

REPORT ON HOSPITAL INFOBOT

1. Introduction

With the rise in demand for quick and accessible healthcare information, automating responses through a chatbot has become essential. Hospital InfoBot is a rule-based chatbot developed using Python Flask that helps users inquire about various hospital services such as doctor availability, appointment procedures, visiting hours, and contact information.

This chatbot is designed for integration into a hospital's website, providing a convenient, responsive way for patients and visitors to get answers to frequently asked questions.

2. Objectives

- To create a simple and efficient hospital information chatbot using Python and Flask.
- To simulate real-time communication between the hospital and patients.
- To handle general queries related to services, appointments, doctors, and more.
- To ensure easy scalability and upgradability for future enhancements.

3. System Architecture

Components

- **Frontend:** HTML page with JavaScript to send user input to the backend and display chatbot responses.
- **Backend:** Flask web application handling request routing and logic.
- **Data File:** intents.json, which stores predefined patterns and responses for each intent (or topic).

4. Functionality and Working

Main Modules

- **home():** Loads the home page (index.html).
- **chat():** Accepts a message from the user via /get route and returns the chatbot's response.
- **get_response(user_input):** Compares the user's input with predefined patterns in the JSON file and returns a response.

Intent Matching Logic

- The user input is converted to lowercase.
- The chatbot loops through each intent and pattern in intents.json.

- If a pattern is found in the user message, a random response from the matched intent is returned.
- If no pattern matches, a default message is returned.

5. Dataset (intents.json)

Each intent contains:

- **tag:** Category of the question (e.g., greeting, appointments)
- **patterns:** Sample user phrases
- **responses:** Predefined chatbot replies

Tag	Patterns (Sample)	Responses (Sample)
greeting	"hi", "hello", "how are you"	"Hello! Welcome to Hospital InfoBot..."
appointments	"book appointment", "schedule visit"	"You can book an appointment by calling..."
doctors	"doctor list", "available specialists"	"We have specialists in..."
services	"what services", "healthcare services"	"We offer general checkups, surgery..."
contact	"hospital phone number", "email"	"Call us at +1-800-HOSPITAL or email..."

6. Testing and Evaluation

- The chatbot was tested manually using different inputs based on the patterns defined in the intents.json.
- It successfully responded to relevant inputs with appropriate answers and returned a fallback message when inputs were not recognized.

Test Case Examples:

Input Message	Expected Output	Result
"hi"	Greeting response	Pass
"book doctor"	Appointment information	Pass
"hospital phone number"	Contact information	Pass
"random unknown input"	Default fallback message	Pass

7. Advantages

- Fast and lightweight
- No external API or database required
- Easy to customize and maintain
- Works locally or on a web server

8. Limitations

- Cannot understand complex or misspelled queries
- Not context-aware (stateless interaction)
- No learning capability or AI integration
- Limited to predefined inputs only

9. Future Enhancements

- Integrate NLP using spaCy, NLTK, or transformers for intelligent intent detection
- Use a machine learning classifier to dynamically identify user intents
- Add contextual memory for multi-turn conversations
- Improve frontend design with chat bubbles, avatars, and typing animations
- Deploy on cloud platforms like Heroku, Render, or AWS

10. Conclusion

Hospital InfoBot demonstrates how rule-based chatbots can be effectively used to provide basic hospital information. While limited in intelligence, its simplicity, ease of use, and flexibility make it a valuable foundation for building more advanced conversational systems in healthcare environments.