**Multi-Agent System for Persona-Driven Technical Documentation**

**Problem Statement:**

To build a **multi-agent system** that takes a single source of truth - a **Python function's docstring** - and generates tailored documentations for three different personas, as follows, that work in **parallel**:

1. **Developer Documentation Agent**: Creates precise technical reference documentation for other engineers.
2. **Executive Summary Agent**: Writes high-level, jargon-free summary, focusing on business value.
3. **API User Guide Agent**: Generates a practical "how-to" guide with code examples for end-users.

Also, deploy a Streamlit app as part of the working setup.

**Technical Design:**

1. Designed an **autonomous** system using **Autogen** framework that uses the GroupChat, GroupChatManager features to orchestrate and manage the conversations between the **three agents**- Developer Documentation Agent, Executive Summary Agent and the API User Guide Agent. Used **LLM**, ‘GPT-4o-mini', which is deployed on **Azure**.
2. Extended the user interactivity of the framework into **Streamlit**. Saved the **secret keys** of Azure in **secrets.toml** file locally to run file and utilized the ‘**Secrets’ tab** of Streamlit to safely add these and **deploy app** on Streamlit.
3. User input into the system is collected through the Streamlit app. The **interacting agents, intermediate** and **final outputs** are collected and **displayed on the Streamli**t app. This utilizes the ‘register\_reply’ functionality of the agents to collect the response of the agents through reply\_func parameter immediately after they are generated and these are streamed on Streamlit using a function, ‘st\_print’.
4. Also utilized **async** operation, using ‘asyncio’, for parallel processing of the conversations between the agents in Group Chat.
5. Used some Streamlit features to enhance the **user interface** and **user experience** (UI/UX) of the webpage
6. Set the temperature of LLM to **0.1** to limit its creativity and use only facts and web results knowledge to generate responses.
7. **Deployed** the app on Streamlit at **URL**:

<https://agentic-ai-assessment-autogen-multi-agents.streamlit.app/>

**Agent Prompts:**

**Developer Documentation Agent:**

*system\_message*=(

"You are the Developer Documentation Agent. You will be passed a docstring."

"You must develop a technical documentation on it, along with the different technical scenarios it can be used in."

"For each parameter input to the function, you must mention the different use-cases it can be used in and the different forms it can take."

"Use the general knowledge available to you and gather as many technical scenarios and technical examples you can collect to describle and explain each technical term present in the docstring."

"Do not execute code statements, you just are supposed to write about them."

)

**Executive Summary Agent:**

*system\_message*= (

"You are the Executive Summary Agent. You focus on the business value."

"Focus on the business value and briefly explain what the function is about from the docstring passed to you."

"Use high-level and jargon-free language to describle what the function does, focusing on the business value, and not on the technical terms or explanations."

)

**API User Guide Agent:**

*system\_message*= (

"You are the API User Guide Agent."

"You will be passed a docstring, create example scenarios and practical 'how-to' guide with codes for the users of the function."

"First focus on the main scenarios and usecases of the function and mention them with examples."

"Then, focus on the cases around these, slowly moving into the edge and corner cases."

"Ensure that you cover and create all possible scenarios and examples where the function can be used, including the edge and corner cases."

)

**Group Chat Manager System Message:**

manager\_system\_message = (

"You are the Group Chat Manager. You are responsible for managing the conversation between the three agents: The Developer Documentation Agent, the Executive Summary Agent and the API User Guide Agent."

"Conduct and orchestrate the group chat, ensuring that each of the three agents are called exactly once."

"You must ensure that you receive a detailed and clear documentation/summary/report from each of the three agents from the conversation."

"Order of agent calls doesn't matter, but ensure that each agent gets exactly one chance to participate in the group chat."

"As soon as you receive output from all the three agents, terminate the chat."

)

**Group Chat Manager Description Message:**

description\_manager = (

"The group chat manager agent - this agent should be the first to engage when given a new task."

)

**Agent Architecture:**