Faculty of Engineering

Compulsory Elective 2 Course

Guardian Pharmacy Bot

October 2024

Le Thai Duong 10421012
Tran Cao Gia Bao 10421005

Computer Science Engineering Program

Plagiarism Declaration

With the exception of any statement to the contrary, all the material presented in this report is the result of my own efforts. In addition, no parts of this report are copied from other sources. I understand that any evidence of plagiarism and/or the use of unacknowledged third party materials will be dealt with as a serious matter.

Signed

1. Le Thai Duong
2. Tran Cao Gia Bao

Contents

List of Figures

Chapter 1

Introduction

Due to the development in Artificial Intelligence and Natural Language Processing, development in chatbots has become a very important aspect of online communication. Their usage varies from domains of technical support and customer service to that of personal help and educational support. The idea of the project is to develop a chatbot using Chainlit—a platform that simplifies the development of chat-based interfaces and allows developers to build with ease interactive AI-powered applications.

The major objective of the Guardian Pharmacy Bot is to offer users convenient, effective, and professional drug information with other product specifications for health needs. In particular, some of the goals are:

- **Comprehensive drug and product information**: The chatbot will be designed in such a way that it will help users make selections from a variety of health products and answer questions on drugs, including their uses, dosage, and side effects.
- **Chainlit for an Intuitive User Interface**: A basic user interface using Chainlit will provide a capability to ask questions, look through categories of products, and get prompt responses easily. This would seriously reduce headaches in interacting with this chatbot and generally improve the user experience.
- **User Authentication and Memories for Personalized Support**: This includes user authentication, so that the chatbot can detect whether the user has interacted before with it. In addition, they will have the ability to remember past conversations. This feature

is going to help in tuning discussions through offering relevant product suggestions or follow-up questions derived from previous consultations.

- Deploy the Bot on a Reliable Cloud Service for Accessibility and Scalability: The deployment of a chatbot on a reputed cloud service, such as AWS, makes it always available to the user and capable of responding to n number of queries at a time, hence effectively supporting both small- and large-scale interactions. It will provide customers with the ability to receive trustworthy information about drugs on the spot, while through the capability provided by the Guardian Pharmacy Bot, pharmacies can enhance their service efficiency, increase customer interaction, and be assured of compliance with the industry standard.

By harnessing the capabilities of the Guardian Pharmacy Bot, customers can access reliable drug information instantly, while the pharmacy can enhance service efficiency, improve customer engagement, and ensure compliance with industry standard

Chapter 2

Project Overview

The Guardian Pharmacy Bot extend was made to mirror an intelligently drug store partner for Guardian Pharmacy. Its primary objective is to help customers in investigating and buying a assortment of wellbeing things, counting vitamins, supplements, medicine drugs, over-the-counter pharmaceuticals, and restorative hardware. The chatbot, which is based on Chainlit and is driven by OpenAI's GPT-3.5-turbo, is planned to grant a user-friendly, intelligently involvement that effectively reacts to shopper request and makes item proposals based on their wellness and wellbeing prerequisites. A intensive question-and-answer (Q&A) highlight that permits shoppers to inquire questions approximately things, costs, and utilize data is one of its essential highlights. The chatbot has a memory component that spares conversational history so it can relevantly reply follow-up questions, progressing client engagement and customization. Clients must input their username and their password in order to ensure that as it were authorized clients are able to get to customized intelligent. With the offer assistance of Chainlit, the chatbot can show an organized client interface (UI) so as to offer assistance customers explore through different item categories and services.

This chatbot's target gathering of people comprises of patients and their relatives who need to effectively browse, inquire questions almost, and perhaps put orders for Guardian Pharmacy things. Clients looking for medicine data, comparing over-the-counter arrangements, examining supplements to upgrade wellbeing, and getting hardware specifics with conveyance alternatives are a few of the expected utilize cases. The project's expected comes about and deliverables compare with specific evaluated components, counting a solid Q&A framework that can give

detailed answers, an instinctive client interface (UI) through Chainlit, customized memory maintenance for improved conversational stream, and sending on a cloud stage like AWS for reliable and available utilize. By accomplishing these objectives, the Guardian Pharmacy Bot will be a valuable apparatus for customers, giving a smooth, instructive, and locks in involvement in line with the pharmacy's reason to bolster wellbeing and prosperity.

Chapter 3

Project Management

3.1 Timeline and Development Milestones

1. **Week 3 (October 3rd)**: The initial phase consisted of learning Chainlit through tutorial videos on YouTube, followed by implementing the core program logic. A message handler was created using `@cl.on_message` to process user input appropriately. Additionally, a markdown file was created for structured data presentation, and a prompt file was developed to guide the chatbot's responses, aligning it with user inquiries.

2. **Week 4 (October 10th)**: The second phase focused on refining the user interface (UI), making the chatbot more user-friendly and interactive. The `app.py` file was updated to retain conversation history, allowing the chatbot to remember previous interactions with the use of `@cl.on_chat_resume`. A JSON file was also developed to store message data, effectively functioning as a database for the chatbot's memory component.

3. **Week 5 (October 17th)**: The final phase involved rigorous testing on the localhost to ensure functionality and resolve any issues. Once testing was complete, the project was deployed on AWS to provide a stable, cloud-hosted environment accessible to users.

3.2 Technologies Used

- **Visual Studio Code:** Coding and development.
- **Chainlit:** Management of chatbot functionalities and UI development.

- **AWS:** Cloud service provider for reliable chatbot deployment.

3.3 Resource Allocation and Responsibilities

During the first week, I handled foundational tasks to establish the initial setup. In the second week, my teammate joined the development process, and we divided the tasks evenly, each contributing 50% to UI improvements and memory integration. In the final week, my teammate took responsibility for deploying the project to AWS.

3.4 Quality Assurance Strategies to Ensure Performance and Reliability

To ensure high performance and reliability, a range of quality assurance strategies was implemented. Various use cases were tested to verify response accuracy, memory retention, and user authentication. Additionally, stress testing assessed performance under heavy usage, and edge cases were reviewed to ensure the chatbot could handle unexpected inputs effectively. Through continuous debugging and iterative improvements, we optimized the bot's functionality and user experience, ensuring reliable interactions for Guardian Pharmacy customers.

Chapter 4

Requirement Analysis

4.1 Functional Requirements

1. **Q&A Ability**: The Chatbot can respond to the users' questions about health products. This is the prime interaction point and functionality of Guardian Pharmacy Bot.
2. **User Authentication**: Required for the robot to recognize the user in case of return. Such functionality allows every new user interaction with the robot to be secure while keeping history for further interactions.
3. **Memory Capabilities**: The bot remembers past interactions to provide personalized recommendations, or follow-up responses based on the interaction history of a user.
4. **Cloud Deployment**: The bot must be deployed on an appropriate cloud platform, like AWS so that the accessibility is very high and multiple users can use it concurrently.

4.2 Non-Functional Requirements

1. **Scalability**: The bot should be capable of scaling its operations depending on demand by the end user, that is, handle large pools of concurrent users and requests.
2. **Responsiveness**: Response time has to be low, even under heavy loads, in order to achieve fluent user experiences.
3. **Security**: Since the bot will be handling potentially sensitive health-related information, data security will be one of the most important factors.

Chapter 5

Implementation

5.1 Step-by-Step Guide to QA Functionality Implementation with Chainlit

1. **Setup Chainlit Environment**: Create a new project directory and install the necessary libraries, including `chainlit` and `openai`. In this directory, set up essential files: `app.py` for the main program, a `.env` file for environment variables, and `prompt.py` to define bot instructions.

2. **Q&A Logic**: In `app.py`, the `@cl.on_message` function defines how the chatbot processes incoming user messages. The message handler forwards queries to OpenAI's GPT-3.5-turbo model, generating responses based on user input. In `prompt.py`, `system_instruction` guides the bot's replies, tailoring them to address typical questions about Guardian Pharmacy products.

3. **Connect OpenAI for Responses**: An OpenAI client instance is created, using the `ask_order` function for interaction handling. The function receives user queries, sends them to the model, and provides real-time responses, ensuring accuracy and relevance for each Q&A session.

5.2 Integration of Authentication for Personalized Memory

1. **Add Authentication**: In `app.py`, use the `@cl.password_auth_callback` decorator to introduce a basic login system where users enter a password to access the bot. This

approach ensures that each user has secure access to their personalized chat session.

2. **Introduce Memory for Personalized Answers**: With `@cl.on_chat_resume`, enable the bot to remember past conversations. A `memory.json` file can store each user's conversation history, allowing the bot to reference previous interactions and making follow-up questions more seamless and personalized.

3. **Development of Chainlit UI**: Chainlit supports UI customization, allowing for a user-friendly layout with welcome messages and product lists. Using `@cl.set_starters`, define initial messages to orient users, such as "Welcome to Guardian Pharmacy Bot! ", showcasing product categories and sample queries.

5.3 Code Snippets and Examples of Key Functions

1. **Message Handler**: In `app.py`, handle incoming messages with `@cl.on_message`

```python
# Handles new user messages and saves them to memory.json
@cl.on_message
async def main(message: cl.Message):
    # Load the conversation history from JSON
    conversation_history = load_memory()

    # Append the user's message and the assistant's response to memory
    messages.append({"role": "user", "content": message.content})
    response = ask_order(messages)
    messages.append({"role": "assistant", "content": response})

    # Save the updated conversation history to JSON
    conversation_history.append({"role": "user", "content": message.content})
    conversation_history.append({"role": "assistant", "content": response})
    save_memory(conversation_history)

    # Send the assistant's response back to the user
    await cl.Message(content=response).send()
```

2. **Ask Order Function**: This function connects to OpenAI's API and retrieves responses.

```python
def ask_order(messages, model="gpt-3.5-turbo",temperature=0):
    response = client.chat.completions.create(
        model= model,
        messages=  messages,
        temperature= temperature
    )

    return response.choices[0].message.content
```

3. **Authentication Setup**:

```python
@cl.password_auth_callback
def auth_callback(username: str, password: str):
    # Dictionary storing usernames and passwords for authentication
    users = {
        "Duong": "10421012",
        "Bao": "10421005"
    }

    # Verifies credentials and returns user metadata if authenticated
    if username in users and users[username] == password:      You, 6 days ago • mai là
        return cl.User(
            identifier=username, metadata={"role": username, "provider": "credentials"}
        )
    else:
        return None
```

4. **Memory Storage**:

```python
@cl.on_chat_resume
async def resume_conversation(messages: Optional[list] = None):
    persisted_messages = load_memory()
    if persisted_messages:
        # Send all previous messages stored in memory to re-establish context
        for msg in persisted_messages:
            await cl.Message(content=msg["content"], role=msg["role"]).send()
    if messages:
        # Respond to the latest message from the persisted conversation
        last_message = messages[-1]
        response = ask_order(messages)
        await cl.Message(content=response).send()
```

Chapter 6

Deployment

Chapter 7

Conclusion

The key development features of the Guardian Pharmacy Bot are an advanced Q&A related to health queries, user authentication in a secure manner, memories allowing for personalized interaction with the user, and deployment on the cloud. The bot can provide both accurate and speedy responses while offering a personalized experience to returning users.

During the development process, many challenges were noted, such as data accuracy for health-related content that required great verification. Similarly, great optimization was required for adding memory capabilities and having low response times under heavy user load. Extensive security measures aimed at ensuring user data safety have also been intensified, given the sensitivity of health information.

In the future, several extensions are likely to improve the performance of the bot. Expanding the database on drug information will increase relevance and precision, multilingual support will expand access, and more sophisticated machine learning models will considerably improve personalization and responsiveness. Integration with a host of health services and resources is likely to follow.

This has brought home just how important it is to balance functionality, user experience, and security in healthcare technology. Allowing the creation of a truly responsive, reliable, and secure tool, the Guardian Pharmacy Bot will provide much-needed assistance for users in ways that provide access to health-related information and showcase the possibilities with AI in healthcare. The project provides a good example of AI in an application and reaches out to those interested in developing intelligent health support tools.