

To create a job search engine for Business Analytics positions in India using Python, you can leverage web scraping techniques along with APIs or job portals like Naukri, LinkedIn, or Indeed. Here's a simple example using BeautifulSoup and requests to scrape job listings from a job portal.

Step-by-Step Process:

Install Required Libraries: You will need requests for sending HTTP requests and BeautifulSoup for scraping job details from web pages.

```
In [1]: pip install requests beautifulsoup4 pandas
```

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in c:\users\purnangshu roy\appdata\roaming\python\python311\site-packages (2.32.3)
Requirement already satisfied: beautifulsoup4 in c:\programdata\anaconda3\lib\site-packages (4.12.2)
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages (1.5.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests) (2023.7.22)
Requirement already satisfied: soupsieve>1.2 in c:\programdata\anaconda3\lib\site-packages (from beautifulsoup4) (2.4)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2022.7)
Requirement already satisfied: numpy>=1.21.0 in c:\programdata\anaconda3\lib\site-packages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

Example Code for Scraping: Here's a real-time example of scraping job postings from a job portal (like Naukri.com or Indeed). Example Code for Scraping: Here's a real-time example of scraping job postings from a job portal (like Naukri.com or Indeed).

```
In [2]: import requests
from bs4 import BeautifulSoup
import pandas as pd

def extract_job_details(job):
    title = job.find('a', class_='title').text.strip()
    company = job.find('a', class_='company').text.strip()
    location = job.find('span', class_='location').text.strip()
    summary = job.find('div', class_='summary').text.strip()
    return {'Title': title, 'Company': company, 'Location': location, 'Summary': summary}
```

```
def scrape_jobs(keyword, location, num_pages=1):
    jobs = []
    base_url = f'https://www.indeed.com/jobs?q={keyword}&l={location}'
    headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"}

    for page in range(num_pages):
        response = requests.get(base_url + f'&start={page*10}', headers=headers)
        soup = BeautifulSoup(response.content, 'html.parser')
        job_cards = soup.find_all('div', class_='jobsearch-SerpJobCard')

        for job in job_cards:
            job_data = extract_job_details(job)
            jobs.append(job_data)

    return jobs

def save_to_csv(jobs, filename='jobs.csv'):
    df = pd.DataFrame(jobs)
    df.to_csv(filename, index=False)
    print(f'Saved {len(jobs)} jobs to {filename}')

if __name__ == "__main__":
    # Scraping jobs for Business Analytics in India
    job_listings = scrape_jobs('Business Analytics', 'India', num_pages=2)
    save_to_csv(job_listings)
```

Saved 0 jobs to jobs.csv

Explanation:Explanation: Step 1: We define a function extract_job_details to extract relevant details (title, company, location, and summary) from a job card. Step 1: We define a function extract_job_details to extract relevant details (title, company, location, and summary) from a job card. Step 2: The function scrape_jobs accepts a job keyword (e.g., "Business Analytics") and location (e.g., "India"). It sends an HTTP request to Indeed or a similar job portal and parses the response HTML using BeautifulSoup. Step 2: The function scrape_jobs accepts a job keyword (e.g., "Business Analytics") and location (e.g., "India"). It sends an HTTP request to Indeed or a similar job portal and parses the response HTML using BeautifulSoup. Step 3: The results are stored in a list and saved to a CSV file using save_to_csv. Step 3: The results are stored in a list and saved to a CSV file using save_to_csv. Output:Output: The code will create a CSV file (jobs.csv) containing job titles, companies, locations, and summaries of jobs related to Business Analytics in India. The code will create a CSV file (jobs.csv) containing job titles, companies, locations, and summaries of jobs related to Business Analytics in India. Future Enhancements: Future Enhancements: Use APIs like LinkedIn API for fetching more structured job data. Use APIs like LinkedIn API for fetching more structured job data. Add features like filtering based on salary, experience level, or job type.

Add features like filtering based on salary, experience level, or job type. Make sure to check the terms and conditions of the website you are scraping to ensure you are compliant. Make sure to check the terms and conditions of the website you are scraping to ensure you are compliant.

In []: