Scientific Article Scraper - Complete Documentation

# 1. Project Overview

This project is a scientific article web scraping and management system designed to automate the extraction, storage, and analysis of PubMed articles. It includes a keyword-driven scraper, a PDF downloader, metadata parser, MongoDB integration, citation generator, and a user-friendly UI for article review and analytics.

# 2. System Requirements

To set up and run this project, ensure the following requirements are met:

* - Operating System: Windows / Linux / macOS
* - Python 3.7 or higher installed
* - pip (Python package manager)
* - MongoDB installed and running locally or on a remote server

# 3. Required Python Libraries

Install the required packages using pip:

pip install -r requirements.txt

* The key libraries include:
* - requests
* - beautifulsoup4
* - pymongo
* - flask
* - pandas
* - openpyxl
* - python-docx
* - lxml
* - htmx
* - tqdm

# 4. Project Setup

1. Clone or extract the project folder.  
2. Navigate to the folder in terminal.  
3. Install dependencies using pip.  
4. Ensure MongoDB is running.  
5. Run the scraper via SciCom.py.  
6. Start the web interface with Flask by running app.py.

# 5. Functional Modules

- SciCom.py: Main script that performs scraping and exports CSV.  
- app.py: Flask application for article review and analytics.  
- pdf\_utils.py: Handles PDF download from PMC.  
- pubmed\_utils.py: Extracts metadata from PubMed.  
- db\_utils.py: Interfaces with MongoDB.  
- config.py: Stores configuration variables.  
- utils.py: Utility functions for sanitization and citations.  
- templates/: HTML pages including base, index, and analytics.  
- static/: Contains logos and styling resources.

# 6. Using the Scraper

1. Prepare keywords.csv with a column 'keyword'.  
2. Prepare abbreviations.csv with 'Term' and 'Abbreviation'.  
3. Run SciCom.py.  
4. It will fetch new PubMed articles, download PDFs, store in MongoDB, and export results as CSV.  
5. Citations will be saved in the 'citations' folder.

# 7. Web Interface

- Home Page: View and approve/reject articles.  
- Move to Folder: Manually move approved PDFs to local folders.  
- Undo: Revert approval/rejection.  
- Analytics Page: View summary of article stats with graphs.

# 8. Output Files and Folders

- pdfs/: Contains all downloaded PDFs.  
- citations/: Contains grouped text files of citations.  
- exports/: Stores CSV exports after each run.  
- logs/: MongoDB stores run\_logs collection with timestamps and errors.

# 9. Maintenance and Extension

- Add new keywords or abbreviations by updating the respective CSV files.  
- No code changes are required after new CSV updates.  
- Monitor run\_logs in MongoDB to troubleshoot errors.