FinDocGPT

1. Challenge Tackled

Financial analysts and investors often spend hours extracting key insights from lengthy, complex PDF/DOCX financial documents. Existing tools are slow, fragmented, and lack real-time forecasting. FinDocGPT solves this by enabling users to upload financial documents, instantly extract and summarize key sections, perform Q&A, and forecast stock prices, all within an intuitive web app.
Target users: Financial analysts, investment firms, and individual investors.

2. Tools / ML Models Used

- Python 3.10 - Core programming language
- Streamlit - Web app framework for fast deployment
- PyPDF2 & python-docx - Document parsing
- scikit-learn (TF-IDF) - Information retrieval for Q&A
- yFinance API - Real-time stock data and forecasting
- pandas, numpy - Data processing
- Matplotlib, Plotly - Dynamic charts and visual analytics
- NLTK - Text processing

3. What Worked Well

- TF-IDF Q&A retrieval returned accurate answers for diverse financial document queries.
- yFinance integration delivered real-time stock prices and forecasts instantly.
- Interactive dashboards allowed users to visualize trends without leaving the app.

4. What Was Challenging

- Parsing mixed-format financial documents with inconsistent layouts - solved using a combined PyPDF2 + python-docx pipeline.
- Ensuring fast query responses while handling large files - optimized TF-IDF vectorization and caching.
- Maintaining accurate forecasts in volatile stock scenarios - implemented rolling-average smoothing.

5. How We Spent Our Time

- 0-3h: Problem definition, requirement gathering
- 3-8h: Document parsing + TF-IDF Q&A module
- 8-16h: yFinance integration + stock forecasting
- 16-20h: Dashboard UI with Streamlit, chart integration
- 20-24h: Debugging, polishing, video + report preparation

Optional Reflection

If we had 24 more hours, we'd integrate LLaMA or GPT-4 for deeper natural language analysis and multilingual document support.