FinSight

Finsight can be used to analyze financial data, identify trends, and provide insights to investors, analysts, and financial institutions

(using multi-agent collaboration system)

In this project, we will use 4 agents

- Data Analyst Agent
- Trading Strategy Agent
- Execution Agent
- Risk Management Agent

Using the 4 agents, we work towards 4 tasks

- Data Analysis Task (for Data Analyst)
- Strategy Development Task (for Trading Strategy agent)
- Execution Planning Task (for Execution Agent)
- Risk Assessment Task (for Risk Management Agent)

In this project, we will use a **Hierarchial Process**, where crewAI creates a manager to delegate task to all the 4 agents

Step 01. Installing crewai and crewai_tools libraries

Note: crewai requires Python version >=3.10 and <=3.13

In [ ]:

```
import sys
```

In [ ]:

```
!pip install crewai
```

In [ ]:

```
!pip install crewai_tools
```

In [ ]:

```
# Warning control
import warnings
warnings.filterwarnings('ignore')
```

Step 02. Importing modules, LLMs, tools

NOTE: To create Agent class objects, we need to specify OPEN_AI model name and api key as environment variables

NOTE: To run crewai's SerperDevTool, we need SERPER_API_KEY as environment variables

In [ ]:

```
from crewai import Crew, Agent, Task

import os
```

```
os.environ['OPENAI_MODEL_NAME'] = "gpt-4-turbo"
os.environ['OPENAI_API_KEY'] = "Your OPEN AI Api Key"
os.environ['SERPER_API_KEY'] = "Your Server API key"
```

crewAI Tools

In [ ]:

```
from crewai_tools import ScrapeWebsiteTool, SerperDevTool

search_tool = SerperDevTool()
scrape_tool = ScrapeWebsiteTool()
```

Step 03. Creating Agents

Agent 1: Data Analyst Agent

In [ ]:

```
data_analyst_agent = Agent(
    role="Data Analyst",
    goal="Monitor and analyze market data in real-time "
         "to identify trends and predict market movements.",
    backstory="Specializing in financial markets, this agent "
              "uses statistical modeling and machine learning "
              "to provide crucial insights. With a knack for data, "
              "the Data Analyst Agent is the cornerstone for "
              "informing trading decisions.",
    allow_delegation=True,
    verbose=True,
    tools=[search_tool, scrape_tool]
)
```

Agent 2: Trading Strategy Agent

In [ ]:

```
trading_strategy_agent = Agent(
    role="Trading Strategy Developer",
    goal="Develop and test various trading strategies based "
         "on insights from the Data Analyst Agent.",
    backstory="Equipped with a deep understanding of financial "
              "markets and quantitative analysis, this agent "
              "devises and refines trading strategies. It evaluates "
              "the performance of different approaches to determine "
              "the most profitable and risk-averse options.",
    allow_delegation=True,
    tools=[search_tool, scrape_tool],
    verbose=True
)
```

Agent 3: Execution Agent

In [ ]:

```
execution_agent = Agent(
    role="Trade Advisor",
    goal="Suggest optimal trade execution strategies "
         "based on approved trading strategies.",
    backstory="This agent specializes in analyzing the timing, price, "
              "and logistical details of potential trades. By evaluating "
              "these factors, it provides well-founded suggestions for "
              "when and how trades should be executed to maximize "
              "efficiency and adherence to strategy.",
    allow_delegation=True,
    tools=[search_tool, scrape_tool],
    verbose=True
)
```

Agent 4: Risk Management Agent

In [ ]:

```
risk_management_agent = Agent(
    role="Risk Advisor",
    goal="Evaluate and provide insights on the risks "
         "associated with potential trading activities.",
    backstory="Armed with a deep understanding of risk assessment models "
              "and market dynamics, this agent scrutinizes the potential "
              "risks of proposed trades. It offers a detailed analysis of "
              "risk exposure and suggests safeguards to ensure that "
              "trading activities align with the firm’s risk tolerance.",
    allow_delegation=True,
    tools=[search_tool, scrape_tool],
    verbose=True
)
```

Step 04. Creating Tasks

In [ ]:

```
# Task for Data Analyst Agent: Analyze Market Data
data_analysis_task = Task(
    description=(
        "Continuously monitor and analyze market data for "
        "the selected stock ({stock_selection}). "
        "Use statistical modeling and machine learning to "
        "identify trends and predict market movements."
    ),
    expected_output=(
        "Insights and alerts about significant market "
        "opportunities or threats for {stock_selection}."
    ),
    agent=data_analyst_agent,
)
```

In [ ]:

```
# Task for Trading Strategy Agent: Develop Trading Strategies
strategy_development_task = Task(
    description=(
        "Develop and refine trading strategies based on "
        "the insights from the Data Analyst and "
        "user-defined risk tolerance ({risk_tolerance}). "
        "Consider trading preferences ({trading_strategy_preference})."
    ),
    expected_output=(
        "A set of potential trading strategies for {stock_selection} "
        "that align with the user's risk tolerance."
    ),
    agent=trading_strategy_agent,
)
```

In [ ]:

```
# Task for Trade Advisor Agent: Plan Trade Execution
execution_planning_task = Task(
    description=(
        "Analyze approved trading strategies to determine the "
        "best execution methods for {stock_selection}, "
        "considering current market conditions and optimal pricing."
    ),
    expected_output=(
        "Detailed execution plans suggesting how and when to "
        "execute trades for {stock_selection}."
    ),
    agent=execution_agent,
)
```

In [ ]:

```
# Task for Risk Advisor Agent: Assess Trading Risks
risk_assessment_task = Task(
    description=(
        "Evaluate the risks associated with the proposed trading "
        "strategies and execution plans for {stock_selection}. "
        "Provide a detailed analysis of potential risks "
        "and suggest mitigation strategies."
    ),
    expected_output=(
        "A comprehensive risk analysis report detailing potential "
        "risks and mitigation recommendations for {stock_selection}."
    ),
    agent=risk_management_agent,
)
```

Step 06. Creating Crew

Creating the Crew

- The `Process` class helps to delegate the workflow to the Agents (kind of like a Manager at work)
- In the example below, it will run this hierarchically.
- `manager_llm` lets you choose the "manager" LLM you want to use.

In [ ]:

```
from crewai import Crew, Process
from langchain_openai import ChatOpenAI
```

In [ ]:

```
# define the crew with agent and tasks
financial_trading_crew = Crew (
    agents=[data_analyst_agent,
            trading_strategy_agent,
            execution_agent,
            risk_management_agent],

    tasks=[data_analysis_task,
           strategy_development_task,
           execution_planning_task,
           risk_assessment_task],

    verbose=True,
    process=Process.hierarchical,
    manager_llm=ChatOpenAI(model="gpt-3.5-turbo", temperature=0.7)
)
```

Running the crew

In [ ]:

```
# specify the input for crew
financial_trading_inputs = {
    'stock_selection': 'AAPL',
    'initial_capital': '100000',
    'risk_tolerance': 'Medium',
    'trading_strategy_preference': 'Day Trading',
    'news_impact_consideration': True
}
```

```
### this execution will take some time to run
```

In [ ]:

```
result = financial_trading_crew.kickoff(inputs=financial_trading_inputs)
```

- Display the final result as Markdown.

In [ ]:

```
from IPython.display import Markdown
Markdown(result)
```

Out[ ]:

The comprehensive risk analysis report and mitigation recommendations for day trading AAPL are as follows:

Risk Analysis:

- **Volume and Liquidity**: Fluctuating trading volumes and liquidity patterns in AAPL indicate potential increased trade costs and price slippage risks.
- **Volatility**: Market shifts influenced by broader economic indicators and policy decisions may lead to sudden price changes in AAPL.
- **Price Segment**: High performance attracting speculative trading in AAPL can result in increased price volatility.
- **Macro Factors**: International trade adjustments and political impacts can indirectly affect market sentiment and AAPL's stock performance.
- **News and Events**: Any news related to AAPL or its sector can cause sharp movements in stock price.

Mitigation Strategies:

1. **Two-Day High/Low Strategy**: Buying breakouts from a two-day high or selling breakdowns from a two-day low with a 1:2 risk/reward ratio and strict stop-loss measures.
2. **Trading Ranges During Low Volatility**: Utilizing oscillators to identify overbought and oversold levels during low volatility periods.
3. **GAP-FILL Strategy**: Trading the gaps between the previous day's close and the current day's open, aiming to fill these gaps with strategic entry and exit points.

These strategies emphasize preparation, discipline, and risk management to navigate the risks associated with day trading AAPL effectively. Traders should stay informed about market conditions, economic indicators, and company-specific news to make informed decisions and optimize their trading experience.

In [ ]:

```

```