ChatGPT Interactive CLI Chat Application

A Python-based command-line interface (CLI) application that enables interactive conversations with ChatGPT through the Mimo AI API. This project provides a seamless chat experience with conversation thread management capabilities.

Features

- **Interactive Chat Interface**: Real-time conversation with ChatGPT directly from your terminal
- **Thread Management**: Maintain conversation context across multiple messages
- **Multi-Thread Support**: Start new conversation threads while preserving thread history
- **Simple Commands**: Easy-to-use commands for navigation and control
- **Stateful Conversations**: Automatic thread ID tracking for contextual responses

Configuration

API Key Setup

1. Obtain your Mimo OpenAI API key from Mimo AI
2. Set the API key as an environment variable:

On Linux/Mac:

bash

```
export MIMO_OPENAI_API_KEY="your-api-key-here"
```

On Windows:

cmd

```
set MIMO_OPENAI_API_KEY=your-api-key-here
```

Or add to your shell profile for persistence:

bash

```
echo 'export MIMO_OPENAI_API_KEY="your-api-key-here"' >> ~/.bashrc
source ~/.bashr
```

Running the Application

1. Open the Jupyter Notebook:

```
jupyter notebook "ChatGPT Project.ipynb"
```

2. Run the cells in order:
   - Cell 1: Import dependencies and initialize API connection
   - Cell 2: Set up request headers
   - Cell 3: Define the message sending function and initialize variables
   - Cell 4: Start the interactive chat loop

Converting to Python Script

To run as a standalone Python script:

python

```
python chatgpt_cli.py
```

Project Structure

```
ChatGPT-CLI-Project/
│
├── ChatGPT Project.ipynb    # Main Jupyter Notebook
├── README.md               # This file
└── requirements.txt        # Python dependencies
```

How It Works

Architecture Overview

The application follows a simple request-response pattern:

1. **Initialization**:
   - Loads API key from environment variables
   - Sets up HTTP headers for authentication
   - Initializes thread tracking variables
2. **Message Flow**:

```
User Input → send_message() → Mimo API → ChatGPT → Response → Display
```

3. **Thread Management**:
   - Each conversation maintains a unique `threadId`
   - The API automatically manages conversation context
   - Users can start new threads to reset context

Use Cases

- **Learning & Education**: Quick access to AI assistance for coding questions
- **Brainstorming**: Generate ideas and explore concepts interactively
- **Code Debugging**: Get help with programming challenges
- **Research**: Quick information lookup and explanations
- **Writing Assistance**: Draft content, improve text, or get suggestions

Contributing

Contributions are welcome! Here's how you can help:

1. Fork the repository
2. Create a feature branch (`git checkout -b feature/AmazingFeature`)
3. Commit your changes (`git commit -m 'Add some AmazingFeature'`)
4. Push to the branch (`git push origin feature/AmazingFeature`)
5. Open a Pull Request

License

This project is licensed under the MIT License - see the LICENSE file for details.

Acknowledgments

- Powered by Mimo AI
- Built with Python and the `requests` library
- Inspired by the need for simple CLI access to ChatGPT

Contact

For questions or feedback, please open an issue on GitHub.