Roteiro de Atividade Prática

Nome: Turma: .

**Título da atividade: Criação do *bot***

**Objetivos**

Criar o *bot* conforme vídeos vistos em aula.

**Lista de materiais**

* Computador com navegador de internet.
* Caderno para anotações.
* Caneta.

**Procedimento experimental**

**Passo 1: Preparação do ambiente**

Certifique-se de ter todos os recursos necessários instalados em seu computador. Isso inclui Python, Flask, OpenAI API, um editor de texto (como Visual Studio Code, Sublime Text ou qualquer outro de sua preferência) e acesso à internet.

Criando e ativando o ambiente virtual Windows:

python -m venv chatbot chatbot\Scripts\activate.

Instando as bibliotecas:

pip install numpy openai python-dotenv tiktoken flask opencv-python uuid.

**Passo 2: Baixando o código-base**

Acesse o repositório do projeto no GitHub fornecido: <https://github.com/alura-cursos/Projeto---3573/archive/128d8b517ec1e2613f7fa7926e277e4853d0d9b3.zip>.

Clone o repositório em seu ambiente local ou baixe o código como um arquivo ZIP e extraia-o.

**Passo 3: Configurando o ambiente virtual**

Abra o terminal ou prompt de comando e navegue até o diretório onde você baixou ou clonou o código do projeto.

Crie um novo ambiente virtual executando o comando:

python -m venv chatbot.

Ative o ambiente virtual criado do diretório .venv

chatbot\Scripts\activate.

**Caso já esteja utilizando o VSCode, certifique-se de que o ambiente virtual esteja selecionado como Interpretador Python. Use o atalho Ctrl+Shift+P e selecione o Interpretador Python dentro do diretório *bin* no diretório .venv.**

**Passo 4: Instalando as dependências**

Com o ambiente virtual ativado, instale as dependências do projeto executando o comando pip install -r requirements.txt.

**Passo 5: Configurando as variáveis de ambiente**

Crie um arquivo .env no diretório raiz do projeto.

Adicione suas credenciais da OpenAI API no formato OPENAI\_API\_KEY=SuaChaveDeAPI.

**Passo 6: Executando o servidor Flask**

No terminal, execute o comando “python app.py” para iniciar o servidor Flask.

**Passo 7: Acessando o *Chatbot* no Navegador**

Abra seu navegador da web e acesse http://localhost:5000 para interagir com o *chatbot*.

**Passo 8: Testando a funcionalidade**

Experimente enviar algumas mensagens para o *chatbot* e observe as respostas geradas.

Certifique-se de testar diferentes tipos de perguntas e verifique se o *chatbot* responde conforme o esperado.

**Passo 9: Explorando o código**

Abra o código-fonte do projeto em seu editor de texto e explore os diferentes arquivos e funções.

Preste atenção especial às partes relacionadas à integração com a OpenAI API e à manipulação das mensagens do usuário.

**Assista ao vídeo “Conectando o *chatbot* com o *front-end*” para auxiliá-lo nesta atividade:**

ALURA. Python e GPT: crie seu *chatbot* com IA. 05 Conectando o *chatbot* com o *front-end*. Disponível em: <https://cursos.alura.com.br/course/python-gpt-crie-chatbot-com-ia/task/147898.> Acesso em: 26 ago. 2024.

**Código-base do vídeo “Criação do *bot*”**

Agora,iremos explorar um código prático para aprofundar nossos conhecimentos e interagir com o *bot*. Para facilitar o processo e torná-lo mais eficiente, decidimos disponibilizar antecipadamente a base do código a seguir.

Assegure-se de copiar e colar este código no Bloco de Notas ou em qualquer editor de texto de sua preferência para que possamos explorá-lo em detalhes.

**Código-base:**

def bot(prompt):

maximo\_tentativas = 1

repeticao = 0

while True:

try:

prompt\_do\_sistema = f"""

Você é um chatbot de atendimento a clientes de um e-commerce.

Você não deve responder perguntas que não sejam dados do e-commerce informado!

"""

response = cliente.chat.completions.create(

messages=[

{

"role": "system",

"content": prompt\_do\_sistema

},

{

"role": "user",

"content": prompt

}

],

temperature=1,

max\_tokens=256,

top\_p=1,

frequency\_penalty=0,

presence\_penalty=0,

model = modelo)

return response

except Exception as erro:

repeticao += 1

if repeticao >= maximo\_tentativas:

return "Erro no GPT: %s" % erro

print('Erro de comunicação com OpenAI:', erro)

sleep(1)

**Assista ao vídeo “Criação do *bot*” para auxiliá-lo nesta atividade:**

ALURA. Python e GPT: crie seu *chatbot* com IA. 08 Criação do *bot*. Disponível em: <https://cursos.alura.com.br/course/python-gpt-crie-chatbot-com-ia/task/147899>. Acesso em: 26 ago. 2024.