Web Scraper for Job Listings

Overview:

Develop a web scraper that collects job listings from a popular job portal, such as Indeed or LinkedIn. The scraper will extract job titles, company names, locations, and summaries, then store this data in a CSV file.

Key Skills:

- Web scraping using BeautifulSoup and Requests
- Data storage using CSV or Pandas
- Exception handling

Code Snippet:

```
import requests
from bs4 import BeautifulSoup
import csv
import time

# Function to get job listings from a page
def get_job_listings(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    jobs = soup.find_all('div', class_='jobsearch-SerpJobCard')
    job_listings = []
    for job in jobs:
        title = job.find('h2', class_='title').text.strip()
```

```
        company = job.find('span', class_='company').text.strip()
        location = job.find('div', class_='location').text.strip()
        summary = job.find('div', class_='summary').text.strip()
        job_listings.append([title, company, location, summary])
    return job_listings

# Main script with pagination
def main():
    base_url = 'https://www.indeed.com/jobs?q=software+developer&l='
    page = 0
    all_jobs = []
    while True:
        url = base_url + '&start=' + str(page)
        jobs = get_job_listings(url)
        if not jobs:
            break
        all_jobs.extend(jobs)
        page += 10
        time.sleep(1)  # To avoid getting blocked

    # Save to CSV
    with open('job_listings.csv', 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(['Title', 'Company', 'Location', 'Summary'])
        writer.writerows(all_jobs)
```

```
    print("Job listings saved to job_listings.csv")


if __name__ == "__main__":
    main()
```