CHATBOT

```
def chatbot():
    print("ChatBot: Hello! Welcome to ShopEase Support. How can I help you today?")
    print("Type 'exit' to end the chat.\n")

    while True:
        user_input = input("You: ").lower()

        if "exit" in user_input:
            print("ChatBot: Thank you for visiting ShopEase. Have a great day!")
            break
        elif "order" in user_input or "track" in user_input:
            print("ChatBot: You can track your order from the 'My Orders' section under your account.")
        elif "refund" in user_input or "return" in user_input:
            print("ChatBot: Refunds are processed within 5-7 business days after we receive your return.")
        elif "cancel" in user_input:
            print("ChatBot: You can cancel your order within 30 minutes of placing it.")
        elif "delivery" in user_input or "shipping" in user_input:
            print("ChatBot: We usually deliver orders within 3-5 working days.")
        elif "payment" in user_input:
            print("ChatBot: We accept credit cards, debit cards, UPI, and net banking.")
        elif "contact" in user_input or "help" in user_input:
            print("ChatBot: You can contact our support at support@shopease.com or call 1800-123-456.")
        else:
            print("ChatBot: I'm sorry, I didn't understand that. Can you please rephrase?")
```

Start the chatbot

chatbot()

Sample Interaction:

vbnet

```
ChatBot: Hello! Welcome to ShopEase Support. How can I help you today?
You: Where is my order?
ChatBot: You can track your order from the 'My Orders' section under your account.
You: What is the delivery time?
ChatBot: We usually deliver orders within 3-5 working days.
You: exit
ChatBot: Thank you for visiting ShopEase. Have a great day!
```

Possible Viva Questions for this Practical:

1. **What is a chatbot, and how does it work?**
    - A chatbot is a software application that simulates human conversation through text or voice. It interacts with users to answer their queries or provide information based on predefined rules, keywords, or machine learning models.
2. **What are the limitations of the chatbot implemented here?**
    - The chatbot is rule-based and relies on exact keywords or phrases. It cannot handle complex or context-dependent conversations. It lacks advanced features such as Natural Language Processing (NLP) or machine learning to understand variations in user input.
3. **Why did you use the while True: loop in the chatbot?**
    - The while True: loop is used to create an infinite interaction until the user types "exit" to end the chat. It allows the chatbot to continuously respond to user queries.
4. **What is the significance of converting the user input to lowercase (user_input.lower()) in the code?**
    - By converting the input to lowercase, the chatbot can handle user inputs case-insensitively. For example, "Order" and "order" will be treated the same, making the bot more flexible.
5. **How does the chatbot decide which response to provide?**

- The chatbot uses conditional checks (if-elif statements) to match keywords or phrases in the user input. Depending on the matched keyword (e.g., "order", "refund", "payment"), it provides a predefined response.

6. **What would happen if a user types something the chatbot doesn't recognize?**
    - If the chatbot doesn't recognize any of the predefined keywords in the user's input, it will respond with "I'm sorry, I didn't understand that. Can you please rephrase?"
7. **What are the advantages of using a rule-based chatbot like this one?**
    - Rule-based chatbots are simple to implement, fast to deploy, and reliable for handling specific, structured tasks. They don't require complex algorithms or large datasets.
8. **How would you improve this chatbot to make it more advanced?**
    - To improve this chatbot, we could integrate Natural Language Processing (NLP) for better understanding of user input, add machine learning models for learning from interactions, or include more interactive elements like buttons or voice support.
9. **Can this chatbot handle multiple users at once?**
    - No, this chatbot can only handle one user interaction at a time. To handle multiple users simultaneously, the chatbot needs to be deployed on a server with concurrent session handling or use multi-threading.
10. **What is the role of the input() function in this chatbot?**
    - The input() function is used to capture user input during the conversation. It waits for the user to type something and then stores the input in the user_input variable for further processing.
11. **What happens when the user types 'exit' in the chatbot?**
    - When the user types "exit," the chatbot prints a farewell message and breaks the loop, ending the conversation.