```
In [4]: # '''
        # return_policy.txt
        # ------------------
        # return_policy:
        # Our return policy allows returns within 30 days of purchase.
        # for Tablet , the return policy allows return within 10 days of pu

        # order_tracking:
        # You can track your order status using your order ID.

        # product_info:
        # Product information includes specifications and features.

        # orders.json
        # ------------
        # {
        #     "123": {
        #         "customer_name": "John Doe",
        #         "order_date": "2022-01-01",
        #         "products": ["Laptop", "Monitor"],
        #         "total": 1500.00
        #     },
        #     "124": {
        #         "customer_name": "Jane Doe",
        #         "order_date": "2022-01-15",
        #         "products": ["Phone", "Keyboard"],
        #         "total": 800.00
        #     },
        #     "125": {
        #         "customer_name": "Bob Smith",
        #         "order_date": "2024-11-30",
        #         "products": ["Tablet", "Laptop"],
        #         "total": 2500.00
        #     }
        # }


        # product_info.json
        # ------------------
        # {
        #     "Laptop": "A high-performance laptop with 16GB RAM and 512GB 
        #     "Phone": "A smartphone with a 6.1-inch display and 128GB stor
        #     "Tablet": "A tablet with a 10.5-inch display and 256GB storag
        #     "Monitor": "A 24-inch full HD monitor.",
        #     "Keyboard": "A mechanical keyboard with RGB backlighting."
        # }


        # order_status.json
        # ------------------
        # {
```

```
#     "123": "Delivered",
#     "124": "Processing",
#     "125": "Delivered",
#     "126": "Cancelled",
#     "127": "Returned",
#     "128": "Out for Delivery"
# }

# '''
#!pip install faiss-cpu langchain langchain_openai langchain_commun
```

In [30]:
```
from langchain.tools import Tool
import json
from langchain_openai import ChatOpenAI
import os
from datetime import date

def get_order_status_tool(order_id):
    if not order_id.startswith("order_id = "):
        return "Please provide an order ID in the format: 'order_id

    try:
        order_id = order_id.replace("order_id = ", "")
        with open('order_status.json', 'r') as file:
            order_status = json.load(file)
        return order_status.get(str(order_id), "Order ID not found"
    except Exception as e:
        print(f"Error reading order_status.json: {e}")
        return "Order ID not found"

def save_order_status_returned_tool(order_id):

    if not order_id.startswith("order_id = "):
        return "Please provide an order ID in the format: 'order_id

    try:
        # Read the existing data from the file
        try:
            with open('order_status.json', 'r') as file:
                order_status = json.load(file)
        except FileNotFoundError:
            order_status = {}
        except json.JSONDecodeError:
            order_status = {}

        # Update the order status
        order_id = order_id.replace("order_id = ", "")
        order_status[str(order_id)] = "Returned"

        # Write the updated data back to the file
        with open('order_status.json', 'w') as file:
            json.dump(order_status, file, indent=4)
```

```python
        return "Order status saved successfully."

    except Exception as e:
        print(f"Error saving order_status.json: {e}")
        return "Failed to save order status."

def get_product_info_tool(product_name):
    if not product_name.startswith("product_name = "):
        return "Please provide a product name in the format: 'produ

    try:
        product_name = product_name.replace('product_name = ', '')
        print(f" product_name {product_name}")
        with open('product_info.json', 'r') as file:
            product_info = json.load(file)
        return product_info.get(product_name, "Product not found")
    except Exception as e:
        print(f"Error reading product_info.json: {e}")
        return "Product not found"

def get_order_info_tool(order_id):
    if not order_id.startswith("order_id = "):
        return "Please provide an order ID in the format: 'order_id

    try:
        order_id = order_id.replace("order_id = ", "")
        with open('orders.json', 'r') as file:
            orders = json.load(file)
        return orders.get(str(order_id), "Order ID not found")
    except Exception as e:
        print(f"Error reading orders.json: {e}")
        return "Order ID not found"

def get_today_date(today):
    from datetime import datetime
    today = datetime.now().strftime("%Y-%m-%d")
    return f"today is {today}"


input_tool = Tool(
    name="get_input_from_user",
    func=input,
    description="Get the input from the user."
)

order_status_tool = Tool(
    name="get_order_status",
    func=get_order_status_tool,
    description="Get the status of an order given an order ID."
)

product_info_tool = Tool(
    name="get_product_info",
    func=get_product_info_tool,
    description="Get information about a product given its name."
)

order_info_tool = Tool(
    name="get_order_info",
    func=get_order_info_tool,
    description="Get information about an order given its ID."
)

order_status_returned_tool = Tool(
    name="save_order_status_returned",
    func=save_order_status_returned_tool,
    description="Change the status of the order for the given order
)

get_today_date_tool = Tool(
    name="get_today_date",
    func=get_today_date,
    description="Get today's date in the format 'YYYY-MM-DD'."
)

os.environ["OPENAI_API_KEY"] = "sk-proj-Z-abcdef"
llm = ChatOpenAI(model="gpt-4o", temperature=0)

#https://python.langchain.com/docs/how_to/agent_executor/#using-lang

from langchain.vectorstores import FAISS
# from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.document_loaders import TextLoader
from langchain.llms import OpenAI
from langchain.document_loaders import PyMuPDFLoader
from langchain_openai import OpenAIEmbeddings

# Load and index documents
loader = TextLoader("return-policy.txt")
documents = loader.load()

# Split documents into chunks
text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chun
split_docs = text_splitter.split_documents(documents)

# Create embeddings and build FAISS index
embeddings = OpenAIEmbeddings()
vector_store = FAISS.from_documents(split_docs, embeddings)
retriever = vector_store.as_retriever(search_type="similarity_score_

from langchain.tools.retriever import create_retriever_tool
from langchain.agents import AgentExecutor, load_tools

tool_search = create_retriever_tool(
    retriever=retriever,
    name="search_return_policy",
    description="Searches and returns policy, product, etc."
)

tools = load_tools([], llm=llm)
tools.append(tool_search)
tools.append(input_tool)
tools.append(order_status_tool)
tools.append(product_info_tool)
tools.append(order_info_tool)
tools.append(order_status_returned_tool)
tools.append(get_today_date_tool)

from langchain import hub
prompt = hub.pull("hwchase17/react")

# only creates the logical steps for us
from langchain.agents import AgentExecutor, create_react_agent

# https://python.langchain.com/v0.1/docs/modules/agents/agent_types/
# https://python.langchain.com/v0.1/docs/modules/agents/agent_types/

react_agent = create_react_agent(llm, tools, prompt)
agent_executor = AgentExecutor(
    agent=react_agent, tools=tools, verbose=True, handle_parsing_er
)

def prompt_user_for_info(query):
    response = agent_executor.invoke({"input": query})
    if "Please provide" in response["output"]:
        new_input = input(response["output"])  # Prompt the user fo
        return prompt_user_for_info(new_input)
    else:
        return response
```

```
/opt/conda/lib/python3.10/site-packages/langsmith/client.py:241: La
ngSmithMissingAPIKeyWarning: API key must be provided when using ho
sted LangSmith API
  warnings.warn(
```

In [32]:
```
query = "get the product information of the order?"
final_response = prompt_user_for_info(query)
print(final_response)
```

```
> Entering new AgentExecutor chain...
To get the product information of an order, I first need to know th
e order ID.

Action: get_input_from_user
Action Input: "Please provide the order ID to retrieve the product
information."
```

```
125I have the order ID. Now, I need to retrieve the order informati
on using this ID to get the product details.

Action: get_order_info
Action Input: 125Please provide an order ID in the format: 'order_i
d = <your_order_id>'  . for the correct format, don't ask the user
again. you prepare the correct format if you have enough informatio
nI need to format the order ID correctly to retrieve the order info
rmation.

Action: get_order_info
Action Input: order_id = 125{'customer_name': 'Bob Smith', 'order_d
ate': '2024-11-30', 'products': ['Tablet', 'Laptop'], 'total': 250
0.0}I have retrieved the order information. The products in the ord
er are a Tablet and a Laptop.

Final Answer: The products in the order are a Tablet and a Laptop.

> Finished chain.
{'input': 'get the product information of the order?', 'output': 'T
he products in the order are a Tablet and a Laptop.'}
```

In [36]:
```
query = "how many days before I ordered?"
final_response = prompt_user_for_info(query)
print(final_response)
```

```
> Entering new AgentExecutor chain...
To determine how many days have passed since you placed your order,
I need to know the order date. I can find this information if you p
rovide me with your order ID.

Action: get_input_from_user
Action Input: "Please provide your order ID so I can check the orde
r date."
```

```
125I have the order ID. Now, I need to retrieve the order informati
on to find out the order date.

Action: get_order_info
Action Input: 125Please provide an order ID in the format: 'order_i
d = <your_order_id>'  . for the correct format, don't ask the user
again. you prepare the correct format if you have enough informatio
nI need to format the order ID correctly to retrieve the order info
rmation.

Action: get_order_info
Action Input: order_id = 125{'customer_name': 'Bob Smith', 'order_d
ate': '2024-11-30', 'products': ['Tablet', 'Laptop'], 'total': 250
0.0}I have the order date, which is November 30, 2024. Now, I need
to find out today's date to calculate the number of days since the
order was placed.

Action: get_today_date
Action Input: Nonetoday is 2024-12-05I have both the order date and
today's date. The order was placed on November 30, 2024, and today
is December 5, 2024. To calculate the number of days since the orde
r was placed, I subtract the order date from today's date.

Final Answer: You ordered 5 days ago.

> Finished chain.
{'input': 'how many days before I ordered?', 'output': 'You ordered
5 days ago.'}
```

In [38]:
```python
query = "Return my product which I was ordered?"
final_response = prompt_user_for_info(query)
print(final_response)
```

```
> Entering new AgentExecutor chain...
To assist you with returning your product, I need to gather some in
formation about your order. Let's start by getting the order ID.

Action: get_input_from_user
Action Input: "Please provide your order ID so I can assist you wit
h the return process."
```

```
123I have the order ID. Now, I need to check the status of the orde
r to proceed with the return process.

Action: get_order_status
Action Input: 123Please provide an order ID in the format: 'order_i
d = <your_order_id>'. for the correct format, don't ask the user ag
ain. you prepare the correct format if you have enough informationI
t seems I need to format the order ID correctly to proceed. Let me
do that.

Action: get_order_status
Action Input: order_id = 123DeliveredThe order with ID 123 has been
delivered. To proceed with the return process, I need to check the
return policy to ensure the product is eligible for return.

Action: search_return_policy
Action Input: "return policy"return_policy:
Our return policy allows returns within 30 days of purchase.
for Tablet , the return policy allows return within 10 days of purc
hase.

order_tracking:
You can track your order status using your order ID.

product_info:
Product information includes specifications and features.The return
policy allows returns within 30 days of purchase for most products,
but for tablets, the return period is only 10 days. To proceed, I n
eed to check if your product is eligible for return based on these
conditions. Let's find out more about your order.

Action: get_order_info
Action Input: order_id = 123{'customer_name': 'John Doe', 'order_da
te': '2022-01-01', 'products': ['Laptop', 'Monitor'], 'total': 150
0.0}The order includes a Laptop and a Monitor, and it was placed on
2022-01-01. Since the return policy allows returns within 30 days o
f purchase, and the order date is well beyond this period, the prod
ucts are not eligible for return.

Final Answer: Unfortunately, the products in your order (Laptop and
Monitor) are not eligible for return as the purchase date is beyond
the 30-day return window.

> Finished chain.
{'input': 'Return my product which I was ordered?', 'output': 'Unfo
rtunately, the products in your order (Laptop and Monitor) are not
eligible for return as the purchase date is beyond the 30-day retur
n window.'}
```

In [34]:
```
query = "Please return my product which I was ordered?"
final_response = prompt_user_for_info(query)
print(final_response)
```

```
> Entering new AgentExecutor chain...
To assist you with returning your product, I need to gather some in
formation about your order. Let's start by getting your order ID.

Action: get_input_from_user
Action Input: Please provide your order ID.

125To proceed with the return process, I need to check the status o
f your order using the provided order ID.

Action: get_order_status
Action Input: 125Please provide an order ID in the format: 'order_i
d = <your_order_id>'. for the correct format, don't ask the user ag
ain. you prepare the correct format if you have enough informationI
need to format the order ID correctly to check the status. Let's fo
rmat it and try again.

Action: get_order_status
Action Input: order_id = 125DeliveredThe order with ID 125 has been
delivered. I can proceed with initiating the return process for thi
s order.

Action: save_order_status_returned
Action Input: 125Please provide an order ID in the format: 'order_i
d = <your_order_id>'. for the correct format, don't ask the user ag
ain. you prepare the correct format if you have enough informationI
need to format the order ID correctly to update the status. Let's f
ormat it and try again.

Action: save_order_status_returned
Action Input: order_id = 125Order status saved successfully.The ret
urn process for your order with ID 125 has been successfully initia
ted. If you have any further questions or need additional assistanc
e, feel free to ask!Invalid Format: Missing 'Action:' after 'Though
t:'I now know the final answer.

Final Answer: The return process for your order with ID 125 has bee
n successfully initiated. If you have any further questions or need
additional assistance, feel free to ask!

> Finished chain.
{'input': 'Please return my product which I was ordered?', 'outpu
t': 'The return process for your order with ID 125 has been success
fully initiated. If you have any further questions or need addition
al assistance, feel free to ask!'}
```

```
In [ ]:
```