# # 🤖 Chatbot Marketplace - Atendimento com Streamlit e openAI

# Este é um projeto de chatbot interativo construído com [Streamlit](https://streamlit.io/), voltado para simular um atendimento ao cliente em um marketplace de eletrônicos.

# ## 🚀 Funcionalidades

# - Interface web interativa com Streamlit.

# - Histórico de conversas com rolagem automática.

# - Botão \*\*Limpar Chat\*\* para limpar apenas o histórico.

# - Botão \*\*Reiniciar Chat\*\* para resetar o histórico e o chatbot.

# - Botão \*\*Fechar Bot\*\* que encerra o atendimento e desativa a interface de entrada.

# ## Estrutura

# chatbot\_streamlit/

# ├── app.py # Interface Streamlit

# ├── chatbot\_logic.py # Lógica RAG + LLM fallback

# ├── respostas\_padrao.py # Base de conhecimento (perguntas/respostas)

# ├── products.py # Lista de produtos fictícios

# ├── proposta.py # Geração de proposta

# ├── requirements.txt # Dependências do projeto

# └── README.md # Documentação

# ## Como rodar

# ```bash

# pip install -r requirements.txt

# streamlit run app.py

# ## Requisitos

# streamlit

# openai

# python-dotenv

# ## Instalação

# 1. Clone o repositório ou crie a pasta do projeto e adicione os arquivos conforme estrutura acima.

# 2. Crie e ative um ambiente virtual (recomendado):

# python -m venv venv

# source venv/bin/activate # Linux/Mac

# venv\Scripts\activate # Windows

# 3. Instale as dependências

# pip install -r requirements.txt

# ## Execução do Chatbot

# Rode o comando para iniciar o app Streamlit:

# streamlit run app.py

# ## Arquivos Principais

# app.py : Interface da aplicação, captura o input do usuário e exibe as respostas do chatbot.

# chatbot\_logic.py : Inicializa o modelo "gpt-3.5-turbo".

# Gera respostas baseadas no histórico da conversa para manter o contexto.

# products.py : Contém a base de dados dos produtos do marketplace fictício, com nomes e preços.

# ## Fluxo de Conversa

# O usuário inicia perguntando sobre interesse em comprar algum produto.

# O bot sugere categorias e produtos disponíveis.

# O usuário escolhe um produto.

# O bot confirma a escolha e pede as informações (nome, e-mail, telefone).

# O bot gera uma proposta personalizada e fornece um link de checkout fictício.

# ## Pontos de Melhoria

# Adicionar persistência do histórico em banco de dados.

# Implementar autenticação e segurança.

# Permitir múltiplos produtos no pedido.

# Ajustar prompt do LLM para respostas mais específicas e naturais.