A. Approach:

1. **Setup & Initialization**
    - Used Python along with these key tools:
        - **Selenium** – to render and crawl pages with JavaScript-driven content (like “View More” links).
        - **BeautifulSoup** – for HTML parsing to extract all <a> tags.
        - **webdriver_manager** – for seamless ChromeDriver installation and version management.
        - **time** – to manage page load delays.
        - **urllib.parse (urljoin)** – to convert relative URLs to absolute ones.
    - Script begins at Finploy’s base URL: https://www.finploy.com .
    - **ChatGPT** – used for guidance on code structure, crawling logic, and sitemap generation formatting.
2. **Dynamic Crawling Loop**
    - Navigate to each URL using Selenium and wait briefly (time.sleep(2)), ensuring all JS-rendered links are fully visible.
    - Parse and collect all link targets using BeautifulSoup.
3. **URL Normalization & Filtering**
    - Normalize URLs via urljoin for consistency.
    - Discard external links and anchor fragments (those containing #).
    - Retain only links that point within the Finploy domain.
4. **De-duplication & Traversal Management**
    - Maintain two collections—visited_urls and urls_to_visit—to prevent redundant crawling and looping.
5. **Sitemap Generation**
    - Once crawling completes, compile and write all unique URLs into a sitemap.xml file that adheres to the sitemaps.org protocol.

B. Challenges Faced & How I Resolved Them:

- **Duplicate URLs**
    - *Challenge:* The same URL could appear multiple times across different pages, leading to redundant entries and wasted crawling.
    - *Solution:* Used a Python set() to store visited URLs and efficiently check against them before enqueuing new URLs.
- **Dynamic Content Load Delays**
    - *Challenge:* Some pages loaded links only after executing JavaScript, causing some links to be missed if parsed too early.
    - *Solution:* Introduced time.sleep(2) post-navigation to allow full rendering before scraping begins.
- **Browser Driver Management**
    - *Challenge:* Manually managing ChromeDriver version mismatches and setup environments added unnecessary complexity.
    - *Solution:* Integrated webdriver_manager for automatic and reliable handling of the correct driver version.

C. Code Samples or GitHub Repo Link:

You can find the full source code, installation instructions, and sample output in my GitHub repository:

GitHub Repository: https://github.com/Yash-Hadashi/finploy-sitemap-generator