PROJECT TITLE

Automated Job Vacancy Scraper and Email Notification System

Sachin SIngh

28th April, 2025

Project Title: Automated Job Vacancy Scraper and Email Notification System

🎯 Objective:

Automatically scrape a job vacancy website, detect new vacancies, and send email alerts daily, scheduled via a cron job on a Linux machine.

Tools Used in the Project:

1. **Python Programming Language**
    - **Purpose:** Writing the core logic for scraping, data processing, and email automation.
2. **Python Libraries:**
    - `requests`
        - **Purpose:** Fetching HTML content from job websites via HTTP requests.
    - `BeautifulSoup` **(from** `bs4`**)**
        - **Purpose:** Parsing and extracting specific data from the HTML content.
    - `pandas`
        - **Purpose:** Handling job data and comparing listings using Excel files.
    - `smtplib`
        - **Purpose:** Sending email notifications via Gmail SMTP server.

- `email.mime`
    - **Purpose:** Creating email messages with attachments (Excel files).

3. **Excel (Spreadsheet)**
    - **Purpose:** Storing and comparing previous and current job listings.
4. **Gmail (SMTP Server)**
    - **Purpose:** Sending automated email alerts with job vacancies.
5. **Kali Linux (Operating System)**
    - **Purpose:** Developing and running the script, and scheduling it with **cron jobs**.
6. **Cron Jobs (Linux)**
    - **Purpose:** Scheduling the Python script to run daily at a specific time.
7. **Text Editor / IDE (Integrated Development Environment)**
    - **Purpose:** Writing and developing the Python code.
        - Examples: **Visual Studio Code**, **Sublime Text**, **PyCharm**.

🛠️ Steps Followed:

1. Web Scraping

- Used **Python** and **BeautifulSoup** to scrape job vacancy listings from the target website.
- Extracted vacancy titles, descriptions, and links.

```python
import requests
from bs4 import BeautifulSoup

url = 'https://example.com/jobs'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

jobs = soup.find_all('div', class_='job-post')
for job in jobs:
    title = job.find('h2').text
    link = job.find('a')['href']
    print(title, link)
```

2. Detecting New Vacancies

- Stored previous job posts in a **local JSON** or **text file**.
- Compared current vacancies with previous ones to find **newly published** jobs.

```python
import json

with open('previous_jobs.json', 'r') as f:
    old_jobs = json.load(f)

# Compare and find new ones
new_jobs = [job for job in scraped_jobs if job not in old_jobs]
```

3. Sending Email Alerts

- I used **smtplib** to email my personal Gmail account when new vacancies were found.

```python
import smtplib
from email.mime.text import MIMEText

sender = "your_email@gmail.com"
receiver = "your_email@gmail.com"
password = "your_app_password"

message = MIMEText('New job posted: ' + job_link)
message['Subject'] = 'New Job Alert'
message['From'] = sender
message['To'] = receiver

with smtplib.SMTP_SSL('smtp.gmail.com', 465) as server:
    server.login(sender, password)
    server.sendmail(sender, receiver, message.as_string())
```

4. Scheduling with Cron

- Scheduled the Python script to run every day at 11:00 AM.

Crontab Setup:

```
crontab -e
```

Entry:

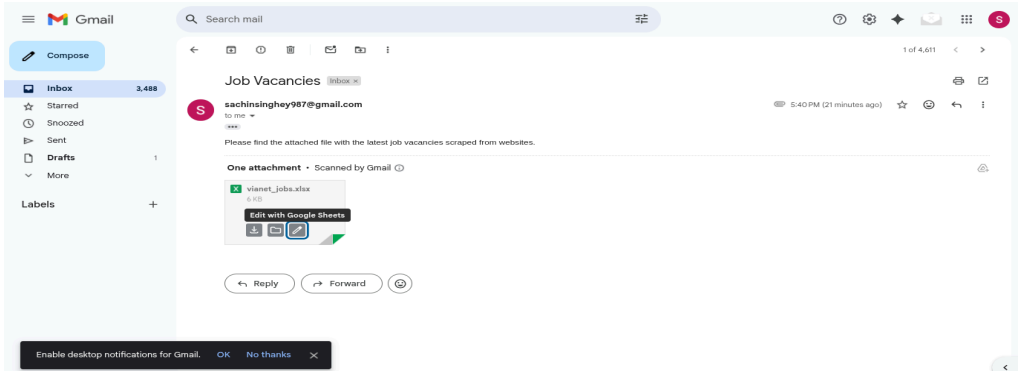
```
0 10 * * * /usr/bin/python3 /path/to/job_scraper.py
```

📸 Screenshot: Running Python Script Successfully

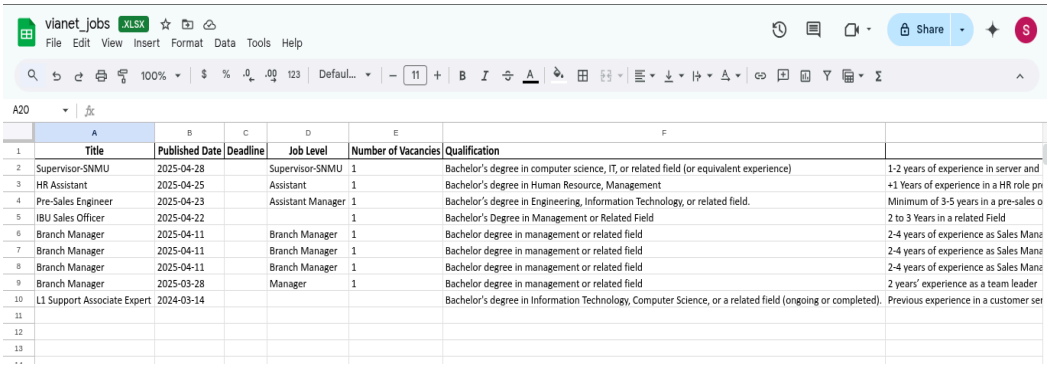
```
┌──(venv)─(blink㉿kali)-[~/Downloads/python]
└─$ python websites.py

Starting the script...
Attempting to scrape data...
Successfully fetched the page! Response code: 200
✅ Scraped 9 jobs and saved to 'vianet_jobs.xlsx'
Attempting to send email with subject: Vianet Job Vacancies
Attaching file: vianet_jobs.xlsx
Email sent to sachinsinghey987@gmail.com successfully!
Scraping and email sending completed!
Starting the script...
Attempting to scrape data...
Successfully fetched the page! Response code: 200
✅ Scraped 9 jobs and saved to 'vianet_jobs.xlsx'
Attempting to send email with subject: Vianet Job Vacancies
Attaching file: vianet_jobs.xlsx
Email sent to sachinsinghey987@gmail.com successfully!
```

5. Email Notification Example



6. Excel Sheet Screenshot



vianet_jobs .XLSX

File Edit View Insert Format Data Tools Help

Share

A20

	A	B	C	D	E	F	
1	Title	Published Date	Deadline	Job Level	Number of Vacancies	Qualification	
2	Supervisor-SNMU	2025-04-28		Supervisor-SNMU	1	Bachelor's degree in computer science, IT, or related field (or equivalent experience)	1-2 years of experience in server and
3	HR Assistant	2025-04-25		Assistant	1	Bachelor's degree in Human Resource, Management	+1 Years of experience in a HR role pr
4	Pre-Sales Engineer	2025-04-23		Assistant Manager	1	Bachelor's degree in Engineering, Information Technology, or related field.	Minimum of 3-5 years in a pre-sales o
5	IBU Sales Officer	2025-04-22			1	Bachelor's Degree in Management or Related Field	2 to 3 Years in a related Field
6	Branch Manager	2025-04-11		Branch Manager	1	Bachelor degree in management or related field	2-4 years of experience as Sales Mana
7	Branch Manager	2025-04-11		Branch Manager	1	Bachelor degree in management or related field	2-4 years of experience as Sales Mana
8	Branch Manager	2025-04-11		Branch Manager	1	Bachelor degree in management or related field	2-4 years of experience as Sales Mana
9	Branch Manager	2025-03-28		Manager	1	Bachelor degree in management or related field	2 years' experience as a team leader
10	L1 Support Associate Expert	2024-03-14				Bachelor's degree in Information Technology, Computer Science, or a related field (ongoing or completed).	Previous experience in a customer se
11							
12							
13							

***NOTE**: This project demonstrates a practical use case of web scraping combined with daily automation to enhance productivity and opportunity tracking.*

Signature:

Email: sachinsinghey987@gmail.com