Automated Grant Opportunity Data Collection

Objective

The goal of this task is to assess your ability to write automated code to extract structured data from unstructured or semi-structured web pages. You are required to build a scraper/crawler that takes a list of websites as input and outputs a clean, deduplicated list of grant opportunities.

Input

The input will be a list of target websites (URLs) where grant opportunities are listed (see **Test URLs** section below).

Requirements

1. Output Schema

Your script must output a JSON list where each object represents a single grant opportunity. The object must strictly follow the schema below.

JSON Structure:

```
{
  "agency": "String",
  "application_link": "String",
  "description": "String",
  "ein": Number or Null,
  "last_update": "String" or Null,
  "name": "String",
  "notice_id": "String",
  "organizations": "String",
  "published_date": "String" or Null,
  "response_date": "String" or Null,
  "url": "String"
}
```

Field Definitions:

Field	Type	Description
agency	String	The specific name of the funding agency or division (e.g., **"DARPA"**, **"NSF"**, **"Department of Energy"**). Do not use generic labels like "Federal Agency". For foundation, just state foundation
application_link	String	The direct URL to the application form or portal. If not found, use "NA".
description	String	A comprehensive text summary of the grant, including eligibility or purpose.
ein	Number	The organization's Employer Identification Number (Tax ID). Use null if not found.
last_update	String/Null	The date the grant information was last modified on the source site.
name	String	The official title of the grant opportunity.
notice_id	String	A unique identifier for the specific opportunity. If the site does not provide an ID, generate one using the format [EIN]_[Index] (e.g., 113227901_1).

organizations	String	The full name of the organization providing the grant.
published_date	String/Null	The date the grant was originally posted.
response_date	String/Null	The application deadline.
url	String	The specific source URL where this opportunity was found.

2. Constraints & Logic

- **Automation:** The process must be fully automated. No manual copy-pasting is allowed.
- **Deduplication:** The final list must not contain duplicate grants. If the same grant appears on multiple input pages, it should only appear once in the output.
- **Robustness:** The code should handle potential errors (e.g., 404s, timeouts, redirects) gracefully without crashing.

Deliverables

1. **Source Code:** A Python script (or Node.js) that performs the extraction.
2. **Output File:** A .jsonl file containing the extracted data from the test URLs provided.
3. **Instructions:** A brief README explaining how to run your script and install dependencies.

Test URLs

(Please run your script against the following URLs for this task)

1. **Marsh Foundation:** https://www.wellsfargo.com/private-foundations/marsh-foundation
2. **Leach Foundation:** https://www.leachfoundation.org/whatwesupport
3. **DeVos Foundation:** https://dmdevosfoundation.org/#about
4. **Ludwick Foundation:** https://www.ludwickfoundation.org/about.htm
5. **Ludlow Foundation:** http://ludlowfoundation.org/grants.html
6. **Michael Foundation:** https://www.michaelfound.org/
7. **Kuyper Foundation:** https://kuyperfoundation.org/grant-guidelines/
8. **Bowsher-Booher Foundation:** https://www.wellsfargo.com/private-foundations/bowsher-booher-foundation/
9. **Barceloux-Tibessart Foundation:** https://www.barceloux-tibessart.org/

10. **Mill Foundation:** http://www.themillfoundation.org/
11. **Sarkeys Foundation:** https://www.sarkeys.org/grant-guidelines
12. **Fasken Foundation:** https://www.faskenfoundation.org/giving-guidelines
13. **Carnahan-Jackson Foundation:** https://www.chautauquagrants.org/local-funding/carnahan-jackson-foundation
14. **PCI Foundation:** https://www.pci-foundation.org/proposal-guidelines-checklist-faq