**1.Introduction**

* **Purpose:** The main goal of the chatbot is to provide users with personalized medicine and diet recommendations based on their symptoms.
* **Scope:** The chatbot will assist users in identifying potential treatments and dietary suggestions by analyzing their input symptoms.
* **Objective:** To enhance user experience by automating health-related queries and providing reliable recommendations swiftly.

**2. Background/Problem Statement**

* **Context:** Many individuals seek quick advice for health issues but may not have immediate access to medical professionals. This chatbot aims to fill that gap by offering 24/7 assistance.
* **Existing Solutions:** While there are health-related chatbots available, they often lack specificity in recommendations or have limited symptom coverage.

**3. Requirements Analysis**

* **User Requirements:** Users expect quick, accurate recommendations and easy interaction with the chatbot for follow-up questions.
* **Functional Requirements:** The chatbot should interpret symptoms, provide corresponding medicine suggestions, and recommend diets based on the analyzed data.
* **Non-functional Requirements:** The system must be user-friendly, secure, and capable of handling multiple user requests simultaneously.

**4. Design and Architecture**

* **System Architecture:** The architecture includes a Flask backend for processing user inputs, a database for symptom data, and a machine learning model for predictions.
* **Conversation Flow:** A flow diagram illustrates how the chatbot interacts with users, responding based on different symptom inputs.
* **NLP Design:** The chatbot employs Natural Language Processing to accurately understand and categorize symptoms provided by users.
* **Technology Stack:** The main technologies used are Flask for the web framework, Scikit-learn for machine learning, and pandas for data handling.

**5. Development and Implementation**

* **Development Process:** Stages include system design, coding, and extensive testing to ensure reliability and accuracy.
* **Algorithms:** A Random Forest algorithm is used for predicting the most suitable medicines and dietary recommendations based on input symptoms.
* **Training Data:** The model is trained on a dataset of symptoms, medicines, and diets, ensuring it learns to provide accurate recommendations.

**6. Conversation Handling**

* **Intent Management:** The chatbot identifies user intents, such as seeking medicine or diet advice, based on the symptoms entered.
* **Entity Extraction:** Key entities, like symptom names, are extracted to refine the recommendations.
* **Fallback Scenarios:** The chatbot is designed to handle ambiguous user inputs by asking clarifying questions to guide them effectively.

**7. Integration**

* **Platform Integration**: Explain which platforms the chatbot integrates with (e.g., Messenger, Slack, WhatsApp, websites) and how.
* **API and External Services:** Discuss any external APIs integrated (e.g., payment gateways, weather APIs, etc.).

**8. Testing and Evaluation**

* **Types of Tests:** Describe the different types of tests conducted (e.g., functional testing, user acceptance testing, performance testing).
* **Test Cases:** Provide examples of typical test cases for verifying the chatbot's responses.
* **User Feedback:** If the bot has been tested with real users, summarize their feedback and how it helped improve the bot.

**9. Challenges and Limitations**

* **Technical Challenges**: Mention any technical difficulties faced during development (e.g., NLP issues, system integration problems) and how you addressed them.
* **Limitations:** Discuss any limitations in the chatbot’s functionality or design, such as language support or scalability.

**10. Results and Performance**

* **KPIs:** Measure performance based on Key Performance Indicators (e.g., response accuracy, success rate of intent recognition, response speed).
* **User Engagement**: If applicable, discuss metrics related to user interaction, such as the number of sessions or average session time.

**11. Future Improvements**

* **Upgrades:** Potential improvements include expanding the dataset to cover more symptoms and integrating advanced NLP features.
* **Scalability:** Plans for scaling the chatbot to accommodate more users and enhance its capabilities are outlined.

**12. Conclusion**

* **Summary:** The project successfully builds a functional chatbot that automates health queries, achieving the intended objectives.
* **Reflection:** The development process highlighted the importance of user feedback and iterative design, confirming that project goals were met.

**13. Appendix**

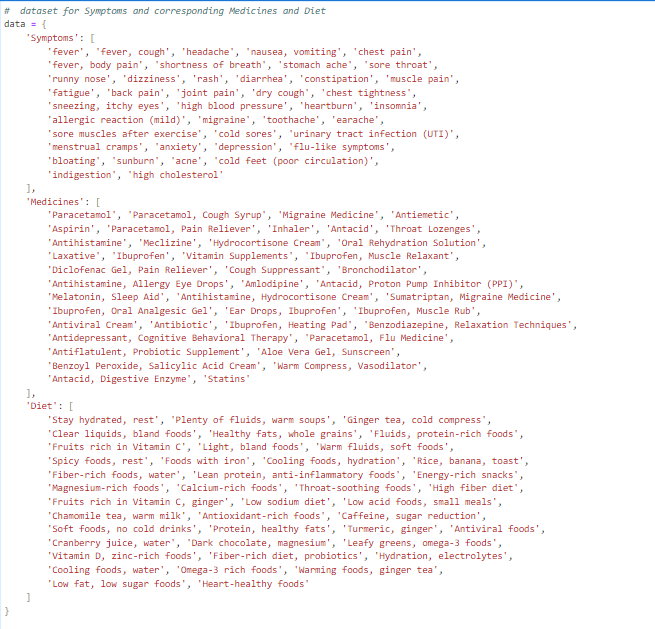
* **Code Snippets:** Relevant code sections, particularly for the prediction function and Flask routes, are included.
* **Flow Diagrams:** Detailed flowcharts illustrating the chatbot's architecture and interaction paths are provided.

**14. References**

* **Flask Documentation:** Guidance on routing and templates.
* **Scikit-learn Documentation:** Information on machine learning algorithms.
* **pandas Documentation:** Data manipulation and analysis resources.
* **Random Forest Overview:** Insights into the Random Forest algorithm.
* **NLP Resources:** Best practices for chatbot development.

**PYTHON CODE**



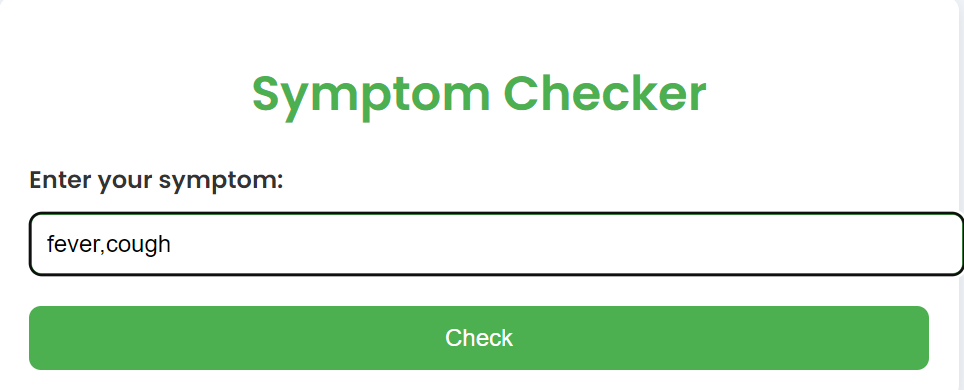


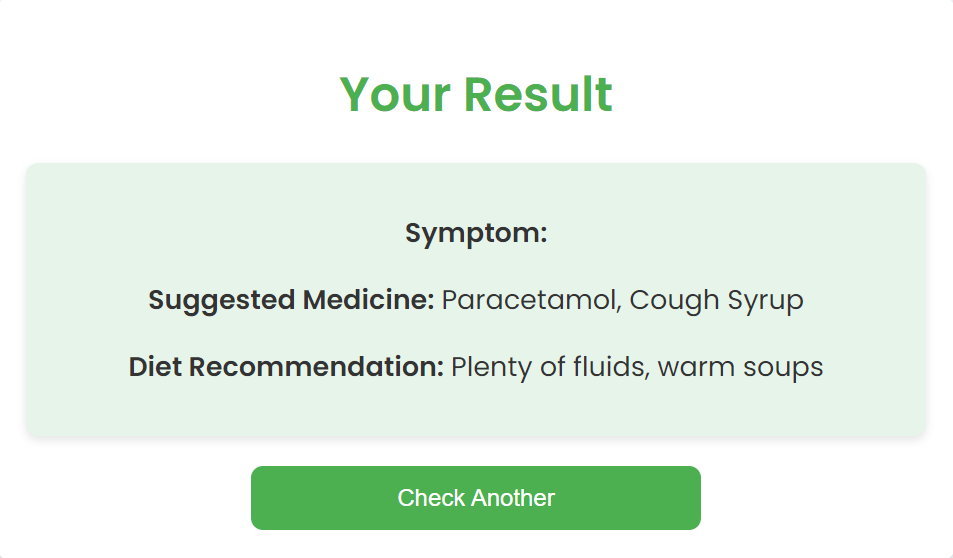


**Output:**

****

**User interface**

****

****