I wanted to share some insights about the adjustments I made to your code to address the issues you raised.

Firstly, regarding the error in retrieving the content from the assistant API response, the issue stemmed from the **"format_response()"** function. It was struggling to access the elements within the response due to a mismatch in how it interacted with the data structure. To resolve this, I took a step to better understand the response structure by utilizing the **'print'** function, which helped me identify the correct access points. Consequently, I tweaked the **"format_response()"** function to accurately retrieve the response content.

Now, onto the second issue involving the frontend layout adjustments to fix the user input in the bottom. I opted to utilize Chat elements from Streamlit, which offers useful commands for building conversational apps. I found a helpful resource in their documentation that outlines implementing the OpenAI API using a conversational layout. You can access it here: [Link to the documentation]

To rectify this, I removed the **"chat_display"** function that utilized formatted strings. Instead, I reworked the **"get_response"** function to manage the chat conversation's frontend layout. Additionally, I introduced the **"st.spinner("Thinking...")"** feature to enhance the user experience while waiting for the assistant's reply.

Addressing the primary tasks led me to discover a couple of smaller issues during testing. One of these was the persistence of conversation history when switching between assistants. To ensure a seamless user experience, I made adjustments within the **"main"** function to reset the conversation history when changing assistants.

Furthermore, I encountered a thread creation issue. Previously, when switching assistants, the conversations remained within the same thread, resulting in mixing dialogues. To prevent this confusion, I modified the **"create_and_run_thread"** function. Now, each change of assistant initiates a new thread for separate conversations, avoiding any mixing.

Lastly, while testing with the provided assistant API keys, I encountered an error suggesting either invalid keys or non-existent assistants. However, when using my own assistant keys, everything functioned smoothly. I recommend double-checking the validity of the provided assistant keys and ensuring that the respective assistants are created and accessible.

I sincerely hope that these adjustments align with your expectations and significantly improve the functionality and user experience of your code. If there's anything else I can assist you with or if further clarification is needed, please don't hesitate to reach out.

Best regards,