Generative-AI Project

Title: AI-Power Fashion Designing Agent

Introduction:

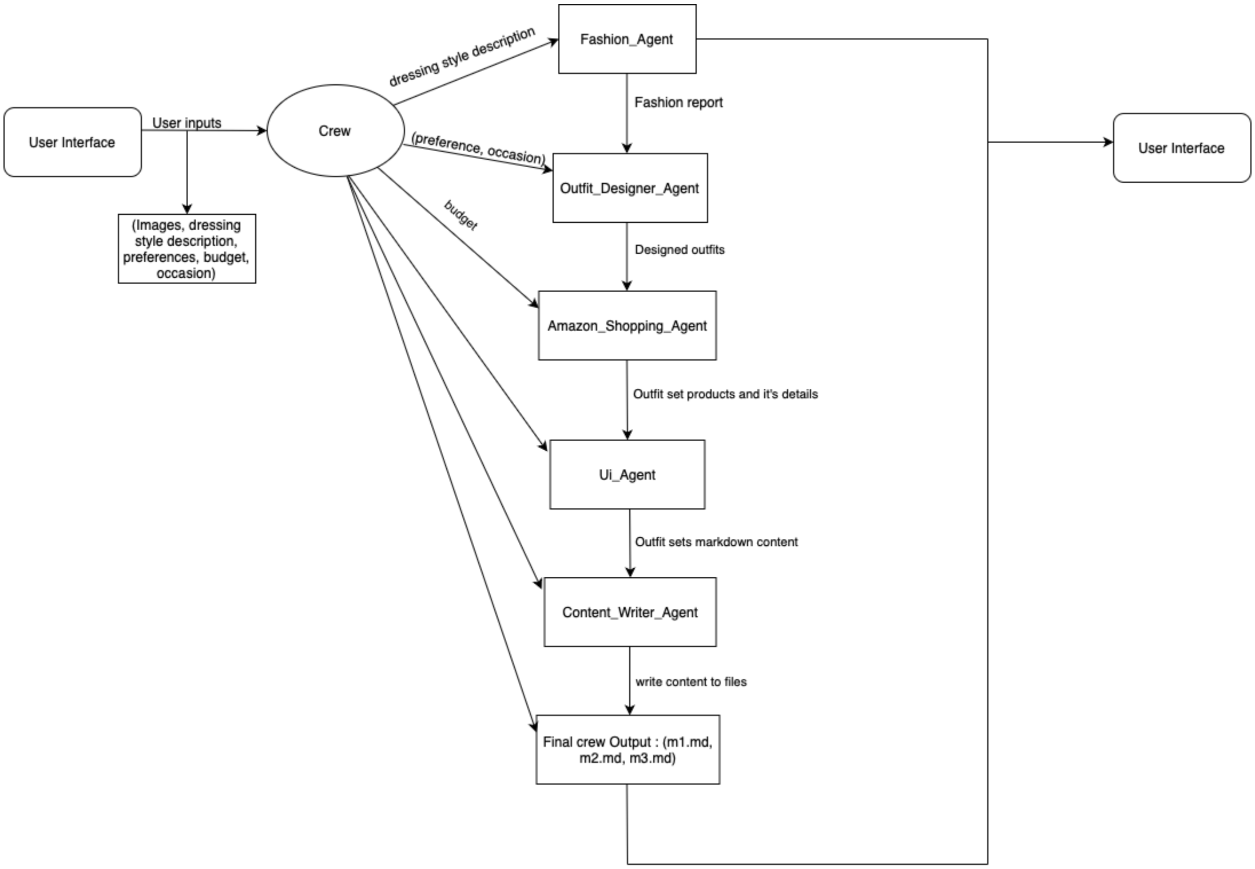
This an AI-Powered Fashion Agent, which create the 3 new innovate complete outfits sets (top wear, bottom wear, footwear, accessories) and provide well represented blog with complete details of each outfit set with summary of outfit, title, price of each product, each product link and product images.

These outfits created based on the user uploaded images, user dressing style description, user budget, user preference and user occasion.

Existing Solution:

These are the few existing solutions (Stylebook, clad well, and smart closely) with some limitations: understanding complex fashion cultural preferences, aesthetics.

My Approach:



Domain knowledge:

The project is related to Fashion, getting domain knowledge in it important and helpful for building the application. Initially I started exploring details about fashion designing, how to describe persons fashion style, how to design outfits etc. Based on this knowledge I start building my agents.
First given pro

User-interface:

Based upon the project requirement, I created different fields (Upload images, dressing descript, budget, preferences, reason for shopping) for collecting user data which is required to pass it to backend running crew agents. Once user click the submit button with provided details then I designed in a way that in my code, to start the crew execution with the user provided data as input to crew.

I created one dedicated directory "images" and implemented code in a way that, store user uploaded images in this directory.

Along with different fields, I created 4 buttons on top of the interface i.e;

Set back: Responsible for going back between pages.

Get saved Outfits: Responsible for getting and display user saved outfits.

Clear images: Responsible for clearing the user uploaded images in images directory in project file structure.

Reset: Responsible for clearing the current user data and make application ready for new user request.

Save outfit: Responsible for saving the user outfit under name of user.

Backend system:

For this project, I used AI agent to implement the whole project. By using crewai I created crew with 5 agents (Fashion_agent, Outfit_designer_agent, Amazon_shopping_agent, Ui_agent and content_writer_agent).

Fashion_agent: I designed this agent in a way, to analyze the user uploaded images and finding all details of fashion factors(these factors are designed in domain knowledge phase) and make them to create one fashion report with details of fashion factors, summary of user dressing style, and key fashion terms.

For this agent, I created one custom crewai tool (fashion tool) by inheriting the crewai tool base class. This tool is responsible for all things which are mentioned in Fashion_agent.

Within this tool, used the vision model(gpt-4o). This agent uses the description of user dressing style content in its task.

Outfit_designer_agent: I designed this agent in a way, to create the 3 new innovative outfits based on the fashion report (created by Fashion_agent), user occasion, and preferences.

This agent uses the user provided preference, reason for shop content in its task. And this agent gives the 3 new outfit sets, in each outfit set it provide details of (top, bottom, footwear, accessories) and outfit summary.

E.g.: Top: Red color polo shirt, bottom: Black color jeans, footwear : white sneakers, accessories : silver color wrist watch.

Amazon_shopping_agent: I designed this agent in way, to perform the big task “shopping”.

For this agent I created one custom crewai tool (amazon tool), which is responsible for doing the shopping all given items over amazon.com and picks the items for the provided outfits.

Picking up the items over search result is challenging, so I come up with an idea called “human like selection”. First, I created list of outfit combination from the search results of each item of outfit and then classified the outfit combination list into two based on user provided budget i.e., (budget-based outfit and fashion-based outfit). Fashion based outfits provide the reason regarding why this product selected.!

Fashion based outfits are more accurate because I used vision model on the outfits sets to pick best item for each product based on the outfit summary.

Budget based outfits are random choice from the list of outfit sets.

At final, this agent provides the result of Fashion based outfit and budget-based outfit for each outfit with complete details about each item of outfit (product name, product URL, product price, and product image URL).

Ui agent: I designed this agent in a way, to create the well organized and represented markdown content for each outfit set provide by the Amazon_shopping_agent.

Created the set of rules in pseudo code format which make sure markdown content created in same specified format every time.

At final, this agent provides the markdown content for each outfit set.

Content_writer_agent: I designed this agent in a way, to write the markdown content of each outfit set into corresponding markdown file.

After the execution of all agents, the next part of the code read the markdown files content and display it on user interface, user can see the well organized and represented complete details of 3 outfit sets.

Save and interaction:

To save and retrieve the outfit set, to add this feature in project I used python file concept i.e., I created one button "save outfit set", once user click the button it asks to enter name, once name entered new directory going to create with user provide name and create one file under it and write the content of that outfit set.

To retrieve the saved outfits, designed one button called "Get saved outfit", once click then it asks to enter name and with user provided name it searches the any directory is present with user provided name or not. If present it read all the file in it and display to user-on-user interface.

Explanation of code:

In app.py file, have 4 methods i.e; (main(), crew(), agents_task(), clear_images())

The application starts with main(), crew is responsible for creating the crew with list of agents and task.

The agents_task() is responsible for creating the agents and tasks and clear_images is responsible for deleting the images under images directory.

The fashion.py file is custom crewai tool created by inheriting the crewai tool base class. This is responsible for analyzing the user images and creating report which is used by the Fashion_analyst_agent.

The amazontool.py is custom crewai tool created by inheriting the crewai tool base class. It has following methods i.e;

Serper_search() – responsible for searching the over internet for the given query.

Get_number() – responsible for filtering the price value of e.g.:'$98 ioio" -→ 98.

Get_budget_based_outfits() – responsible for creating the list of outfit combinations by taking the input parameter of each item products list.

Prepare_product_type_items(): responsible for creating the dictionary of key as product name and value as product image URL. This dictionary is used for vision model for selecting best product.

Get_best_fashion_item(): responsible for selecting the best product.

Search_outfit_list(() : responsible for getting the details of best selected product.

Get_final_fashion_outfit() : responsible for creating the final output string for fashion based outfit.

Shop_items() : responsible for shopping all items by using the above methods(), which return the final output sting for the outfit set.

Limitations:

I restricted myself to shop items only in amazon.com

Automation of payment process (I couldn't accommodate time myself to work on this module).

Future Scope:

I had few ideas i.e; building the RAG pipeline on top of this, in general most of the common people follow typical same dressing style, so what I am thinking to do is create the vectors for the user inputs and store them in vector database. When other user provides the similar inputs instead going and start the crew, start search it in vector database and show results to user if it matches with more confidence else start the crew execution.

I am thinking by doing in this way, we can reduce the number of Api hit's for OpenAI model and serper search which saves budget.

Fine tuning the LLM's on more fashion outfit designing which can give more better results.