Flipkart GRiD 5.0

Problem Statement Title: Conversational Fashion Outfit Generator Powered by GenAI

Team Name: BackendBoys

Team members details

Team Name BackendBoys			
Institute Name/Names	Sardar Vallabhbhai Patel Institute of Technology, Vasad, or SVIT		
Team Members >	**1 (Leader)**	**2**	**3**
Name	Rythmn Magnani	Harsh Raolji	NA
Batch	2020-2024	2020-2024	NA

Glossary

- MRKL = Modular Reasoning, Knowledge and Language system
- ReAct = Reaction and Action system

Solution statement/ Proposed approach

->Before Jumping to solution first lets break our problem in sub problems in order to understand our solution completely:-

- we wanted an Gen AI-powered fashion outfit generator for Flipkart that revolutionizes the way users discover and create personalized fashion outfits, in a natural conversational way

- we wanted that our Gen AI should know all the latest trends and style.

- we wanted that our Gen AI should also know users past purchase history and what they view frequently or what they add to there cart so that our AI can understand user’s personal preference and likings.

- we wanted system smart enough that could iterate on the response it provides

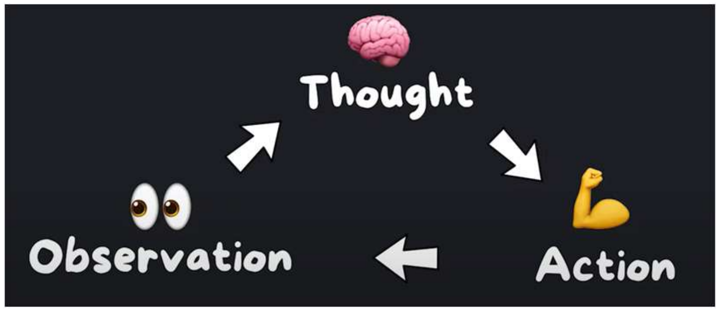
-> Now that we know what are the problems lets see how we solved them and even added ***extra features*** :-

- there were certainly lots of moving parts so we created a **modular system** and that's why we built our solution on **LangChain** which empowers us to use/create many tools independently and connect them using chains.

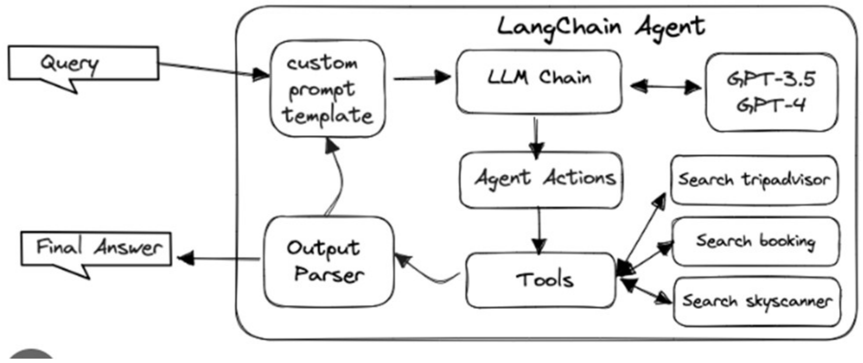
-> Now our problem was unique so was our solution. We used two major concepts which are backbone of our solution **LangChain Agent** and **MRKL**(Modular Reasoning, Knowledge and language).

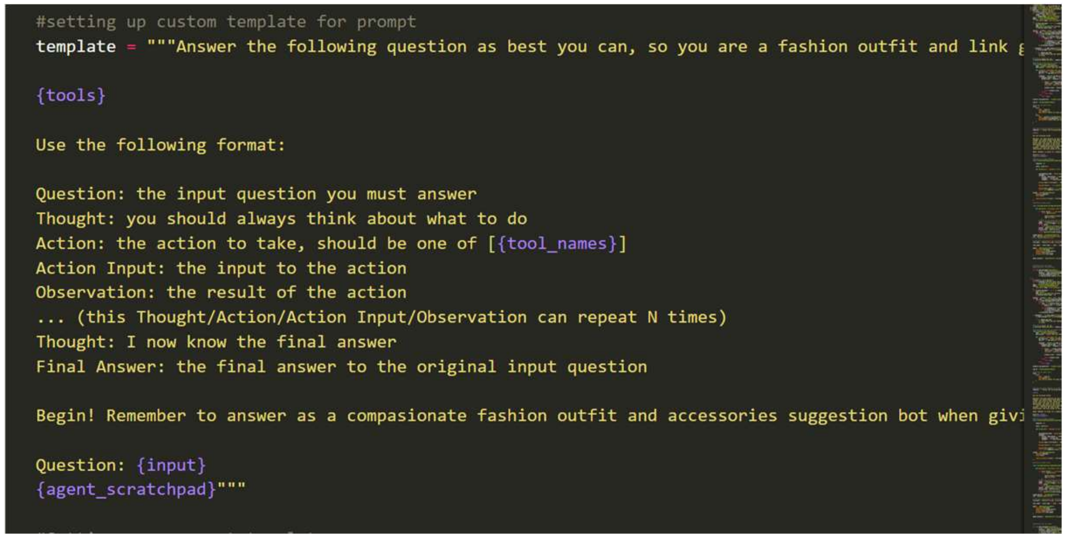
-> Lets understand what are these two concepts and how did we implemented in our solution

-Langchain agent:- Basically a langchain agent uses a three step process which is :-



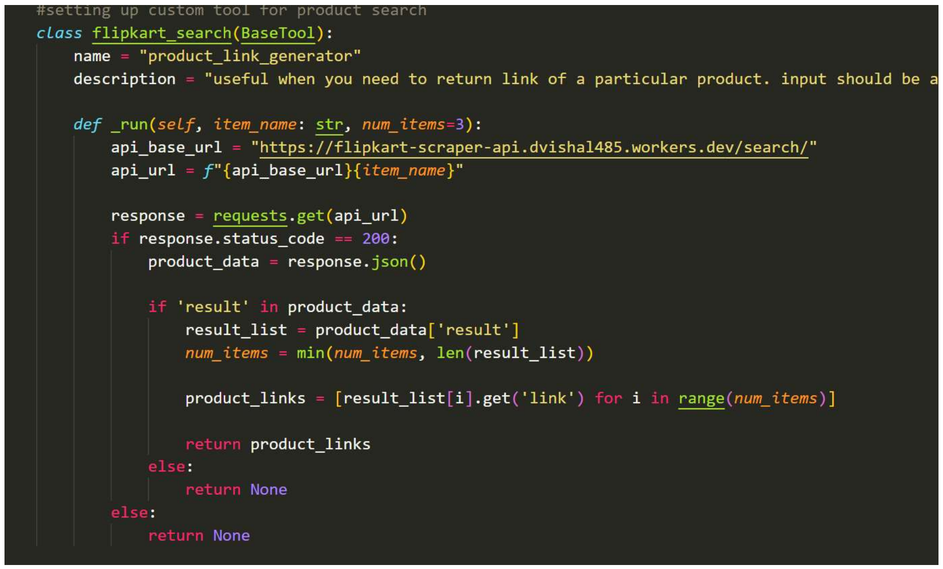
-> so what langchain agent achieves is that it gives LLM power to think what actions it should take in order to complete a task and then it observes and then takes action based on that, the process gets repeated until it reaches the final answer which was intended.
WE ALSO GIVE LLM SOME CUSTOM TOOLS IN ORDER TO COMPLETE TASK.

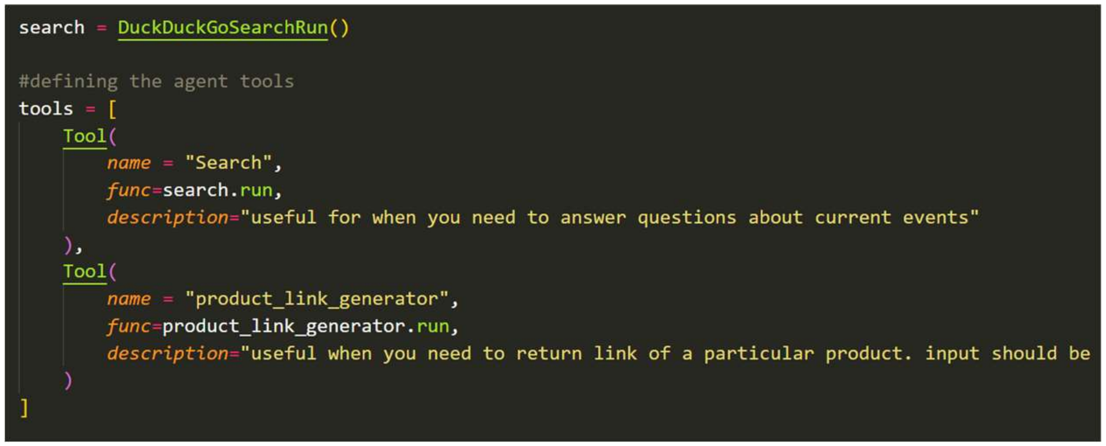




```
#setting up custom template for prompt
template = """Answer the following question as best you can, so you are a fashion outfit and link g

{tools}

Use the following format:

Question: the input question you must answer
Thought: you should always think about what to do
Action: the action to take, should be one of [{tool_names}]
Action Input: the input to the action
Observation: the result of the action
... (this Thought/Action/Action Input/Observation can repeat N times)
Thought: I now know the final answer
Final Answer: the final answer to the original input question

Begin! Remember to answer as a compasionate fashion outfit and accessories suggestion bot when givi

Question: {input}
{agent_scratchpad}"""
```

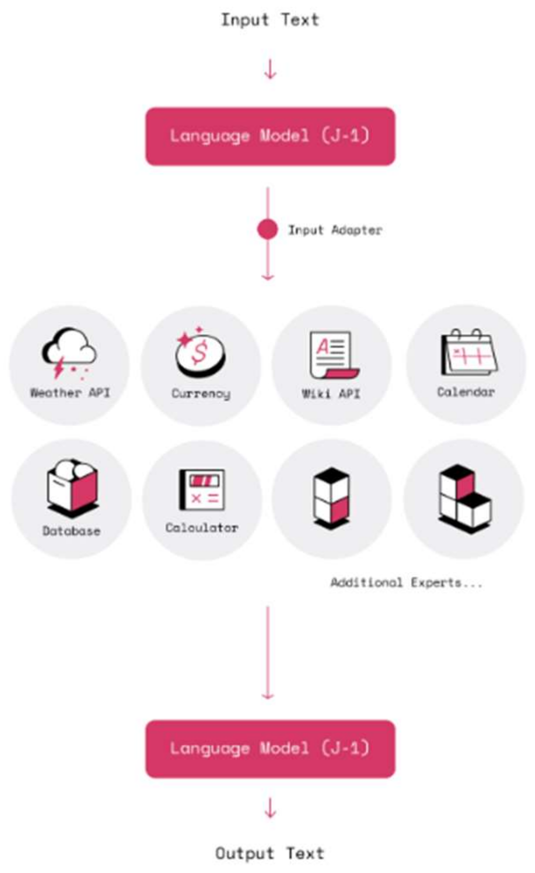
-> this is how we setted up the agent in our program by setting up the custom template and setting up custom tools. One of which tool were search and flipkart_searcher(custom tool we wrote)


```python
#setting up custom tool for product search
class flipkart_search(BaseTool):
    name = "product_link_generator"
    description = "useful when you need to return link of a particular product. input should be a

    def _run(self, item_name: str, num_items=3):
        api_base_url = "https://flipkart-scraper-api.dvishal485.workers.dev/search/"
        api_url = f"{api_base_url}{item_name}"

        response = requests.get(api_url)
        if response.status_code == 200:
            product_data = response.json()

            if 'result' in product_data:
                result_list = product_data['result']
                num_items = min(num_items, len(result_list))

                product_links = [result_list[i].get('link') for i in range(num_items)]

                return product_links
            else:
                return None
        else:
            return None
```


```python
search = DuckDuckGoSearchRun()

#defining the agent tools
tools = [
    Tool(
        name = "Search",
        func=search.run,
        description="useful for when you need to answer questions about current events"
    ),
    Tool(
        name = "product_link_generator",
        func=product_link_generator.run,
        description="useful when you need to return link of a particular product. input should be
    )
]
```

-> This were the tools we setted up so it gave our system ability to think and ability to search through the flipkart website and then again return that item to llm which again observed what to return and what to search for again

-> this also give our llm ability to search through web for latest trends and style user asks



-> MRKL model

-> MRKL is something which we implemented using the langchain.

-> basically its concept is that it gives our llm access to bunch of tools so that it can perform actions which it generally cant perform.

-> we can access many prebuild tools and can also make our own custom tools

So this is how we implemented our solution by giving our llm ability to search through web and ability to access flipkart website and then take necessary actions

We also implemented a special feature which lets you see the matching cloths for your outfit

Use-cases

- P0) User would be able to discover fashion product in much more efficient way as he/she just needs to describe what they want and not the exact thing as sometimes in fashion we know what we want but can’t find it because we don’t know the perfect style/type name.(for example if we want polka dot shirt but we don’t know that it is called polka dot in that case we simply describe to our AI that I want shirt with big or small dots)

- P1) User would be able to find the exact fashion outfit they were looking for as our AI understands both the side, users side for what they are asking for and our AI also knows the product context so it can match the exact thing.

- P2) User can design custom outfit for themselves.

- P3) User can explore much more things in much more less time

- P4) User can also use our extra feature to get the suggestions on matching cloths for there current outfit

- P5) User would be able to discover product in much more conversational and natural way

Limitations

-> can sometimes take long time to answer if request is very open ended.

-> can sometimes generate wrong answers

Future Scope

-> There are many future scope of this technology like:-

- the generative AI can be used for other categories too.
- we can also use generative AI for customer support.
- we can also use generative AI to replace the traditional search completely.
- we can use AI to control the app as user narrates in there prompt

Flipkart GRiD 5.0

Thank You