

# Medical Chatbot: Project Documentation

Nitesh Singh (22B0919)

## What's This About?

Imagine having a chatbot that understands your health concerns and offers simple, helpful advice. That's exactly what this project aims to deliver! This chatbot is designed to recognize common symptoms, medications, and even specific conditions using Natural Language Processing (NLP). It provides practical responses while keeping the interaction engaging and informative.

## Key Features

- 1. Smart Entity Recognition:**
  - Powered by spaCy, the chatbot identifies health-related terms like symptoms (**fever**, **headache**) and medications (**Paracetamol**, **Ibuprofen**).
  - Custom patterns ensure it recognizes terms relevant to medical conversations.
- 2. Helpful Responses:**
  - The chatbot matches recognized terms to predefined advice, offering users quick tips or reassurance.
  - Responses vary, so it feels less robotic and more natural.
  - A built-in database adds extra knowledge about conditions like diabetes and hypertension.
- 3. Fallbacks that Work:**
  - When it doesn't understand, the chatbot asks for clarification or offers keyword-based suggestions.
  - You'll never get a dead-end response.
- 4. Easy Testing:**
  - Includes five test cases to ensure it works smoothly for common scenarios like symptoms, medications, or general health queries.
- 5. Simple and Interactive:**
  - A terminal-based interface lets you chat naturally.
  - Type **exit** anytime to end the session.

## How It Works

- 1. Preprocessing :**
  - Your input is cleaned up (e.g., lowercase conversion, extra spaces removed) so the chatbot focuses on what matters.
- 2. Entity Detection:**

- spaCy's NLP engine identifies key terms in your query (like **fever** or **Paracetamol**).

### 3. Smart Response Logic:

- For each symptom or medication, a response is picked from a dictionary. The chatbot also adds variety by randomly selecting different phrasing for the same advice.
- If it doesn't recognize something, fallback suggestions kick in using a keyword-matching strategy.

## A Quick Example

- **You Say:** "I have a fever and headache."
- **Chatbot Thinks:** "It's a fever and headache situation."
- **Chatbot Responds:** "For fever: Stay hydrated and consider taking paracetamol if recommended by a doctor. For headache: Try to relax and consider over-the-counter pain relief."

## Bonus Features

- **Keyword-Based Advice:**
  - Mention "diabetes," and the chatbot offers tailored tips like "Maintain a healthy diet and monitor blood sugar."
- **Keeps It Interesting:**
  - It alternates responses to feel more conversational.

## How to Use It

1. Run the script in your Python environment.
2. Type in your health query (e.g., symptoms or medication).
3. Get advice instantly or follow up with more details.
4. Exit anytime with the command **exit**.

## Final Thoughts

This project bridges simple rule-based logic with modern NLP to create a chatbot that's both functional and user-friendly. Future upgrades could include real-time API integrations for up-to-date medical info or support for multiple languages, making it even more versatile and useful.