Awesome based on your assignment brief, I put together a working starter kit you can run locally. It hits all the key requirements: Streamlit UI, PDF/Excel parsing, TF-IDF retrieval, and Q&A via a local Ollama model (e.g., `llama3.1:8b` or `mistral`). Grab it here:

Download the project (ZIP)

What you get

- `app.py` — Streamlit app (upload PDFs/XLSX → process → ask questions).
- `parser.py` — PDF text + basic table extraction with `pdfplumber`; Excel parsing with `pandas`.
- `qa.py` — TF-IDF retrieval over extracted chunks; prompts an Ollama SLM with the retrieved context.
- `utils.py` — cleaning + chunking helpers.
- `config.json` — tweak model/host and retrieval params.
- `requirements.txt` — all Python deps.
- `README.md` — copy-paste setup in minutes.
- `sample_data/` — a toy Excel sheet and a tiny "notes" placeholder PDF to test the flow.

How to run (quick)

1. Install Ollama and pull a model:

```
ollama pull llama3.1:8b   # or: ollama pull mistral
```

2. Create a venv and install deps:

```
python -m venv .venv
source .venv/bin/activate     # Windows: .venv\Scripts\activate
pip install -r requirements.txt
```

3. Launch:

```
streamlit run app.py
```

4. Results can be obtained even in Command Prompt by using the following commands:

```
Ollama serve

Ollama run llama2
```

Then upload your financial PDFs/XLSX and ask questions like "What was total revenue in 2023?"

Notes for the rubric

- **Local-only**: No cloud calls—Ollama runs on `localhost:11434`.
- **Conversational Q&A**: The app keeps retrieved chunks and presents the LLM answer, with an expanded to inspect which chunks were used.
- **Error handling**: Basic safeguards with Streamlit status blocks and try/except around parsing/LLM calls.
- **Extendable**: Swap TF-IDF for embeddings, add Camelot/Tabula for stronger PDF tables if you have Java/Ghostscript, or persist a vector index if you want.