```
# Importing necessary modules
import os

from dotenv import load_dotenv

from swarms import Agent
from swarm_models import OpenAIChat
from swarms_memory import ChromaDB
from swarms.prompts.visual_cot import VISUAL_CHAIN_OF_THOUGHT
from swarms import tool

# Loading environment variables from .env file
load_dotenv()

# Getting the Gemini API key from environment variables
gemini_api_key = os.getenv("GEMINI_API_KEY")
openai_api_key = os.getenv("OPENAI_API_KEY")

llm = OpenAIChat(
    openai_api_key=openai_api_key,
    max_tokens=1000,
    temperature=0.2,
)

# Making an instance of the ChromaDB class
memory = ChromaDB(
```

```
    metric="cosine",
    n_results=3,
    multimodal=True,
    # docs_folder="images",
    output_dir="results",
)


# Defining tool by creating a function and wrapping it with the @tool decorator and
# providing the necessary parameters and docstrings to show the usage of the tool.
@tool
def make_new_file(file: str, content: str):
    """
    Make a new file.

    This function creates a new file with the given name.

    Parameters:
        file (str): The name of the file to be created.

    Returns:
        dict: A dictionary containing the status of the operation.
    """
    with open(file, "w") as f:
        f.write(f"{content}")
```

```
# Initializing the agent with the Gemini instance and other parameters
agent = Agent(
    llm=llm,
    agent_name="Multi-Modal RAG Agent",
    agent_description=(
        "This agent fuses together the capabilities of Gemini and"
        " Visual Chain of Thought to answer questions based on the"
        " input image."
    ),
    max_loops="auto",
    autosave=True,
    sop=VISUAL_CHAIN_OF_THOUGHT,
    verbose=True,
    # tools=[make_new_file],
    long_term_memory=memory,
)


# Defining the task and image path
task = (
    "What is the content of this image, return exactly what you see"
    " in the image."
)
img = "images/Screenshot_48.png"
```

```
# Running the agent with the specified task and image

out = agent.run(task=task, img=img)

print(out)
```