**Intelligent Web Crawler & Analyzer**

**Project Overview:**

his project is a smart web crawler and analyzer designed to:

* Check a website’s crawlability (robots.txt, sitemaps, crawl delay)
* Extract key metadata and content (titles, descriptions, links, images)
* Detect if the site is JavaScript-heavy or has APIs/RSS feeds
* Visualize results in a Streamlit dashboard
* Provide recommendations for the best crawling approach

The goal is to help users understand how to best access and extract data from any website, using ethical and efficient crawling methods.

**Setup Instructions:**

1.Clone the repository:

  git clone <your-repo-url>

2. Create and activate a virtual environment:

   python -m venv venv

   venv\Scripts\activate

3.Install dependencies:

   pip install -r requirements.txt

   pip install plotly wordcloud matplotlib

   python -m playwright install

5.Run the dashboard:

  python -m streamlit run src/dashboard/app.py

**How Each Part Works**

* **Crawlability Specialist (robots\_analyzer.py):**
* Fetches and analyzes robots.txt for allowed/disallowed paths, crawl delay, and sitemap links.
* Summarizes crawlability rules for the target website.
* **Content Extractor (content\_extractor.py):**
* Uses BeautifulSoup to extract titles, descriptions, links, and images from web pages.
* Handles pagination and retry logic for robust extraction.
* **JS & API Handler (js\_handler.py):**
* Detects if the site is JavaScript-heavy and uses Playwright for dynamic content.
* Checks for open APIs and RSS feeds.
* **Dashboard (app.py):**
* Streamlit app that visualizes crawlability score, top extracted data, and recommendations.
* Provides interactive analysis and visualizations.
* **Documentation & Deployment:**
* This README and a final report.
* Instructions for running and deploying the project.

**How to Use the Dashboard**

**1.Start the dashboard with:**

python -m streamlit run src/dashboard/app.py

**2.Enter the target website URL in the sidebar**

**3.click “Analyze Website”**

**4.Explore the tabs**

* Crawlability Analysis: See robots.txt, crawl delay, sitemaps, and a crawlability score.
* Content Analysis: View extracted titles, descriptions, links, images, and visualizations.
* JS & API Analysis: Check if the site is JavaScript-heavy, and see detected APIs/RSS feeds.
* Recommendations: Get suggestions for the best crawling approach

**Findings:**

* **Tested Site:** https://www.bonappetit.com/
* **Crawlability:**
* robots.txt: accessible
* Crawl delay: 60.0 seconds
* Sitemaps: <https://www.bonappetit.com/sitemap.xml>
* <https://www.bonappetit.com/branded-sitemap.xml>
* <https://www.bonappetit.com/feed/google-latest-news/sitemap-google-news>
* <https://www.bonappetit.com/categories-sitemap.xml>
* <https://www.bonappetit.com/contributors-sitemap.xml>
* <https://www.bonappetit.com/sitemap-archive-1.xml>
* <https://www.bonappetit.com/bundles-sitemap.xml>
* **Content Extraction:**
* Title:  Bon Appétit: Recipes, Cooking, Entertaining, Restaurants | Bon Appétit
* Description: Cook with confidence. Enjoy your food. Find recipes, search our encyclopedia of cooking tips and ingredients, watch food videos, and more.
* Number of links: 115
* Number of images: 73
* JS/API Analysis:
* JavaScript-heavy: NO
* APIs found: N/A
* RSS feeds:  <https://www.bonappetit.com/feed/rss>
* Recommendations:



**Challenges & Solutions:**

* **Python environment issues:**

Resolved by using the official Python installer and updating the PATH variable.

* **robots.txt not accessible:**

Some sites block bots or do not provide robots.txt. This is handled gracefully in the dashboard.

* **Playwright errors:**

Fixed by using a compatible Python version (official python.org installer, not Microsoft Store) and installing Playwright browse

**Team Contributions**

* **Member 1(Menna Alkholy 233085):**Crawlability analysis (robots.txt, sitemaps), Documentation & deployment (README, setup, report)
* **Member 2(Sara Hossam 230131):**Content extraction (BeautifulSoup, pagination)
* **Member 3(Ismail Aser 236467):**JS/API handling (Playwright, API/RSS detection)
* **Member 4 (Eyad Mohammed 233019):**Dashboard & visualization (Streamlit, charts)