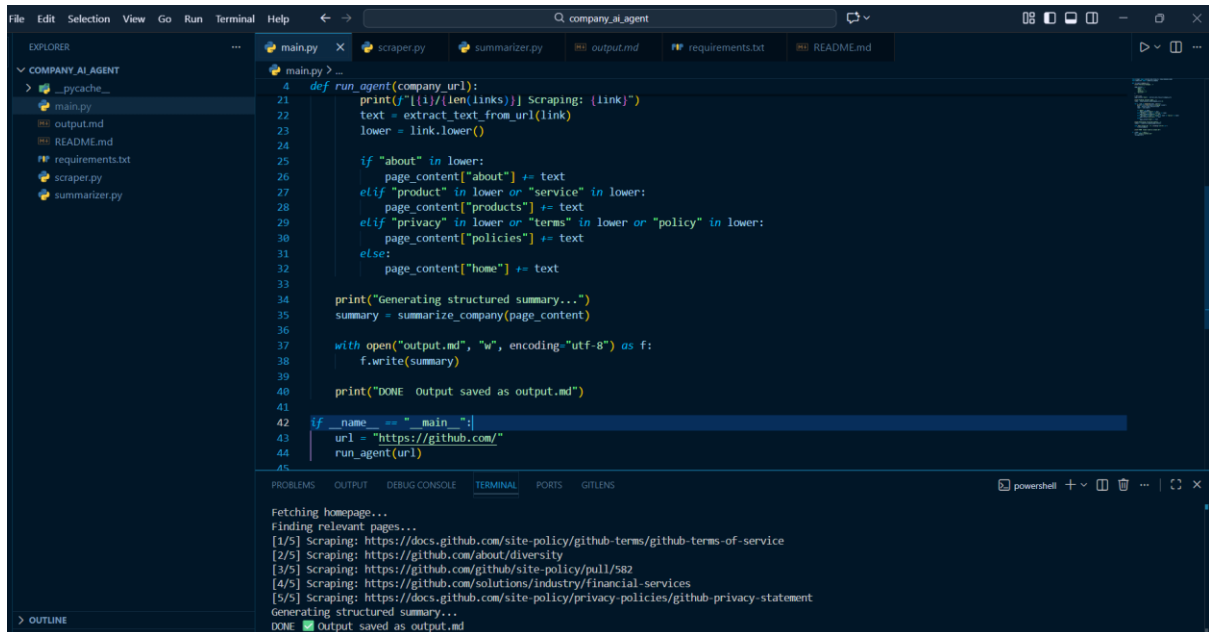


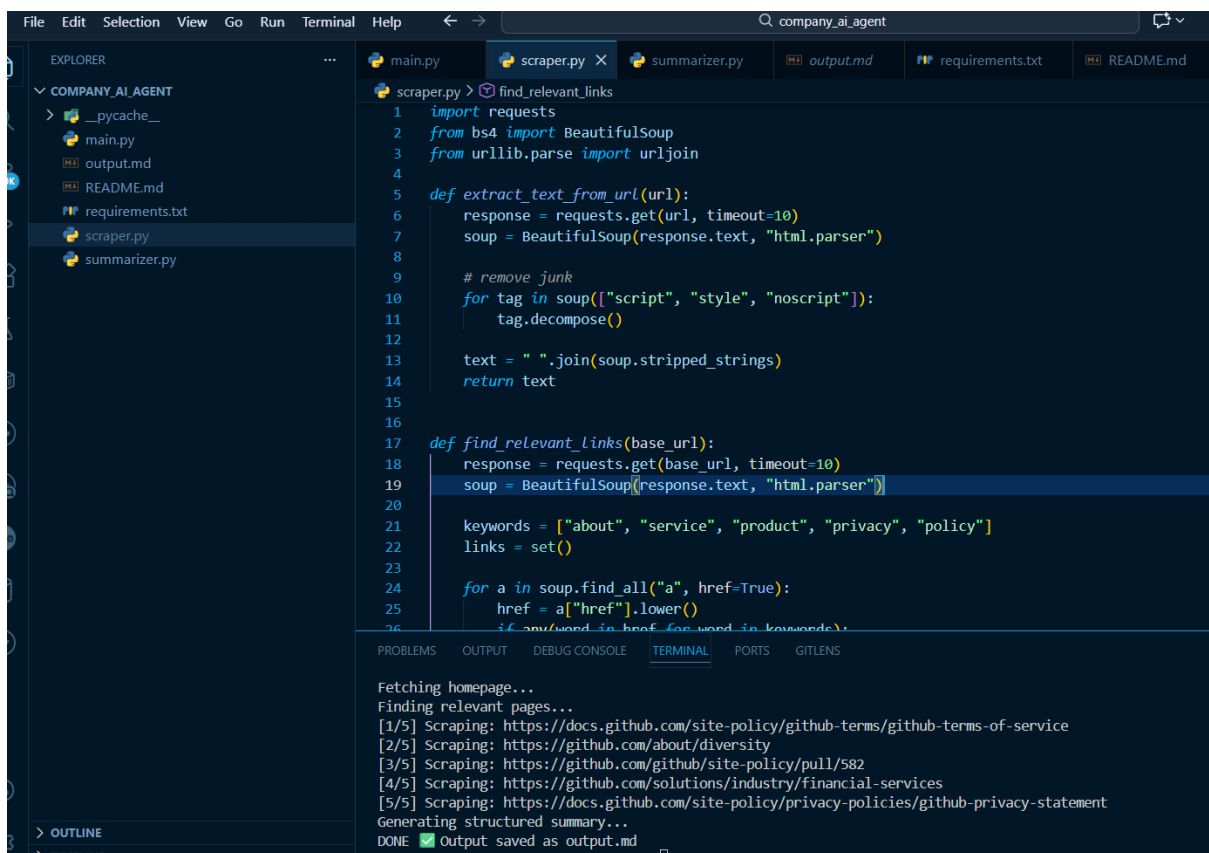
## Main.py:



```
4 def run_agent(company_url):
21     print(f"[1/{len(links)}] Scraping: {link}")
22     text = extract_text_from_url(link)
23     lower = link.lower()
24
25     if "about" in lower:
26         page_content["about"] += text
27     elif "product" in lower or "service" in lower:
28         page_content["products"] += text
29     elif "privacy" in lower or "terms" in lower or "policy" in lower:
30         page_content["policies"] += text
31     else:
32         page_content["home"] += text
33
34     print("Generating structured summary...")
35     summary = summarize_company(page_content)
36
37     with open("output.md", "w", encoding="utf-8") as f:
38         f.write(summary)
39
40     print("DONE Output saved as output.md")
41
42 if __name__ == "__main__":
43     url = "https://github.com/"
44     run_agent(url)
45
```

Fetching homepage...  
Finding relevant pages...  
[1/5] Scraping: https://docs.github.com/site-policy/github-terms/github-terms-of-service  
[2/5] Scraping: https://github.com/about/diversity  
[3/5] Scraping: https://github.com/github/site-policy/pull/582  
[4/5] Scraping: https://github.com/solutions/industry/financial-services  
[5/5] Scraping: https://docs.github.com/site-policy/privacy-policies/github-privacy-statement  
Generating structured summary...  
DONE Output saved as output.md

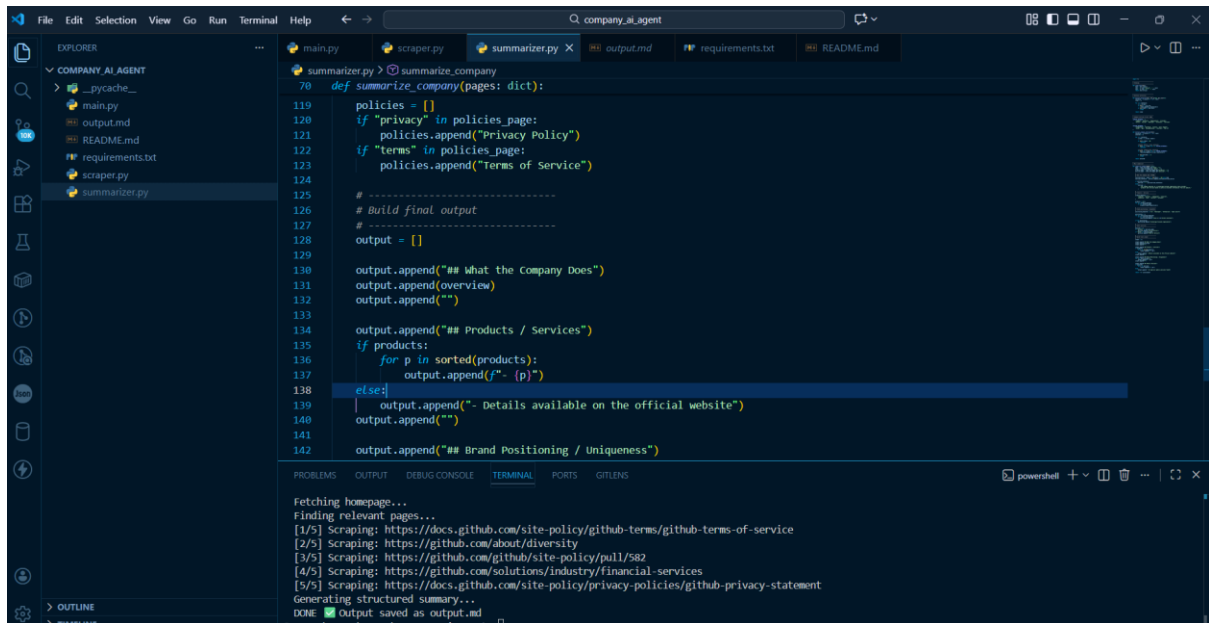
## Scraper.py:



```
1 import requests
2 from bs4 import BeautifulSoup
3 from urllib.parse import urljoin
4
5 def extract_text_from_url(url):
6     response = requests.get(url, timeout=10)
7     soup = BeautifulSoup(response.text, "html.parser")
8
9     # remove junk
10    for tag in soup(["script", "style", "noscript"]):
11        tag.decompose()
12
13    text = " ".join(soup.stripped_strings)
14    return text
15
16
17 def find_relevant_links(base_url):
18     response = requests.get(base_url, timeout=10)
19     soup = BeautifulSoup(response.text, "html.parser")
20
21     keywords = ["about", "service", "product", "privacy", "policy"]
22     links = set()
23
24     for a in soup.find_all("a", href=True):
25         href = a["href"].lower()
26         if any(word in href for word in keywords):
27             links.add(urljoin(base_url, href))
28
29     return links
30
```

Fetching homepage...  
Finding relevant pages...  
[1/5] Scraping: https://docs.github.com/site-policy/github-terms/github-terms-of-service  
[2/5] Scraping: https://github.com/about/diversity  
[3/5] Scraping: https://github.com/github/site-policy/pull/582  
[4/5] Scraping: https://github.com/solutions/industry/financial-services  
[5/5] Scraping: https://docs.github.com/site-policy/privacy-policies/github-privacy-statement  
Generating structured summary...  
DONE Output saved as output.md

## Summarizer.py:



The screenshot shows a Visual Studio Code editor window with the file explorer on the left displaying the project structure for 'COMPANY AI AGENT'. The main editor displays the 'summarizer.py' script, which defines a function 'summarize\_company' that takes a dictionary of pages and generates a structured summary. The terminal at the bottom shows the execution of the script, which scrapes data from various GitHub pages and saves the output to 'output.md'.

```
def summarize_company(pages: dict):
    policies = []
    if "privacy" in pages:
        policies.append("Privacy Policy")
    if "terms" in pages:
        policies.append("Terms of Service")

    # Build final output
    output = []

    output.append("## What the Company Does")
    output.append(overview)
    output.append("")

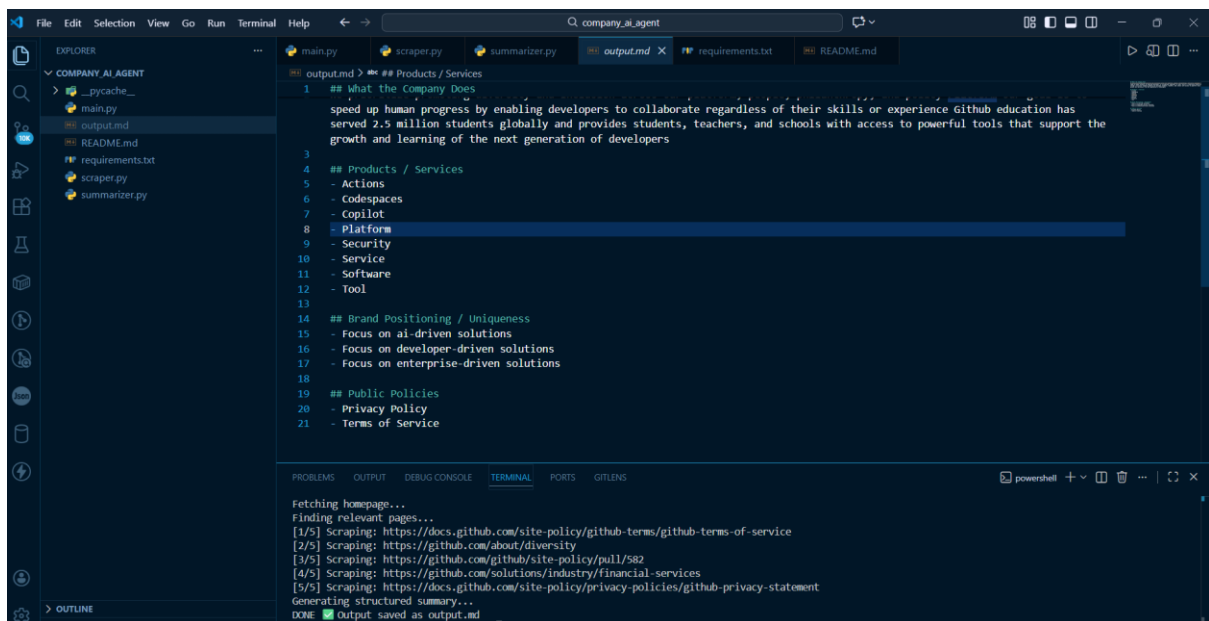
    output.append("## Products / Services")
    if products:
        for p in sorted(products):
            output.append(f"- {p}")
    else:
        output.append("- Details available on the official website")
    output.append("")

    output.append("## Brand Positioning / Uniqueness")
```

Terminal Output:

```
Fetching homepage...
Finding relevant pages...
[1/5] Scraping: https://docs.github.com/site-policy/github-terms/github-terms-of-service
[2/5] Scraping: https://github.com/about/diversity
[3/5] Scraping: https://github.com/github/site-policy/pull/582
[4/5] Scraping: https://github.com/solutions/industry/financial-services
[5/5] Scraping: https://docs.github.com/site-policy/privacy-policies/github-privacy-statement
Generating structured summary...
DONE Output saved as output.md
```

## Output Proof:



The screenshot shows the 'output.md' file generated by the script. It contains a structured summary of the scraped data, including sections for 'What the Company Does', 'Products / Services', 'Brand Positioning / Uniqueness', and 'Public Policies'.

```
## What the Company Does
speed up human progress by enabling developers to collaborate regardless of their skills or experience Github education has served 2.5 million students globally and provides students, teachers, and schools with access to powerful tools that support the growth and learning of the next generation of developers

## Products / Services
- Actions
- Codespaces
- Copilot
- Platform
- Security
- Service
- Software
- Tool

## Brand Positioning / Uniqueness
- Focus on ai-driven solutions
- Focus on developer-driven solutions
- Focus on enterprise-driven solutions

## Public Policies
- Privacy Policy
- Terms of Service
```

Terminal Output:

```
Fetching homepage...
Finding relevant pages...
[1/5] Scraping: https://docs.github.com/site-policy/github-terms/github-terms-of-service
[2/5] Scraping: https://github.com/about/diversity
[3/5] Scraping: https://github.com/github/site-policy/pull/582
[4/5] Scraping: https://github.com/solutions/industry/financial-services
[5/5] Scraping: https://docs.github.com/site-policy/privacy-policies/github-privacy-statement
Generating structured summary...
DONE Output saved as output.md
```