

```
import streamlit as st
import pandas as pd
import google.generativeai as genai

# =====
# Configure Gemini API
# =====
genai.configure(api_key="AIzaSyCAEww4n_Uihmrfj2p7mYpgv_nWDKR7ueI")

model = genai.GenerativeModel("gemini-2.5-flash")

# =====
# Streamlit UI
# =====
st.title("Gemini Pro Financial Decoder")
st.write("Upload financial documents to view charts and AI-generated summaries.")

# File uploaders
balance_sheet = st.file_uploader(
    "Upload Balance Sheet (CSV or XLSX)",
    type=["csv", "xlsx"]
)

profit_loss = st.file_uploader(
    "Upload Profit and Loss Statement (CSV or XLSX)",
    type=["csv", "xlsx"]
)

cash_flow = st.file_uploader(
    "Upload Cash Flow Statement (CSV or XLSX)",
```

```
type=["csv", "xlsx"]  
)  
  
# ======  
# Helper Functions  
# ======  
  
def load_file(file):  
    if file is not None:  
        if file.name.endswith(".csv"):  
            return pd.read_csv(file)  
        elif file.name.endswith(".xlsx"):  
            return pd.read_excel(file)  
    return None  
  
def generate_summary(title, data):  
    data_text = data.head(3).to_string()  
  
    prompt = f"""  
You are a financial analyst.  
Analyze the following {title} data and give a clear, simple summary.  
Highlight key trends and important insights.  
  
Data:  
{data_text}  
"""  
  
    response = model.generate_content(prompt)  
    return response.text  
  
def display_data_chart_and_summary(data, title):
```

```
st.subheader(title)
st.dataframe(data)

numeric_data = data.select_dtypes(include=["number"])
if not numeric_data.empty:
    st.line_chart(numeric_data)

if st.button(f'Generate AI Summary for {title}'):
    with st.spinner("Generating AI summary..."):
        # 🔴 IMPORTANT FIX: send only small data to Gemini
        summary = generate_summary(title, data.head(2))
        st.markdown("### AI Summary")
        st.write(summary)

# =====
# Display Sections
# =====

if balance_sheet:
    bs_data = load_file(balance_sheet)
    display_data_chart_and_summary(bs_data, "Balance Sheet")

if profit_loss:
    pl_data = load_file(profit_loss)
    display_data_chart_and_summary(pl_data, "Profit and Loss Statement")

if cash_flow:
    cf_data = load_file(cash_flow)
    display_data_chart_and_summary(cf_data, "Cash Flow Statement")
```