

Python Rule-Based Chatbot

Firdaush Alam

Contents

1	Project Overview	2
2	Author Information	2
3	Features	2
4	Functionality & Code Structure	2
4.1	Main Functions	2
4.1.1	JSON Responses Loader	2
4.1.2	User Onboarding	2
4.1.3	Chatbot Conversation Loop	2
5	Dependencies	3
6	Time and Space Complexity	3
7	Implementation Notes	3
8	How to Run	3
9	Sample Execution	3
10	Professional Considerations	5
11	Future Enhancements	5
12	License	5
13	Acknowledgements	5

1 Project Overview

A lightweight Python chatbot that reads over 100 predefined Q&A pairs from a JSON file and interacts with users via CLI. The chatbot personalizes conversation by asking for a name, supports exact-match queries, and gracefully handles unknown inputs.

2 Author Information

Firdaush Alam

Python Developer Intern @ BroskiesHub

GitHub: <https://github.com/firdaush25>

Portfolio: <https://firdaushalamportfolio.netlify.app/>

LinkedIn: <https://www.linkedin.com/in/firdaush-alam>

3 Features

- **User Personalization:** Asks user's name and customizes replies
- **JSON-Based QA Mapping:** Easily scalable from external file
- **Input Normalization:** Case-agnostic and whitespace-tolerant
- **Clear Instructions:** Interaction prompts and exit guidance
- **Fallback Responses:** Handles unmatched queries gracefully

4 Functionality & Code Structure

4.1 Main Functions

4.1.1 JSON Responses Loader

Parses `chatbot_responses.json` into a dictionary using Python's built-in `json` module.

4.1.2 User Onboarding

Greets and stores user name. Displays interaction rules including question formatting and exit command.

4.1.3 Chatbot Conversation Loop

Continuously prompts for questions, normalizes inputs with `.lower()` and `.strip()`, uses dictionary lookup for response ($O(1)$), and exits on "bye" command.

5 Dependencies

- Python 3.x
- Built-in Python module: `json`

6 Time and Space Complexity

Operation	Time Complexity	Space Complexity
JSON Loading	$O(N)$	$O(N)$
Query Lookup (per input)	$O(1)$	$O(1)$

Where N = number of Q&A pairs in JSON.

7 Implementation Notes

- Keys are matched post-normalization (`lower()` + `strip()`)
- Questions must end with `?` to match dataset formatting
- JSON file can be updated independently of Python code
- Modular, beginner-friendly structure

8 How to Run

1. Ensure `chatbot.py` and `chatbot_responses.json` are in the same directory
2. Run the chatbot:

```
python chatbot.py
```

9 Sample Execution

Below is a screenshot of the chatbot in action:

```
C:\Windows\System32\cmd.exe  X  +  v
A:\Broskies Hub internship\Task 8>python chatbot.py
Chatbot: Hi! I'm your simple chatbot.
May i know your Name.Firdaush
Hi Firdaush! Now U May ask your Questions.
Firdaush don't forget to Add ? after your Question
Firdaush, For exit type 'bye'
Ask: hi
Chatbot: Hi there! What can I do for you?
Ask: Hello
Chatbot: Hello! How can I assist you today?
Ask: how are you?
Chatbot: I'm a bot, but I'm doing well! How about you?
Ask: who discovered gravity?
Chatbot: Sir Isaac Newton is credited with gravitational theory.
Ask: what is the speed of light?
Chatbot: Approximately 299,792 kilometers per second.
Ask: how do i stay healthy
Chatbot: Sorry, I didn't understand that. Please try something else.
Ask: what careers involve coding?
Chatbot: Software development, data science, and more.
Ask: what is blockchain?
Chatbot: Blockchain is a distributed ledger for digital transactions.
Ask: can you tell me a fun fact?
Chatbot: Honey never spoils and can last thousands of years.
Ask: what is your purpose?
Chatbot: To help answer your questions using Python.
Ask: where is the nearest restaurant?
Chatbot: I don't have location info, try a map app.
Ask: how's the traffic?
Chatbot: I can't check traffic, try a navigation app.
Ask: bye
Chatbot: Goodbye! Have a great day!

A:\Broskies Hub internship\Task 8>
```

Figure 1: Terminal output of the Python chatbot answering various user queries.

Text version of the interaction:

```
Chatbot: Hi! I'm your simple chatbot.
May I know your Name. Firdaush
Hi Firdaush! Now U May ask your Questions.
Firdaush don't forget to Add ? after your Question
Firdaush, For exit type 'bye'
Ask: what careers involve coding?
Chatbot: Software development, data science, and more.
Ask: can you tell me a fun fact?
Chatbot: Honey never spoils and can last thousands of years.
Ask: bye
Chatbot: Goodbye! Have a great day!
```

10 Professional Considerations

- No user data stored; names used only within session
- Fully compatible with Windows, macOS, and Linux
- Clean architecture promotes scalability and readability
- Ideal for academic and personal use projects

11 Future Enhancements

- Add fuzzy matching and NLP support
- Optional GUI/Web front-end
- Dynamic responses via ML or external APIs

12 License

This project is open-source for educational and personal use.

13 Acknowledgements

Inspired by classic rule-based chatbot designs and Python tutorials. Built with attention to clarity, maintainability, and user experience.

”The End”