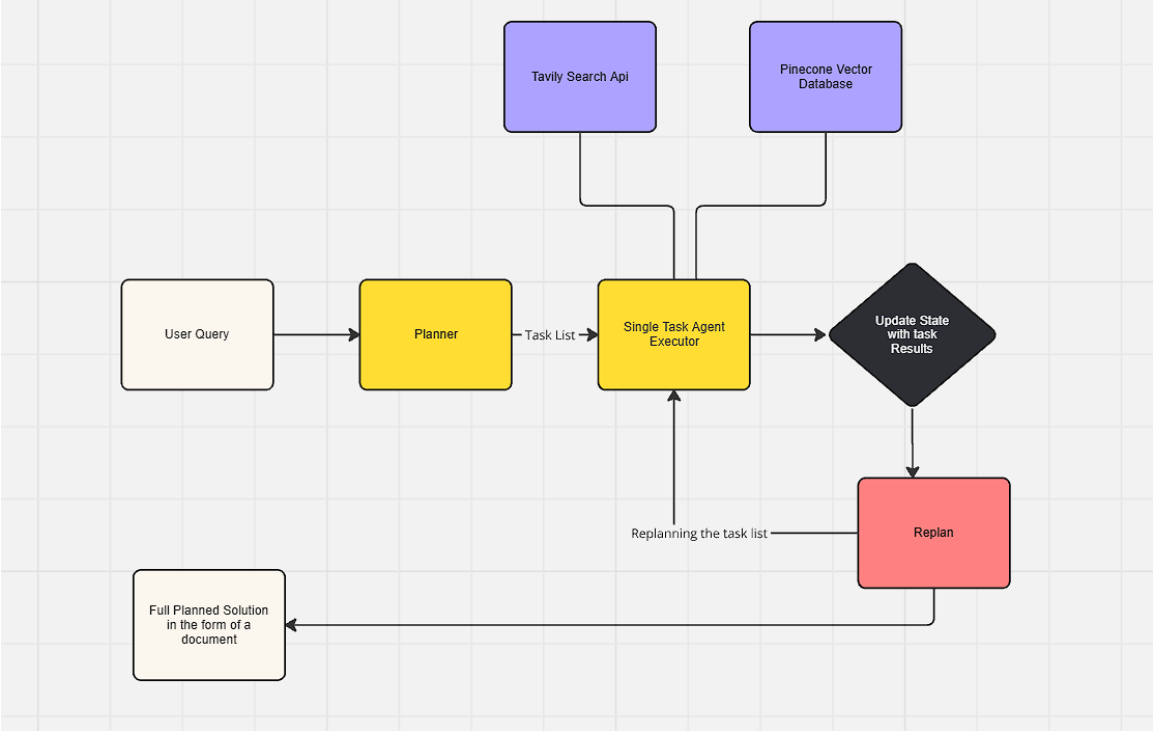
Welcome to SnowPlan

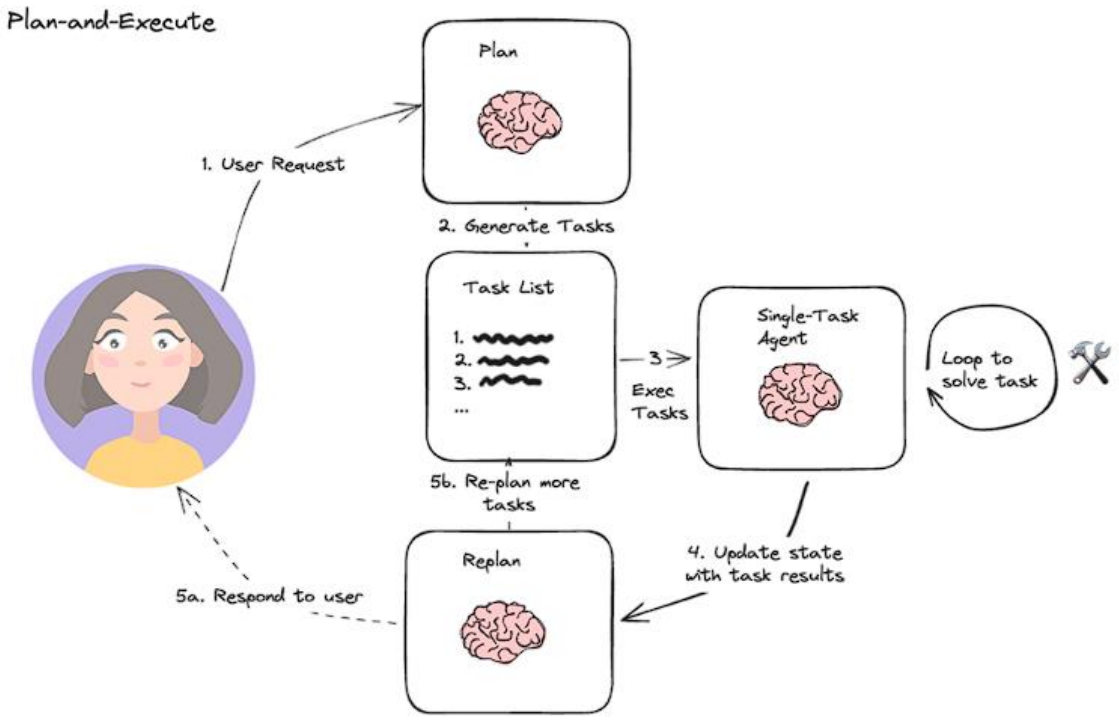
Welcome to SnowPlan, a fully automated planning agent designed to tackle complex tasks by developing and executing a multi-step plan. This agent is built to be both flexible and efficient, adjusting its plan as needed after each step, unlike traditional ReAct-style agents that think one step at a time.

Overview

The core idea behind SnowPlan is to create a detailed multi-step plan and execute it sequentially. After completing each task, the agent revisits the plan to assess progress and make necessary adjustments. This approach ensures that the agent stays on track and adapts to any new information or challenges that arise during execution.

Architecture





Advantages of the Plan-and-Execute Style Agent

Explicit Long-term Planning:

Long-term planning can be challenging for even the most advanced LLMs (Large Language Models). SnowPlan addresses this by structuring the planning process explicitly, allowing the agent to keep the end goal in sight while managing intermediate steps effectively.

Optimized Model Usage:

SnowPlan is designed to optimize the use of different models. Smaller or less powerful models can handle execution steps, while larger, more capable models are reserved for the critical task of planning. This leads to more efficient use of computational resources.

Getting Started

Prerequisites

Before you begin, ensure you have the following installed:

Python 3.7 or higher

pip (Python package manager)

API Keys:

OPENAI_API_KEY: Required for interacting with OpenAI's models.
PINECONE_API_KEY: Required for managing vector databases.
Tavily_api_key: Required for Tavily services.

Dataset Vectorization:

You need to perform vectorization of your dataset before running the agent. This can be done by executing the following script:

python vectorize_docs.py

taking a look at our pinecone vector DB - Watch Video

Installation and Setup

Clone the Repository:

Clone this repository to your local machine using:

git clone https://github.com/manjunath-ab/llm_automata.git
cd snowplan

Install Dependencies:

Navigate to the snowPlan folder and install the required dependencies:

pip install -r requirements.txt

This will install all necessary libraries and tools, the base LLM model (Gpt4o mini) is included using the OPENAI_API_KEY and the orchestration framework (Langraph).

Run the Vectorization Script:

Before running the main application, ensure that your dataset is properly vectorized by executing:

python vectorize_docs.py

Run the Application:

Once the dependencies are installed and your dataset is vectorized, you can start the SnowPlan agent through a Streamlit interface by running:

streamlit run main.py

How It Works

Planning Phase:

The agent uses Gpt4o mini to develop a comprehensive plan. This plan outlines the steps required to achieve the desired outcome.

Execution Phase:

The agent follows the plan, executing each step using the appropriate model. After completing each task, the agent revisits the plan to update it based on the new context or any changes in the environment.

Dynamic Adjustment:

Unlike traditional ReAct agents, SnowPlan continuously adapts its strategy, ensuring it remains aligned with the overall objective.

Conclusion

The LLautomata project represents a significant advancement in the integration of AI technologies for mental health support. By combining generative AI, RAG, LLMs, and the LangChain framework, the platform provides a sophisticated and empathetic environment

where users can engage in meaningful conversations that support their emotional well-being. The ability of the system to dynamically plan, execute, and replan tasks, while integrating relevant information from external knowledge sources, ensures that it remains responsive and effective in addressing the diverse needs of its users.

Future Scope

1. Enhanced Emotional Intelligence:

- Future iterations could focus on improving the emotional intelligence of the AI, enabling it to better understand and respond to nuanced emotional cues. This might involve training the LLMs on datasets specifically curated for emotional understanding and empathy.

2. Personalization:

- The platform could evolve to offer more personalized interactions by learning from individual user behaviors and preferences. By incorporating user feedback and historical interaction data, the AI could tailor its responses and plans to better suit the needs of each user.

3. Multimodal Interaction:

- Expanding the platform to support multimodal interactions, such as voice and text or even incorporating visual inputs, could enhance the user experience. This would allow users to interact with the system in a more natural and flexible manner.

4. Integration with Professional Services:

- As the platform matures, integrating it with professional mental health services could provide a more comprehensive support system. Users could be seamlessly referred to human therapists or counselors when the AI detects situations that require professional intervention.

5. Broader Knowledge Base Integration:

- Expanding the retrieval mechanisms to include a broader range of knowledge bases could make the platform more versatile. This would enable it to provide information and support on a wider array of topics, further enhancing its usefulness.

6. Scalability and Real-time Adaptation:

- Future development could focus on making the platform more scalable and capable of real-time adaptation. This would involve optimizing the AI's performance to handle large volumes of interactions simultaneously and adapting to new user inputs and external data sources as they emerge.

By exploring these avenues, the LLautomata project has the potential to become an even more powerful tool in the realm of mental health, offering scalable, personalized, and emotionally intelligent support to a wide range of users.