```python
import streamlit as st
from langchain.agents.agent_types import AgentType
from langchain_experimental.agents import create_csv_agent
from langchain_google_genai import ChatGoogleGenerativeAI
import pandas as pd
import io
import re
import matplotlib.pyplot as plt

# ------------------------------------------------------------------------------------------------------------------------

api_key = "AIzaSyARCmZdn4ld4emtSQDDeKjDqhC4EAiyED8"

st.title(":rainbow[Querying and Plot Graphs with LLMs]")
csv_file = st.file_uploader('Load Your CSV File Here...', type=['csv'])
# ------------------------------------------------------------------------------------------------------------------------
if csv_file is not None:
    try:
        df = pd.read_csv(csv_file)
        st.dataframe(df.head())

        query = st.text_input('Enter Your Query: ')
        csv_buffer = io.StringIO()
        df.to_csv(csv_buffer, index=False)
        csv_buffer.seek(0)

        agent = create_csv_agent(
            ChatGoogleGenerativeAI(google_api_key=api_key, model='gemini-1.5-pro-latest'),
            csv_buffer, verbose=True,
            agent_type=AgentType.ZERO_SHOT_REACT_DESCRIPTION,
            allow_dangerous_code=True
        )

        button = st.button('Submit')
        # ------------------------------------------------------------------------------------------------------------------------
        if button:
            response = agent.invoke(query)
            output = response['output']
            st.divider()
            st.subheader('Response:')
            st.write(output)

            code_block = re.search(r'```python\n(.*?)\n```', output, re.DOTALL)
            if code_block:
                code_to_execute = code_block.group(1)
                try:
                    # Execute the code to generate the plot
                    exec_globals = {'df': df, 'plt': plt}
                    exec(code_to_execute, exec_globals)
                    # Display the plot using st.pyplot
                    st.pyplot(exec_globals['plt'])
                except Exception as e:
                    st.error(f'An error occurred while generating the plot: {e}')
            else:
                st.info('No valid code block found in the response.')
        # ------------------------------------------------------------------------------------------------------------------------

    except Exception as e:
        st.error(f'An Error occurred: {e}')

else:
    st.info('Please upload a CSV file to proceed')
```