```
import os

from dotenv import load_dotenv

from swarm_models import GPT4VisionAPI
from swarms.prompts.personal_stylist import (
    ACCESSORIES_STYLIST_AGENT_PROMPT,
    BEARD_STYLIST_AGENT_PROMPT,
    CLOTHING_STYLIST_AGENT_PROMPT,
    HAIRCUT_STYLIST_AGENT_PROMPT,
    MAKEUP_STYLIST_AGENT_PROMPT,
)
from swarms.structs import Agent

# Load environment variables
load_dotenv()
api_key = os.getenv("OPENAI_API_KEY")

# Initialize GPT4VisionAPI
llm = GPT4VisionAPI(openai_api_key=api_key)

# User selfie and clothes images
user_selfie = "user_image.jpg"
clothes_image = "clothes_image2.jpg"

# User gender (for conditional agent initialization)
```

```
user_gender = "man"  # or "woman"

# Initialize agents with respective prompts for personal styling
haircut_stylist_agent = Agent(
    llm=llm,
    sop=HAIRCUT_STYLIST_AGENT_PROMPT,
    max_loops=1,
    multi_modal=True,
)

# Conditional initialization of Makeup or Beard Stylist Agent
if user_gender == "woman":
    makeup_or_beard_stylist_agent = Agent(
        llm=llm,
        sop=MAKEUP_STYLIST_AGENT_PROMPT,
        max_loops=1,
        multi_modal=True,
    )
elif user_gender == "man":
    makeup_or_beard_stylist_agent = Agent(
        llm=llm,
        sop=BEARD_STYLIST_AGENT_PROMPT,
        max_loops=1,
        multi_modal=True,
    )
```

```
clothing_stylist_agent = Agent(
    llm=llm,
    sop=CLOTHING_STYLIST_AGENT_PROMPT,
    max_loops=1,
    multi_modal=True,
)

accessories_stylist_agent = Agent(
    llm=llm,
    sop=ACCESSORIES_STYLIST_AGENT_PROMPT,
    max_loops=1,
    multi_modal=True,
)

# Run agents with respective tasks
haircut_suggestions = haircut_stylist_agent.run(
    (
        "Suggest suitable haircuts for this user, considering their"
        " face shape and hair type."
    ),
    user_selfie,
)

# Run Makeup or Beard agent based on gender
if user_gender == "woman":
    makeup_suggestions = makeup_or_beard_stylist_agent.run(
```

```
        (
            "Recommend makeup styles for this user, complementing"
            " their features."
        ),
        user_selfie,
    )
elif user_gender == "man":
    beard_suggestions = makeup_or_beard_stylist_agent.run(
        (
            "Provide beard styling advice for this user, considering"
            " their face shape."
        ),
        user_selfie,
    )

clothing_suggestions = clothing_stylist_agent.run(
    (
        "Match clothing styles and colors for this user, using color"
        " matching principles."
    ),
    clothes_image,
)

accessories_suggestions = accessories_stylist_agent.run(
    (
        "Suggest accessories to complement this user's outfit,"
```

```
        " considering the overall style."
    ),
    clothes_image,
)
```