Ways to get LinkedIn Profile Data

Comprehensive Guide to Fetching User Profiles (Free and Paid)

Goals

Retrieve profile information

- Educational
- Employment i.e Professional
- Demographic data

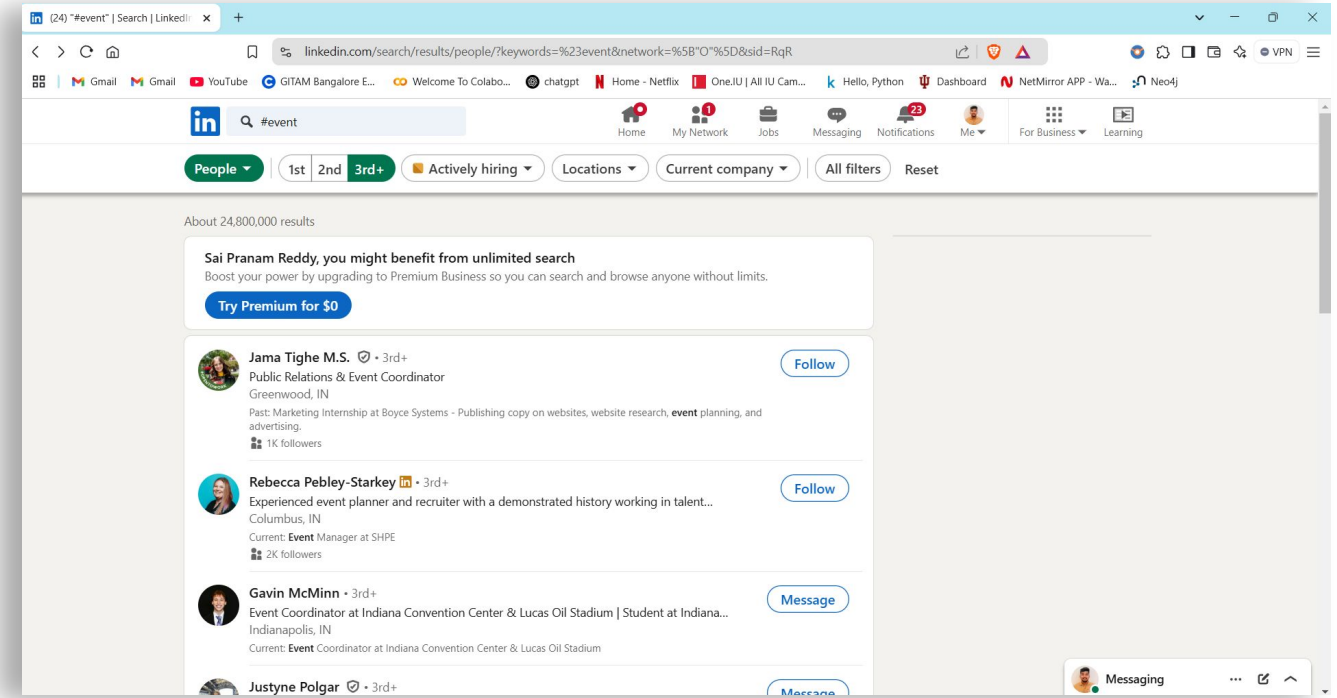
Key concerns

- Legal
- Privacy
- Linkedin Terms of Service

Methods explored

- Manual extraction - click through and copy information needed.
- Google Dorking - Targeted google searches with (“site:linkedin.com/in”) to gather public profile url and data.
- Click Bot - Using Selenium or Puppeteer.
  - User mimic
  - Headless
- Chrome Extension - Phantom Buster e.g.
- LinkedIN Sales Navigator + Manual / Click Bot
- LinkedIn official API
- Third Party APIs

Manual Extraction of LinkedIn Profiles With Tourism and Events

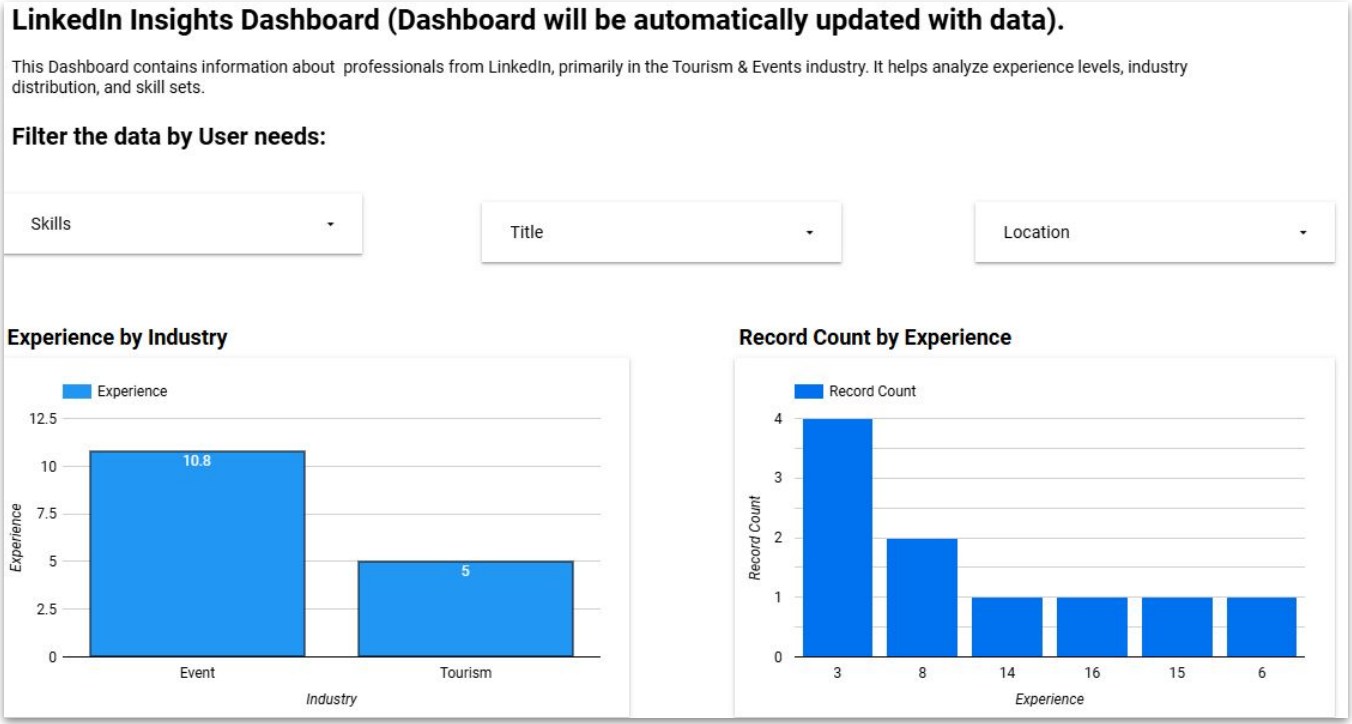


Data Collected

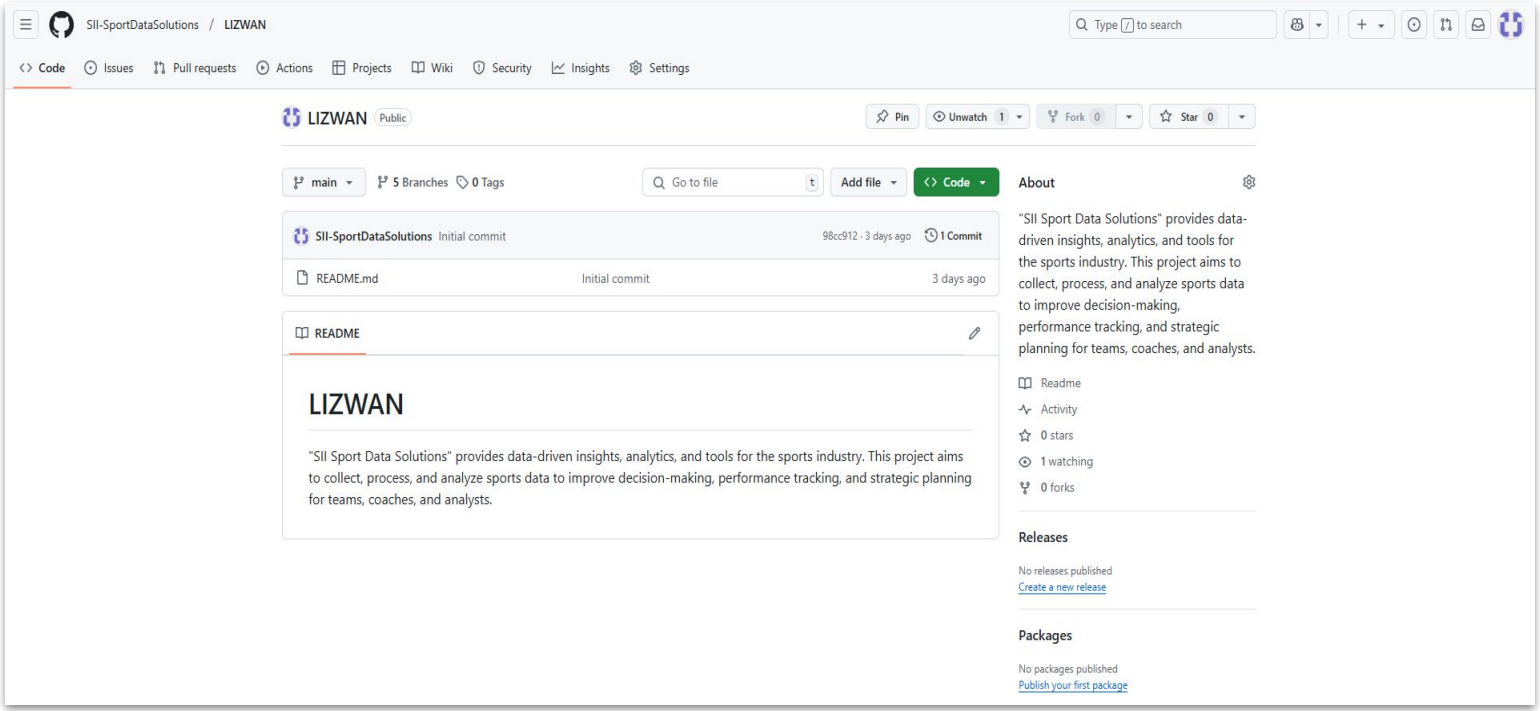
Dataset Columns & Their Indications

1. **Name** → Represents the individual's full name.
2. **Title** → Indicates the professional role or job position of the individual.
3. **Company** → Indicates the organization where the individual is currently employed.
4. **Location** → Shows the geographical area where the individual is based.
5. **Industry** → Specifies the field or sector in which the individual works.
6. **Skills** → Lists the key abilities and expertise the individual possesses.
7. **Experience (Years)** → Represents the total years of professional work experience.
8. **LinkedIn URL** → Provides a direct link to the individual's LinkedIn profile for reference.

Dashboard


LinkedIn Insights Dashboard (Dashboard will be automatically updated with data).
This Dashboard contains information about professionals from LinkedIn, primarily in the Tourism & Events industry. It helps analyze experience levels, industry distribution, and skill sets.
Filter the data by User needs:
Skills
Title
Location
Experience by Industry
Experience
12.5
10
7.5
5
2.5
0
10.8
5
Event
Tourism
Industry
Record Count by Experience
Record Count
4
3
2
1
0
3
8
14
16
15
6
Experience

Github Page For SII



Databook

Datacreated: Link

Dashboard:Link

GITHUB PAGE: LINK

Google Dorking

Many LinkedIn profiles are **publicly indexed by Google**, you can **use Google search queries (Google Dorks) to find profiles**.

Example query: site:linkedin.com/in/ "Sports" "San Francisco"

site:linkedin.com/in/ "Sports" "San Francisco"

All Images News Videos Maps Shopping Forums More Tools

LinkedIn · Becky Biniek
1.5K+ followers

Becky Biniek - San Francisco, California, United States

Sports Partnerships & Marketing Leader | Golden State Valkyries, Meta & San Francisco Giants Alum · Location: San Francisco · 500+ connections on LinkedIn.

LinkedIn · Tracy Hughes
1K+ followers

Tracy Hughes - Silicon Valley Sports Ventures

Menlo Park, California, United States · Silicon Valley Sports Ventures

Silicon Valley Sports Ventures has partnered with NFL Hall of Fame Quarterback and San Francisco 49er Legend Steve Young to accelerate SyncThink - the Palo Alto ...

LinkedIn · Bonta Hill
1.7K+ followers

Bonta Hill - NBC Sports Bay Area & California

San Francisco, California, United States · NBC Sports Bay Area & California

I'm a graduate of San Francisco... · Experience: NBC Sports Bay Area & California · Education:

Serp API

We can then extract **publicly visible** profile snippets using **SerpAPI** or manual techniques.

```
from serpapi import GoogleSearch

params = {
    "engine": "google",
    "q": 'site:linkedin.com/in/ "Sports" "San Francisco"',
    "api_key": "ad70ca8121a44b5fe4a714c5d02e792e4f648fa881d9b43b44994d382b1f9232",
    "num": 5
}

search = GoogleSearch(params)
results = search.get_dict()

profiles = []
for result in results.get("organic_results", []):
    profiles.append(result["link"])

print("Extracted LinkedIn Profiles:")
print(profiles)
```

```
Extracted LinkedIn Profiles:
['https://www.linkedin.com/in/beckybiniek', 'https://www.linkedin.com/in/vernon-vern-glenn-822335267', 'https:/
```

Results are not clear

```
Extracted LinkedIn Profiles:
['https://www.linkedin.com/in/beckybiniek', 'https://www.linkedin.com/in/vernon-vern-glenn-822335267', 'https://www.linke
```

It only gives links that are also visible when done with google dorks.

Converting it into csv

```
import re
import csv
from serpapi import GoogleSearch


SERPAPI_KEY = "ad70ca8121a44b5fe4a714c5d02e792e4f648fa881d9b43b44994d382b1f9232"
query = 'site:linkedin.com/in/ "Sports" "San Francisco" "studied at"'

params = {
    "engine": "google",
    "q": query,
    "api_key": SERPAPI_KEY,
    "num": 10
}


search = GoogleSearch(params)
results = search.get_dict()

# CSV File
csv_filename = "linkedin_profiles.csv"
with open(csv_filename, "w", newline="", encoding="utf-8") as file:
    writer = csv.writer(file)
    writer.writerow(["Name & Job Title", "City", "Work Experience", "Education", "LinkedIn URL"])


    for result in results.get("organic_results", []):
        title = result.get("title", "")  # Extract Name & Job Title
        snippet = result.get("snippet", "")  # Extract Full Snippet Text
        link = result.get("link", "")  # Extract LinkedIn Profile URL


        city_match = re.search(r'in (.+?)(\.|$)', snippet)
        city = city_match.group(1) if city_match else "N/A"

        work_match = re.search(r'worked at (.+?)(\.|$)', snippet)
        work_experience = work_match.group(1) if work_match else "N/A"
```

Detailed Results

linkedin_profiles.csv

1 to 10 of 10 entries Filter

Name & Job Title	City	Work Experience	Education	LinkedIn URL
Robert Juarez - Sales Consultant - Sunglass Hut	Delta College in 2017, and have studied at	N/A	.	https://www.linkedin.com/in/robert-juarez-95128812a
Matt Connolly - ONE Championship	N/A	N/A	N/A	https://www.linkedin.com/in/mattdconnolly
Nona Rhaburn, M.S. - Game Day Assistant - Los Angeles ...	Sports Management	N/A	the University of San Francisco and received my M	https://www.linkedin.com/in/nona-rhaburn
Marshall Payne - Sacramento Kings	Barcelona, Spain for a semester during junior year	N/A	N/A	https://www.linkedin.com/in/marshallpayne
Pablo Sandoval - San Francisco Bay Area	N/A	N/A	N/A	https://www.linkedin.com/in/sandovalpablo
Ameera T. - San Francisco, California, United States	N/A	N/A	N/A	https://www.linkedin.com/in/ameeranteal
Chuck Jacobs - Football Player - Baltimore Ravens	N/A	N/A	Utah State where I graduated with a Bachelors of Science	https://www.linkedin.com/in/chuck-jacobs-1ba880b6
Marlen Alvarado - NBC Sports Bay Area & California	the industry	N/A	N/A	https://www.linkedin.com/in/marlenalvarado
Gina Izerel - Graphics/Realtime Artist - FOX Sports	Game Art and Animation	N/A	Northeastern University with a major in Game Art and Animation	https://www.linkedin.com/in/ginaizerel
Tameem Tutakhil - Clinician Worker 1 - Telecare Corporation	my time there	N/A	City College of San Francisco (CCSF) for 5 years	https://www.linkedin.com/in/tameem-tutakhil-b1120a158

Show 10 per page

But not as detailed as we need them

This method only gives us some parameters. For the full profile, we will need to login with linkedin account.

But there's another constraint when we login. If we do manual scrapping we are against time.

To speed up the process, we can explore automation: Method 2: LinkedIn Scraper + Selenium

If we have LinkedIn login access, we can use Selenium to simulate a browser, log in, and extract full profiles.

Risk

Using a Selenium Bot goes against Linkedin ToS. If found, the linkedin account may lead to a ban.

Workaround? Use a alternate account?

Drawbacks, multiple accounts might be needed as they might blocked one by one each session.

Creative Workarounds

Parse Resumes for Data: **Pyresparser**

Manually parse resumes for the information we are looking for.

Cons: We need to have access to a large number of resumes for this.

Manually parsing

Websites that have resumes

Platform	Access Type	Notes
Indeed (indeed.com/resumes)	Requires login	Search by job title, location
Monster (monster.com)	Paid	Resume access for recruiters
LinkedIn Jobs (linkedin.com/jobs)	Manual	Some profiles allow resume downloads
GitHub (Developers) (github.com)	Free	Search for resumes in repos
Google Drive/Docs	Public	Google Dorks to find uploaded resumes

Workaround for resumes

Solution: Modify the Google Dork Automation to search profiles with a PDF attachment named something similar to resume/CV.

site:docs.google.com "resume" "linkedin.com/in" filetype:pdf

site:github.com "resume" "CV" "pdf"

site:docs.google.com ("resume" OR "cv") ("linkedin.com/in") ("Sports" OR "Fitness) ("San Francisco" OR "Indianapolis") filetype:pdf

Again, Ethical Concerns

These queries search the profiles that have these in their profile but wont allow access to the pdf resume files unless you are logged in.

Hence getting resumes in possible **but by hand.**

Automation of this is not possible as it will lead to breaching Linkedin's ToS.

ClickBot - Using Selenium

What is it ?

Programmatically log in (or reuse your session cookies) with tools like Python’s Requests combined with BeautifulSoup or Scrapy to download the HTML of profile pages.

ClickBot - Using Selenium

Challenges:

LinkedIn’s heavy anti-bot measures (dynamic content, login requirements, and rate limiting) make this approach challenging without advanced techniques.

ClickBot - Using Selenium - User mimic

Advantages

- Mimics Human Behavior: Allows for random delays, scrolling, and clicking to evade detection.
- Faster than manual but slower than headless
- Better control of data extracted

Disadvantages

- Privacy - Data needs to managed responsibility
- Breaches LinkedIn TOS
- New accounts need to be created just for getting the data

ClickBot - Using Selenium - Headless

Advantages

- Gets all the information required.
- Fastest way

Disadvantages

- Privacy - Data needs to managed responsibility
- Breaches LinkedIn TOS
- New accounts need to be created just for getting the data
- High chance of getting detected and account getting banned.

Code Demo - Example data extracted

Linkedin-extraction-bs4.docx

Browser Extensions for Scraping

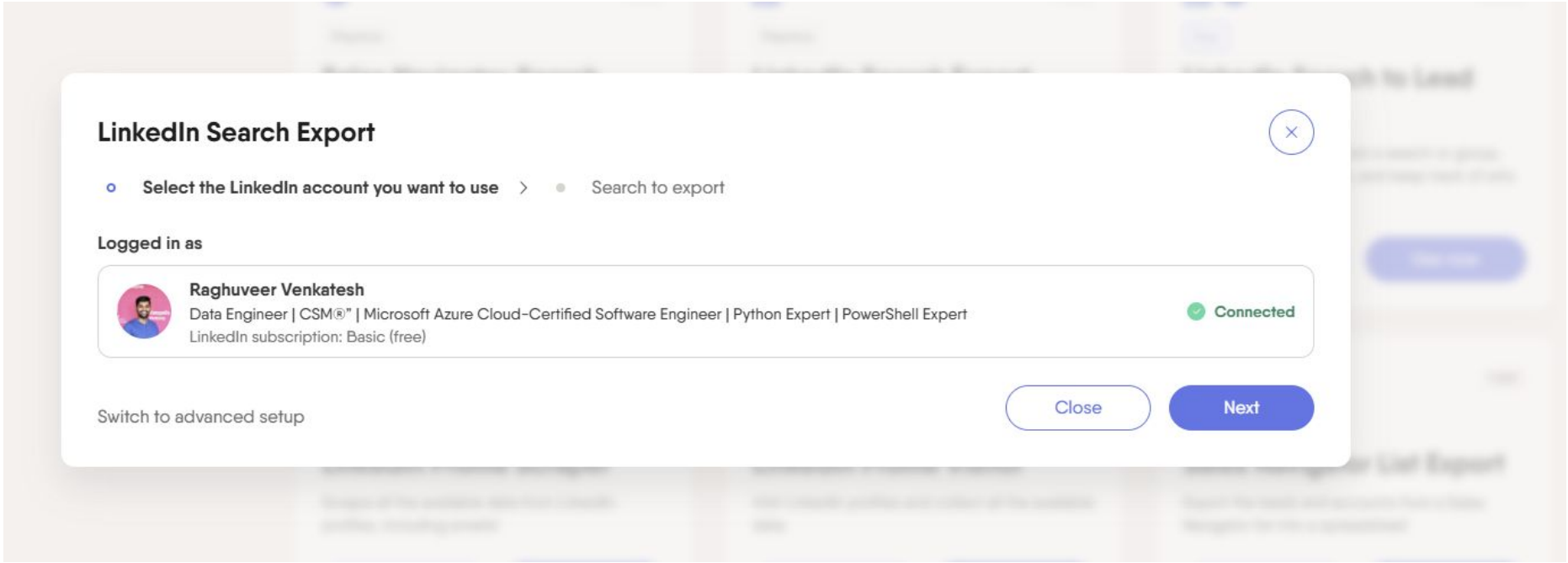
(e.g. Phantom Buster)

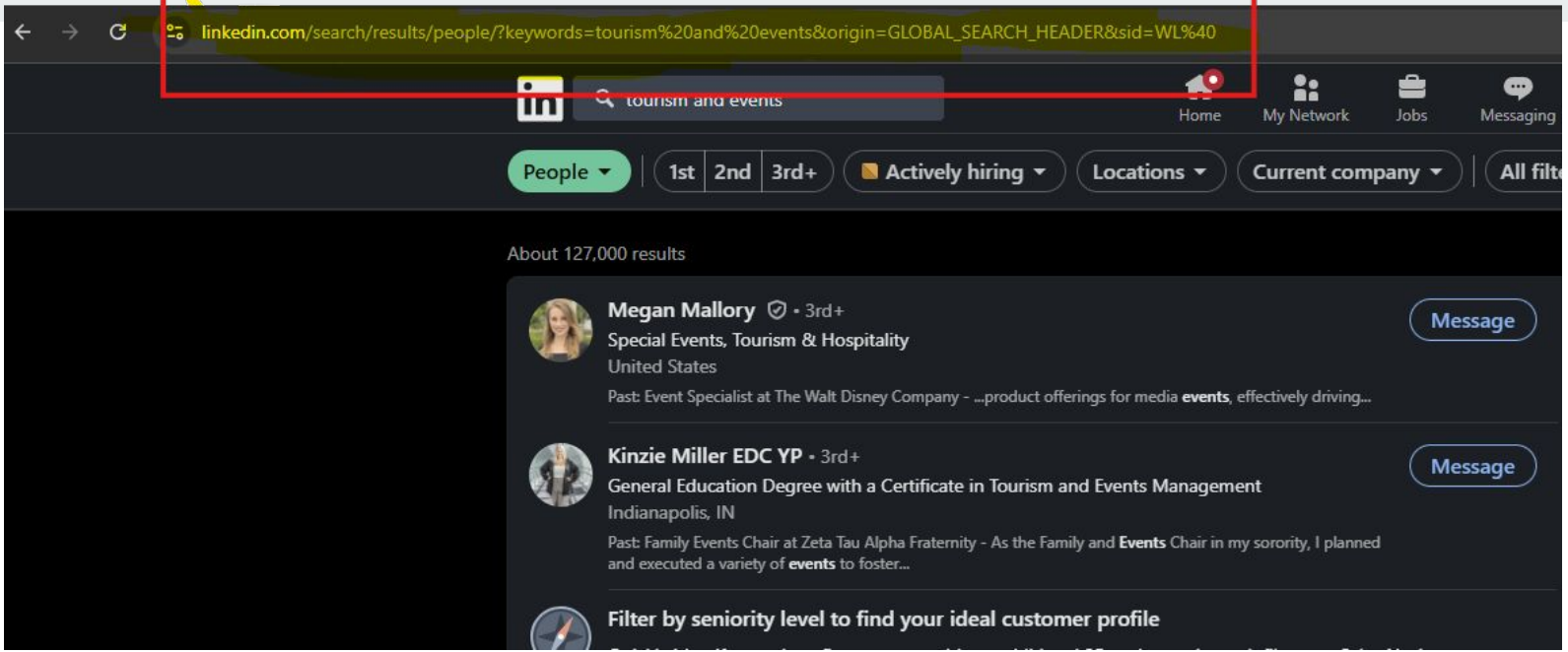
Advantages

- No code option - Easy to use

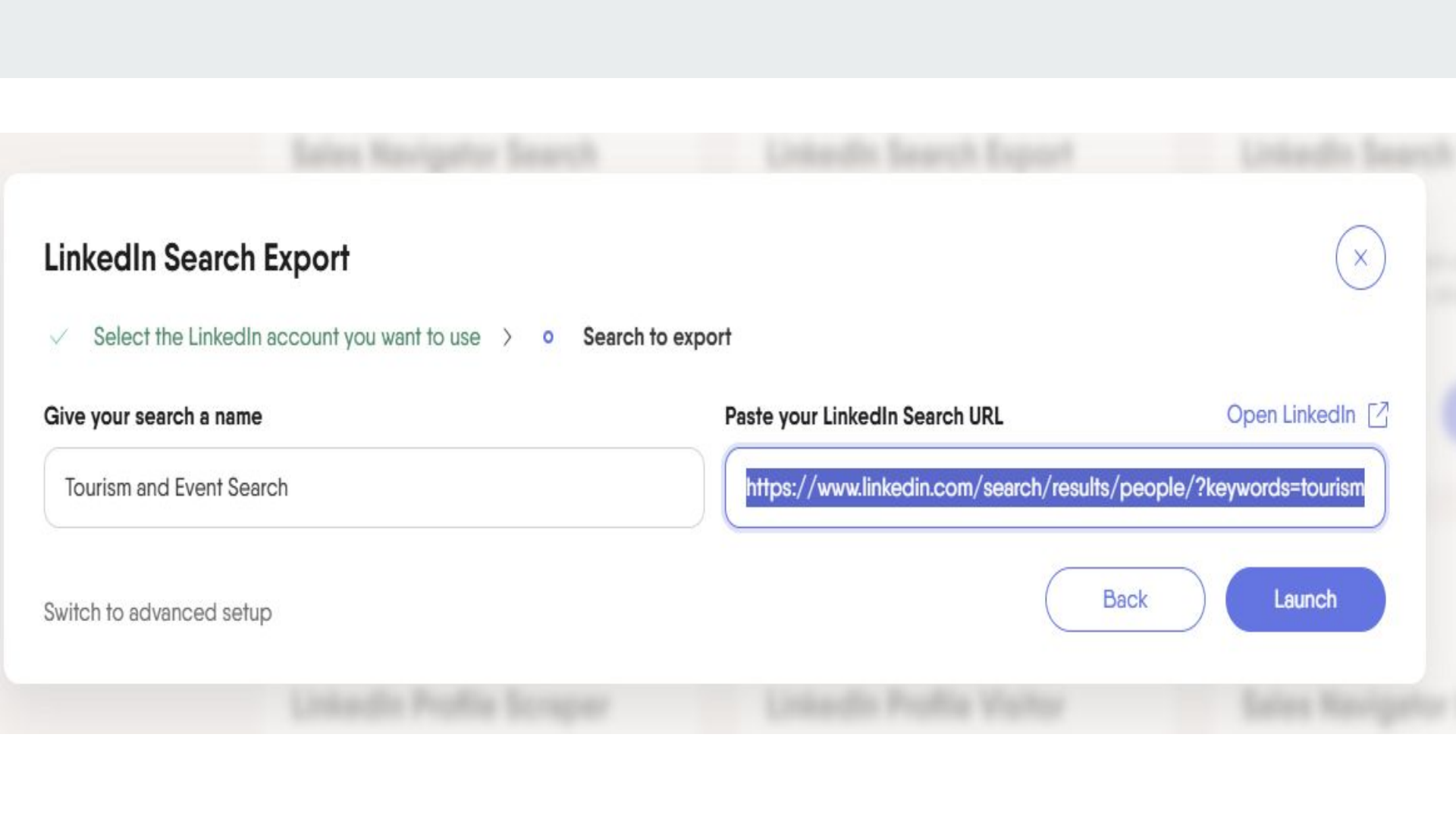
Disadvantages

- No flexibility to extract desired fields.
- Account at risk so alternate accounts need to be created.
- Violates TOS and privacy concerns when extracting bulk data.
- Limited information available without paying


LinkedIn Search Export
Select the LinkedIn account you want to use
Search to export
Logged in as
Raghuveer Venkatesh
Data Engineer | CSM®" | Microsoft Azure Cloud-Certified Software Engineer | Python Expert | PowerShell Expert
LinkedIn subscription: Basic (free)
Connected
Switch to advanced setup
Close
Next



https://www.linkedin.com/search/results/people/?keywords=tourism%20and%20events&origin=SWITCH_SEARCH_VERTICAL&sid=s36


LinkedIn Search Export
Select the LinkedIn account you want to use
Search to export
Give your search a name
Tourism and Event Search
Paste your LinkedIn Search URL
Open LinkedIn
https://www.linkedin.com/search/results/people/?keywords=tourism
Switch to advanced setup
Back
Launch

How the data looks

Linkedin_Data_Browser Extension.csv

LinkedIn API

- Linkedin People Profile API
- 3rd Party API
- Case Opened - Case details | LinkedIn Help

People API

- Custom pricing (reported starting at 59$ - 2999$/month)
- Full profile data: Work history, education, skills, recommendations, and contact info.
- Higher rate limits (e.g., 15,000 requests/day under the Professional plan)
- LinkedIn Partnership: Requires approval via LinkedIn's Partner Program, which involves a lengthy application process

Is Indiana University a LinkedIn partner ?

Sales Navigator - Marketing API - Not very Useful! I think

Mostly for sales prospecting and B2B outreach

Pricing: Dynamic based on usage

3rd Party API services

Provider	Pricing	Features	Limitations
Proxycurl	$10/one-time	Fresh profile data (≤29 days old), no partnership required	Limited to public profiles
Unipile	€5/account/month	Multi-platform integration (LinkedIn + Gmail/WhatsApp)	Requires account linking
PhantomBuster	$56/month	Automation of LinkedIn actions (e.g., scraping)	Risk of account bans
People Data Labs	$98/month	Bulk historical data	Outdated information