Table of Contents

1. Overview

This code repository is about developing an LLM chatbot using ChatGPT for Singapore's Finance Minister's Budget 2024. The code is based on the **LlamaIndex** framework that is designed for building RAG applications and consists of the following NLP capabilities:

1) **Event-driven Workflows**
    - This is the main paradigm chosen to build the orchestration agent and other agents that the orchestration agent coordinates with.
2) **Routing (or Tool Selection) via Zero-shot Classification**
    - This is the technique used by the orchestration agent. Before answering the query, the orchestration agent determines if the query is related to Budget 2024 and then chooses a suitable tool agent to answer. If the query is not related to Budget 2024, the orchestration agent will firmly state that it cannot provide an answer.
3) **RAG with Reranking**
    - This is the technique used by two of the tool agents. Each agent is connected to separate vector stores which serves as knowledge bases for different sections of information regarding Budget 2024. Each RAG agent also consists of a Reranker to improve the relevance of the retrieved documents by sorting the documents and further filtering out irrelevant ones.
4) **Function Calling Agents**
    - One of the tool agents uses function calling to overcome issues such as having information that is hard to be interpreted after parsing the PDF file into text. An example is the Medisave Bonus that is shown in a table with merged header cells. Information from the table is reorganised as conditional statements for the function calling agent to interpret, enabling the Budget 2024 Chatbot to provide more accurate information to the user.
    - While this code repository only uses function calling for Medisave Bonus, function calling can potentially be used to overcome other issues as well. Further examination on the data and iterations of query testing can be done to determine the best approach to overcome any issues. Function calling serves as one of the approaches that can be considered.

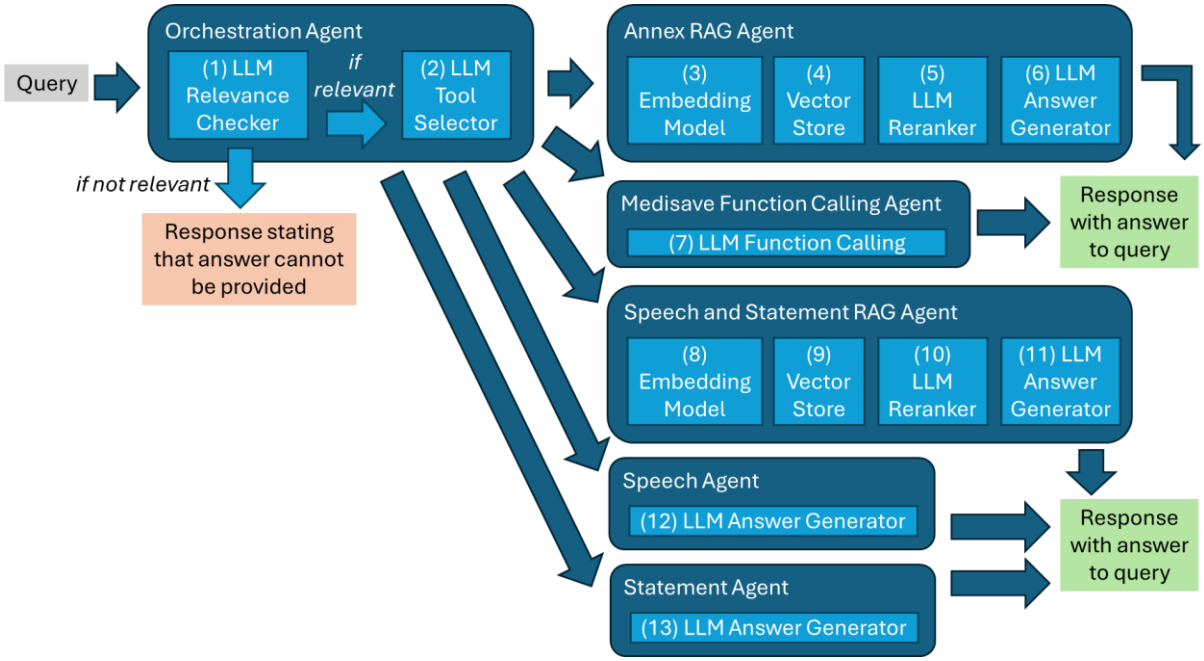
ChatGPT can be interpreted as the family of models offered by OpenAI. For this code repository, the OpenAI LLM used is **gpt-4o-mini** and the OpenAI text embedding model used is **text-embedding-3-small**.

Streamlit is used as the front-end framework for users to interact with via a web interface. The vector store used for RAG is a simple in-built memory vector store offered by LlamaIndex' s default setting.

2. Architecture Diagrams

2.1. Overall Querying Pipeline

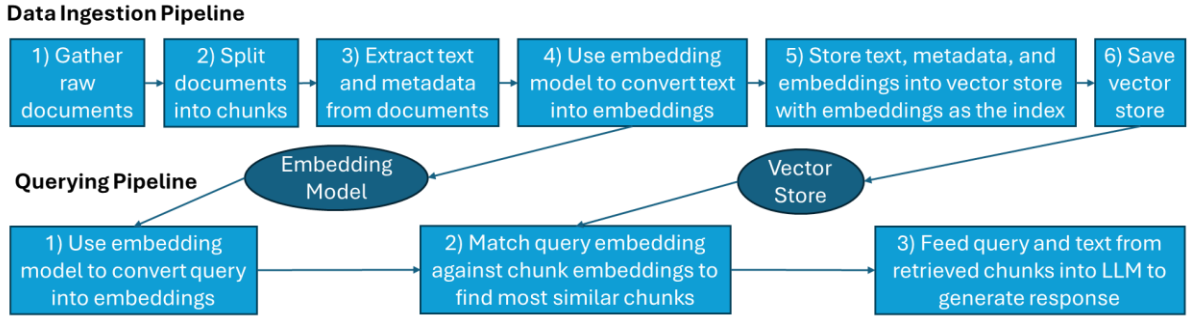
The Budget 2024 chatbot consists of the following components. Further details of the prompts and chunking strategies used can be found in Section 4 and Section 5.



2.2. More Detailed Pipelines for RAG

RAG is used to ground the LLM's outputs to the organization's knowledge base so that the LLM does not spew off outputs which may be inaccurate based on its knowledge cut-off. Instead of uploading the entire knowledge base for every query, RAG can be used to break up documents into smaller chunks and only have relevant chunks retrieved to answer queries. Therefore, with RAG, the number of input tokens used to answer queries can be reduced, allowing organizations to enjoy cost savings as a result.

RAG consists of two main pipelines. There is a data ingestion pipeline for setting up an RAG application and a querying pipeline that takes in queries for processing. Further details of the chunking strategies used can be found in Section 4.

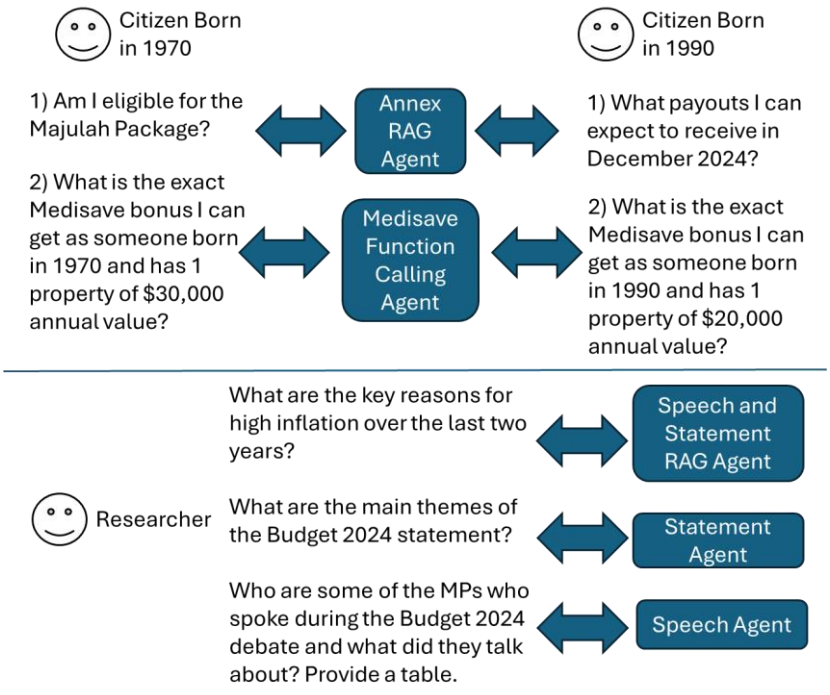


3. Prompts for Assumed Budget 2024 Chatbot Users

Two personas are assumed from the 3 sample queries given. The two personas are:

1) Common Citizen
    - This user is interested to know about the benefits given from Budget 2024 and would like to know details such as the eligibility, timeline and payout.
    - Sample Prompts:
        i. Am I eligible for the Majulah Package?
        ii. What are the payouts I can expect to receive in December 2024?
        iii. What is the exact Medisave bonus I can get as someone born in 1990 and has 1 property of $20,000 annual value?
            - Question iii serves as a possible follow-up question for either Question i or Question ii.
2) Researcher
    - This user would like to study the economic trends, policy considerations, and other details from the parliamentary proceedings for Budget 2024.
    - Sample Prompts:
        i. What are the key reasons for high inflation over the last two years?
        ii. What are the main themes of the Budget 2024 statement?
        iii. Who are some of the MPs who spoke during the Budget 2024 debate and what did they talk about? Provide a table.

The various agent tools used in the Budget 2024 chatbot cater for the above two personas as seen from the chain of sample queries below. Different tools may be used for different questions, but they all appear as one unified chatbot to the user.



4. Comparison of Chunking Strategies

Based on observations made when using LlamaIndex's default RAG method (i.e. treats every PDF page as one document and splits each page into smaller chunks if exceed 512 tokens), several steps were taken to address the shortcomings observed. The descriptions below roughly capture the thought processes that led to the final approach.

1) **Deciding how to parse annex PDFs into the vector store**
    - Query: Am I eligible for the Majulah Package?
    - Result: You are eligible for the Majulah Package if you were born in 1973 or earlier, have an average monthly income within the specified range, live in a residence with an Annual Value of $25,000 and below, and own not more than one property.
    - Observation: The answer gives extra info that is pertaining only to Earn and Save bonus (i.e. Annual value 25,000 and below), which makes the answer to the question wrong because the eligibility criteria for Majulah Package is not so stringent. Only the Earn and Save bonus component has this additional criterion. The LLM made this mistake because it only sees the first page of the Annex F2 document and not the entire Annex F2 document.
    - Decision: Do not split up annex PDF documents into pages. Instead, store each annex PDF in its entirety in the vector store. After all, each annex PDF is not too long and fits within the 8191 token limit of the embedding model.
2) **Deciding on number of RAG agents**
    - Query: What are the key reasons for high inflation over the last two years?
    - Observation: The default RAG method can answer this question correctly.
    - Decision:
        - Have separate RAG agents with its own vector store
            - One for annexes from the budget statement. This RAG agent can use the chunking strategy described in (1). [Annex RAG Agent]
            - The other for the budget statement (without annexes) and the budget debate round up speech. This RAG agent can use the default chunking strategy. [Speech and Statement RAG Agent]
3) **Handling infographics**
    - Query: What are the payouts I can expect to receive in December 2024?
    - Observation: The query cannot be answered correctly because the page containing this information is from a PDF document of infographics that cannot be parsed properly.
    - Decision:
        - Filter out information from the infographics PDF document that is not covered by other documents (i.e. calendar summary, individual summary, household summary)

- Manually curate the filtered information into text files and add to the annex rag agent

4) Handling questions that require the document to be read in entirety

- Query: List the MPs mentioned in the debate round up speech and also list down what they spoke about.
- Observation: Chunks from the Speech and Statement RAG agent cannot fully answer this question as the MPs are spread out across different chunks.
- Decision:
  - Create two more agent tools. One for the budget statement and one for the budget debate round up speech.
  - Both agent tools will provide the entire text as context for the LLM to answer queries.
  - This is possible because each of the text files are well below the 128,000 token limit of the LLM.

5) Handling tables with merged cells

- Query: What is the Medisave bonus I can get as someone born in 1970 and own 1 property with $30,000 annual value?
- Observation: The LLM seems to be unable to interpret the merged table cells properly and gave incorrect information as a result.
- Decision:
  - Create a function calling tool that consists of conditional statements which reorganise the information found in the nested table
  - There can be one combined function calling tool for the Medisave information in both Annex F2 and Annex F3 because they have similar conditions as seen below.
  - Annex F2 Majulah Package

Table 3: Quantum for the MSB

Singaporeans born in	Own not more than 1 property		Own more than 1 property
	AV of residence		
	Not more than $25,000	More than $25,000	
1959 or earlier	$750		
1960-1973	$1,500	$750	

  - Annex F3 One-time Medisave Bonus

Table 1: Bonus Quantum

Singaporeans born in	Own not more than 1 property		Own more than 1 property
	AV of residence		
	Not more than $25,000	More than $25,000	
1974-1983	$300		$200
1984-2003	$200		$100

5. Prompt Templates and Other Component Details

Below are the prompts and details of each agent component in the chatbot.

5.1. Orchestration Agent

Component 1 uses the following prompt template to check if the query is relevant to Budget 2024.

```
RELEVANCE_CHECK = """
Use the description of each tool to determine if the query can be answered by any tool.
Return a boolean on whether the query can be answered by any tool and an explanation of the decision.

Tools:
{tool_descriptions}

Query: {query}
"""
```

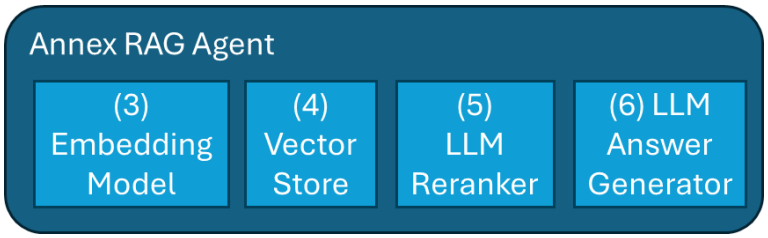
If query is irrelevant, Component 1 uses the following prompt template to explain to a user why the query cannot be answered. The template takes into account the role and tone to be adopted for various situations.

```
IRRELEVANT_RESPONSE = """
You are public officer communicating information about the Singapore Budget 2024.
You are explaining to a user why the query cannot be answered by any tool.

In your explanation, use the following pointers:
- The query is not related to Singapore's Budget 2024
- Query: {query}
- Explanation: {explanation}

Adopt a professional tone and provide a clear and concise explanation in a friendly and polite manner.
If the question is rude or inappropriate, respond with a firm but diplomatic message that is not overly friendly.
"""
```

If the query is relevant, the query is passed on to Component 2 which uses the following default prompt to select the best tool to answer the query.

```
# Default llamaindex prompt used by LLM router.
# Putting here for ease of reference.
DEFAULT_SINGLE_SELECT_PROMPT_TMPL = (
    "Some choices are given below. It is provided in a numbered list "
    "(1 to {num_choices}), "
    "where each item in the list corresponds to a summary.\n"
    "---------------------\n"
    "{context_list}"
    "\n---------------------\n"
    "Using only the choices above and not prior knowledge, return "
    "the choice that is most relevant to the question: '{query_str}'\n"
)
```

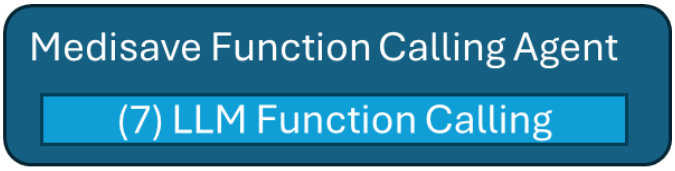
5.2. Annex RAG Agent



Component 3 is used to embed the incoming query. The embedded query is then matched against Component 4 to find the **top 3 most relevant chunks**. The top 3 most relevant chunks are then passed to Component 5 to select the **top 1 chunk** (i.e. the most appropriate annex document). The top 1 chunk and the query are then passed to Component 6 to generate the answer via the following template which considers the URL citation:

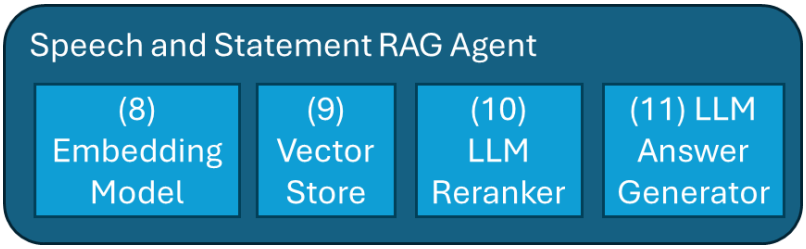
```
ANNEX_RAG_PROMPT_TMPL = """
{query}

In your response to the above query using the given context,
provide the following URLs in the response as reference:

{url_str}
"""
```

5.3. Medisave Function Calling Agent



Component 7 does function calling with a function that consists of a bunch of conditional statements. Each result to be returned via the conditional statement consists of a citation to Annex F2 or Annex F3.

5.4. Speech and Statement RAG Agent



Component 8 is used to embed the incoming query. The embedded query is then matched against Component 9 to find the **top 5 most relevant chunks**. The top 5 most relevant chunks are then passed to Component 10 to select the **top 3 chunks**. The top 3

chunks and the query are then passed to Component 11 to generate the answer via the following template which considers the URL citation:

```
SPEECH_RAG_PROMPT_TMPL = """
{query}

In your response to the above query, cite the paragraph numbers where the information can be found.
Also, provide the following URLs in the response as reference:

{url_str}
"""
```

5.5. Speech Agent and Statement Agent

Component 12 and Component 13 have similar structures. Each of them takes in the entire PDF document and uses the following prompt template to cite the paragraph numbers and the URL.

```
ENTIRE_TEXT_PROMPT_TMPL = """
Context:
{context}

Query:
{query}

In your response to the above query using the given context,
provide the following URLs in the response as reference:

{url_str}
"""
```

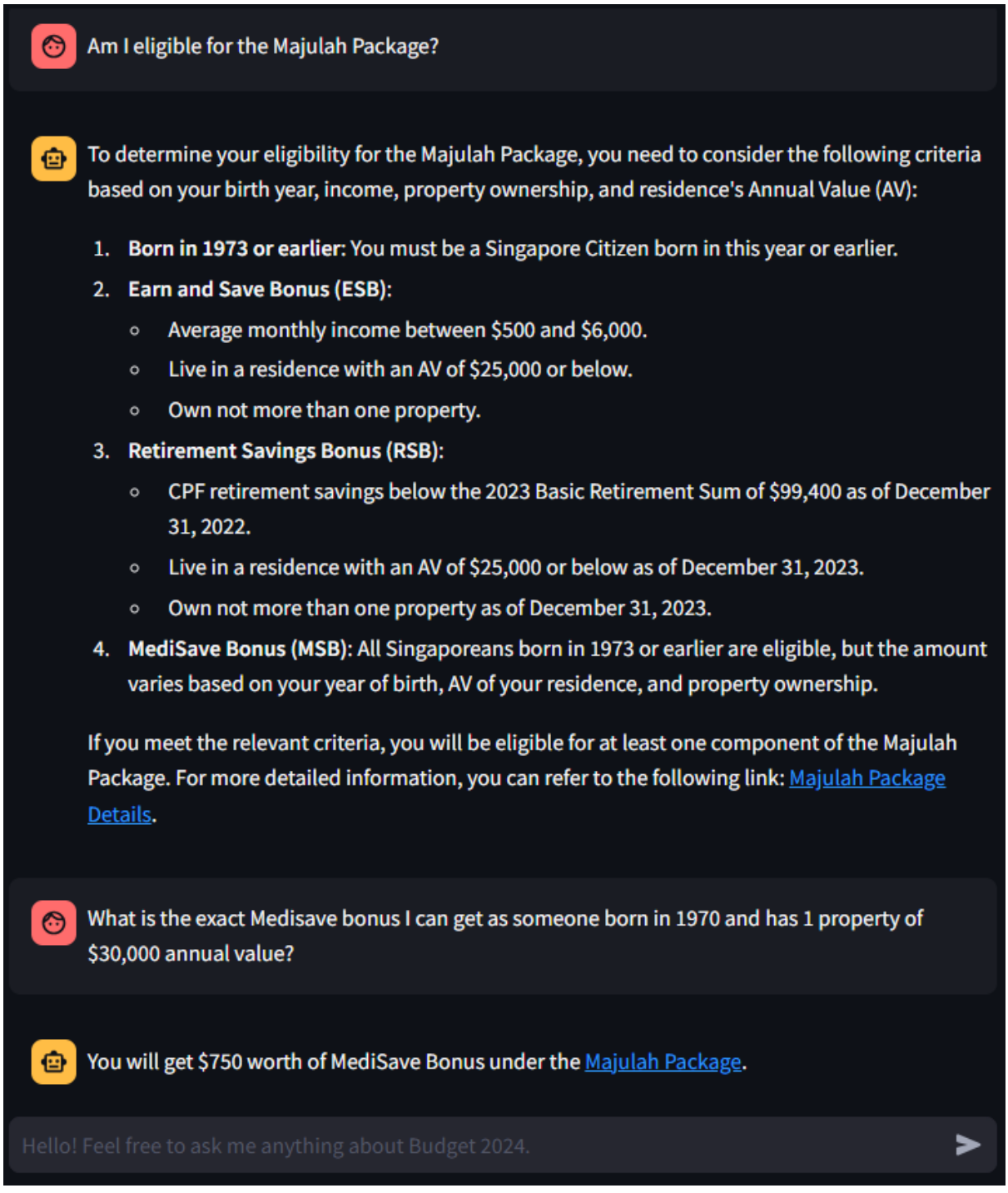
6. Possible Future Enhancements for Production

- Cloud Deployment
  - It is possible to deploy this chatbot web application via public cloud providers such as AWS, Azure, GCP, etc.
  - EC2 on AWS which can be seen as a server-oriented option is a possibility.
  - Serverless options include AWS Fargate or GCP cloud run
  - My experience in deploying AI web applications on the cloud can be found in a paper I published with my SMU professors at the Machine Learning with Applications (MLWA) journal.

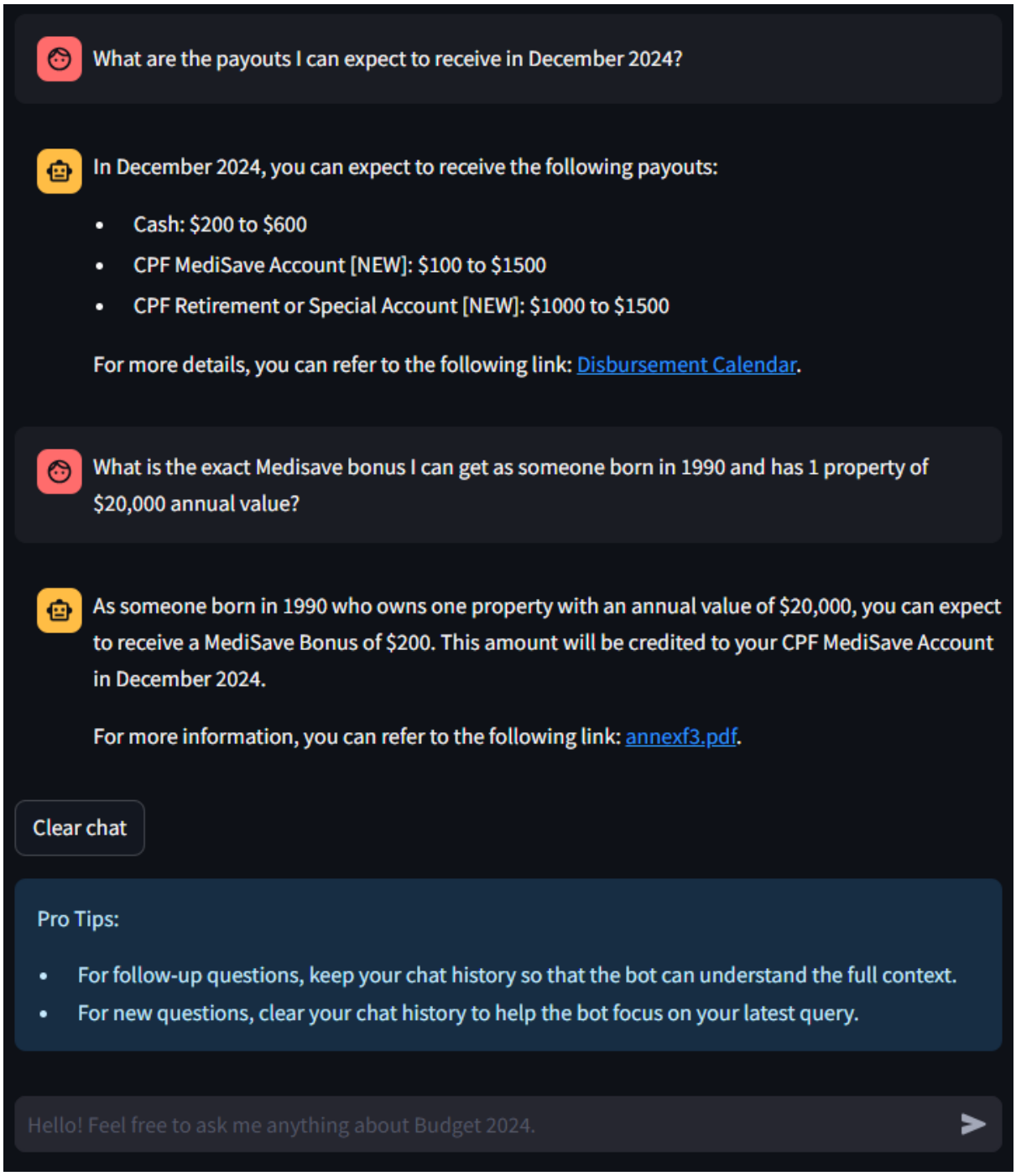
- Analytics via Tracing Observability
    - With the LlamaIndex framework, tracing can be done via LlamaTrace which is built upon the Arize Phoenix project.
    - This allows AI engineers to be able to effectively debug, observe and evaluate the chatbot application.
- REST APIs
    - It is possible to separate out Streamlit as a frontend web app that interacts with the backend REST API server.
    - This backend REST API can be built with Python frameworks such as Flask or FastAPI.
    - With this separation of frontend and backend components, compliance with the certain architecture standards that require frontend components and backend components to be in separate tiers can be achieved.
    - My experience in deploying REST APIs for AI web applications can also be found in the paper I published.
- On-premise Deployment Considerations
    - There may be a need to deploy applications that are on-premise with an air-gapped environment that cannot access the internet.
    - In such an environment, OpenAI's LLMs and embedding models cannot be used since these models can only be accessed by calling APIs that are hosted on OpenAI's servers.
    - Alternatives to OpenAI's LLMs can be Meta's open-sourced Llama family of models as well as open-sourced variants that are built on top of Llama.
    - Alternatives to OpenAI's embedding models include open-sourced models that can be found on HuggingFace.
    - The local instance of Arize Phoenix can be used as well for analytics via tracing purposes.

Annex: Screenshots of Chatbot Web App

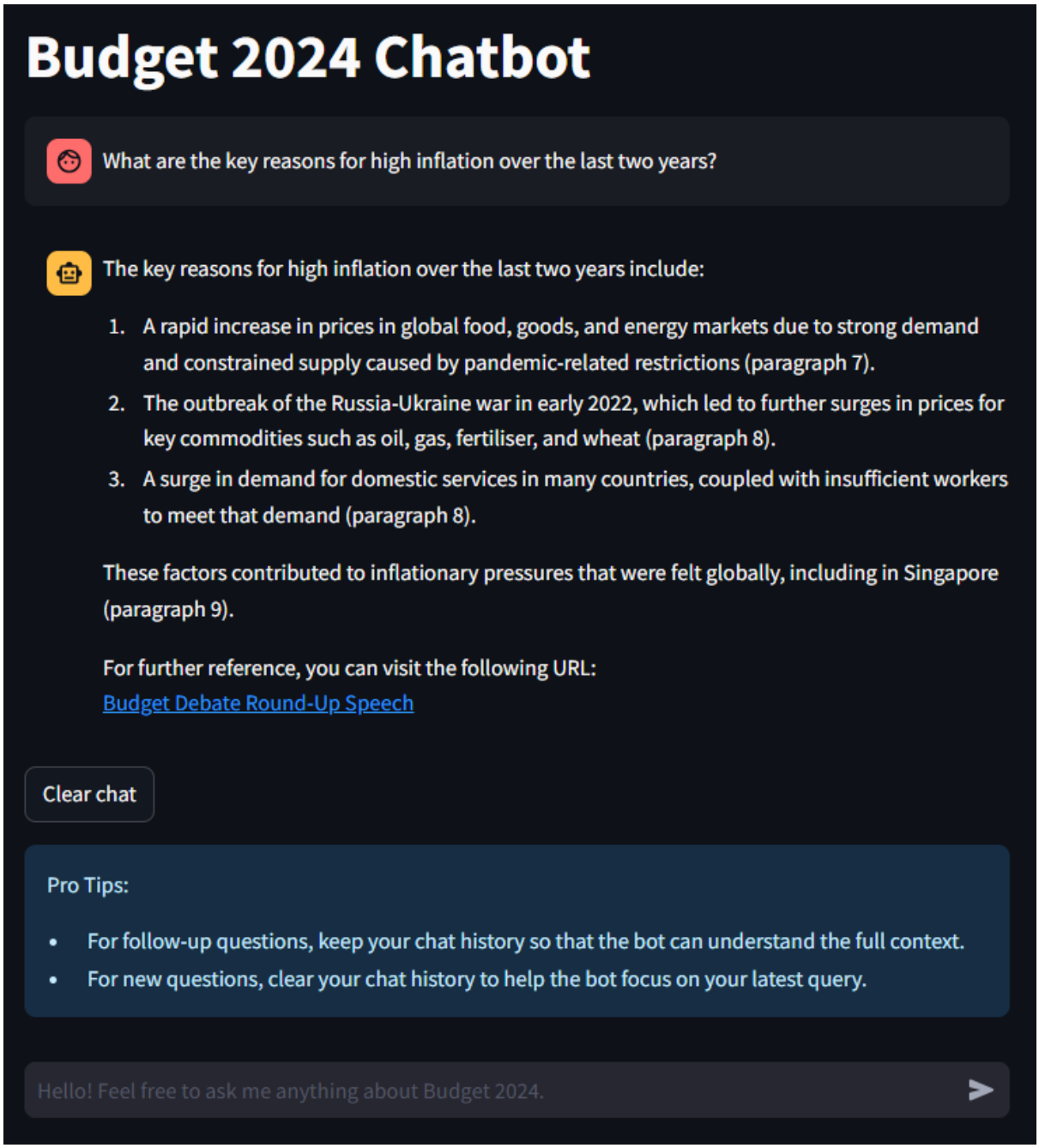
Question on Majulah Package with follow-up question



Question on December payout with follow-up question



Question on inflation



Question on budget 2024 statement

1. **Tackling Immediate Challenges**: The government acknowledges the pressures faced by households due to rising living costs and aims to provide immediate financial support through enhanced assistance measures.
2. **Pursuing Better Growth and Jobs**: There is a strong emphasis on fostering a vibrant economy through innovation, productivity, and attracting quality investments to create good jobs for Singaporeans.
3. **Equipping Workers for Life**: The budget highlights the importance of continuous skills upgrading and lifelong learning, particularly for mid-career workers, to adapt to changing job markets and technological advancements.
4. **Creating Paths Towards Equality and Mobility**: The government aims to address income inequality and enhance social mobility through targeted support for lower-wage workers and initiatives to uplift families from less privileged backgrounds.
5. **Providing Assurance for Families and Seniors**: There are plans to strengthen support systems for families at various life stages, including childcare, education, housing, and retirement adequacy.
6. **Forging a Stronger and More United Nation**: The budget emphasizes national resilience, security, and community solidarity, highlighting the importance of social cohesion in facing external challenges.
7. **Securing Our Fiscal Position**: The statement underscores the need for fiscal responsibility and sustainability to ensure that the government can continue to invest in the economy and social programs.

These themes reflect a comprehensive approach to addressing both immediate and long-term challenges faced by Singaporeans, aiming to build a more resilient and inclusive society.

For more detailed information, you can refer to the full Budget 2024 statement at the following URL: Budget 2024 Statement.

Hello! Feel free to ask me anything about Budget 2024.

Question on list of MPs from budget debate speech