	https://wv	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES
26	Post #25	https://wv	NO	NO	YES	NO	YES	NO	NO	YES	NO	YES	YES

Programming the Chatbot

```
pi@raspberrypi:~/Desktop/ChatBot$ python3 chatbot.py  
  
from sentence_transformers import SentenceTransformer, util  
from dict_keywords import keywords  
from flask import Flask, request, jsonify  
from flask_cors import CORS  
  
app = Flask(__name__)  
CORS(app)  
  
with open('links_and_captions.txt', 'r', encoding='utf8') as f:  
    lines = f.readlines()
```

Programming the Chatbot

```
jobs = [{'link': lines[i].strip(), 'caption': lines[i+1].strip() if i+1 < len(lines) else ""} for i in range(0, len(lines), 2)]  
  
model = SentenceTransformer('all-MiniLM-L6-v2')  
  
job_descriptions = [job['caption'] for job in jobs]  
  
job_vectors = model.encode(job_descriptions, convert_to_tensor=True)  
  
returned_job_indices = []  
context = ""  
last_query = ""
```

Programming the Chatbot

```
def search_jobs(query):
    global last_query
    query_words = set(query.upper().split())
    keyword_keys = [key for key in keywords if key in query_words]

    for key in keyword_keys:
        query += " " + ".join(keywords[key])

    query_vector = model.encode(query, convert_to_tensor=True)
    cos_similarities = util.pytorch_cos_sim(query_vector, job_vectors)
    cos_similarities_list = cos_similarities.tolist()[0]

    for index in returned_job_indices:
        if index < len(cos_similarities_list):
            cos_similarities_list[index] = -1

    if len(returned_job_indices) >= len(cos_similarities_list):
        return None

    top_job_index = cos_similarities_list.index(max(cos_similarities_list))
    returned_job_indices.append(top_job_index)
    return jobs[top_job_index]
```

Programming the Chatbot

Hello! I am Aids. I can help you find jobs and internships. Type "Get started" to begin.

```
def generate_response(query):
    global context, last_query, returned_job_indices

    if query.lower() == "get started":
        context = "asked_for_job_details"
        last_query = ""
        returned_job_indices = []
        return "Could you please provide details about the type of job or internship you're looking for?"
```

Programming the Chatbot

```
elif context == "asked_for_job_details":  
    if query.lower() == 'next':  
        if len(returned_job_indices) < len(jobs):  
            job = search_jobs(last_query)  
            if job is not None:  
                job['link'] = job['link'].replace('Link: ', '')  
                return f"I found a job that might match your query!<br/><br/>Please visit this link: <a href='{job['link']}'>{job['link']}            else:  
                context = ""  
                last_query = ""  
                return "I'm sorry, I couldn't find any more jobs that match your query. Type 'Get started' to begin."  
        else:  
            context = ""  
            last_query = ""  
            return "I'm sorry, I couldn't find any more jobs that match your query. Type 'Get started' to begin."  
    elif query.lower() == 'yes':  
        context = ""  
        last_query = ""  
        return "Great! If you need more assistance, type 'Get started'."
```

Programming the Chatbot

```
else:  
    last_query = query  
    job = search_jobs(last_query)  
    if job is not None:  
        context = "asked_for_job_details"  
        job['link'] = job['link'].replace('Link: ', '')  
        return f"I found a job that might match your query!<br/><br/>Please visit this link: <a href='{job['link']}'>{job['link']}    else:  
        context = "asked_for_job_details"  
        return "I'm sorry, I couldn't find any jobs that match your query. Could you please provide more details about the type of job or internship you're looking for?"
```

I found a job that might match your query!

Please visit this link: [https://www.facebook.com/groups/1215947428519908/posts/cft_\[0\]=AZUnX4qcC5K0a06AGmhGfSRnSOy7l03WXojTy7lLXxBVsAerlHX1iXCAEPIlrl4yZYITSONWsOenrGvwUWqj8I3GepHQqash0VgPiy1Z0gGoMqL3yhyY9jl6b5rl2mVc3td2Jfyro5n&tn=%2CO%2CP-R](https://www.facebook.com/groups/1215947428519908/posts/cft_[0]=AZUnX4qcC5K0a06AGmhGfSRnSOy7l03WXojTy7lLXxBVsAerlHX1iXCAEPIlrl4yZYITSONWsOenrGvwUWqj8I3GepHQqash0VgPiy1Z0gGoMqL3yhyY9jl6b5rl2mVc3td2Jfyro5n&tn=%2CO%2CP-R)

Please indicate your next step by typing one of the following options:

Yes if this job meets your expectations and you don't need to explore other options.

Next if you want to view another job based on your initial preferences.

Refine if you'd like to add more specific criteria to better suit your job search.

Restart if you prefer to begin a new search for a different job role.

Programming the Chatbot

```
elif context == "asked_for_job_details":  
    if query.lower() == 'next':  
        if len(returned_job_indices) < len(jobs):  
            job = search_jobs(last_query)  
            if job is not None:  
                job['link'] = job['link'].replace('Link: ', '')  
                return f"I found a job that might match your query!<br/><br/>Please visit this link: <a href='{job['link']}'>{job['link']}            else:  
                context = ""  
                last_query = ""  
                return "I'm sorry, I couldn't find any more jobs that match your query. Type 'Get started' to begin."  
        else:  
            context = ""  
            last_query = ""  
            return "I'm sorry, I couldn't find any more jobs that match your query. Type 'Get started' to begin."  
    elif query.lower() == 'yes':  
        context = ""  
        last_query = ""  
        return "Great! If you need more assistance, type 'Get started'."
```

Programming the Chatbot

```
elif query.lower() == 'refine':
    context = "refine_search"
    return "Please provide additional details to refine your job search."
elif query.lower() == 'restart':
    context = ""
    last_query = ""
    return "Let's start a new search. Type 'Get started' to begin."
else:
    last_query = query
    job = search_jobs(last_query)
    if job is not None:
        context = "asked_for_job_details"
        job['link'] = job['link'].replace('Link: ', '')
        return f"I found a job that might match your query!<br/><br/>Please visit this link: <a href='{job['link']}'>{job['link']}
```

Programming the Chatbot

```
elif context == "refine_search":  
    last_query += " " + query  
    job = search_jobs(last_query)  
    if job is not None:  
        context = "asked_for_job_details"  
        job['link'] = job['link'].replace('Link: ', '')  
        return f"I found a job that might match your query!<br/><br/>Please visit this link: <a href='{job['link']}'>{job['link']}    else:  
        context = "refine_search"  
        return "I'm sorry, I couldn't find any jobs that match your new criteria. Could you please provide more details to refine your job search?"
```

Programming the Chatbot

```
else:  
    context = ""  
    return "Type 'Get started' to begin."
```

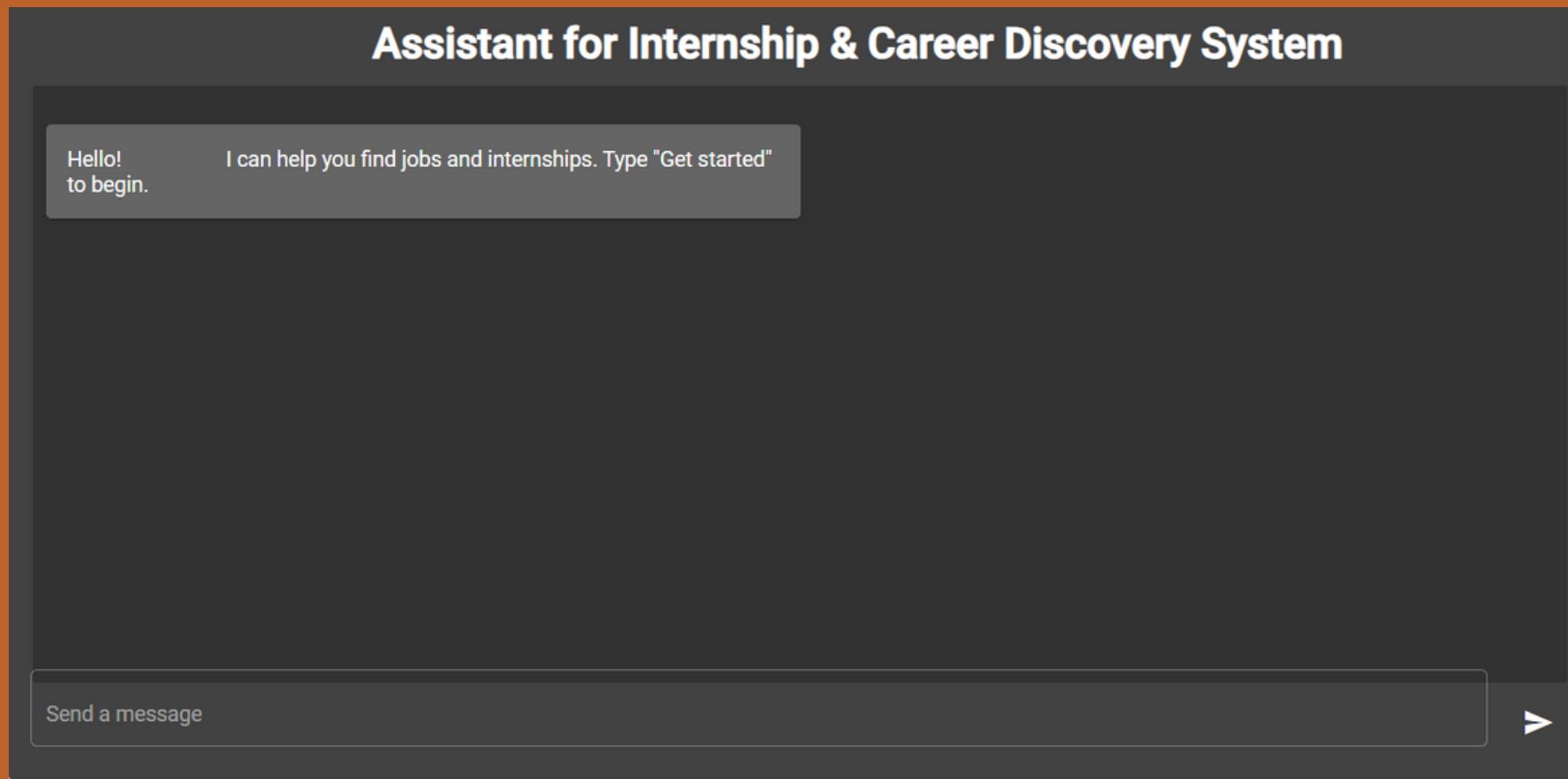


Programming the Chatbot

```
@app.route('/api/ask', methods=['POST'])
def ask():
    x = request.get_json()
    response = generate_response(str(x['prompt']))
    return jsonify(str(response))

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

HTML Code



```
▼ app
  ◁ app.component.html
  ↗ app.component.scss
  TS app.component.spec.ts
  TS app.component.ts
  TS app.model.ts
  TS app.module.ts
  TS app.service.spec.ts
  TS app.service.ts
  > assets
  > environments
  ★ favicon.ico
  ◁ index.html
  TS main.ts
  TS polyfills.ts
  ↗ styles.scss
  TS test.ts
```



The word "Live Demo" is written in a large, white, sans-serif font. A thin white oval surrounds the text. Two small, five-pointed starburst shapes, one at the top left and one at the bottom left, extend from the oval's curve towards the text. The background is dark gray.

Live Demo

RECOMMENDATIONS

FREQUENT AND
AUTOMATED
SCRAPING

APPLICATION
INTEGRATION

UTILIZATION
OF CHATGPT

SCRAPING PROCESS

- Automated scraping at a fixed interval (weekly, daily, bimonthly)
- Updating and continuously writing a txt file used by the cleaning process -> also impacts the csv file continuously written in the dataset clean up
- More efficient use of ChatBot -> gives the most recent posts for the user



APPLICATION INTEGRATION

- Running the code as a Facebook or Telegram ChatBot

```
from flask import Flask, request

app = Flask(__name__)

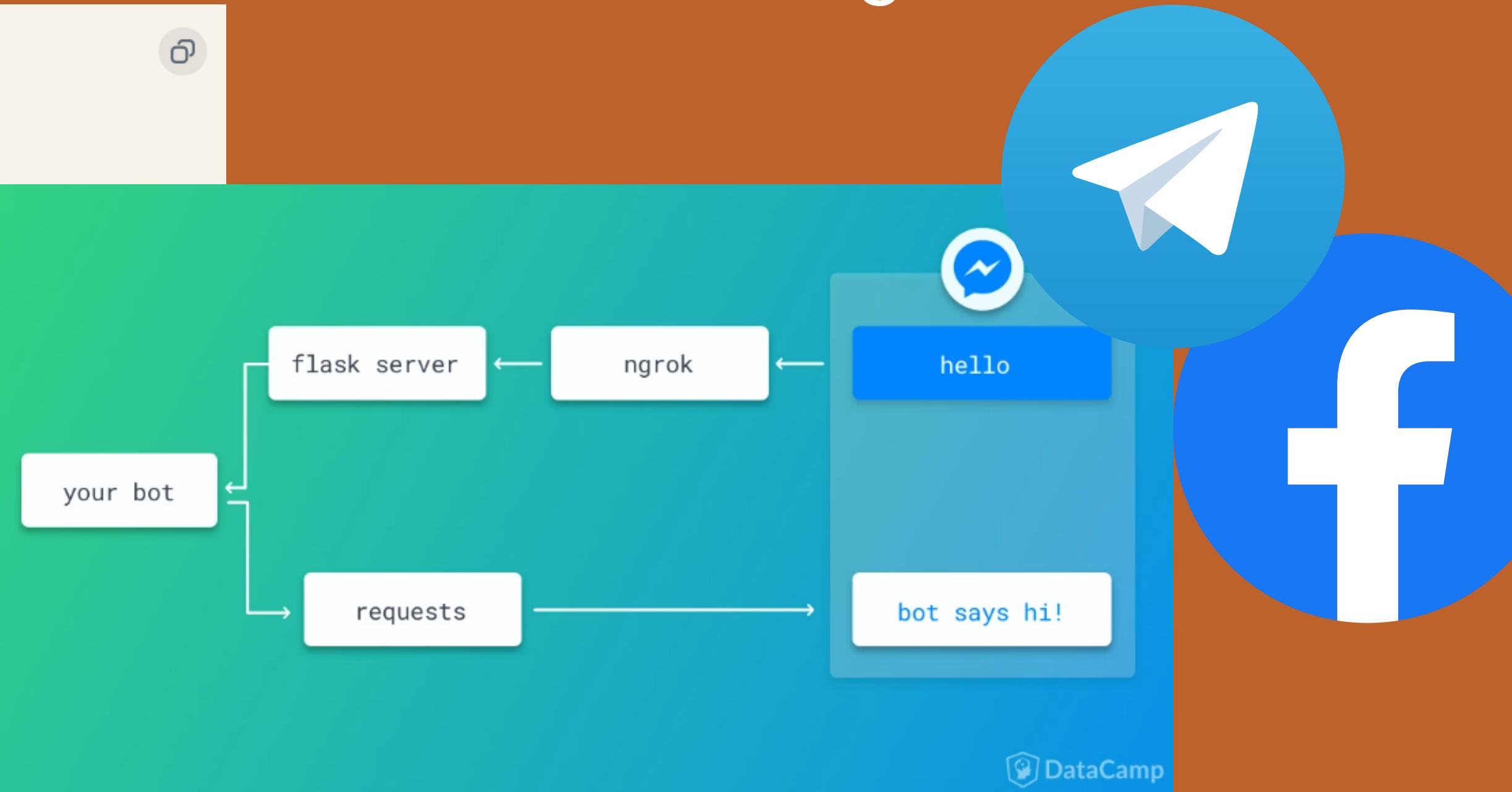
FB_API_URL = 'https://graph.facebook.com/v2.6/me/messages'
VERIFY_TOKEN = '# <paste your verify token here>'
PAGE_ACCESS_TOKEN = '# paste your page access token here>'

def get_bot_response(message):
    """This is just a dummy function, returning a variation of
    the user said. Replace this function with one connected to
    return "This is a dummy response to '{}'.format(message)

def verify_webhook(req):
    if req.args.get("hub.verify_token") == VERIFY_TOKEN:
        return req.args.get("hub.challenge")
    else:
        return "incorrect"

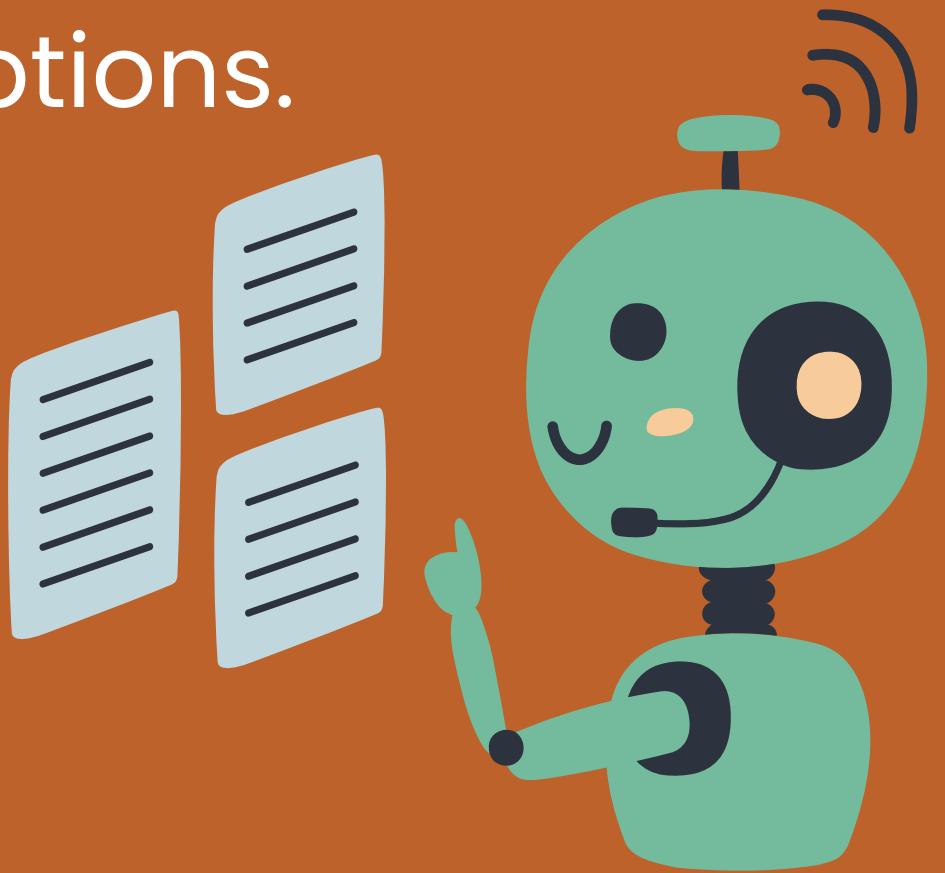
def respond(sender, message):
    """Formulate a response to the user and
    pass it on to a function that sends it."""

```



UTILIZATION OF CHATGPT

- Using the AI word-processing of ChatGPT to more accurately categorize posts and identify key information from the captions.



based on this text:

[PAID INTERNSHIP][MARKETING][GRAPHIC DESIGN]

Remedy is looking for talented and creative individuals to join our team as a Marketing Intern! Your role will be focused on developing Remedy and its sister company's brand language and content through graphics, content creation and planning, and multimedia design.

You will be making impactful and valuable contributions to the business' marketing by:

- Interested in the skin and wellness industry, e-commerce, graphic design, and digital marketing
- Up to date on trends and social media
- Good communication and presentation skills
- Excellent problem-solving skills

what is the company looking for?

what is the company name?

what is the general job description?

what are the qualifications required?

```
import openai
openai.api_key = 'sk-vXsr7JA0IqQHLMbrC0sWT3B1bkFJhI139b6X4E29o9DBW6hx'

# list models
models = openai.Model.list()

# print the first model's id
print(models.data[0].id)

# create a chat completion
chat_completion = openai.ChatCompletion.create(
    message="based on this text:\n\n[PAID INTERNSHIP][MARKETING][GRAPHIC DESIGN]\n\nRemedy is looking for talented and creative individuals to join our team as a Marketing Intern! Your role will be focused on developing Remedy and its sister company's brand language and content through graphics, content creation and planning, and multimedia design.\n\nYou will be making impactful and valuable contributions to the business' marketing by:\n\n- Interested in the skin and wellness industry, e-commerce, graphic design, and digital marketing\n- Up to date on trends and social media\n- Good communication and presentation skills\n- Excellent problem-solving skills\n\nwhat is the company looking for?\nwhat is the company name?\nwhat is the general job description?\nwhat are the qualifications required?",
```

```
import openai

def chat_with_chatgpt(prompt, model="gpt-3.5-turbo"):
    response = openai.Completion.create(
        engine=model,
        prompt=prompt,
        max_tokens=100,
        n=1,
        stop=None,
        temperature=0.5,
    )

    message = response.choices[0].text
    return message
```