

GEMINI PRO FINANCIAL DECODER

Module: LLM & Styling Logic (llm_handler + styles)

Overview

This module provides two key functionalities for the Gemini Pro Financial Decoder system:

1. **LLM Initialization and Summary Generation**
2. **Custom UI Styling via Streamlit-compatible CSS**

It includes support for Google Gemini Pro model initialization, financial document summarization using prompt templates, and aesthetic enhancements for the Streamlit application interface.

Functionality

◆ **initialize_llm()**

- Initializes the Gemini Pro LLM via langchain_google_genai.GoogleGenerativeAI.
- Supports secure key injection using st.secrets["API_KEY"].
- Uses Streamlit's @st.cache_resource to avoid redundant initialization.
- Handles fallback and displays an error message on failure.

◆ **generate_summary()**

- Accepts:
 - prompt_type: one of "balance_sheet", "profit_loss", "cash_flow"
 - data: a Pandas DataFrame
 - llm: the initialized LLM instance
- Generates a summary based on the top 10 records of the data using the appropriate prompt template.
- Returns LLM-generated insights or an error message.

◆ **Prompt Templates**

- Stored in PROMPT_TEMPLATES as PromptTemplate instances.
- Covers:
 - **Balance Sheet**
 - **Profit & Loss Statement**
 - **Cash Flow Statement**
- Each template asks the LLM to extract clear insights or takeaways.

◆ **load_css()**

- Returns a CSS block as a string.
 - Provides styling for:
 - Background gradients
 - Button hover effects
 - Sidebar appearance
 - Message cards (error/success)
 - Data tables
 - Injected into the app using st.markdown(load_css(), unsafe_allow_html=True)
-

Workflow Summary

1. initialize_llm() configures the LLM once on app load.
 2. Financial data is passed through generate_summary().
 3. A formatted prompt is created using the corresponding template.
 4. The prompt is sent to the Gemini Pro LLM via LangChain integration.
 5. The LLM response is displayed in the UI.
 6. UI appearance is managed using the styles returned by load_css().
-

Contribution

This module was designed and implemented to enable real-time financial summarization and elegant visual styling for the Gemini Pro Financial Decoder. The LLM interface is modular and configurable, while the CSS component ensures a cohesive and modern user experience across all Streamlit screens.