WebWeave: Semantic Site Crawler & Data Extractor

Intelligently parses and classifies content from any website into structured domains

Overview.

This web application allows users to input any URL, and the backend system crawls and extracts structured information from the target site. It intelligently identifies key content blocks such as contact info, company overview, services, legal sections, and more using keyword-driven logic. The extracted content is cleaned, deduplicated, formatted, and stored securely in a MongoDB database for future retrieval or chatbot integration.

Objective.

Build a tool that extracts meaningful content from arbitrary websites
Tag extracted data by category (overview, services, team, legal, contact, etc.)
Maintain a consistent file-saving pipeline
Use MongoDB to persist results for future LLM or RAG-based workflows
Return an easily consumable output (SHA-256 key) to identify crawls uniquely

Tools.

- **Backend Engine**: Python
- **Web Crawling**: Selenium (ChromeDriver)
- **HTML Parsing**: BeautifulSoup
- **Regex Matching**: Python re module
- **Security**: SHA-256 hashing for deduplication
- **Database**: MongoDB (via PyMongo)
- **Output Formats**: json, .txt
- **Deployment Mode**: Web App integration (e.g., connected to a Node.js frontend)

Process.

1. **User Input**: Web form receives a website URL and email
2. **Crawling**: Selenium loads the root page and discovers internal links
3. **Parsing**: Each page is parsed via BeautifulSoup, removing <script> and <style>
4. **Data Categorization**: Based on predefined keyword buckets
5. **Contact Info**: Regex used to extract emails and phone numbers
6. **Deduplication**: Unique lines filtered to avoid repetition in saved files
7. **File Saving**: Outputs saved as all_data.json and all_data.txt
8. **MongoDB Upload**: Final data pushed with a key = sha256(email + link)
9. **Frontend Response**: Output hash returned as JSON to be used in the web UI

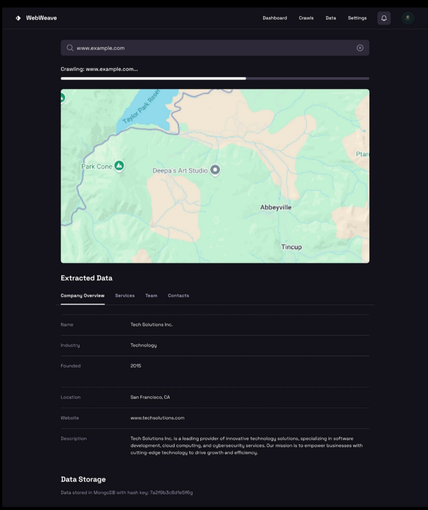
Targeted Categories.

- **Company Overview** – about us, who we are
- **Service Info** – services, solutions, offerings
- **Categories & Subcategories** – industries served
- **Customer Support** – help, support, contact info
- **Legal Info** – terms of use, privacy, cookies
- **News & Updates** – blog posts, announcements
- **Upcoming Events** – webinars, calendars, meetups
- **Portfolio** – case studies, past clients
- **Employees/Team** – bios, leadership pages
- **Emails & Phones** – using regular expression search

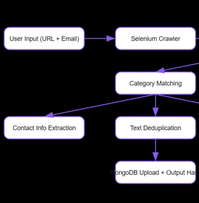
Challenges

- Pages with infinite scrolling or JavaScript rendering needed additional wait logic
- Keyword overlap in categories (e.g., "support" inside legal disclaimers) required careful regex
- Some websites blocked crawling with bot detection — handled via user-agent spoofing (future)
- Memory control for large page link trees (solved with visited URL tracking)
- Long page loads occasionally caused Selenium timeouts

Demo Working



Flowchart



Results, Learnings, and Impact

- Successfully extracted structured content from over 40 corporate and agency websites
- Achieved **>90% categorization coverage** for known sections (About, Services, Contact, etc.)
- Generated fully deduplicated .txt and .json summaries ready for downstream NLP use
- **<10 seconds** median crawl time for most static websites
- Clean integration with web frontend using stdout JSON messaging