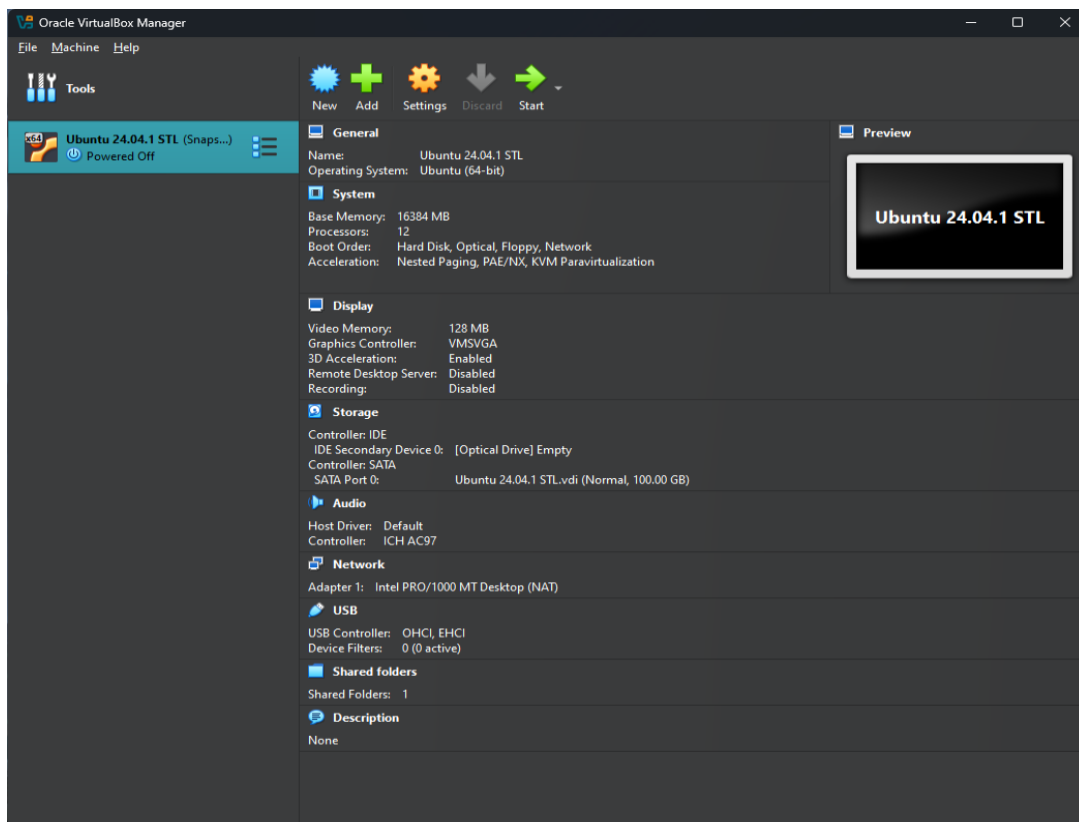


DSCI 560 Lab 1: Report

1 Installation and Setup

1.1 VirtualBox and Ubuntu Installation

- ❖ VirtualBox 7.1 x86_64 was downloaded from official website and installed on local Windows PC.
- ❖ Ubuntu 24.04.1 STL x86_64-bit image file was download from official website and installed on VirtualBox. The resources of the virtual machine were allocated as shown in the snapshot below:

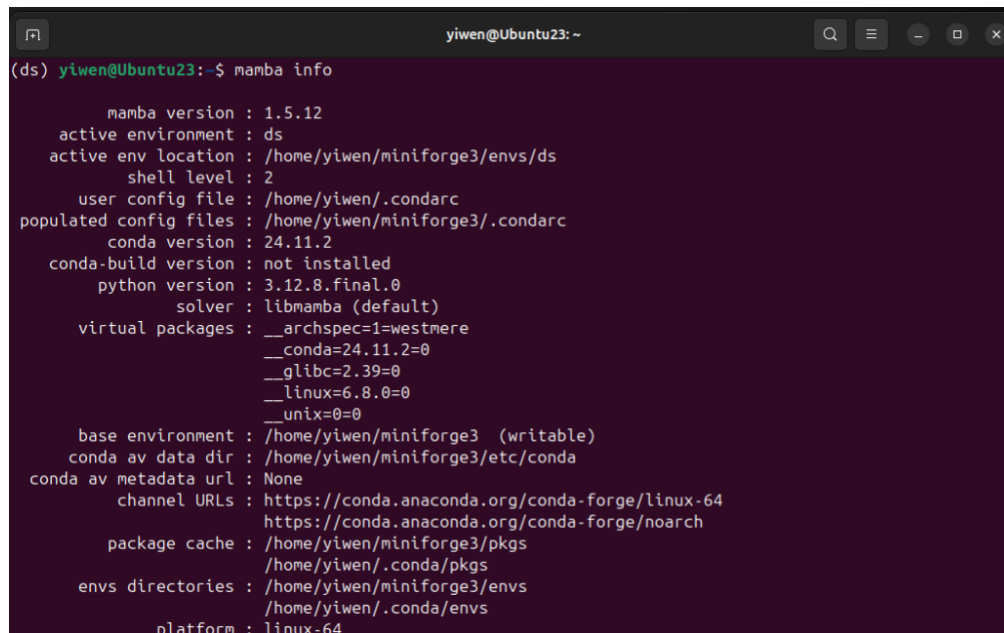


Name: Zhenyu Chen

SID: 2242377315

1.2 Python Environment Setup

- ❖ The Python environment was configured using Miniforge3 script, which was obtained from the official GitHub repository.
 - Subsequently, a new Python environment named “ds” was created with Python version 3.11, and the required libraries were installed.

A terminal window titled 'yiwen@Ubuntu23: ~' showing the output of the 'mamba info' command. The output lists various configuration details for the mamba environment, including version, active environment name, location, shell level, user config file, populated config files, conda version, conda-build version, python version, solver, virtual packages, base environment, conda av data dir, conda av metadata url, channel URLs, package cache, envs directories, and platform.

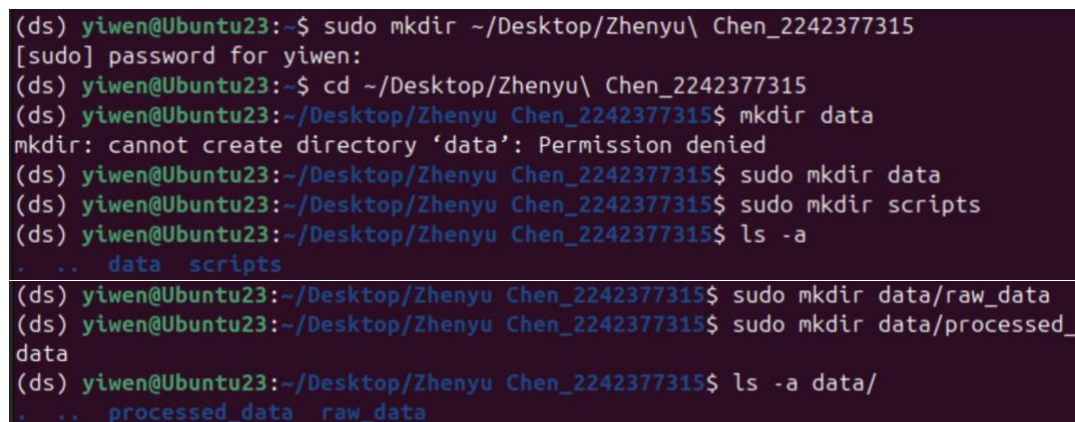
```
(ds) yiwen@Ubuntu23:~$ mamba info

mamba version : 1.5.12
active environment : ds
active env location : /home/yiwen/miniforge3/envs/ds
shell level : 2
user config file : /home/yiwen/.condarc
populated config files : /home/yiwen/miniforge3/.condarc
conda version : 24.11.2
conda-build version : not installed
python version : 3.12.8.final.0
solver : libmamba (default)
virtual packages : __archspec=1=westmere
                  __conda=24.11.2=0
                  __glibc=2.39=0
                  __linux=6.8.0=0
                  __unix=0=0
base environment : /home/yiwen/miniforge3 (writable)
conda av data dir : /home/yiwen/miniforge3/etc/conda
conda av metadata url : None
channel URLs : https://conda.anaconda.org/conda-forge/linux-64
               https://conda.anaconda.org/conda-forge/noarch
package cache : /home/yiwen/miniforge3/pkg
                 /home/yiwen/.conda/pkg
envs directories : /home/yiwen/miniforge3/envs
                  /home/yiwen/.conda/envs
platform : linux-64
```

2 Get Familiar with Linux and Python

2.1 Playing around with Linux Terminal

- ❖ The folder bearing my name and student ID on the Desktop was created using the "sudo mkdir" command, alongside other directories, as illustrated in the snapshots below:

A terminal window showing a series of commands to create a directory structure on the desktop. The user creates a directory named 'Zhenyu Chen_2242377315' on the desktop, then enters it and creates subdirectories 'data' and 'scripts'. Finally, they create 'raw_data' and 'processed_data' subdirectories within 'data' and list the contents of the 'data' directory.

```
(ds) yiwen@Ubuntu23:~$ sudo mkdir ~/Desktop/Zhenyu\ Chen_2242377315
[sudo] password for yiwen:
(ds) yiwen@Ubuntu23:~$ cd ~/Desktop/Zhenyu\ Chen_2242377315
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ mkdir data
mkdir: cannot create directory 'data': Permission denied
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ sudo mkdir data
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ sudo mkdir scripts
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ ls -a
.  ..  data  scripts
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ sudo mkdir data/raw_data
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ sudo mkdir data/processed_data
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ ls -a data/
.  ..  processed_data  raw_data
```

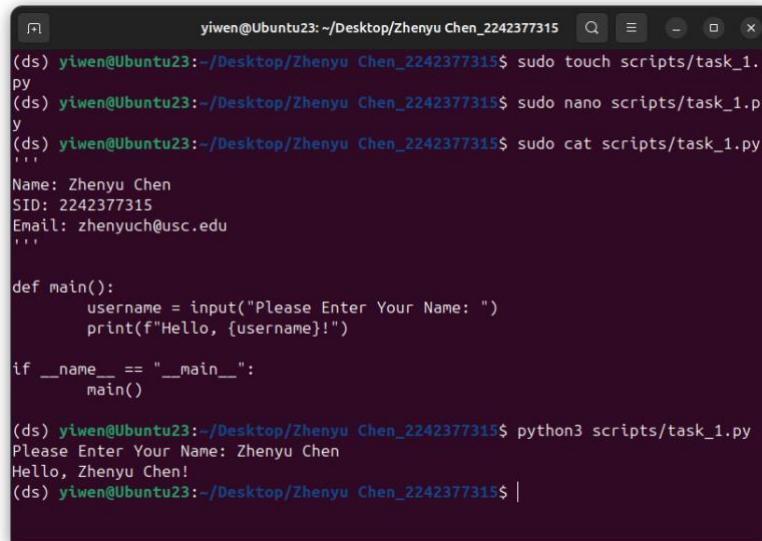
- ❖ A Python script named “task_1.py” was created using the “touch” command, as shown below:

Name: Zhenyu Chen
SID: 2242377315

```
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ sudo touch scripts/task_1.py
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ ls -a scripts/
.  ..  task_1.py
```

2.2 A basic Python Script

- ❖ For the first task, the script was opened and edited using the GNU nano editor via “nano” command.
 - The script defines a main function to read the input username variable and subsequently print a greeting with user’s name.

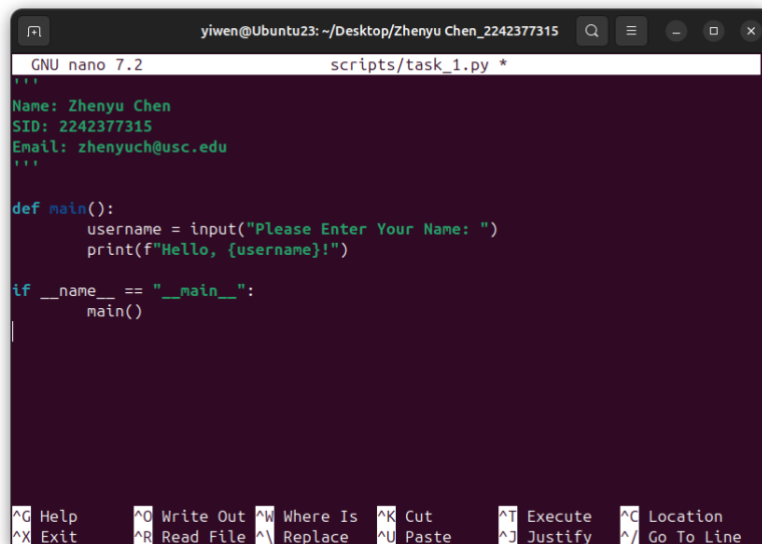


```
yiwen@Ubuntu23: ~/Desktop/Zhenyu Chen_2242377315
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ sudo touch scripts/task_1.py
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ sudo nano scripts/task_1.py
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ sudo cat scripts/task_1.py
'''
Name: Zhenyu Chen
SID: 2242377315
Email: zhenyuch@usc.edu
'''

def main():
    username = input("Please Enter Your Name: ")
    print(f"Hello, {username}!")

if __name__ == "__main__":
    main()

(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ python3 scripts/task_1.py
Please Enter Your Name: Zhenyu Chen
Hello, Zhenyu Chen!
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ |
```



```
GNU nano 7.2 scripts/task_1.py *
'''
Name: Zhenyu Chen
SID: 2242377315
Email: zhenyuch@usc.edu
'''

def main():
    username = input("Please Enter Your Name: ")
    print(f"Hello, {username}!")

if __name__ == "__main__":
    main()

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^A Replace  ^U Paste     ^J Justify  ^_ Go To Line
```

Name: Zhenyu Chen

SID: 2242377315

2.3 Python Web-scraping Task

In the “web_scraper.py” script, the web scraping process is divided into two main parts: Static Data, which can be easily retrieved using the requests library, and Dynamic Data, which requires the selenium library as it is generated and updated through JavaScript.

Implementation Logic:

❖ Static Data Retrieval:

- The static data from the target website was retrieved using the requests library. Most of the required information was captured, except for the market data. The static data was then processed into a “soup” structure powered by the BeautifulSoup4 library.

❖ Dynamic Data Retrieval:

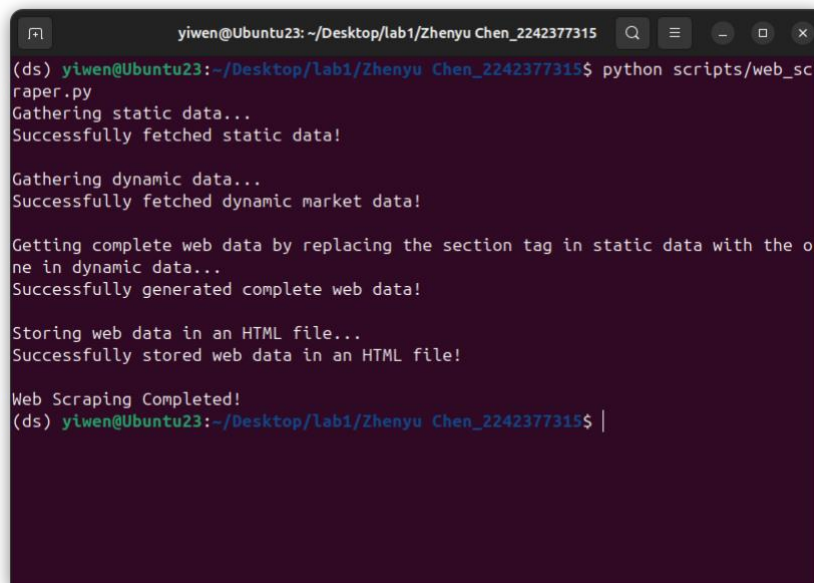
- The dynamic data (primarily the market data within the “section” tag) was extracted using the selenium library, along with the Chrome WebDriver. This data was also converted into a “soup” structure for further processing.

❖ Data Integration:

- After obtaining both static and dynamic data in “soup” structures, the complete dataset was created by replacing the “section” tag in the static data with the corresponding content from the dynamic data.

❖ Data Storage:

- Finally, the combined web data was saved as an HTML file in the “raw_data” folder. The snapshots of the resulting are shown below:

A terminal window with a dark purple background and white text. The window title is "yiwen@Ubuntu23: ~/Desktop/lab1/Zhenyu Chen_2242377315". The command prompt shows the user running "python scripts/web_scraper.py". The script outputs several status messages: "Gathering static data...", "Successfully fetched static data!", "Gathering dynamic data...", "Successfully fetched dynamic market data!", "Getting complete web data by replacing the section tag in static data with the one in dynamic data...", "Successfully generated complete web data!", "Storing web data in an HTML file...", "Successfully stored web data in an HTML file!", and "Web Scraping Completed!". The prompt then shows the user at the shell again.

```
yiwen@Ubuntu23: ~/Desktop/lab1/Zhenyu Chen_2242377315
(ds) yiwen@Ubuntu23:~/Desktop/lab1/Zhenyu Chen_2242377315$ python scripts/web_scraper.py
Gathering static data...
Successfully fetched static data!

Gathering dynamic data...
Successfully fetched dynamic market data!

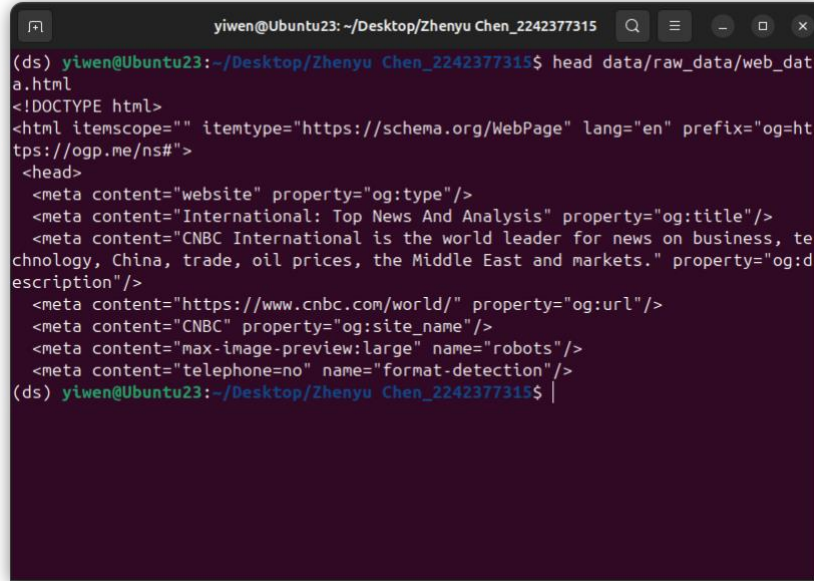
Getting complete web data by replacing the section tag in static data with the one in dynamic data...
Successfully generated complete web data!

Storing web data in an HTML file...
Successfully stored web data in an HTML file!

Web Scraping Completed!
(ds) yiwen@Ubuntu23:~/Desktop/lab1/Zhenyu Chen_2242377315$
```

Name: Zhenyu Chen

SID: 2242377315



```
yiwen@Ubuntu23: ~/Desktop/Zhenyu Chen_2242377315
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ head data/raw_data/web_data.html
<!DOCTYPE html>
<html itemscope="" itemtype="https://schema.org/WebPage" lang="en" prefix="og:https://ogp.me/ns#">
  <head>
    <meta content="website" property="og:type"/>
    <meta content="International: Top News And Analysis" property="og:title"/>
    <meta content="CNBC International is the world leader for news on business, technology, China, trade, oil prices, the Middle East and markets." property="og:description"/>
    <meta content="https://www.cnbc.com/world/" property="og:url"/>
    <meta content="CNBC" property="og:site_name"/>
    <meta content="max-image-preview:large" name="robots"/>
    <meta content="telephone=no" name="format-detection"/>
(ds) yiwen@Ubuntu23:~/Desktop/Zhenyu Chen_2242377315$ |
```

2.4 Data Filtering Task

In the “data_filter.py” script, the program first reads the raw web data created and stored in the previous task. This is accomplished using the BeautifulSoup4 library, which parses the data into a soup structure. The data filtering process is divided into two parts: filtering market data and filtering the latest news data.

Market Data Filtering:

- **Identifying Parent Element:** The parent element containing market data was identified using the “div” tag with the class “MarketsBanner-marketData”.
- **Extracting Market Items:** Sub-elements containing market cards were located using the “a” tag with the class “MarketCard-container”.
- **Extracting Individual Market Item Details:**
 - For each market card, the following details were obtained:
 - ◇ **Symbol:** Retrieved from the “span” tag with the class “MarketCard-symbol”.
 - ◇ **Stock Position:** Retrieved from the “span” tag with the class “MarketCard-stockPosition”.
 - ◇ **Change Percentage:** Retrieved from the “span” tag with the class “MarketCard-changePct”.
- **Data Storage:** After extracting the above details for each market item, the data was saved as a CSV file in the “processed_data” folder. The snapshot of the market result is provided below:

Name: Zhenyu Chen

SID: 2242377315

1	Symbol	Stock Position	Change Percentage
2	DJIA	43,487.83	+0.78%
3	S&P 500	5,996.66	+1.00%
4	NASDAQ	19,630.20	+1.51%
5	RUSS 2K*	2,275.88	+0.40%
6	VIX	15.97	-3.80%

Latest News Data Filtering:

- **Identifying Parent Element:** The parent element containing the latest news data was identified using the ul tag with the class LatestNews-list.
- **Extracting News Items:** Sub-elements containing individual news items were located using the li tag with the class LatestNews-item.
- **Extracting Individual News Item Details:**
 - For each news item, the following details were obtained:
 - ◇ **Timestamp:** Retrieved from the “time” tag with the class “LatestNews-timestamp”.
 - ◇ **Title:** Retrieved from the headline element (“a” tag) with the class “LatestNews-headline”.
 - ◇ **Link:** Retrieved from the same headline element as the title.
- **Data Storage:** After extracting the above details for each market item, the data was saved as a CSV file in the “processed_data” folder. The snapshot of the latest news result is provided below:

1	Timestamp	Title	Link
2	7 Min Ago	I've worked with over 1,000 kids—the emotionally intelligent ones use 6 phrases	https://www.cnbc.com/2025/01/19/kids-with-high-emotional-intelligence-use-these-phrases-therapist.html
3	3 Hours Ago	Trump vowed to declare a national energy emergency. Here's how he might do it	https://www.cnbc.com/2025/01/19/how-trump-could-declare-a-national-energy-emergency-.html
4	3 Hours Ago	Charting the Biden economy: Despite all the growth, a deeply unpopular president	https://www.cnbc.com/2025/01/19/charting-the-biden-economy-deeply-unpopular-despite-growth-and-jobs.html
5	3 Hours Ago	Citigroup picks high-yield stocks to play China as tariffs loom on the horizon	https://www.cnbc.com/2025/01/19/citigroup-picks-high-yield-stocks-to-play-china-as-tariffs-loom.html
6	3 Hours Ago	Berkshire hasn't paid a dividend in nearly 60 years — Could that ever change?	https://www.cnbc.com/2025/01/19/berkshire-hasnt-paid-a-dividend-in-60-years-could-that-ever-change.html
7	3 Hours Ago	Buy these food stocks to ride out the 'Make America Healthy Again' risks	https://www.cnbc.com/2025/01/19/make-america-healthy-again-buy-these-food-stocks-to-ride-out-the-risk.html
8	3 Hours Ago	Earnings playbook: Your guide to this week's biggest reports, including Netflix	https://www.cnbc.com/2025/01/19/earnings-playbook-your-guide-to-this-weeks-biggest-reports-including-netflix.html
9	3 Hours Ago	LA wildfires thrust insurance startup into spotlight as homeowners seek protection	https://www.cnbc.com/2025/01/19/la-wildfires-put-stand-in-spotlight-as-homeowners-look-for-insurance.html
10	11 Hours Ago	Apple, Google remove TikTok from stores as app halts service in U.S.	https://www.cnbc.com/2025/01/18/apple-google-remove-tiktok-from-stores-as-app-halts-service-in-us.html
11	19 Hours Ago	Perplexity AI makes a bid to merge with TikTok U.S.	https://www.cnbc.com/2025/01/18/perplexity-ai-makes-a-bid-to-merge-with-tiktok-us.html
12	22 Hours Ago	Solana surges 12% on launch of Trump-themed meme coin, ether falls	https://www.cnbc.com/2025/01/18/crypto-market-today.html
13	23 Hours Ago	What to expect from travel prices in 2025, and which spots have the best deals	https://www.cnbc.com/2025/01/18/what-to-expect-from-travel-prices-in-2025.html
14	24 Hours Ago	Consumer protection agencies at risk in Trump's second term: What it means for you	https://www.cnbc.com/2025/01/18/how-trumps-second-term-could-mean-the-downfall-of-the-fdic-cfpb.html
15	January 18, 2025	Why the gold boom is causing a surge in illegal mining	https://www.cnbc.com/2025/01/18/why-the-gold-boom-is-causing-a-surge-in-illegal-mining.html
16	January 18, 2025	Google Maps is turning 20 — mapping more countries and adding AI capabilities	https://www.cnbc.com/2025/01/18/google-maps-turns-20-adds-ai-features-new-countries-to-beat-apple.html
17	January 18, 2025	Trump inauguration trades: The sectors that could win in the early days	https://www.cnbc.com/2025/01/18/trump-inauguration-trades-the-sectors-that-could-win-in-the-early-days.html