

## Gemini Pro Financial Decoder - Streamlit Interface

This module (app.py) provides the main user interface for the Gemini Pro Financial Decoder application. Built using Streamlit, it allows users to upload financial documents, initialize the LLM backend, and generate AI-powered summaries and visualizations.

### What This Module Does

- Sets up the Streamlit UI with a page title, layout, and custom header.
- Initializes the LLM used for financial document summarization.
- Provides a sidebar file uploader for:
  - Balance Sheet
  - Profit & Loss Statement
  - Cash Flow Statement
- On button click, processes the uploaded files:
  - Loads each file
  - Summarizes it using the LLM
  - Displays the summary
  - Generates and shows visualizations

### Dependencies

This module depends on:

- streamlit - for building the UI
- file\_uploader.load\_file() - for loading uploaded documents into a DataFrame
- llm\_handler.initialize\_llm() - for setting up the large language model
- llm\_handler.generate\_summary() - for generating text summaries
- visualizer.create\_enhanced\_visuals() - for visual analysis of financial data

## UI Layout Overview

- Header: Centered title and description
- Sidebar: File uploaders for three financial statements
- Main Panel: When "Generate Comprehensive Financial Analysis" is clicked:
  - Each uploaded document is:
    - Parsed
    - Summarized by LLM
    - Visualized using charts/graphs

## LLM Fallback Handling

If the LLM fails to initialize, the app:

- Displays an error
- Stops further execution to avoid crashes

## Sample Code:

```
llm = initialize_llm()
```

if llm is None:

```
    st.error("Cannot proceed without LLM initialization")  
  
    st.stop()
```

## Sample Usage (UI Flow)

1. Launch the app:

```
streamlit run app.py
```

2. Upload your financial documents via the sidebar.

3. Click "Generate Comprehensive Financial Analysis".

4. View AI-generated summaries and visual reports directly on the screen.

## Note

This file does not handle:

- Backend LLM configuration
- File parsing logic
- Visualization details

Those are delegated to separate modules to maintain modularity and separation of concerns.