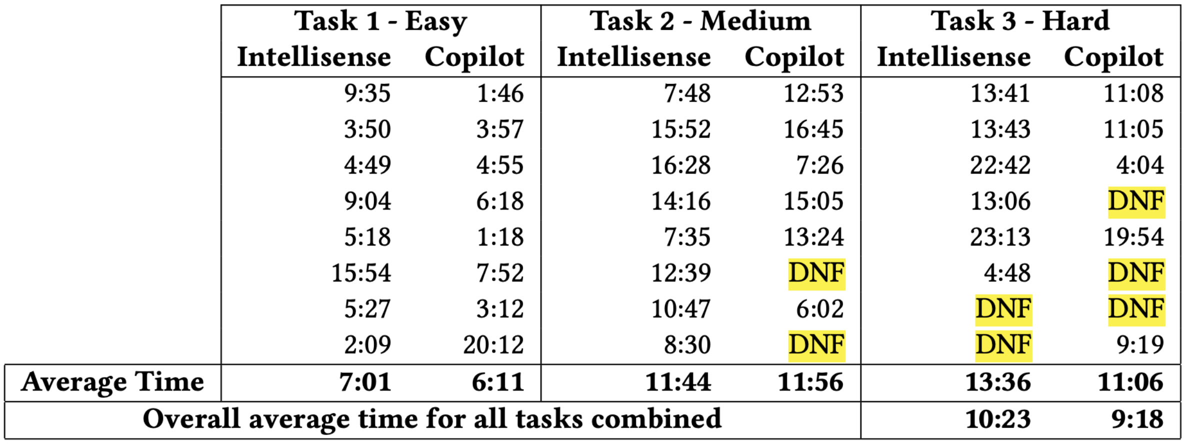
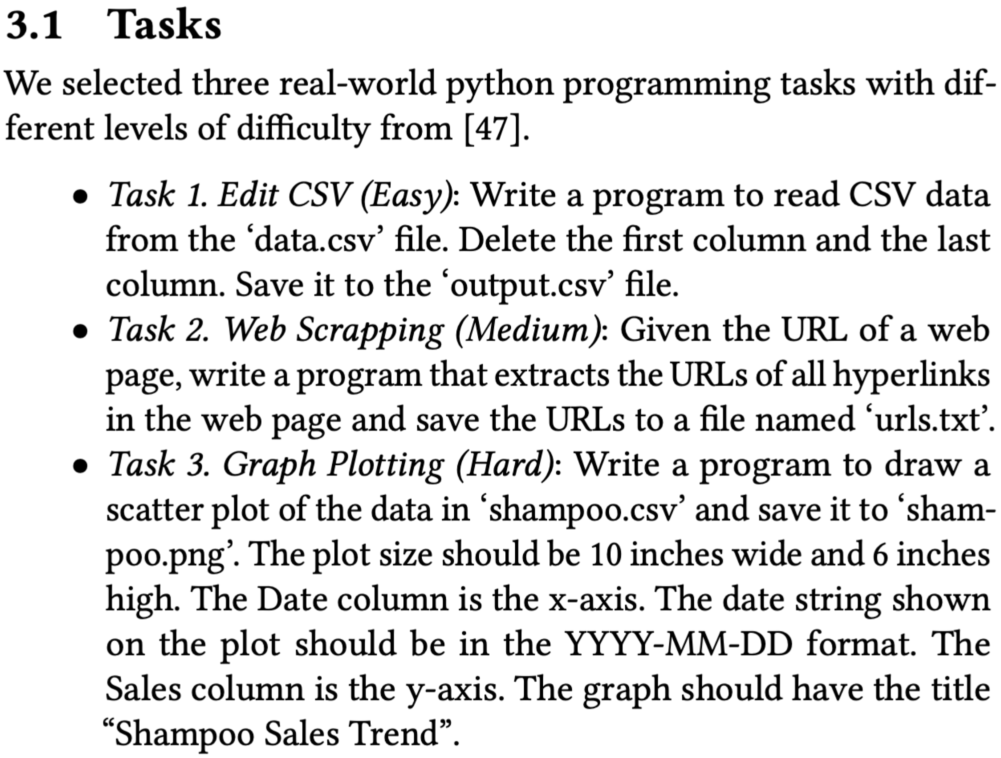
To understand the impact of copilot, I thought I would give copilot workspace a chance to repeat the tasks that were part of this research paper. The paper found that Copilot was not able to help developers save time:



	Task 1 - Easy		Task 2 - Medium		Task 3 - Hard	
	Intellisense	**Copilot**	**Intellisense**	**Copilot**	**Intellisense**	**Copilot**
	9:35	1:46	7:48	12:53	13:41	11:08
	3:50	3:57	15:52	16:45	13:43	11:05
	4:49	4:55	16:28	7:26	22:42	4:04
	9:04	6:18	14:16	15:05	13:06	DNF
	5:18	1:18	7:35	13:24	23:13	19:54
	15:54	7:52	12:39	DNF	4:48	DNF
	5:27	3:12	10:47	6:02	DNF	DNF
	2:09	20:12	8:30	DNF	DNF	9:19
Average Time	**7:01**	**6:11**	**11:44**	**11:56**	**13:36**	**11:06**
Overall average time for all tasks combined					**10:23**	**9:18**

Total time for python developer + non-copilot: 32:15 mins
Total time for python developer + copilot: 29 mins
Total time for me (not a python developer) + copilot-workspace: 37 mins

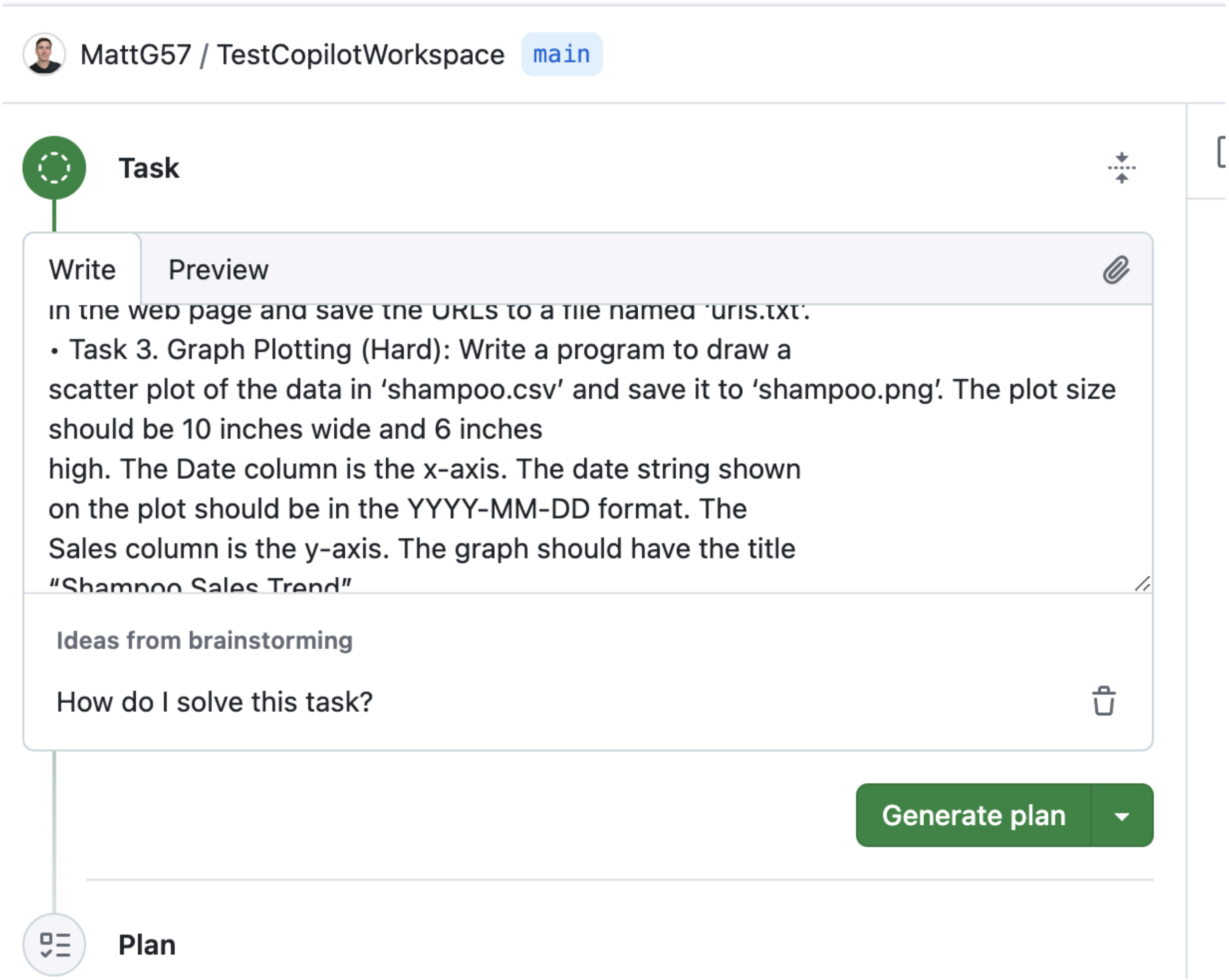
Start: Dec 9, 8:45 PM
Created a blank repository and used Copilot Workspace to create an app that accomplishes the 3 tasks:



3.1 Tasks

We selected three real-world python programming tasks with different levels of difficulty from [47].

- *Task 1. Edit CSV (Easy)*: Write a program to read CSV data from the ‘data.csv’ file. Delete the first column and the last column. Save it to the ‘output.csv’ file.
- *Task 2. Web Scrapping (Medium)*: Given the URL of a web page, write a program that extracts the URLs of all hyperlinks in the web page and save the URLs to a file named ‘urls.txt’.
- *Task 3. Graph Plotting (Hard)*: Write a program to draw a scatter plot of the data in ‘shampoo.csv’ and save it to ‘shampoo.png’. The plot size should be 10 inches wide and 6 inches high. The Date column is the x-axis. The date string shown on the plot should be in the YYYY-MM-DD format. The Sales column is the y-axis. The graph should have the title “Shampoo Sales Trend”.

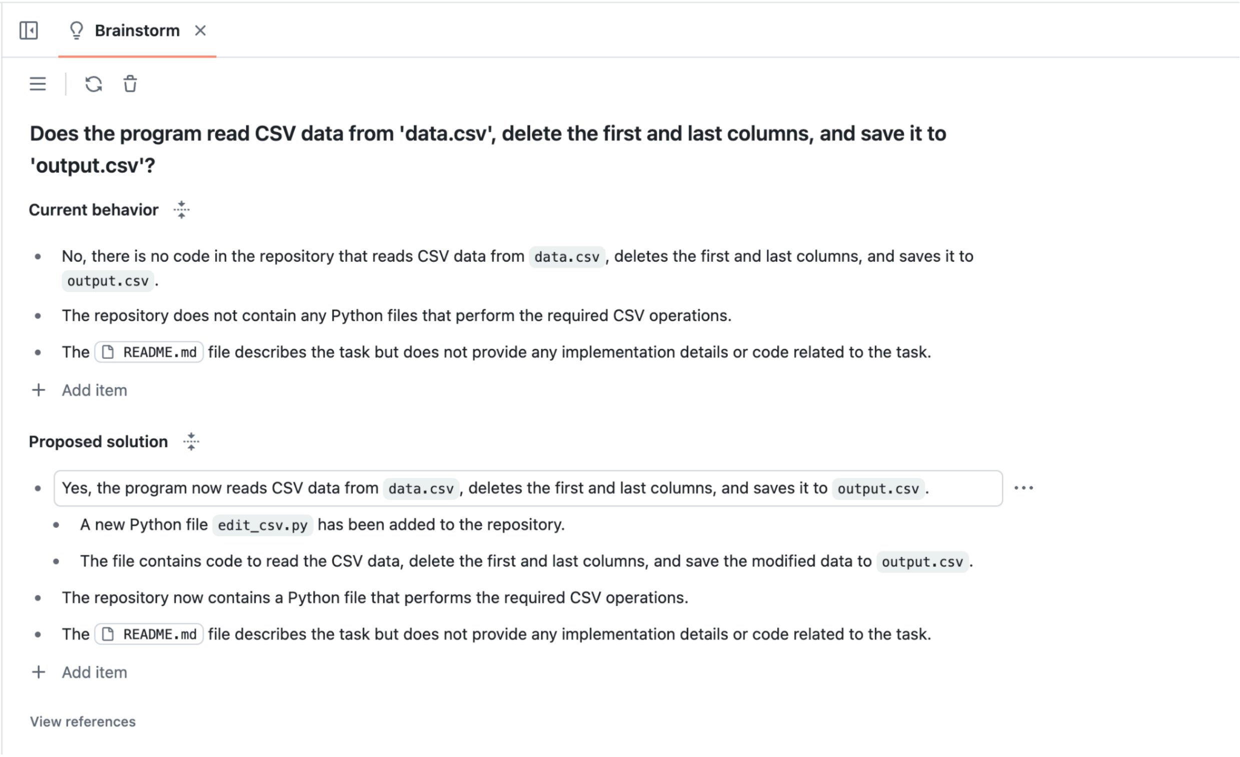


in the web page and save the URLs to a file named 'urls.txt'.

• Task 3. Graph Plotting (Hard): Write a program to draw a
scatter plot of the data in 'shampoo.csv' and save it to 'shampoo.png'. The plot size
should be 10 inches wide and 6 inches
high. The Date column is the x-axis. The date string shown
on the plot should be in the YYYY-MM-DD format. The
Sales column is the y-axis. The graph should have the title
"Shampoo Sales Trend"

Ideas from brainstorming

How do I solve this task?



Does the program read CSV data from 'data.csv', delete the first and last columns, and save it to 'output.csv'?

Current behavior

- No, there is no code in the repository that reads CSV data from `data.csv`, deletes the first and last columns, and saves it to `output.csv`.
- The repository does not contain any Python files that perform the required CSV operations.
- The `README.md` file describes the task but does not provide any implementation details or code related to the task.

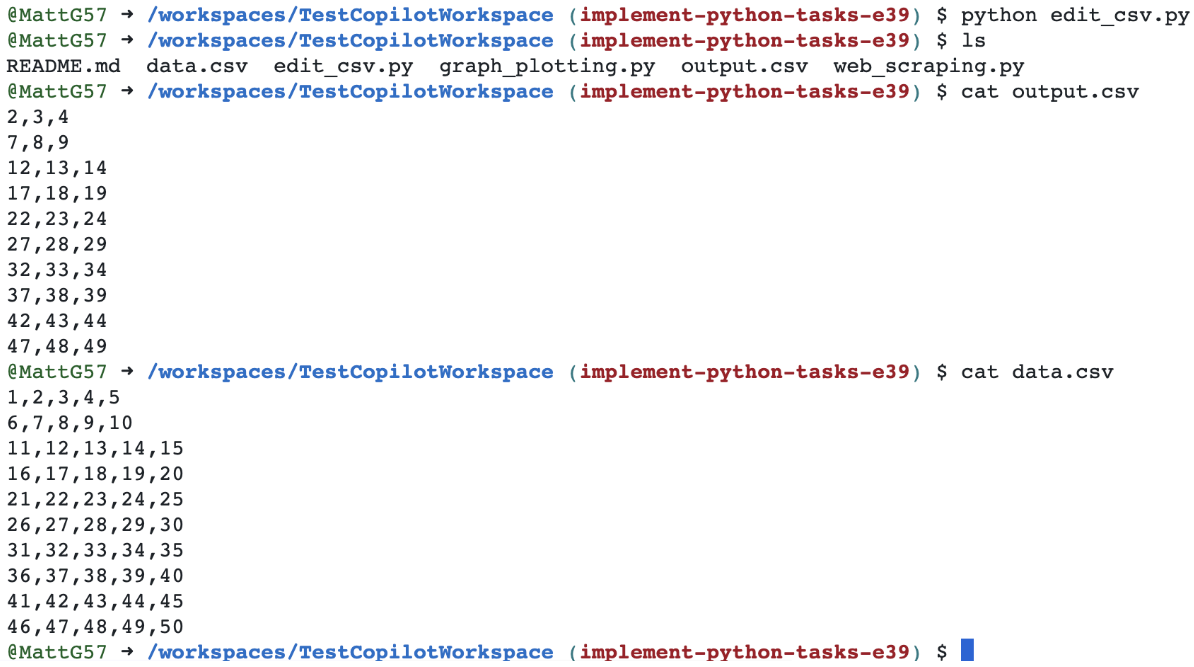
Proposed solution

- Yes, the program now reads CSV data from `data.csv`, deletes the first and last columns, and saves it to `output.csv`.
  - A new Python file `edit_csv.py` has been added to the repository.
  - The file contains code to read the CSV data, delete the first and last columns, and save the modified data to `output.csv`.
- The repository now contains a Python file that performs the required CSV operations.
- The `README.md` file describes the task but does not provide any implementation details or code related to the task.

During Brainstorming, I added a bit more guidance. As I might if this was a real person taking a test.

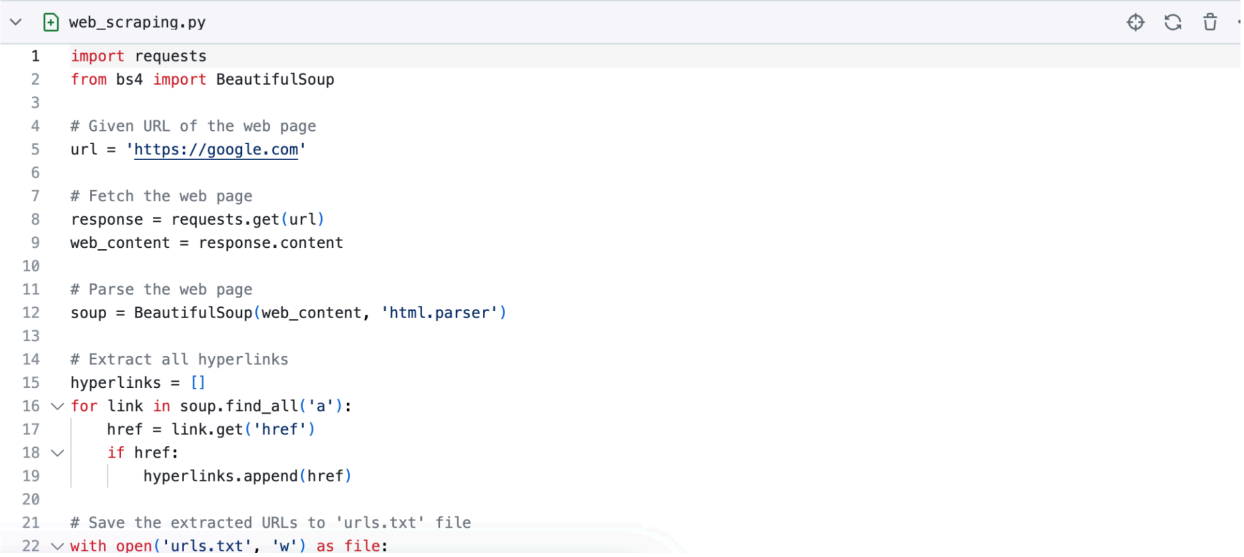
Note:
Copilot correctly spells “Web Scraping” vs the research paper’s misspelling: “Web Scrapping”

Test 1 : passed

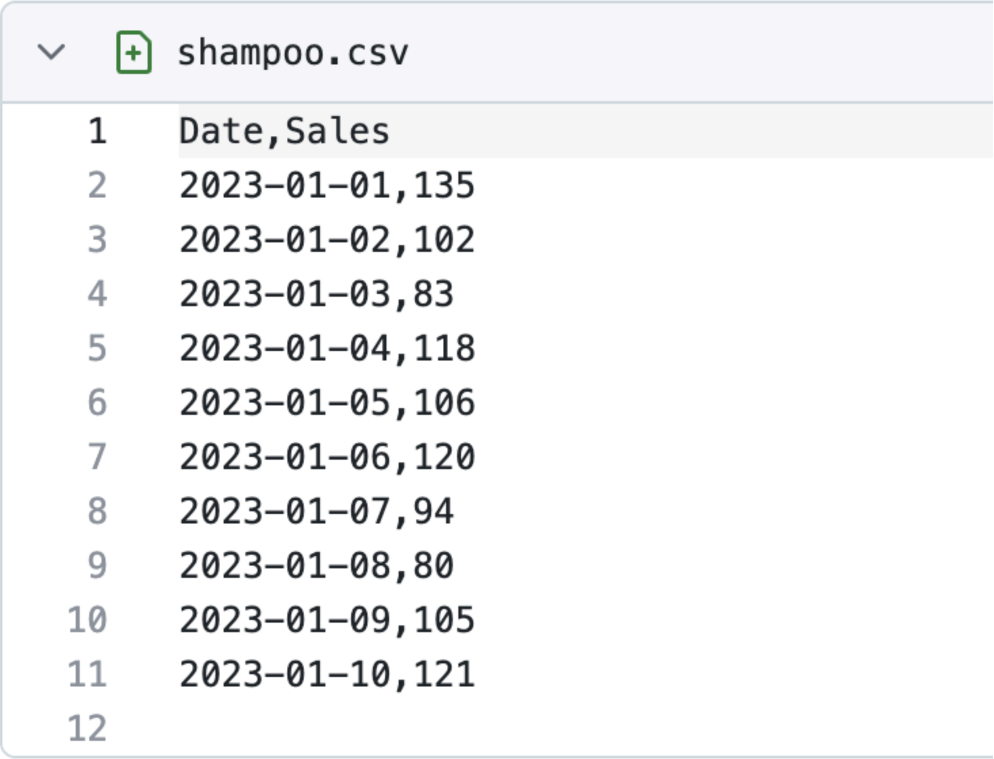


```
@MattG57 ➜ /workspaces/TestCopilotWorkspace (implement-python-tasks-e39) $ python edit_csv.py
@MattG57 ➜ /workspaces/TestCopilotWorkspace (implement-python-tasks-e39) $ ls
README.md  data.csv  edit_csv.py  graph_plotting.py  output.csv  web_scraping.py
@MattG57 ➜ /workspaces/TestCopilotWorkspace (implement-python-tasks-e39) $ cat output.csv
2,3,4
7,8,9
12,13,14
17,18,19
22,23,24
27,28,29
32,33,34
37,38,39
42,43,44
47,48,49
@MattG57 ➜ /workspaces/TestCopilotWorkspace (implement-python-tasks-e39) $ cat data.csv
1,2,3,4,5
6,7,8,9,10
11,12,13,14,15
16,17,18,19,20
21,22,23,24,25
26,27,28,29,30
31,32,33,34,35
36,37,38,39,40
41,42,43,44,45
46,47,48,49,50
@MattG57 ➜ /workspaces/TestCopilotWorkspace (implement-python-tasks-e39) $ 
```

Test 2: passed

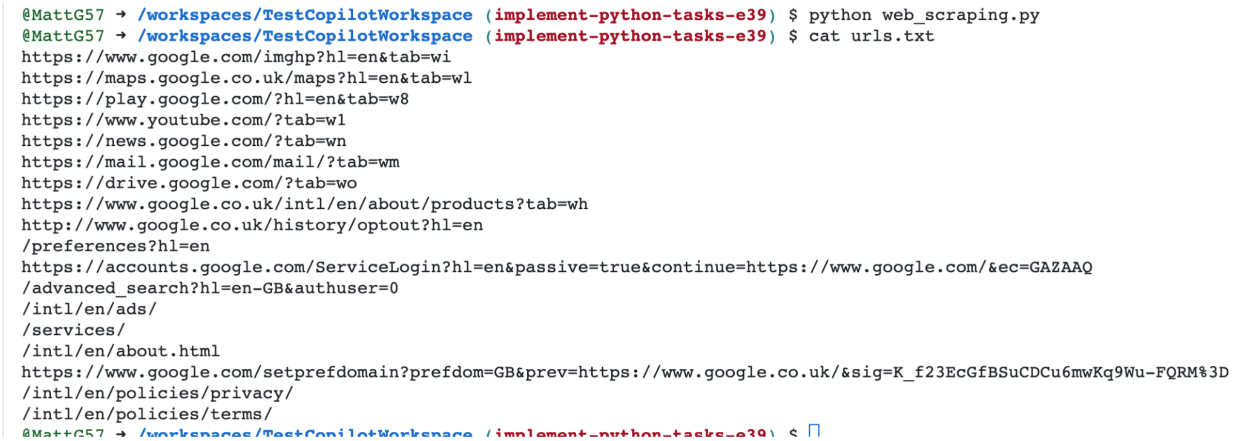


```python
import requests
from bs4 import BeautifulSoup

# Given URL of the web page
url = 'https://google.com'

# Fetch the web page
response = requests.get(url)
web_content = response.content

# Parse the web page
soup = BeautifulSoup(web_content, 'html.parser')

# Extract all hyperlinks
hyperlinks = []
for link in soup.find_all('a'):
    href = link.get('href')
    if href:
        hyperlinks.append(href)

# Save the extracted URLs to 'urls.txt' file
with open('urls.txt', 'w') as file:
```



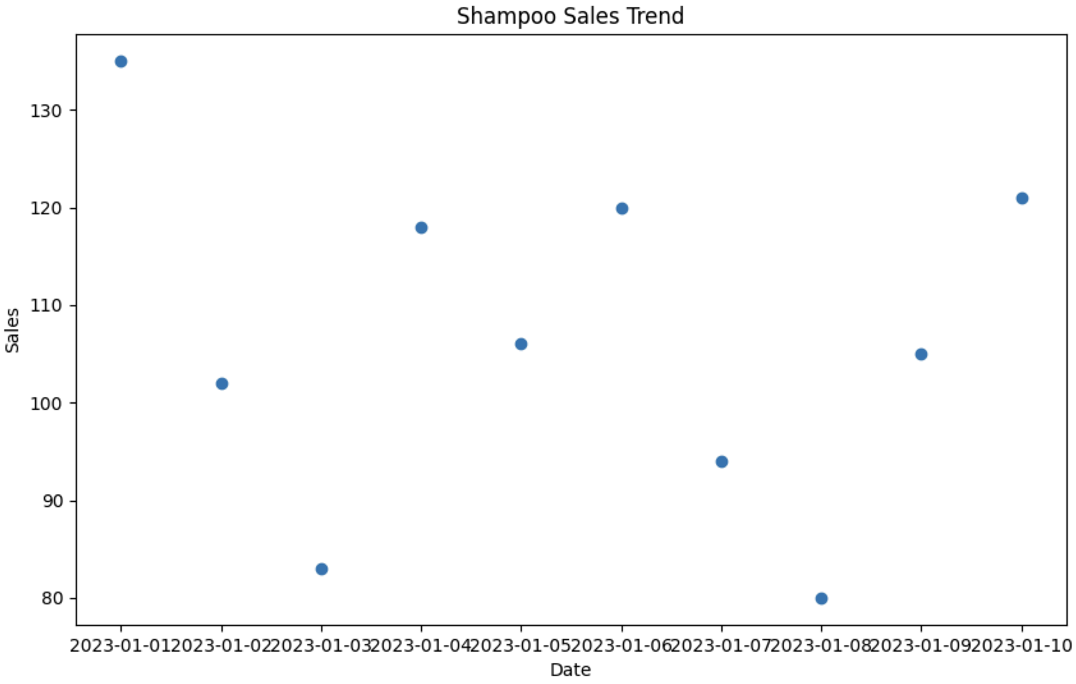
```
Date,Sales
2023-01-01,135
2023-01-02,102
2023-01-03,83
2023-01-04,118
2023-01-05,106
2023-01-06,120
2023-01-07,94
2023-01-08,80
2023-01-09,105
2023-01-10,121

```

Results were correct:



```
@MattG57 → /workspaces/TestCopilotWorkspace (implement-python-tasks-e39) $ python web_scraping.py
@MattG57 → /workspaces/TestCopilotWorkspace (implement-python-tasks-e39) $ cat urls.txt
https://www.google.com/imghp?hl=en&tab=wi
https://maps.google.co.uk/maps?hl=en&tab=wl
https://play.google.com/?hl=en&tab=w8
https://www.youtube.com/?tab=w1
https://news.google.com/?tab=wn
https://mail.google.com/mail/?tab=wm
https://drive.google.com/?tab=wo
https://www.google.co.uk/intl/en/about/products?tab=wh
http://www.google.co.uk/history/optout?hl=en
/preferences?hl=en
https://accounts.google.com/ServiceLogin?hl=en&passive=true&continue=https://www.google.com/&ec=GAZAAQ
/advanced_search?hl=en-GB&authuser=0
/intl/en/ads/
/services/
/intl/en/about.html
https://www.google.com/setprefdomain?prefdom=GB&prev=https://www.google.co.uk/&sig=K_f23EcGfBSuCDCu6mwKq9Wu-FQRM%3D
/intl/en/policies/privacy/
/intl/en/policies/terms/
@MattG57 → /workspaces/TestCopilotWorkspace (implement-python-tasks-e39) $
```

Test3: Passed



Finished: Dec 9, 21:20 PM
(<37 mins total time)

The average in the study was 29 mins.