

Documentation for Poppleton Interactive Chat System

Overview

This program simulates an interactive chat system between a user and a randomly assigned agent. It supports basic chat interactions, admin access, keyword-based responses, random jokes, and the ability to reconnect after disconnection. It also records all conversations with timestamps in a text file for training and reference purposes.

Features

- **Chat with Random Agents:** Users can chat with a randomly assigned agent.
 - **Keyword-based Responses:** Predefined keywords trigger specific responses.
 - **Jokes:** Users can request jokes during the conversation.
 - **Admin Access:** Admins can modify keyword responses, jokes, and random responses.
 - **Conversation Logging:** All conversations are recorded with timestamps for future analysis.
 - **Reconnection Support:** Users can reconnect after disconnection.
-

Functionality

1. SelectionUI_Main

The entry point of the program. It:

- Welcome the user.
 - Assigns a random agent for interaction.
 - Loads responses from a JSON file.
 - Starts the chat loop.
-

2. Username

Prompts the user to enter their name and format it (first letter capitalized).

3. Available_agents

Randomly selects an agent from the predefined list of agents.

4. read_response_file

Loads predefined keywords, jokes, and random responses from a responses.json file. If the file is not found, the program exits.

5. record_conversation

Logs the entire conversation into a text file with the following details:

- **Filename:** <username>_conversation.txt
 - **Header:** Includes the user name, agent name, and timestamp.
 - **Entries:** Each message is prefixed with a timestamp.
-

6. disconnect_agent

Handles user disconnection. It:

- Logs the disconnection message.
 - Prompts the user to reconnect.
 - If the user agrees, starts a new chat session.
-

7. Authentication

Provides an authentication interface for admins. Only valid credentials (admin_id=10037 and admin_pass="Admin") grant access to the admin panel.

8. Admin

The admin panel allows the admin to:

1. Add or modify keyword-based responses.

2. Add jokes.
3. Add random responses.

9. Append_keywords

Allows the admin to add new keywords and their associated responses to the responses.json file.

10. Append_jokes

Allows the admin to add new jokes to the responses.json file.

11. Append_random_responses

Allows the admin to add random responses to the responses.json file.

12. chat_loop

The core of the program. It:

- Handles the conversation between the user and the agent.
 - Triggers specific responses based on user input.
 - Logs each interaction.
 - Detects keywords and delivers corresponding responses.
 - Manages the disconnection process.
-

13. reconnect_session

Allows the user to reconnect after disconnection. A new agent is assigned, and the chat session resumes.

File Structure

1. responses.json

A JSON file containing:

- **Keywords:** A dictionary mapping keywords to their responses.
- **Jokes:** A list of jokes.
- **Random Responses:** A list of responses for unrecognized inputs.

Example:

```
{  
  "keywords": {  
    "hello": "Hello, {name}! How can I help you?",  
    "fees": "The fees are hella expensive, {name}. Save up!"  
  },  
  "jokes": [  
    "Why was the JavaScript developer sad? Because they didn't know how to 'null' their feelings."  
  ],  
  "random_responses": [  
    "Hmm, let me think about it, {name}.",  
    "That's beyond my expertise, but I can learn more, {name}!"  
  ]  
}
```

How to Use

Prerequisites

- Python 3.x installed.
- A valid responses.json file in the same directory.

Steps

3. Implement a GUI for easier interaction.

Author

Developed by Nikhilesh.