Web Scraping and Data Structuring Documentation

# Objective

To extract biomarker-related content from the given Google Sheet link, clean and organize the data using programmatic methods and tools such as Selenium, BeautifulSoup, and Grok (via LLM prompts), and finally convert it into structured formats like CSV, JSON, or Excel.

# Input Source

URL: https://docs.google.com/spreadsheets/d/1gnFSagLXtFyx2UgNa4oSPa4PV12R-LROPHHFFj2ICts/edit?usp=sharing  
Tool Used for Scraping: Selenium  
Libraries Used: selenium – For web navigation, bs4 – For HTML parsing, json, csv, pandas – For data transformation and storage, grok (via prompt) – For logical structuring and disambiguation

# Deliverables

- Scraping.ipynb – Jupyter notebook with the complete scraping logic.  
- cleaning.ipynb – Jupyter notebook for cleaning, flattening nested JSONs, and grouping disease-biomarker relations.  
- biomarker\_data\_combined.json – Raw scraped & combined dataset.  
- disease\_grouped.json – Final cleaned, disease-wise structured output.  
- Report (this file)

# Data Scraped

From the Google Sheet, the following key fields were extracted and cleaned:  
- Biomarker Name  
- Associated Disease(s)  
- Application  
- Biological Source  
- Mechanism / Pathway  
- Other Relevant Terms  
Some entries contained lists of biomarkers grouped under a single disease, which were iteratively flattened using logic written in cleaning.ipynb.

# Cleaning & Structuring Approach

1. Flattening:  
- Parsed each biomarker individually.  
- Normalized nested fields like Biomarkers array into flat dictionaries.  
  
2. Standardization:  
- Unified casing, synonym handling (e.g., LDL (Low-Density Lipoprotein) and LDL Cholesterol)  
- Consolidated spelling variations and duplicates.  
  
3. Disease-wise Grouping:  
- Grouped all relevant information under unique diseases.  
- Used Grok (LLM-based prompt) for:  
 - Merging synonymous biomarker entries  
 - Logical grouping of associated terms  
 - Resolving entity types (disease vs symptom vs process)

# Grok Prompt Used

The Grok LLM was used to assist with high-level grouping using the following sample prompt:  
  
"Given a list of biomarker entries with potential nesting and inconsistent formats, group all information by disease name. For each disease, list all associated biomarkers, applications, biological sources, mechanisms/pathways, and other relevant terms without duplications."  
  
This enabled semantic understanding that traditional rule-based cleaning might miss.

# Output Format

Final dataset is stored in:  
  
- JSON – disease\_grouped.json  
Example:  
{  
 "Disease": "Heart attack",  
 "Associated Biomarkers": "...",  
 "Applications": "...",  
 ...  
}  
  
Optional CSV/Excel export can be done via:  
import pandas as pd  
df = pd.read\_json('disease\_grouped.json')  
df.to\_csv('disease\_grouped.csv', index=False)  
df.to\_excel('disease\_grouped.xlsx', index=False)

# Assumptions

- Repetitive entries of similar biomarkers with slight name changes are assumed to be the same (e.g., HDL and HDL Cholesterol).  
- Some nested structures in JSONs are intended to be flattened for tabular use.  
- Grok's output was validated manually for factual consistency.

# Challenges Faced

- Inconsistent formatting in the original source (Google Sheet).  
- Multiple nested structures required recursive parsing.  
- Logical grouping (disease-centric) was not straightforward due to variation in naming and structure.  
- Grok LLM output needed validation, as LLMs can sometimes hallucinate relationships.

# Submission Deadline

13th April 2025 – All deliverables submitted before this date.

# Final Checklist

Deliverable | Status  
------------|--------  
Python Script / Notebook | Provided (Scraping.ipynb, cleaning.ipynb)  
Raw Structured Data | biomarker\_data\_combined.json  
Final Cleaned Dataset | disease\_grouped.json  
Documentation Report | You’re reading it now!