Project Report: PubMed Fetcher CLI Tool

Title:

PubMed Fetcher CLI – A Command-Line Tool for Research Metadata Retrieval from PubMed

---

1. Objective :

To design and develop a lightweight **command-line tool** that:

- Accepts search queries related to medical/scientific research
- Fetches data from **PubMed API**
- Filters for **non-academic authors**
- Extracts structured metadata (title, author, affiliation, email, date)
- Saves the output to a **CSV file** for offline use or further analysis

---

2. Motivation:

Modern research often involves scanning hundreds of papers. Manually reviewing metadata wastes time and leads to inefficiency.

The motivation was to:

- Speed up literature review tasks
- Identify papers from industry (non-academic)
- Automate metadata extraction
- Make the tool installable and usable from terminal or CI pipelines

---

3. Approach and Methodology

➤ A. Architecture

User Input → CLI Parser → PubMed API (Entrez) → XML Parsing → Filtering → CSV Output

➤ B. Method Breakdown

1. **Input Handling**
   Accepts queries like "covid vaccine" via CLI using argparse.

2. **PubMed ID Fetching**
   Uses esearch API endpoint to retrieve PubMed IDs.

3. **Metadata Fetching**
   Uses efetch API to download full metadata in **XML** format.

4. **Filtering Logic**
   - Detect non-academic authors using keyword-based heuristics
   - Keywords like "inc", "pharma", "biotech" are flagged as industry
   - Academic keywords like "university", "hospital" are excluded

5. **Email & Author Extraction**
   - Uses regex to extract emails
   - Filters corresponding authors with company affiliations

6. **Output Handling**
   - Saves as a well-formatted .csv
   - Allows output path and debug logging via flags

---

4. Tech Stack

Layer	**Tool/Library**
CLI	argparse
API Requests	requests
Data Parsing	xml.etree.ElementTree, re
Data Output	csv, pandas (optional)
Packaging	Poetry
Deployment	TestPyPI

---

5. Project Structure

```
pubmed_fetcher_thanush/
 ├— __init__.py
 ├— fetcher.py      # API interaction
 ├— filter.py       # Parsing, cleaning, filtering
 ├— utils.py        # Text and email utilities
 ├— main.py         # CLI entry point
pyproject.toml      # Poetry config with script setup
```

---

6. Command-Line Usage:

```
get-papers-list "covid vaccine" --file result.csv --max 50 --debug
```

CLI Arguments:

Flag	Description
query	Search keyword (e.g., "cancer vaccine")
--file, -f	Output CSV file path
--max, -m	Max results (default: 100)
--debug, -d	Print verbose output
--help, -h	Show help info

---

7. Sample Output (CSV):

PubMedID	Title	Authors	Affiliations	Emails	Date
12345678	COVID Drug Trial	John Smith	Pfizer Inc.	jsmith@pfizer.com	2024-04-01

PubMedID	Title	Authors	Affiliations	Emails	Date
22345678	AI in Healthcare	Alice Brown	Genentech Ltd.	alice.b@genentech.com	2023-09-10

8. Results:

- Tested queries like:
  covid vaccine, diabetes, ai in healthcare
- Output:
  - 5 to 15 valid papers saved per 100 IDs fetched
  - < 5 seconds average response time
  - Emails found for 60–70% of cases

9. Highlights:

✅ Modular code design
✅ Quick CSV output
✅ Non-academic author detection
✅ Debug-friendly logs
✅ Ready for TestPyPI publication

10. Future Scope:

- ✅ Add support for abstract, keywords, and full-text metadata
- ✅ Integrate NLP to classify paper relevance
- ✅ Export in PDF / BibTeX formats
- ✅ GUI version with Tkinter or Web interface
- ✅ Add unit tests and CI pipelines

11. Developer:

Thanush Shetty
📧 thanushshetty7@gmail.com
💻 GitHub Profile

12. License:

This project is licensed under the **MIT License**.

13. Resources:

- 📚 PubMed API Documentation
- 🧪 TestPyPI Package
- 🧰 Poetry CLI Tool
- 📊 PubMed Query Help