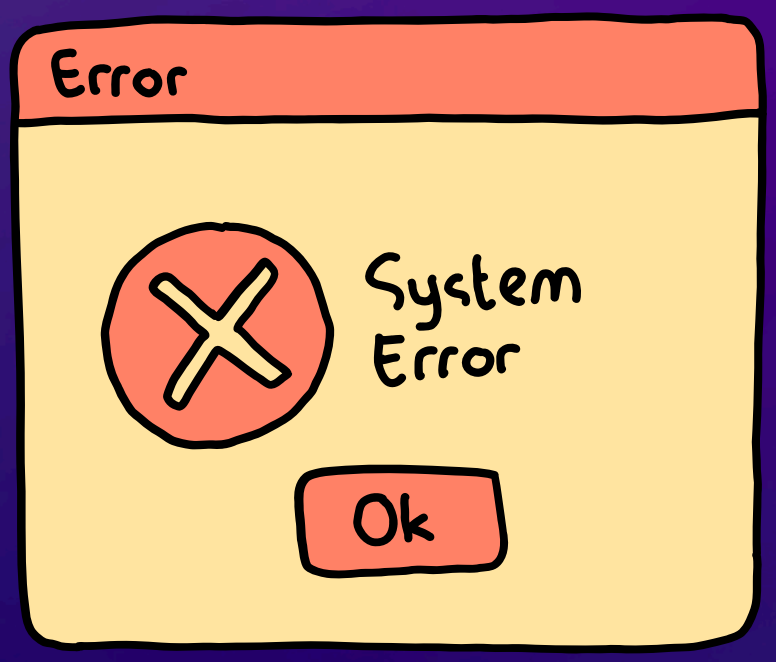
4 MISTAKES
DEVELOPING RELIABLE API WORKFLOWS IN MY PROJECT

That cost me whole week

Skip the struggle I went through



Taha Sayyed
@tahasayyedk00@gmail.com

«SWIPE»

Improper Response Format Parsing for arXiv API

What I did 🤦

I opted for **xml.etree.ElementTree** rather than the documentation-recommended **feedparser**, which led to complexity in parsing the XML response

What I learn 💡

I realized that relying on the API provider's documentation and using their language-appropriate techniques is crucial for smoother development

Using xml.etree.ElementTree make complex

```
entry = root.find('atom:entry', ns)
title = entry.find('atom:title', ns).text
summary = entry.find('atom:summary', ns).text
```

Using feedparser makes it easy

```
entry=feedparser.parse(response_text).entries[0]
title = entry.title
summary = entry.summary
authors = [a.name for a in entry.authors]
```

«SWIPE»

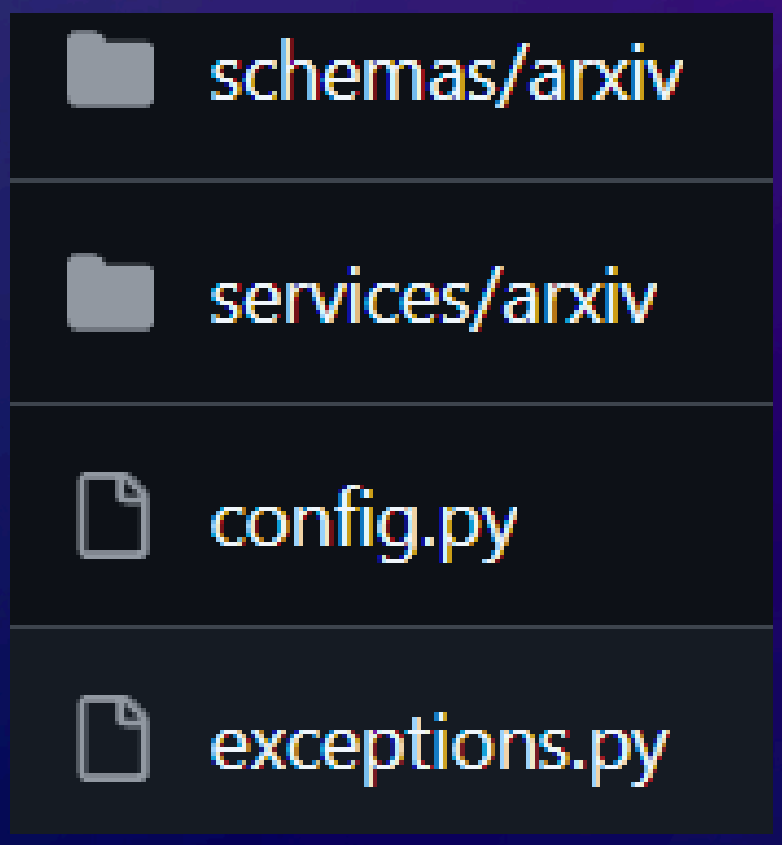
Failed Modular Client Structure

What I did 🤦

I combined the arXiv API response schema with the client logic responsible for fetching, parsing, and rate limiting, instead of separating them into dedicated modules

What I learn 💡

I improved the architecture by separating the API schema, client logic, client factory, and exception handling into dedicated modules such as a factory file for arXiv instances and an exception.py for API errors



«SWIPE»

OOP Absence Increased Complexity

What I did 🤦

I implemented the configuration settings as raw definitions instead of encapsulating them in a class, which resulted in repetitive and unnecessarily lengthy code. Similarly, I defined the API response schema directly without creating a dedicated schema class

What I learn 💡

I learned that configuration and schema definitions should be encapsulated in dedicated classes. This improves reusability, reduces duplication, and keeps the codebase far more maintainable

```python
# Configuration scattered everywhere
BASE_URL = "https://api.example.com"
TIMEOUT = 10

def fetch_data():
    # uses BASE_URL, TIMEOUT directly
    ...
```

Before

```python
class Config:
    BASE_URL = "https://api.example.com
    TIMEOUT = 10

class Client:
    def fetch_data(self):
        url = Config.BASE_URL
```

After

«SWIPE»

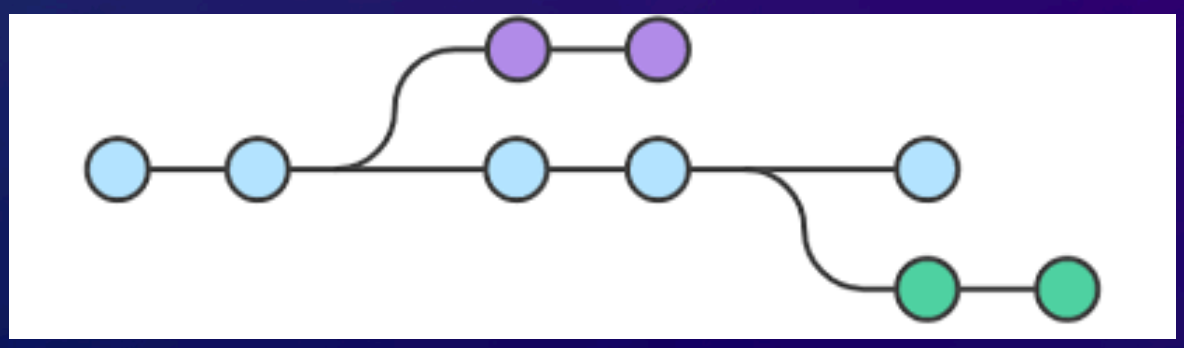
Unmanaged Git Workflow Issues

What I did 🤦

I merged all updates into the master branch instead of following a proper branching strategy with separate branches for the API client, PDF parser, and Ollama service

What I learn 💡

This taught me to follow a structured Git workflow with separate branches for each feature or service. It ensures cleaner commits, safer merges, and more maintainable version control



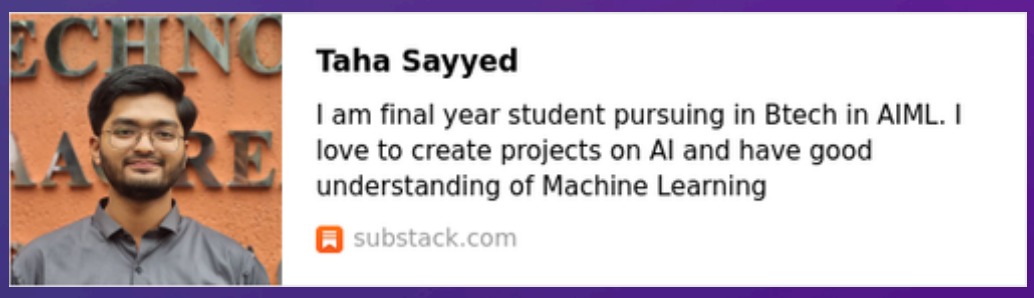
«SWIPE»

WHAT I DO NOW FOR MY PROJECT

- [x] **Follow API docs and recommended parsing libraries.**
- [x] **Build a modular, well-structured client architecture.**
- [x] **Apply OOP principles for cleaner, reusable code.**
- [x] **Use Git branches properly for organized development.**

«SWIPE»

EXPLORE THE FULL BREAKDOWN ON SUBSTACK AND BROWSE MY WORK ON GITHUB.



https://github.com/Taha-Sayyed/arxiv-paper-curator

https://www.linkedin.com/in/taha-sayyed/

Taha Sayyed
@tahasayyedk00@gmail.com