Rufus – Intelligent Web Data Extraction for LLMs

The Objective

To develop Rufus, a tool that can:

Scrape websites in a smart way considering user prompts.

Pulling pertinent data and synthesizing it into structured formats (JSON/CSV).

Deliver clean, ready-to-use data for downstream applications similar to Retrieval-Augmented Generation (RAG) pipelines.

Approach

Repository Setup:

Organized a modular structure with separate components for scraping, synthesis, and API logic.

Configured a virtual environment and documented dependencies to ensure scalability and maintainability.

Core Features:

Crawling: Implemented recursive crawling for all nested links with a maximum depth defined by the user.

Dynamic Content: Integrated with Playwright to scrape pages that are rendered through JavaScript.

Prompt-Based Filtering: The extraction includes only the content that matches the keywords from the prompt.

Error Handling & Logging: Strong error handling added and the logging centralized to rufus.log file.

Output Synthesis:

The data is output to JSON or CSV to fit into RAG workflows.

API Design:

Developed a REST API on FastAPI with the following endpoints:

POST /nested\_scrape: Invoke the scraping of some website.

POST /crawl\_and\_synthesize: Enable crawling and saving data in structured formats.

GET /download: To download generated files.

Testing:

Delivered unit tests for scraping, filtering, and synthesis.

Tested with Swagger and sample inputs for API endpoints as well.

Challenges Solved

Handling nested links and dynamic content. Extracting relevant data based on user-defined prompts. Ensuring robust error handling and user-friendly logging.