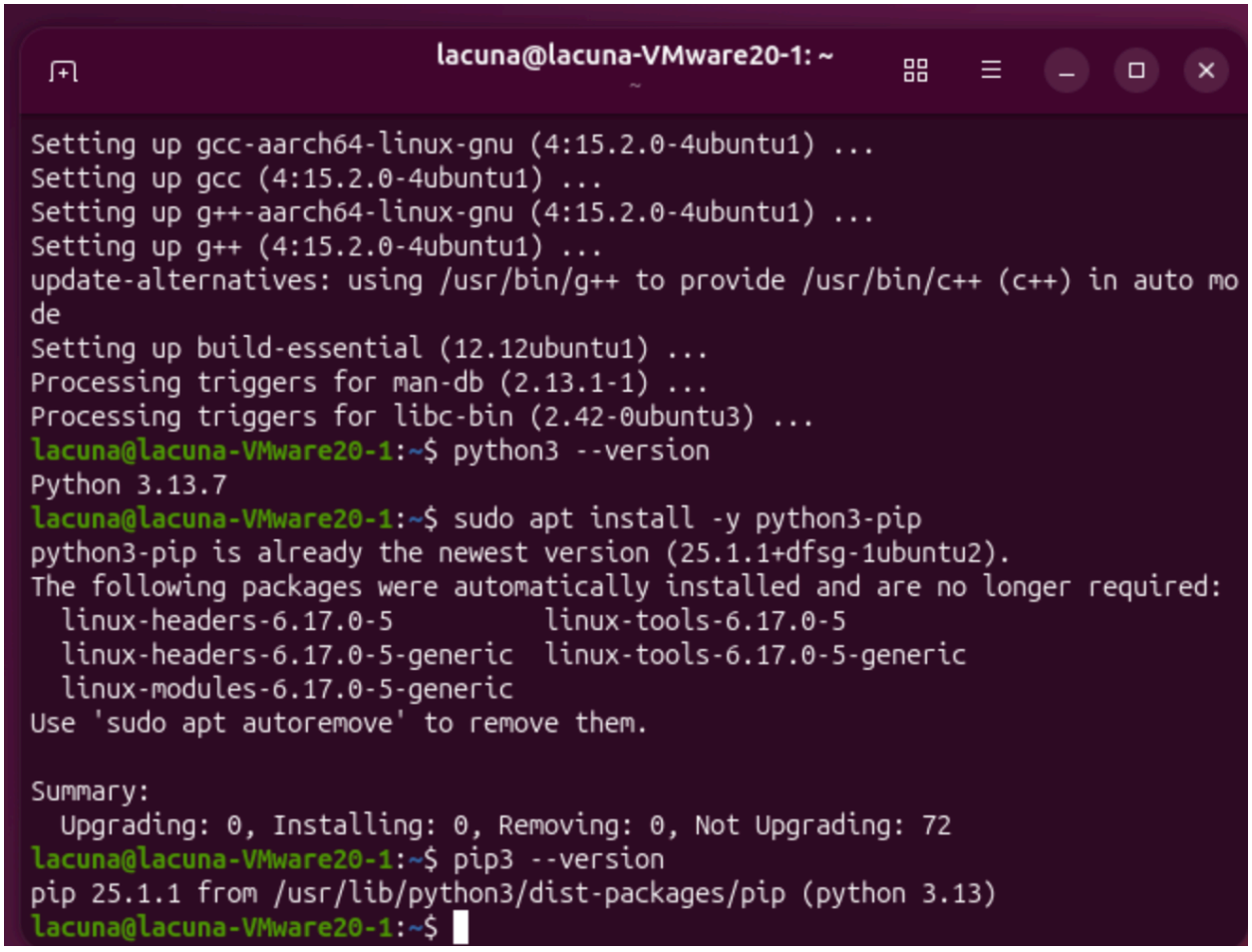
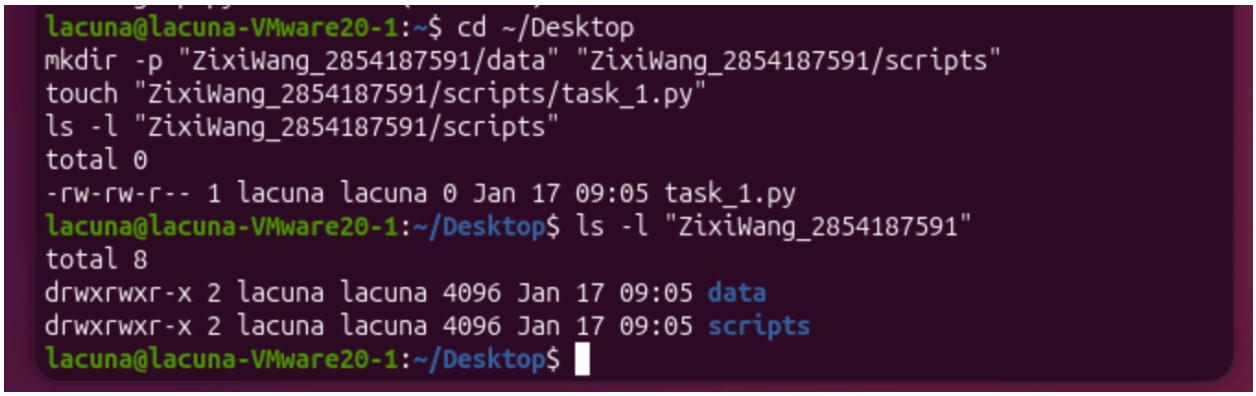
2.1. Playing around with Linux Terminal





It shows I downloaded everything. I have folder data and scripts, and I created task_1.py

2.2. A basic Python Script

```
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ nano task_1.py
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ nano scripts/task_1.py
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ python3 scripts/task_1.py
Enter your name: Zixi
Hello, Zixi!
```

lacuna@lacuna-VMware20-1: ~/Desktop/ZixiWang_2854187591 — nano scripts/task_1.py
~/Desktop/ZixiWang_2854187591

```
 GNU nano 8.4                                    scripts/task_1.py
name = input("Enter your name: ")
print(f"Hello, {name}!")
```

I have task_1.py and can output Hello, [name]!

2.3. Python Web-scraping Task

```
lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ cd ~/Desktop/ZixiWang_2854187591
python3 -m venv .venv
lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ source .venv/bin/activate
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ pip install requests beautifulsoup4
Collecting requests
  Downloading requests-2.32.5-py3-none-any.whl.metadata (4.9 kB)
Collecting beautifulsoup4
  Downloading beautifulsoup4-4.14.3-py3-none-any.whl.metadata (3.8 kB)
Collecting charset_normalizer<4,>=2 (from requests)
  Downloading charset_normalizer-3.4.4-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl.metadata (37 kB)
Collecting idna<4,>=2.5 (from requests)
  Downloading idna-3.11-py3-none-any.whl.metadata (8.4 kB)
Collecting urllib3<3,>=1.21.1 (from requests)
  Downloading urllib3-2.6.3-py3-none-any.whl.metadata (6.9 kB)
Collecting certifi>=2017.4.17 (from requests)
  Downloading certifi-2026.1.4-py3-none-any.whl.metadata (2.5 kB)
Collecting soupsieve>=1.6.1 (from beautifulsoup4)
  Downloading soupsieve-2.8.1-py3-none-any.whl.metadata (4.6 kB)
Collecting typing-extensions>=4.0.0 (from beautifulsoup4)
  Downloading typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
Downloading requests-2.32.5-py3-none-any.whl (64 kB)
Downloading charset_normalizer-3.4.4-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl (147 kB)
Downloading idna-3.11-py3-none-any.whl (71 kB)
Downloading urllib3-2.6.3-py3-none-any.whl (131 kB)
Downloading beautifulsoup4-4.14.3-py3-none-any.whl (107 kB)
Downloading certifi-2026.1.4-py3-none-any.whl (152 kB)
Downloading soupsieve-2.8.1-py3-none-any.whl (36 kB)
Downloading typing_extensions-4.15.0-py3-none-any.whl (44 kB)
Installing collected packages: urllib3, typing-extensions, soupsieve, idna, charset_normalizer, certifi, requests, beautifulsoup4
Successfully installed beautifulsoup4-4.14.3 certifi-2026.1.4 charset_normalizer-3.4.4 idna-3.11 requests-2.32.5 soupsieve-2.8.1 typing-extensi
5.0 urllib3-2.6.3
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ python -c "import requests; import bs4; print('ok')"
ok
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$
```

```
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ ls
data  lab1_zixiwang_2854187591.zip  scripts
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ ls data
processed_data  raw_data
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$
```

```
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ nano scripts/web_scraper.py
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ python3 scripts/web_scraper.py
head -n 10 data/raw_data/web_data.html
grep -n "MarketCard-container" -m 2 data/raw_data/web_data.html
grep -n "LatestNews-list" -m 1 data/raw_data/web_data.html
Fetching: https://www.cnbc.com/world/?region=world
Saved: data/raw_data/web_data.html
<div class="MarketsBanner-marketData">
 <a class="MarketCard-container" href="#">
  <div class="MarketCard-row">
   <span class="MarketCard-symbol">
    .DJI
   </span>
  </div>
  <div class="MarketCard-row">
   <span class="MarketCard-stockPosition">
    49359.33
2: <a class="MarketCard-container" href="#">
19: <a class="MarketCard-container" href="#">
72:<ul class="LatestNews-list">
```

Jan 17 22:12

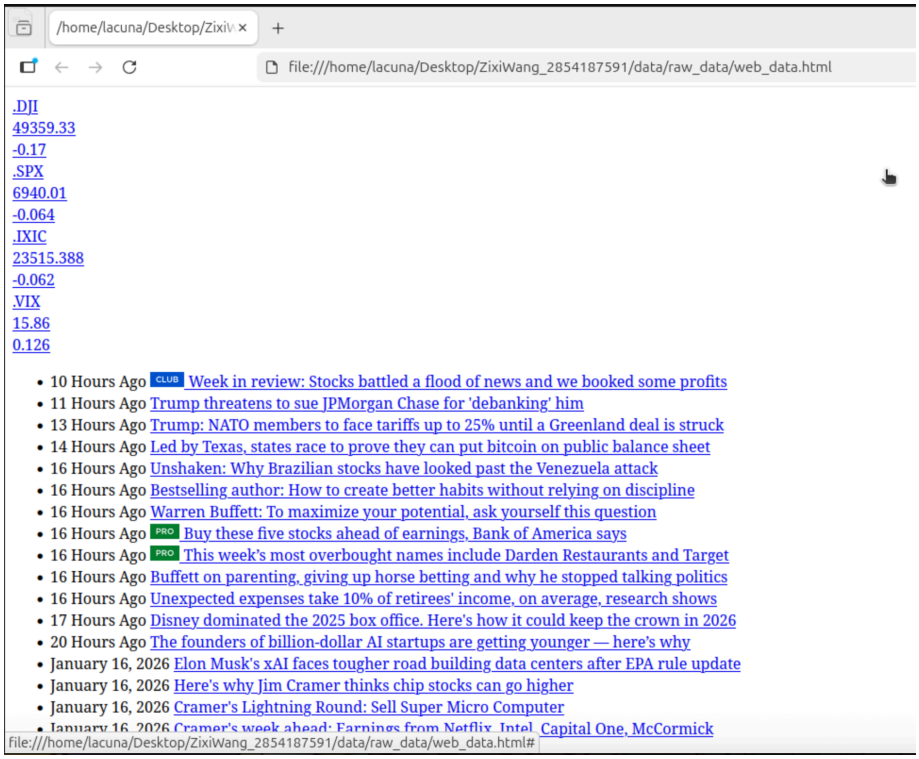
lacuna@lacuna-VMware20-1: ~/Desktop/ZixiWang_2854187591 — nano scripts/web_scraper.py

~/Desktop/ZixiWang_2854187591

GNU nano 8.4 scripts/web_scraper.py

```python
import os
import re
import json
import requests
from bs4 import BeautifulSoup

URL = "https://www.cnbc.com/world/?region=world"
OUT = "data/raw_data/web_data.html"
SYMBOLS = [".DJI", ".SPX", ".IXIC", ".VIX"]

def get_page():
    headers = {"User-Agent": "Mozilla/5.0"}
    r = requests.get(URL, headers=headers, timeout=20)
    r.raise_for_status()
    return r.text

def get_quotes(symbols):
    joined = "|".join(symbols)
    api = f"https://quote.cnbc.com/quote-html-webservice/quote.htm?symbols={joined}&output=json"
    r = requests.get(api, headers={"User-Agent": "Mozilla/5.0"}, timeout=20)
    r.raise_for_status()
    txt = r.text.strip()

    try:
        return json.loads(txt)
    except Exception:
        m = re.search(r"(\{.*\})", txt, flags=re.DOTALL)
        return json.loads(m.group(1)) if m else {}

def build_market_html(quotes_json):
    soup = BeautifulSoup("", "html.parser")
    banner = soup.new_tag("div", **{"class": "MarketsBanner-marketData"})

    q = quotes_json.get("QuickQuoteResult", {}).get("QuickQuote", [])
    mp = {}
    if isinstance(q, list):
        for it in q:
            sym = str(it.get("symbol", "")).strip()
            mp[sym] = (it.get("last", ""), it.get("change_pct", ""))

    for sym in SYMBOLS:
        last, chg = mp.get(sym, ("", ""))
        a = soup.new_tag("a", **{"class": "MarketCard-container", "href": "#"})
        a.append(soup.new_tag("div", **{"class": "MarketCard-row"}))
        a.contents[-1].append(soup.new_tag("span", **{"class": "MarketCard-symbol"}))
        a.contents[-1].contents[-1].string = sym

        a.append(soup.new_tag("div", **{"class": "MarketCard-row"}))
        a.contents[-1].append(soup.new_tag("span", **{"class": "MarketCard-stockPosition"}))
        a.contents[-1].contents[-1].string = str(last)

        a.append(soup.new_tag("div", **{"class": "MarketCard-row"}))
```

```
    for sym in SYMBOLS:
        last, chg = mp.get(sym, ("", ""))
        a = soup.new_tag("a", **{"class": "MarketCard-container", "href": "#"})
        a.append(soup.new_tag("div", **{"class": "MarketCard-row"}))
        a.contents[-1].append(soup.new_tag("span", **{"class": "MarketCard-symbol"}))
        a.contents[-1].contents[-1].string = sym

        a.append(soup.new_tag("div", **{"class": "MarketCard-row"}))
        a.contents[-1].append(soup.new_tag("span", **{"class": "MarketCard-stockPosition"}))
        a.contents[-1].contents[-1].string = str(last)

        a.append(soup.new_tag("div", **{"class": "MarketCard-row"}))
        a.contents[-1].append(soup.new_tag("span", **{"class": "MarketCard-changePct"}))
        a.contents[-1].contents[-1].string = str(chg)

        banner.append(a)

    return banner

def main():
    os.makedirs("data/raw_data", exist_ok=True)
    print("Fetching:", URL)

    html = get_page()
    soup = BeautifulSoup(html, "html.parser")
    latest_news = soup.find("ul", class_="LatestNews-list")

    quotes = get_quotes(SYMBOLS)
    market = build_market_html(quotes)

    with open(OUT, "w", encoding="utf-8") as f:
        f.write(market.prettify() + "\n")
        if latest_news:
            f.write(latest_news.prettify() + "\n")

    print("Saved:", OUT)

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



I created scripts/web_scraper.py, installed the required libraries (pip install requests beautifulsoup4), inspected the CNBC World page (https://www.cnbc.com/world/?region=world) to identify the Market banner and the “Latest News” section tags, and created the folders data/raw_data and data/processed_data. Then I wrote a script using Requests + BeautifulSoup to collect the page content and save the relevant HTML to data/raw_data/web_data.html. Finally, I verified the result in the terminal by printing the first 10 lines of the file (head -n 10 data/raw_data/web_data.html) and checking for key elements using grep (e.g., MarketCard-container and LatestNews-list).

2.4. Data Filtering Task

```
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ nano scripts/data_filter.py
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ python3 scripts/data_filter.py
head -n 5 data/processed_data/market_data.csv
head -n 5 data/processed_data/news_data.csv
Filtering fields...
Storing Market data...
CSV created: data/processed_data/market_data.csv (rows=4)
Storing News data...
CSV created: data/processed_data/news_data.csv (rows=30)
Done.
marketCard_symbol,marketCard_stockPosition,marketCardchangePct
.DJI,49359.33,-0.17%
.SPX,6940.01,-0.064%
.IXIC,23515.388,-0.062%
.VIX,15.86,0.126%
LatestNews_timestamp,title,link
10 Hours Ago,Week in review: Stocks battled a flood of news and we booked some profits,https://www.cnbc.com/2026/01/17/week-in-review-stocks-battled-a-flood-of-news-and-we
-booked-some-profits.html
11 Hours Ago,Trump threatens to sue JPMorgan Chase for 'debanking' him,https://www.cnbc.com/2026/01/17/trump-jpmorgan-chase-debanking.html
13 Hours Ago,Trump: NATO members to face tariffs up to 25% until a Greenland deal is struck,https://www.cnbc.com/2026/01/17/trump-greenland-tariffs-nato.html
14 Hours Ago,"Led by Texas, states race to prove they can put bitcoin on public balance sheet",https://www.cnbc.com/2026/01/17/texas-us-states-budgets-bitcoin-crypto-strat
egic-reserve.html
(.venv) lacuna@lacuna-VMware20-1:~/Desktop/ZixiWang_2854187591$ python3 scripts/data_filter.py
head -n 2 data/processed_data/market_data.csv
Filtering fields...
Storing Market data...
CSV created: data/processed_data/market_data.csv (rows=4)
Storing News data...
CSV created: data/processed_data/news_data.csv (rows=30)
Done.
marketCard_symbol,marketCard_stockPosition,marketCardchangePct
.DJI,49359.33,-0.17%
```

lacuna@lacuna-VMware20-1: ~/Desktop/ZixiWang_2854187591 — nano scripts/data_filter.py

~/Desktop/ZixiWang_2854187591

GNU nano 8.4 scripts/data_filter.py

```
#!/usr/bin/env python3
import os
import csv
from bs4 import BeautifulSoup

IN_FILE = "data/raw_data/web_data.html"
MARKET_CSV = "data/processed_data/market_data.csv"
NEWS_CSV = "data/processed_data/news_data.csv"

os.makedirs("data/processed_data", exist_ok=True)

print("Filtering fields...")

with open(IN_FILE, "r", encoding="utf-8", errors="ignore") as f:
    soup = BeautifulSoup(f.read(), "html.parser")

market = []
for card in soup.select("a.MarketCard-container"):
    sym = card.select_one("span.MarketCard-symbol")
    pos = card.select_one("span.MarketCard-stockPosition")
    pct = (card.select_one("span.MarketCard-changesPct")
           or card.select_one("span.MarketCard-changePct"))

    market.append([
        sym.get_text(strip=True) if sym else "",
        pos.get_text(strip=True) if pos else "",
        pct.get_text(strip=True) if pct else "",
    ])

print("Storing Market data...")
with open(MARKET_CSV, "w", newline="", encoding="utf-8") as f:
    w = csv.writer(f)
    w.writerow(["marketCard_symbol", "marketCard_stockPosition", "marketCardchangePct"])
    w.writerows(market)
print(f"CSV created: {MARKET_CSV} (rows={len(market)})")

news = []
for item in soup.select("ul.LatestNews-list > li.LatestNews-item"):
    t = item.select_one("time.LatestNews-timestamp")
    a = item.select_one("a.LatestNews-headline")

    ts = t.get_text(strip=True) if t else ""
    title = a.get_text(strip=True) if a else ""
    link = (a.get("href") or "").strip() if a else ""

    news.append([ts, title, link])

print("Storing News data...")
with open(NEWS_CSV, "w", newline="", encoding="utf-8") as f:
    w = csv.writer(f)
    w.writerow(["LatestNews_timestamp", "title", "link"])
    w.writerows(news)
print(f"CSV created: {NEWS_CSV} (rows={len(news)})")

print("Done.")
```

Open ~/Desktop/ZixiWang_2854187591/data/processed_data market_data.csv

market_data.csv news_data.csv chromedriver.log

```
marketCard_symbol,marketCard_stockPosition,marketCardchangePct
.DJI,49359.33,-0.17
.SPX,6940.01,-0.064
.IXIC,23515.386,-0.062
.VIX,15.86,0.126
```

Open ~/Desktop/ZixiWang_2854187591/data/processed_data news_data.csv

market_data.csv news_data.csv chromedriver.log

```
LatestNews_timestamp,title,link
10 Hours Ago,Week in review: Stocks battled a flood of news and we booked some profits,https://www.cnbc.com/2026/01/17/week-in-review-stocks-battled-a-flood-of-news-and-we-booked-some-profits.html
11 Hours Ago,Trump threatens to sue JPMorgan Chase for 'debanking' him,https://www.cnbc.com/2026/01/17/trump-jpmorgan-chase-debanking.html
13 Hours Ago,Trump: NATO members to face tariffs up to 25% until a Greenland deal is struck,https://www.cnbc.com/2026/01/17/trump-greenland-tariffs-nato.html
14 Hours Ago,"Led by Texas, states race to prove they can put bitcoin on public balance sheet",https://www.cnbc.com/2026/01/17/texas-us-states-budgets-bitcoin-crypto-strategic-reserve.html
16 Hours Ago,Unshaken: Why Brazilian stocks have looked past the Venezuela attack,https://www.cnbc.com/2026/01/17/unshaken-why-brazilian-stocks-have-looked-past-the-venezuela-attack.html
16 Hours Ago,Bestselling author: How to create better habits without relying on discipline,https://www.cnbc.com/2026/01/17/james-clear-how-to-create-better-habits-without-relying-on-discipline.html
16 Hours Ago,"Warren Buffett: To maximize your potential, ask yourself this question",https://www.cnbc.com/2026/01/17/warren-buffett-to-maximize-your-potential-ask-yourself-this-question.html
```

I created scripts/data_filter.py to read data/raw_data/web_data.html into a Python list (readlines()), parse it with BeautifulSoup, and extract (marketCard_symbol, marketCard_stockPosition, marketCardchangePct) from the Market banner and (LatestNews-timestamp, title, link) from each Latest News entry. The script writes the extracted market data to data/processed_data/market_data.csv and the news data to

data/processed_data/news_data.csv, while printing progress messages such as “Filtering fields…”, “Storing Market data…”, and “CSV created…”. I confirmed the outputs by previewing the CSV files in the terminal using head -n 5 data/processed_data/market_data.csv and head -n 5 data/processed_data/news_data.csv.

my githublink: https://github.com/lacunaxu/DSCI560-Lab1