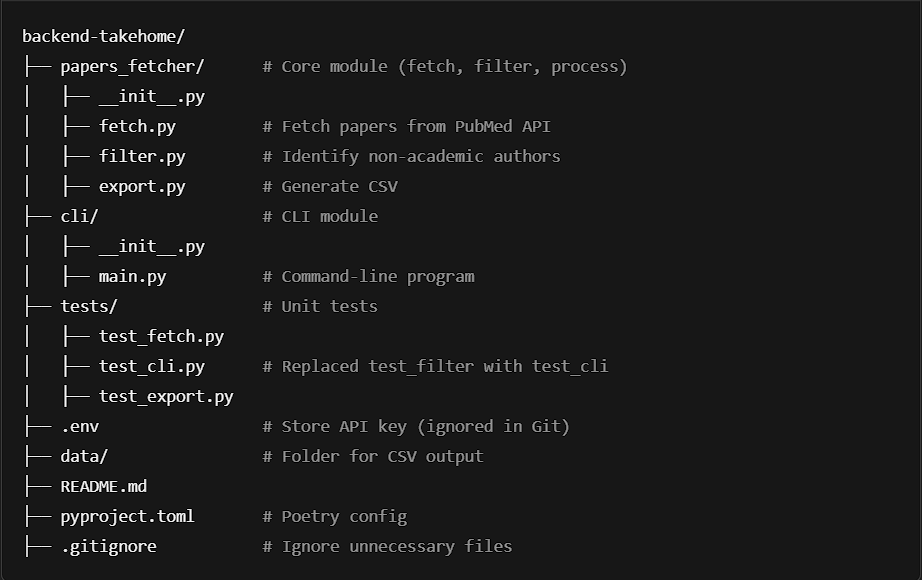
Project Structure



**Requirements Checklist**

**Source of Papers**

* Fetch papers using the PubMed API✅
* Support PubMed's full query syntax✅

**Output Requirements**

* Return results as a CSV file with the following columns:
  + PubmedID✅
  + Title✅
  + PublicationDate✅
  + Non-academic Author(s) ✅
  + Company Affiliation(s) ✅
  + Corresponding Author Email✅

**Command-line Program Features**

* Accept the query as a command-line argument✅
* Provide the following options:
  + -h or --help: Display usage instructions✅
  + -d or --debug: Print debug information during execution✅
  + -f or --file: Specify the filename to save the results✅

**Code Organization and Environment**

* Version Control:
  + Use Git for version control✅
  + Code hosted on GitHub✅
* Dependencies and Setup:
  + Use Poetry for dependency management and packaging✅
  + Running poetry install sets up all dependencies✅
* Execution:
  + Provide an executable command named get-papers-list via Poetry✅

**Documentation**

* Include a README.md file with:
  + How the code is organized✅
  + Instructions on how to install dependencies and execute the program✅
  + Mention any tools used to build the program, along with relevant links✅

**Evaluation Criteria**

* Functional Requirements:
  + Adherence to the problem statement✅
  + Ability to fetch and filter results correctly✅
* Non-functional Requirements:
  + Typed Python: Using types everywhere✅
  + Performance: Efficiency of API calls and processing✅
  + Readability: Clear and maintainable code with appropriate comments and docstrings✅
  + Organization: Logical separation of concerns✅
  + Robustness: Error handling for invalid queries, API failures, or missing data✅

**Bonus Points**

* Break the program into two parts:
  + A module✅
  + A command-line program that uses the module✅
* Publish the module in test-pypi