Shoe Seller

Abstract

A friendly chatbot that guides users step-by-step through purchasing shoes. It collects the desired model, size, and color, confirms the order, and provides a summary.

Table of Contents

# Topics

| topic | description | prompt\_regex |
| --- | --- | --- |
| help\_command | “help” echos the help block | (?i)^help$ |

# Help

> User Documentation   
> ==================   
>   
> The chatbot guides the user step by step through the shoe purchase process.   
> It first asks for the desired model, then the appropriate size, and finally the preferred color.   
> Only after all required inputs are provided does it display an order summary.   
>   
> Overview   
> --------   
> This chatbot helps users order shoes by collecting all necessary order details step by step.   
> It ensures that no input is missing and only completes the process once all required data is gathered.   
>   
> Interaction Flow   
> ----------------   
> 1. \*\*Greeting\*\* – The bot greets the user and begins the ordering process.   
> 2. \*\*Model Selection\*\* – Asks the user for the shoe model.   
> 3. \*\*Size Selection\*\* – Asks for the preferred shoe size.   
> 4. \*\*Color Selection\*\* – Asks for the desired color.   
> 5. \*\*Confirmation\*\* – Once all inputs are collected, the bot summarizes the order and asks for confirmation.   
> 6. \*\*Response Handling\*\* –   
> - If confirmed: shows the full order summary.   
> - If not confirmed: gives a polite response and awaits further action.   
>   
> Input Fields   
> ------------   
> - `model`: A text value representing the type of shoe (e.g., “sneakers”, “loafers”).   
> - `size`: The selected size (e.g., “42”, “US 9”).   
> - `color`: The chosen shoe color (e.g., “black”, “white”).   
> - `confirmation`: A boolean indicating whether the user confirmed the order.   
> - `response\_to\_customer`: Used when the order is not confirmed — e.g., a helpful or polite message.   
>   
> Output Format   
> -------------   
> The bot’s responses vary based on the interaction state:   
> - If order is complete, it outputs a full summary using bullet points.   
> - If the order is incomplete, it provides a friendly message with the current status.   
>   
> Usage Notes   
> -----------   
> - The chatbot handles incomplete input gracefully, prompting only for missing data.   
> - It uses emoji in responses to add a friendly tone (e.g., 🛍, ✅, 💬).   
> - Works well in step-by-step interfaces or conversation-based purchase flows.

# System

You are a friendly and helpful assistant who sells shoes.   
Guide the customer step by step through the purchase process:   
1. First, ask for the desired shoe model.   
2. Then ask for the shoe size.   
3. Finally, ask for the preferred color.   
4. If any information is missing, continue asking until all details are complete.   
5. Once all data is available, summarize the order and ask if the customer wants to complete the purchase.   
6. If the order is confirmed, display the full order details.   
7. If the order is still pending, respond with a friendly message instead.   
  
Let's take it one step at a time.

# Schema

{  
 "type": "object",  
 "properties": {  
 "model": {  
 "type": "string",  
 "description": "The selected shoe model."  
 },  
 "size": {  
 "type": "string",  
 "description": "The chosen shoe size."  
 },  
 "color": {  
 "type": "string",  
 "description": "The desired shoe color."  
 },  
 "confirmation": {  
 "type": "boolean",  
 "description": "Confirmation whether the customer wants to place the order."  
 },  
 "response\_to\_customer": {  
 "type": "string",  
 "description": "A friendly response from the chatbot if the order is not yet completed."  
 }  
 },  
 "additionalProperties": false  
}

# Response

% if RESPONSE.get("confirmation", None):   
🛍 \*\*Your Order:\*\*   
- \*\*Model:\*\* ${RESPONSE.get("model", "undefined")}   
- \*\*Size:\*\* ${RESPONSE.get("size", "undefined")}   
- \*\*Color:\*\* ${RESPONSE.get("color", "undefined")}   
  
✅ Your order has been successfully placed! Thank you for shopping with us.   
% else:  
💬 ${RESPONSE.get("response\_to\_customer", "How can I assist you further?")}  
% endif