```python
import streamlit as st
from io import StringIO
from bs4 import BeautifulSoup
from lxml import html
import pandas as pd

NOISE_WORDS = ['Copy code','Copy']

def convert_df2csv(df, index=True):
    return df.to_csv(index=index).encode('utf-8')

def create_new_cell(contents):
    # https://discourse.jupyter.org/t/how-to-programmatically-add-serveral-new-cells-in-a-notebook-in-jupyterlab/4323
    from IPython.core.getipython import get_ipython
    shell = get_ipython()
    payload = dict(
        source='set_next_input',
        text=contents,
        replace=False,
    )
    shell.payload_manager.write_payload(payload, single = False)

def htm2txt(html_txt):
    return html.fromstring(html_txt).text_content().strip()

def is_noise_word(html_txt):
    return htm2txt(html_txt) in NOISE_WORDS

st.set_page_config(
     page_title='parse Claude chat',
     layout="wide",
     initial_sidebar_state="expanded",
)

INPUT_FILENAME = ""

st.subheader("Convert Claude.AI chat history")

st.markdown("""#### <span style="color:green">Upload a saved chat in HTML file</span>""", unsafe_allow_html=True)
txt_file = st.file_uploader("Upload", key="upload_txt")
if txt_file is not None:
    INPUT_FILENAME = txt_file.name
    # To convert to a string based IO:
    html_txt = StringIO(txt_file.getvalue().decode("utf-8")).read()

    soup = BeautifulSoup(html_txt, "html.parser")
    results = soup.findAll("div", class_="contents")

    cells = []
    for i in range(len(results)):
        v = results[i].prettify()
        if is_noise_word(v): continue
        cells.append(v)

    st.write(len(cells))
    chat_data = []
    for i in range(0, len(cells), 2):
        st.markdown("""##### <span style="color:red">Q:</span>""", unsafe_allow_html=True)
        st.markdown(cells[i], unsafe_allow_html=True)
        st.markdown("""##### <span style="color:blue">A:</span>""", unsafe_allow_html=True)
        st.markdown(cells[i+1], unsafe_allow_html=True)
        chat_data.append([cells[i], cells[i+1]])
```

```
st.markdown("""#### <span style="color:green">Download chat to a CSV file</span>""", unsafe_allow_html=True)
if chat_data and INPUT_FILENAME:
    out_filename = ".".join(INPUT_FILENAME.split(".")[:-1]) + ".csv"
    df_chat = pd.DataFrame(chat_data, columns=["Question", "Answer"])
    # st.dataframe(df_chat)
    st.download_button(
        label="Download",
        data=convert_df2csv(df_chat, index=False),
        file_name=out_filename,
        mime='text/csv',
    )

```