Documentation and Workflow for

Website Chatbot Script

Purpose

The Website Chatbot is a Python-based application that allows users to interact with and extract information from any website using OpenAI's GPT-3.5 model. The chatbot can understand website content and engage in meaningful conversations about it.

Requirements

Dependencies

pip install openai selenium beautifulsoup4 webdriver-manager python-dotenv

API Keys

- OpenAI API Key (REQUIRED):
 - Not included in code: For security reasons, the OpenAI API key is not included in the script.
 - o **User responsibility:** Users must use their own OpenAI API key.
 - How to get an API key:
 - 1. Sign up at https://platform.openai.com.
 - 2. Navigate to API section.
 - 3. Generate your API key.
 - o Two ways to add your API key:
 - 1. Environment variable: Set openal api key in your system.
 - 2. Direct input: Enter when prompted by the script.

Other Requirements

• Chrome WebDriver (automatically installed).

Workflow

1. Script Initialization

- Logging Configuration:
 - o Logs critical warnings and errors.
 - o Suppresses unnecessary logs from dependencies like selenium and webdriver manager.
- Welcome Message:
 - Displays an ASCII banner and introduces the chatbot.
- Environment Variables:

o Loads the OpenAI API key using dotenv.

2. User Input Handling

- Prompts the user for a website URL.
- Validates the URL using urlparse to ensure proper format (e.g., http or https schema).
- Allows user commands:
 - o exit: Quit the script.
 - o new: Switch to a different website.

3. Website Content Extraction

• Selenium WebDriver:

- o Configures ChromeDriver in headless mode for efficient content scraping.
- o Loads the webpage and scrolls to dynamically load all content.

• HTML Parsing:

- o Uses BeautifulSoup to parse the website content.
- o Strips unnecessary tags (e.g., script, style, nav, etc.).
- o Extracts meaningful content such as headers (h1, h2, h3) and paragraphs (p, div).

• Content Cleaning:

- o Joins extracted headers and paragraphs.
- o Removes excessive whitespace.
- o Truncates content to fit within OpenAI's 4000-character limit.

4. Chatbot Context Creation

- Builds an initial context with the extracted content and website domain.
- Ensures the chatbot:
 - o Provides accurate and concise answers.
 - o Engages users with a friendly tone and occasional emojis.
 - o Redirects irrelevant questions to the website's content.

5. OpenAI Chatbot Interaction

Conversation History:

o Maintains a history of up to 10 exchanges for context.

• API Request:

- o Queries OpenAI's GPT model (gpt-3.5-turbo) with the user's question and context
- Uses parameters like:
 - max tokens: Limits response length.
 - temperature: Balances creativity and consistency.
 - presence penalty: Encourages concise responses.

• Response Formatting:

- o Cleans up markdown from API responses.
- o Properly formats numbered or bulleted lists.
- o Adds separators for better readability.

6. Error Handling

- User Input Errors:
 - o Prompts for re-entry of invalid URLs.
- Extraction Failures:
 - o Notifies the user if content extraction fails.
- API Errors:
 - o Logs errors and gracefully informs the user.

Features

Core Functionality

1. Website Content Analysis:

- o Dynamic content loading.
- o Automatic scrolling.
- o Content cleaning and formatting.

2. Interactive Chat:

- o Natural language processing.
- o Context-aware responses.
- o Markdown support.
- o Emoji integration.

3. Navigation Commands:

- o new: Switch to a different website.
- o exit: End program.
- o Empty input handling.

Error Handling

- URL validation.
- Content extraction errors.
- API response errors.
- Keyboard interrupts.
- General exceptions.

Usage Guide

Starting the Program

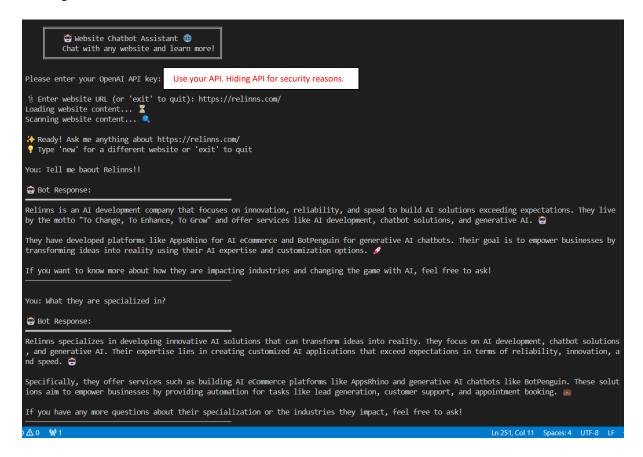
python website chatbot.py

Basic Commands

- Enter URL: Input any valid website URL.
- Ask Questions: Type any question about the website.
- Switch Website: Type new.

• Exit Program: Type exit.

Example Interaction



Best Practices

URL Input

- Always include protocol (http://orhttps://).
- Ensure URL is complete and valid.
- Wait for content loading to complete.

Asking Questions

- Be specific about what you want to know.
- Focus on website-related queries.
- Use clear and simple language.

Performance Considerations

- Large websites may take longer to analyze.
- Complex questions may require more tokens.

• Consider conversation history limits.

Troubleshooting

Common Issues

- 1. API Key Errors:
 - o Check API key validity.
 - o Ensure proper environment variable setup.
- 2. Website Loading Issues:
 - o Verify internet connection.
 - o Check URL accessibility.
 - Allow sufficient loading time.
- 3. Response Errors:
 - o Check input formatting.
 - o Verify website content extraction.
 - o Monitor API rate limits.

Support

For issues related to:

- OpenAI API: Visit OpenAI support.
- Selenium: Check Selenium documentation.
- General bugs: Review error messages and logs.

Security Notes

API Key Security

- NEVER share your OpenAI API key.
- NEVER commit API keys to version control.
- Each user must use their own API key.
- Consider using environment variables for API key storage.

General Security

- No sensitive data is stored.
- Website content is processed locally.
- Secure HTTPS connections preferred.
- Session data is temporary and not persisted.

Thank You

Thank you for taking the time to use and explore the Website Chatbot documentation. I hope you find it helpful and informative!