```
import streamlit as st
import pandas as pd
from openai import OpenAI
from sqlalchemy import create_engine, text
import streamlit_authenticator as stauth
import bcrypt

# --- CONFIGURAÇÕES DE PÁGINA E DESIGN ---
st.set_page_config(page_title="AI DOC PRO", page_icon="📄", layout="wide")

# Estilo CSS para o tema Dourado/Escuro
st.markdown("""
    <style>
    .main { background-color: #0e1117; color: #fafafa; }
    .stButton>button { background-color: #d4af37; color: black; font-weight: bold;
border-radius: 5px; }
    .stTextInput>div>div>input { background-color: #262730; color: white; border: 1px solid
#d4af37; }
    </style>
    """, unsafe_allow_html=True)

# --- CONEXÃO COM BANCO DE DATA (SECRETS) ---
def get_db_connection():
    db_url =
f"postgresql://{st.secrets['db_user']}:{st.secrets['db_pass']}@{st.secrets['db_host']}/{st.secrets
['db_name']}"
    return create_engine(db_url)

# --- SISTEMA DE AUTENTICAÇÃO ---
# Nota: Em um sistema real, os usuários viriam do banco de dados
names = ["Administrador"]
usernames = ["admin"]
passwords = ["admin123"] # O authenticator fará o hash automaticamente

authenticator = stauth.Authenticate(
    {"usernames": {usernames[0]: {"name": names[0], "password": passwords[0]}}},
    "cookie_doc_pro", "key_doc_pro", cookie_expiry_days=30
)

name, authentication_status, username = authenticator.login("Login", "main")

if authentication_status:
    authenticator.logout("Sair", "sidebar")
    st.sidebar.title(f"Bem-vindo, {name}")

    # --- INTERFACE PRINCIPAL ---
    st.title("📄 AI DOC PRO - Gerador Estratégico")
```

```
    aba1, aba2 = st.tabs(["Gerar Documento", "Histórico"])

    with aba1:
        st.subheader("Configurações do Documento")
        tipo_doc = st.selectbox("Tipo de Documento", ["Contrato de Prestação de Serviço",
"Relatório Técnico", "Parecer Jurídico", "Manual de Operação"])
        detalhes = st.text_area("Descreva os detalhes específicos (Nomes, prazos, cláusulas
extras):")

        if st.button("Gerar Documento com IA"):
            if detalhes:
                with st.spinner("A IA está redigindo com precisão técnica..."):
                    try:
                        client = OpenAI(api_key=st.secrets["openai_key"])
                        prompt = f"Aja como um especialista em redação de {tipo_doc}. Crie um
documento formal e exato baseado nestes dados: {detalhes}. Use linguagem técnica e
estruturada."

                        response = client.chat.completions.create(
                            model="gpt-4",
                            messages=[{"role": "user", "content": prompt}]
                        )

                        resultado = response.choices[0].message.content
                        st.success("Documento Gerado!")
                        st.markdown("---")
                        st.markdown(resultado)

                        # Salvar no Histórico (Banco de Dados)
                        engine = get_db_connection()
                        with engine.connect() as conn:
                            conn.execute(text("INSERT INTO historico_documentos (tipo_doc,
conteudo) VALUES (:t, :c)"),
                                         {"t": tipo_doc, "c": resultado})
                            conn.commit()

                    except Exception as e:
                        st.error(f"Erro na geração ou banco de dados: {e}")
            else:
                st.warning("Por favor, insira os detalhes do documento.")

    with aba2:
        st.subheader("Arquivos Salvos na Nuvem")
        try:
            engine = get_db_connection()
            query = "SELECT data_criacao, tipo_doc, conteudo FROM historico_documentos
ORDER BY data_criacao DESC"
            df = pd.read_sql(query, engine)
```

```
            st.dataframe(df, use_container_width=True)
        except:
            st.info("Nenhum histórico encontrado ou banco de dados não configurado.")

elif authentication_status is False:
    st.error("Usuário ou senha incorretos")
elif authentication_status is None:
    st.warning("Por favor, insira suas credenciais para acessar o AI DOC PRO.")
```