Multi-Agent Financial Analysis System

Group Name: AAI-520 Group 14

Team Members: Jagadeesh Kumar Sellappan, Saurav Kumar Subudhi.

GitHub Repository: https://github.com/jagadeesh-usd/AI-Financial-Analysis-Agent.git

Table of Contents

1. Project Overview
2. System Architecture Overview
3. Agent Design and Workflows
    - A. The Planner–Executor Workflow
    - B. Specialist Agent Delegation (Prompt Chaining)
    - C. Evaluator–Optimizer Workflow (Critique–Refinement Loop)
4. Agent Functions and Capabilities
5. Execute Financial Analysis
6. Evaluation and Iteration
7. Streamlit Integration and Demonstration Video
8. Conclusion and Future Work
9. AI Assistance Disclosure

Project Overview

This notebook implements a sophisticated multi-agent financial analysis system using LangChain and LangGraph. The system is designed to automate the process of investment research by coordinating multiple specialized AI agents to perform complex tasks end-to-end.

The core architecture follows a **Planner-Executor** model, augmented with a **Critique-Refinement** loop to ensure the quality and objectivity of the final analysis. This agentic approach moves beyond simple, scripted pipelines by enabling agents to reason, plan, delegate, and self-correct, mimicking the workflow of a human financial research team.

In [33]:
```
# Install necessary packages
# !pip install -q python-dotenv langchain-openai yfinance fredapi edgartools pytrends langgraph
```

Import libraries and load environment variables

In [34]:
```
import json
import os
import pprint
import re
import warnings

import matplotlib.pyplot as plt
import pandas as pd
import requests

from bs4 import BeautifulSoup
from datetime import datetime, timedelta
from dotenv import load_dotenv
from edgar import Company, set_identity
from fredapi import Fred
from langchain.agents import AgentExecutor, create_openai_functions_agent
from langchain_core.prompts import ChatPromptTemplate, MessagesPlaceholder
from langchain_core.tools import tool
from langchain.tools import Tool
from langchain_openai import ChatOpenAI
from pytrends.request import TrendReq
import yfinance as yf
from langgraph.graph import StateGraph, END
from typing import TypedDict, Annotated, List
import operator
```

Load API keys from your .env file

In [35]:
```
load_dotenv()

# Check for the OpenAI API key
if not os.environ.get("OPENAI_API_KEY"):
    raise ValueError(
        "OPENAI_API_KEY not found in environment variables. "
        "Please create a .env file."
    )
```

Helper functions

In [36]:
```
def plot_price_trend(ticker: str):
    """
    Plots the stock's closing price and 50/200-day moving averages for the last
    year. Saves the plot as a PNG for inclusion in reports and displays it in
    the notebook.
    """
    try:
        stock = yf.Ticker(ticker)
        hist = stock.history(period="1y")

        if hist.empty:
            return {"error": "Could not retrieve price history for plotting."}

        plt.figure(figsize=(10, 6))
        plt.plot(hist.index, hist['Close'], label='Closing Price',
                 color='#1f77b4', linewidth=2)
        plt.plot(hist.index, hist['Close'].rolling(window=50).mean(),
                 label='50-Day MA', color='#ff7f0e', linestyle='--')
        plt.plot(hist.index, hist['Close'].rolling(window=200).mean(),
                 label='200-Day MA', color='#2ca02c', linestyle='--')

        plt.title(f"{ticker} Stock Price and Moving Averages (1 Year)",
                  fontsize=14, pad=15)
        plt.xlabel("Date", fontsize=12)
        plt.ylabel("Price (USD)", fontsize=12)
        plt.legend(fontsize=10)
        plt.grid(True, linestyle='--', alpha=0.7)
        plt.tight_layout()

        plt.show()
        return {"message": f"Price trend plot saved as {ticker}"}

    except Exception as e:
        return {"error": f"Failed to plot price trend: {e}"}
```

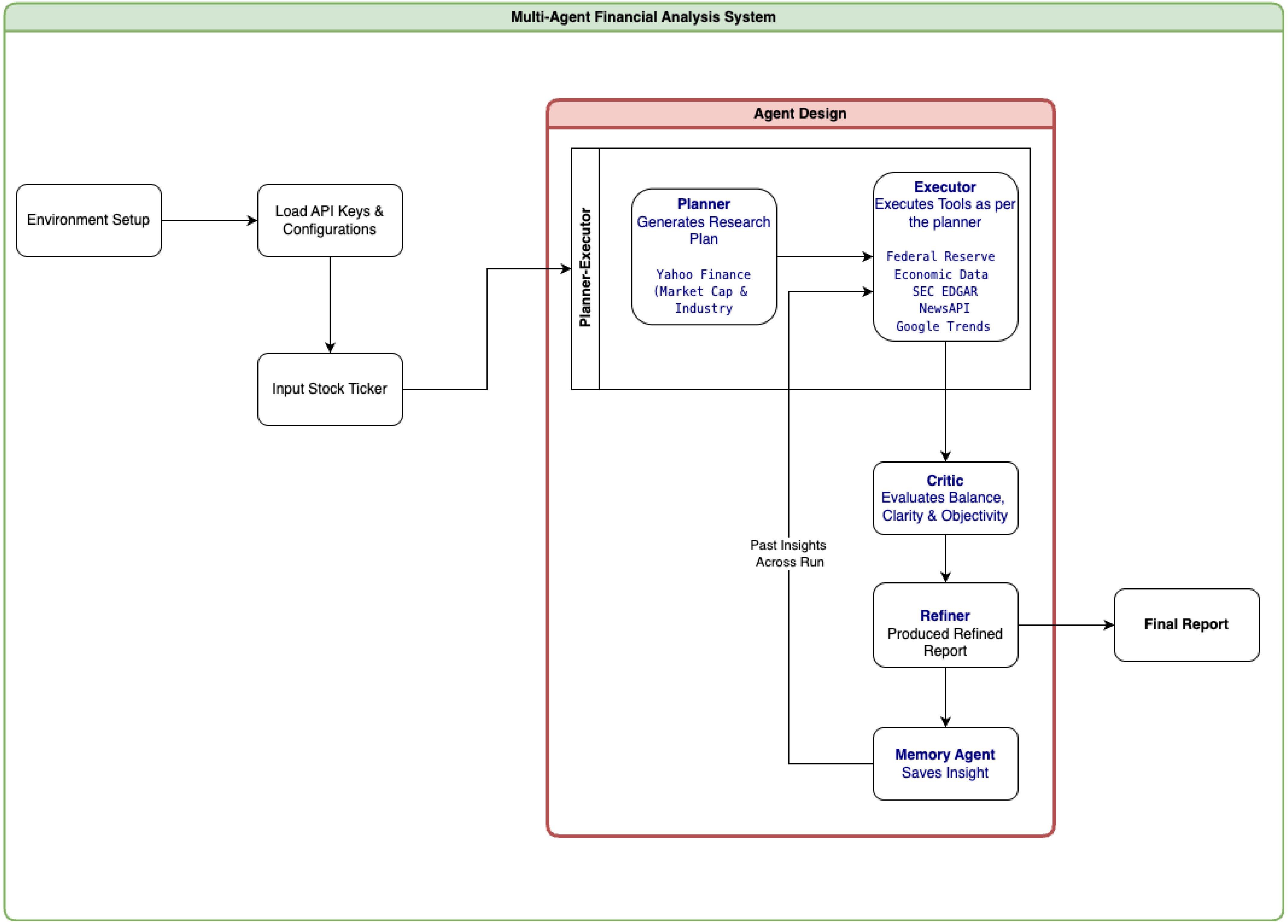
In [37]:

```
import re

def clean_filing_content(text):
    """
    Clean the extracted filing content by removing XBRL/inline tags and
    normalizing whitespace.
    """
    # Remove all XBRL/inline tags (e.g., xbrli:shares, iso4217:USD, us-gaap:*, etc.)
    text = re.sub(r'\b(xbrli|us-gaap|dei|srt|aapl|intc):[^\s>]+', '', text)

    # Remove standalone tags like "P1Y", "c-1", "f-46", etc.
    text = re.sub(r'\b([A-Za-z]+-[0-9]+|[A-Za-z]+[0-9]+|[A-Z]{2,}[A-Za-z]*\d*)\b',
                   '', text)

    # Remove dates in the format "YYYY-MM-DD" if they are standalone
    text = re.sub(r'(?<!\d)\d{4}-\d{2}-\d{2}(?!\d)', '', text)

    # Remove numbers like "0000050863" if they are standalone
    text = re.sub(r'(?<!\d)\d{8,}(?!\d)', '', text)

    # Remove extra whitespace and newlines
    text = re.sub(r'\s+', ' ', text).strip()

    return text
```

System Architecture Overview

The diagram below illustrates the end-to-end **Multi-Agent Financial Analysis System** implemented in this project. This architecture begins with the environment setup and API configuration phase. The **Planner Agent** defines a research plan by selecting relevant tools and datasets based on the stock ticker and market characteristics. The **Executor Agent** then executes these tools to gather market, macroeconomic, and regulatory information.

Within the agent design, the **Critic Agent** evaluates the generated insights for clarity, balance, and objectivity, while the **Refiner Agent** improves the report quality based on critique feedback. The **Memory Agent** stores key insights from previous runs, allowing the system to improve continuously across sessions. Finally,it generates the final, data-driven investment report.



Agent Design and Workflows

This system is built on a foundation of several key agentic workflow patterns that enable complex, autonomous behavior. The primary architectural choice is the **Planner-Executor** pattern, which separates the "thinking" from the "doing" to improve reliability and efficiency.

1. The Planner-Executor Workflow

The workflow begins with a clear separation of duties:

1. **Planning Phase (`planner_node`):** A specialized **Planner Agent** is invoked first. Its sole responsibility is to create a strategic research plan. It performs a single, low-cost action—calling the `get_company_info` tool—to determine the company's market cap and sector. Based on this profile, it dynamically selects the appropriate set of tools for a deeper analysis and outputs its reasoning along with a structured plan (a JSON array of tool names).
2. **Execution Phase (`executor_node`):** The plan is then passed to an **Executor Agent**. This agent's job is to mindlessly and efficiently execute the provided plan. Its internal protocol mandates that it **always** calls `read_notes_from_memory` first to gather historical context. Following that, it sequentially invokes each tool from the planner's list, gathering all the necessary data points before synthesizing an `initial_analysis` .

2. Specialist Agent Delegation (Prompt Chaining)

Within the Execution phase, the `executor_agent` demonstrates another key pattern: delegation to a specialist. When it needs to analyze news, it doesn't process the raw headlines itself. Instead, it calls the **`Financial_News_Analyst` tool**, which is a self-contained agent. This specialist agent performs an entire internal workflow (Ingest → Classify → Summarize) in a single call, returning a concise, high-level summary. This is an example of **Prompt Chaining**.

3. Evaluator–Optimizer Workflow (Critique–Refinement Loop)

A key feature of this agentic system is its built-in mechanism for evaluation and iterative improvement, which is handled by the **Critique-Refinement** loop. This pattern ensures that the initial analysis does not go unchecked and is refined based on objective feedback before being presented as the final report.

The process follows a clear three-step pattern:

1. **Generate (`executor_node`):** The **Executor Agent** first produces an `initial_analysis` based on the data it has gathered. This serves as the first draft.
2. **Evaluate (`critic_node`):** The `initial_analysis` is then passed to the **Critic Agent**. This agent acts as an automated quality assurance step. It evaluates the draft against important criteria (e.g., Is it balanced? Does it present both risks and opportunities?) and produces a structured `critique` .
3. **Refine (`refiner_node`):** Finally, both the `initial_analysis` and the `critique` are passed to the **Refiner Agent**. This agent's sole purpose is to rewrite and improve the initial draft by explicitly addressing the feedback from the critic. The output of this node is the polished `refined_analysis` , which becomes the final report.

This **Generate → Evaluate → Refine** cycle is a powerful form of self-correction that significantly enhances the quality, objectivity, and reliability of the system's output.

Agentic Tools Definition

In [38]:
```
import os
import warnings
from fredapi import Fred

try:
    fred_api_key = os.environ.get("FRED_API_KEY")
    fred = Fred(api_key=fred_api_key)
except Exception:
    warnings.warn(
        "FRED API key not found in Streamlit secrets. The economic data "
        "tool will be disabled."
    )
    fred = None

# --- Memory Component ---
MEMORY_FILE = "memory.json"
```

In [39]:
```
# Define memory tools for learning across runs
@tool
def read_notes_from_memory(ticker: str) -> list[str]:
    """
    Reads past analysis notes for a given stock ticker to provide context
    for a new analysis. Use this tool FIRST before any other tool to get
    historical context.
    """
    if not os.path.exists(MEMORY_FILE):
        return ["No past notes found for this ticker."]

    with open(MEMORY_FILE, 'r') as f:
        try:
            data = json.load(f)
            return data.get(ticker, ["No past notes found for this ticker."])
        except json.JSONDecodeError:
            return ["Memory file is empty or corrupted."]

@tool
def save_note_to_memory(ticker: str, note: str) -> str:
    """
    Saves a single, concise key takeaway from the latest analysis to memory
    for future reference. Use this tool LAST after the analysis is complete.
    The note should be a single sentence.
    """
    if not isinstance(note, str) or not note.strip():
        return "Error: Note must be a non-empty string."

    if (os.path.exists(MEMORY_FILE) and
            os.path.getsize(MEMORY_FILE) > 0):
        with open(MEMORY_FILE, 'r') as f:
            try:
                data = json.load(f)
            except json.JSONDecodeError:
                data = {}
    else:
        data = {}

    if ticker not in data:
        data[ticker] = []

    note_with_date = (f"[**{datetime.now().strftime('%Y-%m-%d')}**] - "
                      f"{note.strip()}")
    data[ticker].append(note_with_date)
    data[ticker] = data[ticker][-3:]

    with open(MEMORY_FILE, 'w') as f:
        json.dump(data, f, indent=4)

    return f"Successfully saved note for {ticker}."
```

In [40]:

```
@tool
def get_company_info(ticker: str) -> dict:
    """Retrieves general information and key metrics for a given stock ticker."""

    stock = yf.Ticker(ticker)
    info = stock.info
    market_cap = info.get('marketCap')

    return {
        "longName": info.get('longName', 'N/A'),
        "marketCap": (f"${market_cap:,}"
                       if isinstance(market_cap, (int, float)) else 'N/A'),
        "sector": info.get('sector', 'N/A'),
    }
```

In [41]:

```
@tool
def get_stock_news(ticker: str) -> list[str]:
    """Fetches the latest news headlines for a given stock ticker."""

    try:
        stock = yf.Ticker(ticker)
        news = stock.news

        if not news:
            return ["No recent news found."]

        return [
            article['content']['title']
            for article in news[:10]
            if 'content' in article and
            'title' in article.get('content', {})
        ]

    except Exception as e:
        return [f"An error occurred: {e}"]
```

In [42]:

```
@tool
def search_specific_news(ticker: str, keywords: list[str]) -> list[str]:
    """
    Searches for news articles about a company from the last 30 days that
    contain specific keywords. Use this to find targeted information like
    'partnerships', 'acquisitions', 'speculation', 'product launch', or
    'regulatory approval'. The `keywords` argument should be a list of
    search terms.
    """

    try:
        api_key = os.environ.get("NEWS_API_KEY")
    except Exception:
        return ["Error: NewsAPI key not found"]

    # Calculate date 30 days ago
    date_from = (datetime.now() - timedelta(days=30)).strftime('%Y-%m-%d')

    # Format keywords for the API query
    query_keywords = " OR ".join(keywords)
    query = f'"{ticker}" AND ({query_keywords})'

    url = (f'https://newsapi.org/v2/everything?'
           f'q={query}&'
           f'from={date_from}&'
           f'sortBy=relevancy&'
           f'apiKey={api_key}')

    try:
        response = requests.get(url)
        response.raise_for_status()
        articles = response.json().get('articles', [])

        if not articles:
            return [
                f"No specific news found for keywords '{', '.join(keywords)}' "
                f"in the last 30 days."
            ]

        # Return a list of the top 5 most relevant headlines
        return [article['title'] for article in articles[:5]]

    except Exception as e:
        return [f"An error occurred while fetching specific news: {e}"]
```

In [43]:

```
@tool
def get_price_summary(ticker: str) -> dict:
    """
    Retrieves a comprehensive price summary and key technical indicators
    for a stock. Provides the latest price, 52-week range, 50-day and
    200-day moving averages, and the 14-day Relative Strength Index (RSI)
    to understand the stock's trend and momentum.
    """

    try:
        stock = yf.Ticker(ticker)
        # Fetch 1 year of data to calculate all indicators
        hist = stock.history(period="1y")
        if hist.empty:
            return {"error": "Could not retrieve price history."}

        # --- Key Price Points ---
        latest_close = hist['Close'].iloc[-1]
        fifty_two_week_high = hist['High'].max()
        fifty_two_week_low = hist['Low'].min()

        # --- Trend Indicators (Moving Averages) ---
        ma_50 = hist['Close'].rolling(window=50).mean().iloc[-1]
        ma_200 = hist['Close'].rolling(window=200).mean().iloc[-1]

        # Determine trend based on MAs
        if latest_close > ma_50 and ma_50 > ma_200:
            trend = "Strong Uptrend"
        elif latest_close > ma_50 and latest_close > ma_200:
            trend = "Uptrend"
        elif latest_close < ma_50 and ma_50 < ma_200:
            trend = "Strong Downtrend"
        elif latest_close < ma_50 and latest_close < ma_200:
            trend = "Downtrend"
        else:
            trend = "Sideways/Consolidating"

        # --- Momentum Indicator (RSI) ---
        delta = hist['Close'].diff()
        gain = (delta.where(delta > 0, 0)).rolling(window=14).mean()
        loss = (-delta.where(delta < 0, 0)).rolling(window=14).mean()
        rs = gain / loss
        rsi = 100 - (100 / (1 + rs))
        latest_rsi = rsi.iloc[-1]

        # Determine overbought/oversold condition
        if latest_rsi > 70:
            rsi_condition = "Overbought"
        elif latest_rsi < 30:
            rsi_condition = "Oversold"
        else:
            rsi_condition = "Neutral"

        return {
            "latest_price": f"${latest_close:.2f}",
            "52_week_high": f"${fifty_two_week_high:.2f}",
            "52_week_low": f"${fifty_two_week_low:.2f}",
            "trend_analysis": {
                "50_day_ma": f"${ma_50:.2f}",
                "200_day_ma": f"${ma_200:.2f}",
                "trend_conclusion": trend
            },
            "momentum_analysis": {
                "rsi_14_day": f"{latest_rsi:.2f}",
                "condition": rsi_condition
            }
        }
    except Exception as e:
        return {"error": f"Failed to calculate price summary: {e}"}
```

In [44]:

```
@tool
def get_financial_ratios(ticker: str) -> dict:
    """
    Calculates and returns key financial ratios for a given stock ticker.
    This provides insights into the company's valuation, profitability,
    and financial health.
    """

    try:
        stock = yf.Ticker(ticker)
        info = stock.info

        ratios = {
            "trailing_pe": info.get("trailingPE", "N/A"),
            "forward_pe": info.get("forwardPE", "N/A"),
            "price_to_book": info.get("priceToBook", "N/A"),
            "price_to_sales": info.get(
                "priceToSalesTrailing12Months", "N/A"
            ),
            "debt_to_equity": info.get("debtToEquity", "N/A"),
            "return_on_equity": info.get("returnOnEquity", "N/A"),
            "profit_margins": info.get("profitMargins", "N/A"),
        }

        return ratios
    except Exception as e:
        return {"error": f"Could not retrieve financial ratios: {e}"}
```

In [45]:

```
@tool
def get_analyst_ratings(ticker: str) -> dict:
    """
    Fetches the latest analyst ratings and price targets for a stock.
    This helps understand the consensus view from market professionals.
    """

    try:
        stock = yf.Ticker(ticker)
        # Use recommendations_summary which is often more consistent
        recommendations = stock.recommendations_summary
        if recommendations.empty:
            return {"message": "No analyst ratings found for this period."}

        # The data is often in a single row, so we access it directly.
        latest_summary = recommendations.iloc[0]

        # Use .get() to safely access keys that might be missing
        strong_buy = latest_summary.get('strongBuy', 0)
        buy = latest_summary.get('buy', 0)
        hold = latest_summary.get('hold', 0)
        sell = latest_summary.get('sell', 0)
        strong_sell = latest_summary.get('strongSell', 0)

        summary = {
            "period": latest_summary.get('period', 'N/A'),
            "buy_ratings": int(strong_buy + buy),
            "hold_ratings": int(hold),
            "sell_ratings": int(sell + strong_sell),
        }

        # Get the latest specific recommendations if available
        rec_details = stock.recommendations
        if not rec_details.empty:
            latest_ratings = rec_details.tail(5)
            firm_col = 'Firm' if 'Firm' in latest_ratings.columns else 'firm'
            grade_col = 'To Grade' if 'To Grade' in latest_ratings.columns else 'toGrade'
            if (firm_col in latest_ratings.columns and
                    grade_col in latest_ratings.columns):
                summary["latest_recommendations"] = (
                    latest_ratings[[firm_col, grade_col]].to_dict('records')
                )
            else:
                summary["latest_recommendations"] = []
        else:
            summary["latest_recommendations"] = []

        return summary
    except Exception as e:
        return {"error": f"Could not retrieve analyst ratings: {e}"}
```

In [46]:

```
@tool
def get_google_trends(ticker: str, timeframe: str = 'today 3-m') -> dict:
    """
    Fetches Google Trends data for a company by first looking up its name
    from the ticker. This is useful for gauging public interest in a
    company or its products.
    """
    try:
        # 1. Look up the company's name from the ticker.
        stock = yf.Ticker(ticker)
        # Use the longName, but fall back to the ticker if it's not available.
        keyword = stock.info.get('longName', ticker)

        pytrends = TrendReq(hl='en-US', tz=360)
        pytrends.build_payload(
            [keyword], cat=0, timeframe=timeframe, geo='', gprop=''
        )
        interest_over_time_df = pytrends.interest_over_time()
        if (interest_over_time_df.empty or
                keyword not in interest_over_time_df.columns):
            return {"message": f"No Google Trends data found for '{keyword}'."}

        avg_interest = interest_over_time_df[keyword].mean()
        peak_interest_date = (
            interest_over_time_df[keyword].idxmax().strftime('%Y-%m-%d')
        )

        return {
            "keyword_searched": keyword,
            "average_interest_score": round(avg_interest, 2),
            "peak_interest_date": peak_interest_date,
            "comment": (
                f"The average interest score for '{keyword}' is "
                f"{round(avg_interest, 2)} out of 100 over the last 3 months."
            ),
        }
    except Exception as e:
        return {"error": f"Could not retrieve Google Trends data: {e}"}
```

In [47]:

```
@tool
def get_economic_data(series_id: str = 'GDP') -> dict:
    """
    Fetches the latest data for a given economic series from FRED. Use
    this to understand the broader economic context. For example, use
    series_id 'GDP' for GDP data.
    """
    if fred is None:
        return {
            "error": "FRED API key not configured. Cannot fetch economic data."
        }

    try:
        data = fred.get_series(series_id)
        latest_value = data.iloc[-1]
        latest_date = data.index[-1].strftime('%Y-%m-%d')
        return {
            "series": series_id,
            "latest_value": f"{latest_value:,.2f}",
            "latest_date": latest_date
        }
    except Exception as e:
        return {
            "error": f"Failed to fetch data for series {series_id}: {e}"
        }
```

In [48]:

```
@tool
def get_latest_filings(ticker: str, top_n: int = 1) -> list[dict]:
    """
    Fetch latest 10-K / 10-Q filings for the ticker.
    Returns a list of dicts with metadata and cleaned content.
    """
    try:
        # Identify to SEC
        set_identity("jagadeesch1981@gmail.com")
        company = Company(ticker.upper())
        filings = company.get_filings().filter(form=["10-K", "10-Q"])
        filings_df = filings.to_pandas()
        results = []
        for _, filing_row in filings_df.head(top_n).iterrows():
            filing_dict = filing_row.to_dict()
            # Prefer the filing's cik; fallback to company.cik if not present
            cik_raw = (
                filing_row.get("cik")
                or getattr(company, "cik", None)
                or filing_row.get("company_info", {}).get("cik")
            )
            if not cik_raw:
                results.append({
                    "error": "No CIK available for this filing.",
                    "filing_meta": filing_dict
                })
                continue
            cik = str(cik_raw).replace("-", "").zfill(10)  # zero-pad to 10 digits
            accession = str(
                filing_row.get("accession_number")
                or filing_row.get("accessionNo")
                or ""
            ).replace("-", "")
            primary_document = (
                filing_row.get("primaryDocument")
                or filing_row.get("primary_doc")
                or ""
            )
            # Construct canonical document URL
            document_url = None
            if accession and primary_document:
                document_url = (
                    f"https://www.sec.gov/Archives/edgar/data/{cik}/"
                    f"{accession}/{primary_document}"
                )
            # Fallback URLs
            fallback_urls = []
            if filing_row.get("filing_href"):
                fallback_urls.append(filing_row.get("filing_href"))
            if filing_row.get("linkToFilingDetails"):
                fallback_urls.append(filing_row.get("linkToFilingDetails"))
            if filing_row.get("filing_url"):
                fallback_urls.append(filing_row.get("filing_url"))
            # Try primary constructed URL first, then fallbacks
            tried_urls = []
            headers = {"User-Agent": "jagadeesch1981@gmail.com"}
            response_text = None
            for url in ([document_url] if document_url else []) + fallback_urls:
                if not url:
                    continue
                tried_urls.append(url)
                try:
                    resp = requests.get(url, headers=headers, timeout=15)
                    resp.raise_for_status()
                    response_text = resp.text
                    used_url = url
                    break
                except requests.exceptions.HTTPError:
                    continue
                except requests.exceptions.RequestException as e:
                    results.append({
                        "error": f"Network error fetching filing: {e}",
                        "filing_meta": filing_dict,
                        "tried_urls": tried_urls,
                    })
                    response_text = None
                    break
            if not response_text:
                results.append({
                    "error": "Could not fetch filing document (404 or unavailable).",
                    "filing_meta": filing_dict,
                    "tried_urls": tried_urls,
                })
                continue
            # Clean the text
            soup = BeautifulSoup(response_text, "html.parser")
            # Remove all XBRL and XML tags
            for tag in soup.find_all(True, {"name": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"xlink": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"xbrli": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"us-gaap": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"dei": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"srt": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"aapl": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"intc": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"ixt": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"ixt-sec": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"xbrldi": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"xsi": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"iso4217": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"ecd": True}):
                tag.decompose()
            for tag in soup.find_all(True, {"xml": True}):
                tag.decompose()
            # Remove all <head> tags
            if soup.head:
                soup.head.decompose()
            # Get the cleaned text
            cleaned_text = soup.get_text(separator=" ", strip=True)
            # Further clean the text using regex
            cleaned_text = clean_filing_content(cleaned_text)
            results.append({
                "form_type": filing_dict.get("form"),
                "filed_at": str(
                    filing_dict.get("filing_date")
                    or filing_dict.get("filingDate")
                ),
                "accession_number": accession,
                "url": used_url,
                "content": cleaned_text,
                "meta": filing_dict,
            })
        if not results:
            return [{"error": f"No recent 10-K or 10-Q filings found for {ticker}."}]
        return results
    except Exception as e:
        return [{"error": f"Failed to fetch SEC filings for ticker {ticker}: {e}"}]
```

Agent Creation Factory

In [49]:

```
def create_agent(llm: ChatOpenAI, tools: list, system_prompt: str):
    """
    Create an agent for processing inputs using the specified language
    model and tools.

    Args:
        llm (ChatOpenAI): The language model to be used by the agent.
        tools (list): A list of tools that the agent can utilize.
        system_prompt (str): A prompt that guides the agent's responses.

    Returns:
        AgentExecutor: An executor instance that manages the agent's
                       operations and interactions with the provided tools.
    """
    prompt = ChatPromptTemplate.from_messages([
        ("system", system_prompt),
        ("human", "{input}"),
        MessagesPlaceholder(variable_name="agent_scratchpad"),
    ])

    agent = create_openai_functions_agent(llm, tools, prompt)

    return AgentExecutor(
        agent=agent,
        tools=tools,
        verbose=False,
        return_intermediate_steps=True
    )
```

Agent Functions and Capabilities

The agents in this system are designed with specific functions that align with the core requirements of an agentic AI.

1. Planning (`planner_agent`)

The system explicitly demonstrates planning capabilities. The **Planner Agent**'s primary function is to create a dynamic, context-aware research plan. It reasons about the company's profile (Penny Stock, Mid-Cap, etc.) and outputs a precise list of tools for the next agent to execute.

2. Dynamic Tool Use (`executor_agent`)

The **Executor Agent** showcases dynamic tool use. It is not hardcoded to run a fixed sequence of tools. Instead, it receives a plan as input and dynamically invokes the tools specified in that plan, allowing its behavior to adapt to the specific research task at hand.

3. Self-Reflection (`critic_agent`)

Self-reflection is implemented through the **Critic Agent**. After the `executor_agent` generates its initial analysis, the `critic_agent` is invoked to assess the quality of that output. It evaluates the analysis against predefined criteria: **Balance**, **Clarity**, and **Objectivity**, providing structured feedback for improvement.

4. Learning Across Runs (Memory Tools)

The system has a persistent memory implemented via the `read_notes_from_memory` and `save_note_to_memory` tools.

- At the **start** of every run, the executor agent is mandated to call `read_notes_from_memory` , ensuring it has historical context.
- At the **end** of the workflow, the `save_memory_node` generates a concise takeaway from the final report and saves it to a JSON file, allowing the system to learn and improve its context for future analyses of the same stock.

Agent Definitions

In [50]:

```python
def get_news_analyst_agent(llm: ChatOpenAI):
    """
    Defines a specialist agent that performs a multi-step news analysis chain.
    """
    # This agent only needs one tool: the ability to get raw news headlines.
    news_analyst_tools = [get_stock_news]

    # This detailed prompt IS the "Prompt Chain". It instructs the agent on
    # the exact sequence of steps.
    news_analyst_prompt = (
        "You are an expert financial news analyst. Your goal is to produce "
        "a concise summary of the latest news sentiment.\n\n"
        "Follow this exact multi-step process:\n"
        "1. **Ingest:** First, use the `get_stock_news` tool to fetch the "
        "latest raw news headlines for the given stock ticker.\n"
        "2. **Classify & Analyze:** Internally, classify the sentiment of each "
        "headline (Positive, Negative, Neutral) and identify the key topics "
        "being discussed (e.g., earnings, partnerships, market trends).\n"
        "3. **Summarize:** Finally, synthesize your findings into a concise, "
        "2-3 sentence summary that captures the overall sentiment and the most "
        "important news points.\n\n"
        "Your final output MUST be only the summary paragraph. Do not output "
        "the list of headlines or your classification details."
    )

    return create_agent(llm, news_analyst_tools, news_analyst_prompt)
```

In [51]:

```python
def get_planner_agent(llm: ChatOpenAI):
    """
    An agent that creates a research plan based on company profile.
    It only decides WHICH tools to use; it does not execute them.
    """
    # This agent only needs one tool to do its job: get_company_info
    planner_tools = [get_company_info]

    planner_system_prompt = (
        "You are an expert financial planning agent. Your goal is to create "
        "a step-by-step research plan for a given company.\n\n"
        "1. First, use the `get_company_info` tool to determine the company's "
        "market cap and sector.\n"
        "2. Based on this information, select the appropriate tools from the "
        "following list that an executor agent should run:\n"
        "    --- MARKET CAP BASED PLANS ---\n\n"
        "    - **Penny Stock (<$50M):** Your scope is strictly limited.\n"
        "      **Required Tools:** `get_price_summary`, `Financial_News_Analyst`.\n"
        "      **Forbidden Tools:** Do not use `get_latest_filings` or "
        "`get_economic_data`.\n\n"
        "    - **Small-Cap ($50M - $2B):** These are growing companies requiring "
        "a balanced analysis.\n"
        "      **Required Tools:** `get_price_summary`, `Financial_News_Analyst`, "
        "`get_latest_filings`, `get_financial_ratios`.\n\n"
        "    - **Mid-Cap ($2B - $10B):** These are established companies. The "
        "analysis must focus on sustainable growth and profitability.\n"
        "      **Required Tools:** `get_price_summary`, `Financial_News_Analyst`, "
        "`get_financial_ratios`, `get_latest_filings`, `get_economic_data`.\n\n"
        "    - **Large-Cap (>$10B):** A comprehensive analysis is required. "
        "You must use all available tools for a complete picture.\n"
        "      **Required Tools:** `get_price_summary`, `Financial_News_Analyst`, "
        "`get_financial_ratios`, `get_latest_filings`, `get_economic_data`, "
        "`get_analyst_ratings`, and `get_google_trends` (if applicable to the "
        "sector).\n\n"
        "    --- SECTOR-BASED FOCUS ---\n\n"
        "    After selecting your tools, use the company's sector to refine your "
        "focus:\n"
        "    - **Technology/Healthcare:** Focus on innovation and competition. "
        "**Use `search_specific_news` with keywords like 'partnership', "
        "'acquisition', and 'product launch'** to find key strategic moves.\n"
        "    - **Financials/Industrials:** Focus on balance sheet health, debt, "
        "and economic indicators like interest rates.\n"
        "    - **Consumer Cyclical/Defensive:** Focus on consumer sentiment and "
        "supply chain news.\n"
        "    - **Utilities/Energy/Real Estate:** Focus on debt, dividends, and "
        "interest rate sensitivity.\n\n"
        "Your final output MUST be a single, valid JSON object with two keys:\n"
        "1. `reasoning`: A text containing your reason to choose all these tools "
        "and what to achieve in a couple of lines.\n"
        "2. `plan`: A JSON array of strings, where each string is the exact name "
        "of a tool to be executed.\n\n"
    )

    return create_agent(llm, planner_tools, planner_system_prompt)
```

In [52]:

```
def get_executor_agent(llm: ChatOpenAI):
    """
    An agent that executes a given research plan.
    It takes a plan and calls the tools in that plan.
    """
    # The specialist news agent needs to be created here for the executor's tool list
    news_analyst_agent = get_news_analyst_agent(llm)

    def run_news_agent(ticker: str):
        return news_analyst_agent.invoke({"input": ticker})

    news_analysis_tool = Tool(
        name="Financial_News_Analyst",
        func=run_news_agent,
        description=(
            "Provides a comprehensive summary of the latest news sentiment for "
            "a given stock ticker."
        )
    )
    # This agent needs all the tools available for execution.
    executor_tools = [
        read_notes_from_memory,
        get_price_summary,
        news_analysis_tool,
        get_economic_data,
        get_latest_filings,
        get_financial_ratios,
        get_analyst_ratings,
        get_google_trends,
        search_specific_news
    ]

    executor_system_prompt = (
        "You are an expert financial research executor. Your goal is to execute a "
        "given plan precisely.\n\n"
        "**Execution Protocol:**\n"
        "1. **MANDATORY FIRST STEP:** Always begin by using the "
        "`read_notes_from_memory` tool to gather historical context for the "
        "ticker. This is a non-negotiable first action.\n"
        "2. **Execute the Provided Plan:** After consulting memory, execute the "
        "tools listed in the user-provided plan, in the order they are given.\n\n"
        "Do not deviate from the plan or add extra tools. Your job is to execute, "
        "not to plan.\n"
        "3. **Synthesize Final Analysis:** After executing your chosen plan, "
        "combine all gathered information into a single, detailed analysis. This "
        "paragraph MUST be your final output.\n\n"
        "**Formatting Instructions:** Do not output markdown. Ensure your final "
        "output is a well-formatted, readable paragraph."
    )

    return create_agent(llm, executor_tools, executor_system_prompt)
```

In [53]:

```
def get_critic_agent(llm: ChatOpenAI):
    critic_prompt = ChatPromptTemplate.from_template(
        "You are a meticulous financial 'Critic' agent. Evaluate an analysis "
        "based on:\n"
        "1. **Balance:** Does it present both risks and opportunities?\n"
        "2. **Clarity:** Is the language clear and concise?\n"
        "3. **Objectivity:** Is the analysis data-driven?\n\n"
        "Provide short, bulleted feedback. If excellent, state 'No major changes "
        "needed'.\n\n"
        "Initial Analysis to Critique:\n{initial_analysis}"
    )

    return critic_prompt | llm
```

In [54]:

```
def get_refiner_agent(llm: ChatOpenAI):
    refiner_prompt = ChatPromptTemplate.from_template(
        "You are a 'Refiner' agent. Your task is to rewrite and improve an "
        "initial financial analysis based on a critique.\n\n"
        "Initial Analysis:\n{initial_analysis}\n\n"
        "Critique:\n{critique}\n\n"
        "Your Final, Rewritten Analysis:\n"
        "**Formatting instructions:** Your final output must be a well-structured "
        "report using markdown. Include a main header (e.g., '### <company name>'), "
        "use bolding for key metrics, and structure the content into clear, "
        "readable paragraphs with sub headers (##### Company Overview, Growth Drivers, "
        "Profitability and Financial Health, Cost Structure and Investments, Risks and "
        "Competitive Landscape, Market Performance and Analyst Sentiment, Recommendation, "
        "Summary)."
    )

    return refiner_prompt | llm
```

In [55]:

```
# Agent State
class AgentState(TypedDict):
    ticker: str
    plan: List[str]
    reasoning: str
    research_steps: Annotated[List[dict], operator.add]
    initial_analysis: str
    critique: str
    refined_analysis: str
    memory_confirmation: str
```

Agent Nodes

In [56]:

```
def planner_node(state):
    """Generates the research plan and reasoning."""
    planner_agent = get_planner_agent(
        ChatOpenAI(model="gpt-4.1-mini", temperature=0)
    )

    # The agent now returns a JSON string
    result_json_string = planner_agent.invoke({
        "input": f"Create a plan for {state['ticker']}"
    })['output']

    # Parse the JSON string into a Python dictionary
    plan_data = json.loads(result_json_string)

    return {
        "reasoning": plan_data["reasoning"],
        "plan": plan_data["plan"]
    }
```

In [57]:

```
def executor_node(state):
    """Executes the research plan."""
    executor_agent = get_executor_agent(
        ChatOpenAI(model="gpt-4.1-mini", temperature=0)
    )

    # Pass both the plan and the ticker to the executor
    result = executor_agent.invoke({
        "input": (
            f"Execute the following plan for the ticker {state['ticker']}:\n\n"
            f"PLAN: {state['plan']}"
        )
    })

    return {
        "research_steps": result['intermediate_steps'],
        "initial_analysis": result['output']
    }
```

In [58]:

```python
def critic_node(state):
    """Critiques the initial analysis."""
    critic_agent = get_critic_agent(
        ChatOpenAI(model="gpt-4.1-mini", temperature=0)
    )

    critique_text = critic_agent.invoke({
        "initial_analysis": state["initial_analysis"]
    }).content

    # st.write(critique_text)
    return {"critique": critique_text}
```

In [59]:

```python
def refiner_node(state):
    """Refines the analysis based on the critique."""
    refiner_agent = get_refiner_agent(
        ChatOpenAI(model="gpt-4.1-mini", temperature=0)
    )

    refined_text = refiner_agent.invoke({
        "initial_analysis": state["initial_analysis"],
        "critique": state["critique"]
    }).content

    # st.write(refined_text)
    return {"refined_analysis": refined_text}
```

In [60]:

```python
def save_memory_node(state):
    """Generates a key insight and saves it to memory."""
    llm = ChatOpenAI(model="gpt-4.1-mini", temperature=0)

    # Use an LLM to generate a concise note from the refined analysis
    note_generation_prompt = ChatPromptTemplate.from_template(
        "You are a note-taking assistant. Based on the following financial analysis, "
        "generate a single, concise sentence that captures the most important "
        "takeaway. This will be saved for future reference.\n\n"
        "Analysis:\n{analysis}\n\n"
        "Your single-sentence takeaway:"
    )

    note_chain = note_generation_prompt | llm
    key_insight = note_chain.invoke({
        "analysis": state["refined_analysis"]
    }).content

    # Save the generated note to memory
    confirmation = save_note_to_memory.invoke({
        "ticker": state["ticker"],
        "note": key_insight
    })

    return {"memory_confirmation": confirmation}
```

Graph Definition

In [61]:

```python
def build_agentic_workflow():
    workflow = StateGraph(AgentState)
    # Add the new nodes
    workflow.add_node("planner", planner_node)
    workflow.add_node("executor", executor_node)
    workflow.add_node("critic", critic_node)
    workflow.add_node("refiner", refiner_node)
    workflow.add_node("save_memory", save_memory_node)

    # Define the new sequence of edges
    workflow.set_entry_point("planner")
    workflow.add_edge("planner", "executor")
    workflow.add_edge("executor", "critic")
    workflow.add_edge("critic", "refiner")
    workflow.add_edge("refiner", "save_memory")
    workflow.add_edge("save_memory", END)

    return workflow.compile()
```

Definition for running workflow and formatting output

In [62]:

```
from rich.panel import Panel
from rich.tree import Tree
from rich.rule import Rule
from rich.syntax import Syntax
from rich.markdown import Markdown
from rich.console import Console
import pprint
import json

console = Console()
```

In [63]:

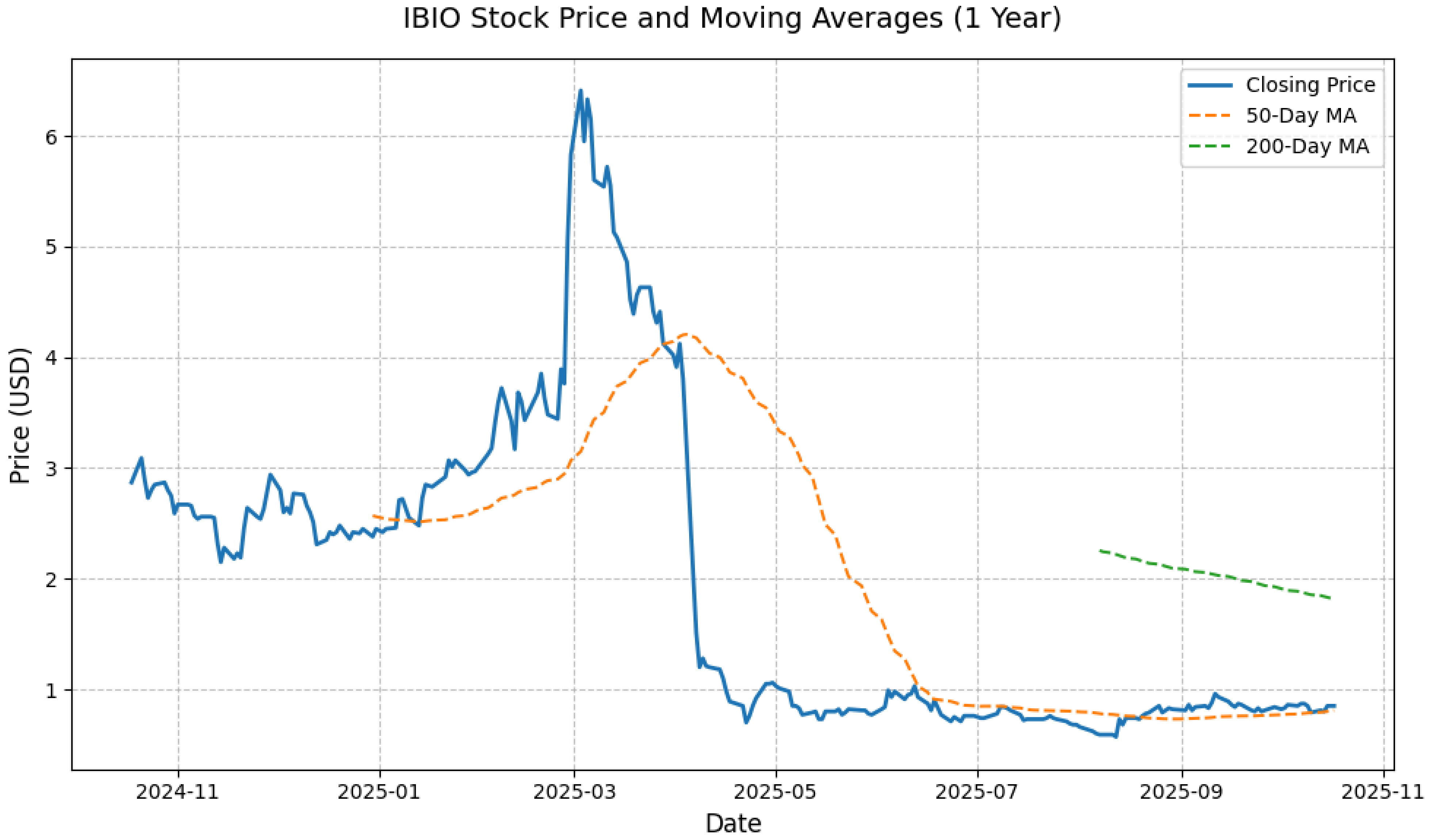
```
def run_workflow(ticker_symbol: str):
    """Executes the workflow and prints a structured, readable report."""
    console.print(Rule(f"[bold cyan]📈 Starting Analysis for "
                       f"{ticker_symbol}[/bold cyan]", style="cyan"))

    plot_price_trend(ticker_symbol)

    agent_workflow = build_agentic_workflow()
    inputs = {"ticker": ticker_symbol}
    final_report = "Analysis could not be generated."

    for event in agent_workflow.stream(inputs):
        for node_name, value in event.items():
            if node_name == "planner":
                reasoning = value.get('reasoning', 'No reasoning provided.')
                plan = value.get('plan', [])
                plan_str = "\n".join(f"- {tool}" for tool in plan)

                content = (f"[bold]Reasoning:[/bold]\n{reasoning}\n\n"
                           f"[bold]Plan:[/bold]\n{plan_str}")
                console.print(Panel(content, title="[bold yellow]🧠 Planner[/bold yellow]",
                                    border_style="yellow"))

            elif node_name == "executor":
                tree = Tree("🛠 Tool Calls", guide_style="cyan")
                research_steps = value.get('research_steps', [])

                for step in research_steps:
                    action, observation = step
                    tool_name = action.tool

                    tool_branch = tree.add(f"[bold green]{tool_name}[/bold green]")

                    if isinstance(observation, dict) or isinstance(observation, list):
                        obs_syntax = Syntax(json.dumps(observation, indent=2,
                                                        default=str), "json",
                                            theme="monokai", line_numbers=False)
                        tool_branch.add(obs_syntax)
                    else:
                        tool_branch.add(str(observation))

                executor_panel = Panel(tree, title="[bold green]🛠 Executor[/bold green]",
                                       border_style="green")
                console.print(executor_panel)

                initial_analysis = value.get('initial_analysis',
                                              'No initial analysis.')
                analysis_panel = Panel(Markdown(initial_analysis),
                                       title="[bold]Initial Analysis[/bold]",
                                       border_style="dim")
                console.print(analysis_panel)

            elif node_name == "critic":
                critique = value.get('critique', 'No critique provided.')
                console.print(Panel(Markdown(critique),
                                    title="[bold magenta]🧐 Critic[/bold magenta]",
                                    border_style="magenta"))

            elif node_name == "refiner":
                final_report = value.get("refined_analysis", final_report)
                console.print(Panel("[italic]Refining the analysis based on critique..."
                                    "[/italic]", title="[bold blue]✍ Refiner[/bold blue]",
                                    border_style="blue"))

            elif node_name == "save_memory":
                confirmation = value.get('memory_confirmation',
                                          'Could not save to memory.')
                console.print(Panel(confirmation,
                                    title="[bold white]💾 Memory[/bold white]",
                                    border_style="white"))

    console.print(Rule("[bold cyan]📜 FINAL REPORT[/bold cyan]", style="cyan"))
    console.print(Panel(Markdown(final_report), border_style="bold bright_white"))
    console.print(Rule("[bold green]✅ Workflow Complete[/bold green]", style="green"))
```

Execute Financial Analysis

In [64]:

```
ticker_to_analyze = "IBIO"
run_workflow(ticker_to_analyze)
```

```
───────────────────────────────────────────── 📈 Starting Analysis for IBIO ─────────────────────────────────────────────
```



🧠 **Planner**

Reasoning:
iBio, Inc. is a penny stock with a market cap under $50M, so the analysis scope is limited to price summary and financial news/analyst insights. Since it is in the Healthcare sector, focusing on innovation and competition is important, so searching for news on partnerships, acquisitions, and product launches will provide strategic insights.

Plan:
- get_price_summary
- Financial_News_Analyst
- search_specific_news

```
╭──────────────────────────────── 🛠 Executor ────────────────────────────────╮
│ 🛠 Tool Calls                                                              │
├── read_notes_from_memory
│   └── [
│         "[**2025-10-19**] - iBio remains a high-risk, early-stage biotech with significant debt and ongoing l
│         "[**2025-10-20**] - iBio is a high-risk, preclinical biotech with ongoing financial losses and debt b
│         "[**2025-10-20**] - iBio is a high-risk, preclinical biotech with promising obesity-focused pipeline
│       ]
├── get_price_summary
│   └── {
│         "latest_price": "$0.85",
│         "52_week_high": "$6.89",
│         "52_week_low": "$0.56",
│         "trend_analysis": {
│           "50_day_ma": "$0.81",
│           "200_day_ma": "$1.82",
│           "trend_conclusion": "Sideways/Consolidating"
│         },
│         "momentum_analysis": {
│           "rsi_14_day": "52.00",
│           "condition": "Neutral"
│         }
│       }
├── Financial_News_Analyst
│   └── {
│         "input": "IBIO",
│         "output": "iBio's latest news reflects a mixed but cautiously optimistic sentiment, with a significan
│         "intermediate_steps": [
│           [
│             "tool='get_stock_news' tool_input={'ticker': 'IBIO'} log=\"\\nInvoking: `get_stock_news` with `{'
│             [
│               "iBio Full Year 2025 Earnings: US$1.75 loss per share (vs US$4.03 loss in FY 2024)",
│               "Insider Buyers At iBio Likely Disappointed With 23% Slide",
│               "Is Ibio, Inc. (IBIO) Among Stocks Insiders Are Buying This Year?",
│               "iBio Expands Pipeline with New Antibody Agreement",
│               "Nano-Cap iBio Expands Obesity Program With New Antibody Using In-Licensed Platform"
│             ]
│           ]
│         ]
│       }
└── search_specific_news
    └── [
          "No specific news found for keywords 'pipeline, obesity, partnership, loss, stock slide' in the last
        ]
╰────────────────────────────────────────────────────────────────────────────╯
╭───────────────────────────── Initial Analysis ─────────────────────────────╮
│ iBio is a high-risk, early-stage biotech company with a history of significant debt and ongoing losses, but
│ recent notes indicate cautious optimism due to pipeline advancements and improved loss forecasts for 2025. The
│ current stock price is $0.85, near its 52-week low of $0.56 and well below its 52-week high of $6.89, with
│ technical indicators showing a sideways or consolidating trend and a neutral momentum (14-day RSI at 52).
│ Recent news sentiment is mixed but leans cautiously optimistic, highlighting a substantial reduction in
│ full-year 2025 losses compared to 2024, signaling improving financial performance. However, insider buying
│ sentiment has been dampened by a recent 23% stock price slide. Positively, the company is actively expanding
│ its pipeline through new antibody agreements and an obesity program, reflecting ongoing innovation and growth
│ potential. Despite this, no specific recent news articles were found in the last 30 days related to pipeline
│ developments, obesity, partnerships, losses, or the stock slide. Overall, iBio remains a speculative investment
│ suited for risk-tolerant investors who are focused on long-term potential rather than near-term stability.
╰────────────────────────────────────────────────────────────────────────────╯
```

```
🧐 Critic

 • Balance: Generally balanced, presenting both risks (debt, losses, stock slide) and opportunities (pipeline
   advancements, loss reduction, new programs). However, the absence of recent news on key developments could
   be emphasized more as a potential risk or uncertainty.
 • Clarity: Language is mostly clear and concise, though some sentences are dense and could be simplified for
   easier reading (e.g., breaking up long sentences).
 • Objectivity: Analysis is data-driven with specific financial metrics, stock price context, and technical
   indicators. The mention of mixed news sentiment and insider buying adds nuance. However, the phrase
   "cautious optimism" is somewhat subjective and could be supported with more concrete data or quotes.

Overall: Minor improvements in clarity and emphasizing the lack of recent news as a risk would enhance the
analysis.

✍️ Refiner
Refining the analysis based on critique...

💾 Memory
Successfully saved note for IBIO.

📜 FINAL REPORT
```

iBio

Company Overview

iBio is an early-stage biotech company characterized by a high-risk profile due to its history of significant debt and ongoing operating losses. The company is focused on developing innovative biologics, with recent strategic efforts aimed at expanding its product pipeline. While still in a developmental phase, iBio's progress has attracted cautious interest from investors who are willing to accept volatility in exchange for potential long-term gains.

Growth Drivers

Recent updates highlight iBio's initiatives to broaden its pipeline, including new antibody agreements and the launch of an obesity program. These efforts demonstrate the company's commitment to innovation and diversification, which could drive future revenue streams. However, it is important to note that no specific news related to these developments, partnerships, or pipeline progress has been reported in the past 30 days, introducing an element of uncertainty regarding near-term catalysts.

Profitability and Financial Health

iBio continues to operate at a loss, but management's forecasts indicate a substantial reduction in full-year losses for 2025 compared to 2024. This improvement suggests progress toward better financial discipline and operational efficiency. Despite this, the company's significant debt burden remains a concern, underscoring the need for careful monitoring of its capital structure and cash flow management.

Cost Structure and Investments

The company's ongoing investments in research and development, particularly in antibody technologies and obesity-related programs, contribute to its current cost base. These expenditures are essential for pipeline advancement but also weigh on near-term profitability. Balancing investment with financial sustainability will be critical as iBio moves forward.

Risks and Competitive Landscape

Key risks include the absence of recent news on pipeline milestones or partnerships, which may signal delays or challenges in development. The stock's recent 23% decline has also dampened insider buying sentiment, potentially reflecting internal concerns or market skepticism. Additionally, the biotech sector's inherent volatility and competitive pressures add layers of uncertainty to iBio's outlook.

Market Performance and Analyst Sentiment

The stock is currently trading at **$0.85**, close to its 52-week low of **$0.56** and significantly below its 52-week high of **$6.89.** Technical indicators suggest a sideways or consolidating trend, with a neutral momentum reflected by a 14-day RSI of **52.** News sentiment is mixed but leans slightly toward cautious optimism, primarily due to the projected reduction in losses for 2025. However, the lack of recent positive news and the sharp price drop have tempered enthusiasm among investors and insiders alike.

Recommendation

iBio remains a speculative investment best suited for risk-tolerant investors with a long-term horizon. While pipeline expansion and improving loss forecasts offer potential upside, the company's financial challenges, lack of recent news flow, and stock volatility present significant risks. Prospective investors should weigh these factors carefully and consider iBio as part of a diversified portfolio focused on early-stage biotech opportunities.

Summary

In summary, iBio shows signs of progress through pipeline development and improved financial outlooks, but the absence of recent news and ongoing financial risks warrant caution. The stock's current technical and sentiment

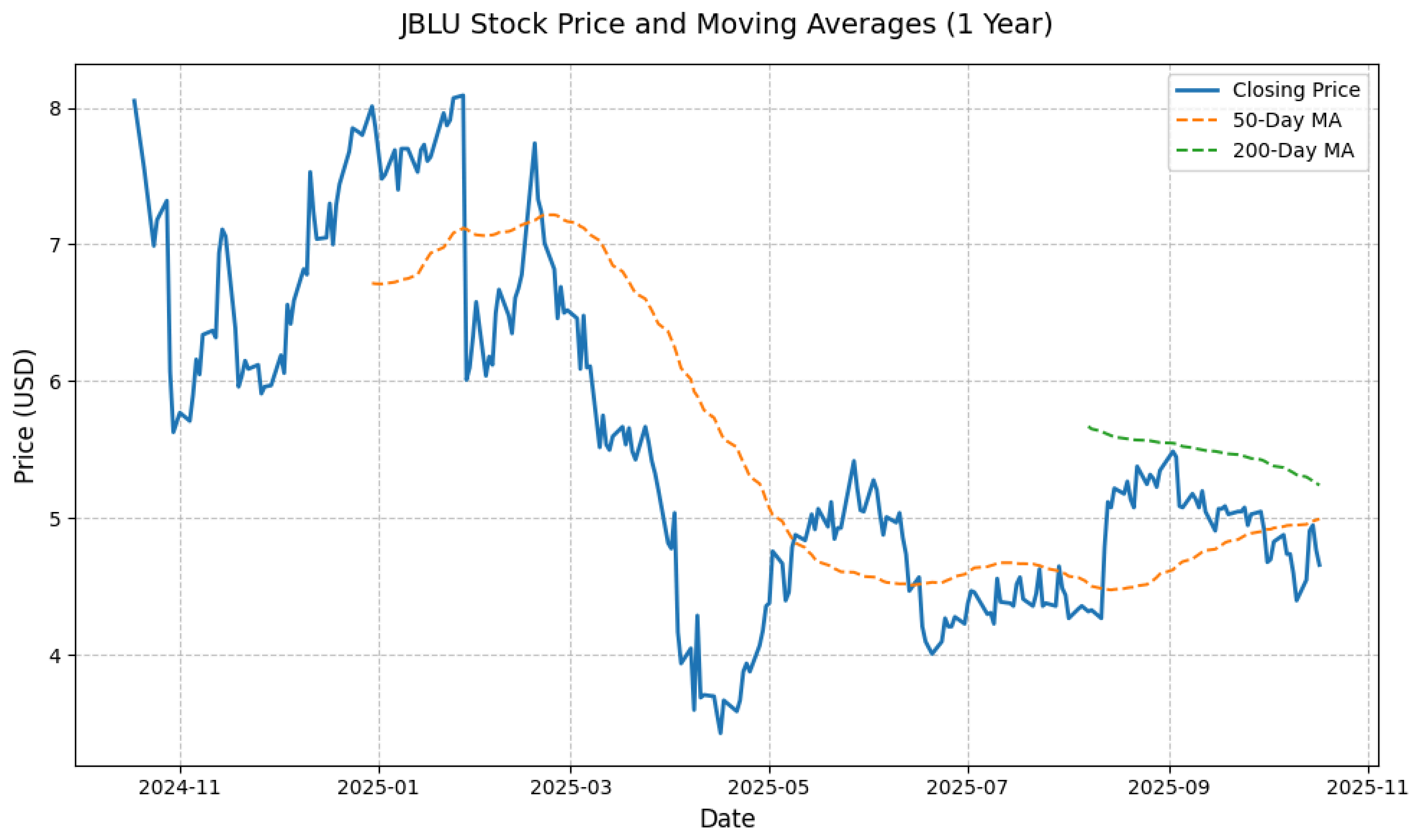
```
indicators reflect uncertainty, making iBio a high-risk, high-reward play for investors willing to accept
volatility in pursuit of future growth.
```

```
✅ Workflow Complete
```

In [65]:

```python
ticker_to_analyze = "JBLU"
run_workflow(ticker_to_analyze)
```

```
📈 Starting Analysis for JBLU
```



```
🧠 Planner
Reasoning:
JetBlue Airways Corporation is a Small-Cap company in the Industrials sector. The analysis should include price
summary, financial news and analyst insights, latest filings, and financial ratios to understand its growth and
financial health. Additionally, focus on balance sheet health, debt, and economic indicators like interest
rates due to its sector.

Plan:
- get_price_summary
- Financial_News_Analyst
- get_latest_filings
- get_financial_ratios
- get_economic_data
```

```
╭──────────────────────────────── 🛠 Executor ────────────────────────────────╮
│ 🛠 Tool Calls
│ ├── read_notes_from_memory
│ │   └── [
│ │         "[**2025-10-19**] - JetBlue Airways is currently struggling with significant financial losses and hig
│ │         "[**2025-10-20**] - JetBlue Airways faces significant near-term financial losses and high leverage am
│ │         "[**2025-10-20**] - JetBlue Airways faces significant financial challenges with ongoing losses and hi
│ │       ]
│ ├── get_price_summary
│ │   └── {
│ │         "latest_price": "$4.66",
│ │         "52_week_high": "$8.31",
│ │         "52_week_low": "$3.34",
│ │         "trend_analysis": {
│ │           "50_day_ma": "$5.00",
│ │           "200_day_ma": "$5.24",
│ │           "trend_conclusion": "Strong Downtrend"
│ │         },
│ │         "momentum_analysis": {
│ │           "rsi_14_day": "39.68",
│ │           "condition": "Neutral"
│ │         }
│ │       }
│ ├── Financial_News_Analyst
│ │   └── {
│ │         "input": "JBLU",
│ │         "output": "JetBlue (JBLU) is facing significant challenges as its shares have dropped 41% in 2025 ami
│ │         "intermediate_steps": [
│ │           [
│ │             "tool='get_stock_news' tool_input={'ticker': 'JBLU'} log=\"\\nInvoking: `get_stock_news` with `{'
│ │             [
│ │               "Airlines cutting fare is 'putting pressure' on low-cost carriers",
│ │               "United is focused on winning brand loyal customers",
│ │               "JetBlue Shares Slide 41% in 2025 Amid Ongoing Industry Turbulence\u2014What Does the Valuation
│ │               "CEO: AI and defense demand are remaking the space economy",
│ │               "Gov't shutdown could have lasting effects on airlines. Here's why.",
│ │               "Delta reports strong Q3 earnings, CEO sees 'significant improvement' in revenue outlook",
│ │               "What could push airlines to tipping point if US shutdown drags on",
│ │               "How the Story Behind JetBlue Is Evolving Amid Analyst Upgrades and Industry Shifts",
│ │               "Breeze Airways Sets First International Flights",
│ │               "Market Chatter: US Rewriting Rule on Protections for Passengers Using Wheelchairs"
│ │             ]
│ │           ]
│ │         ]
│ │       }
│ ├── get_latest_filings
│ │   └── [
│ │         {
│ │           "form_type": "10-Q",
│ │           "filed_at": "2025-07-29",
│ │           "accession_number": "000115846325000089",
│ │           "url": "https://www.sec.gov/Archives/edgar/data/0001158463/000115846325000089/jblu-20250630.htm",
│ │           "content": ": : jblu:aircraft jblu:airframe jblu:engine jblu:pilot_instructor jblu:lawsuit jblu:seg
│ │           "meta": {
│ │             "accession_number": "0001158463-25-000089",
│ │             "filing_date": "2025-07-29",
│ │             "reportDate": "2025-06-30",
│ │             "acceptanceDateTime": "2025-07-29 11:08:26+00:00",
│ │             "act": "34",
│ │             "form": "10-Q",
│ │             "fileNumber": "000-49728",
│ │             "items": "",
│ │             "size": 6742644,
│             "isXBRL": 1,
│             "isInlineXBRL": 1,
│             "primaryDocument": "jblu-20250630.htm",
│             "primaryDocDescription": "10-Q"
│           }
│         }
│       ]
├── get_financial_ratios
│   └── {
│         "trailing_pe": "N/A",
│         "forward_pe": -7.28125,
│         "price_to_book": 0.7038212,
│         "price_to_sales": 0.18545374,
│         "debt_to_equity": 390.988,
│         "return_on_equity": -0.15122,
│         "profit_margins": -0.04224
│       }
└── get_economic_data
    └── {
          "series": "GDP",
          "latest_value": "30,485.73",
          "latest_date": "2025-04-01"
        }
```

Initial Analysis

JetBlue Airways (JBLU) is currently facing significant financial challenges, including ongoing losses and high leverage, as reflected in its recent 10-Q filing for the quarter ended June 30, 2025. The company reported a net loss of $74 million for the second quarter and a net loss of $282 million for the first half of 2025, with operating income declining sharply compared to prior periods. Despite these losses, JetBlue maintains strong liquidity with $3.4 billion in unrestricted cash and equivalents and a $600 million undrawn credit line. The airline is navigating a tough industry environment marked by fare cuts and operational disruptions, including engine groundings related to Pratt & Whitney inspections. JetBlue is actively executing strategic initiatives under its JetForward plan, focusing on service reliability, network optimization, product enhancements, and financial stability. The company is also transitioning its fleet by selling its remaining Embraer aircraft and expanding partnerships, such as the Blue Sky collaboration with United Airlines. Technically, the stock is in a strong downtrend with the latest price at $4.66, below both the 50-day and 200-day moving averages, and a neutral RSI of 39.68. Financial ratios indicate a distressed valuation with no trailing P/E, a negative forward P/E of -7.28, a high debt-to-equity ratio of 391, and negative profitability metrics including a -4.2% profit margin and -15.1% return on equity. The broader economic context shows a U.S. GDP of $30.49 trillion as of April 2025, indicating a large but potentially volatile economic environment. Overall, JetBlue is contending with significant near-term financial and operational headwinds but is pursuing strategic initiatives and maintaining liquidity that could support a recovery if industry conditions stabilize.

🧐 Critic

- **Balance:** Good coverage of both risks (losses, high leverage, operational disruptions, downtrend stock) and opportunities (strong liquidity, strategic initiatives, partnerships).
- **Clarity:** Generally clear and concise; some sentences could be simplified for easier reading (e.g., breaking complex sentences).
- **Objectivity:** Strongly data-driven with specific financial figures, ratios, and market context; avoids subjective language.

No major changes needed.

✍ Refiner

Refining the analysis based on critique...

💾 Memory

Successfully saved note for JBLU.

📜 FINAL REPORT

JetBlue Airways (JBLU)

Company Overview

JetBlue Airways is currently navigating a challenging financial and operational environment, as detailed in its 10-Q filing for the quarter ended June 30, 2025. The airline reported a **net loss of $74 million** in Q2 and a **net loss of $282 million** for the first half of 2025. Operating income has declined sharply compared to previous periods, reflecting ongoing pressures in the airline industry.

Growth Drivers

Despite these setbacks, JetBlue maintains a strong liquidity position with **$3.4 billion in unrestricted cash and equivalents** and an additional **$600 million undrawn credit facility.** The company is actively pursuing its JetForward strategic plan, which emphasizes improving service reliability, optimizing its network, enhancing product offerings, and strengthening financial stability. Fleet modernization efforts include the sale of remaining Embraer aircraft, while partnerships such as the Blue Sky collaboration with United Airlines aim to expand market reach and operational synergies.

Profitability and Financial Health

JetBlue’s financial ratios highlight significant distress. The company currently has **no trailing P/E ratio** and a **negative forward P/E of -7.28,** indicating ongoing losses. Its **debt-to-equity ratio stands at 391,** underscoring high leverage. Profitability metrics remain negative, with a **profit margin of -4.2%** and a **return on equity of -15.1%.** These figures reflect the company’s struggle to generate positive returns amid industry headwinds.

Cost Structure and Investments

Operational disruptions, including engine groundings related to Pratt & Whitney inspections, have added to costs and complexity. The airline is managing fare competition and other industry pressures, which have contributed to margin compression. Investments in network optimization and product enhancements under JetForward are designed to improve long-term cost efficiency and customer experience.

Risks and Competitive Landscape

JetBlue faces significant near-term risks from a volatile industry environment marked by fare cuts, operational challenges, and high leverage. The broader U.S. economic context, with a GDP of **$30.49 trillion** as of April 2025, presents both opportunities and uncertainties that could impact demand and pricing power.

Market Performance and Analyst Sentiment

Technically, JetBlue’s stock is in a pronounced downtrend, trading at **$4.66,** below both its 50-day and 200-day moving averages. The Relative Strength Index (RSI) is neutral at **39.68,** suggesting limited momentum. Market valuation reflects distress, consistent with the company’s financial challenges.

Recommendation

While JetBlue faces significant near-term financial and operational headwinds, its strong liquidity position and strategic initiatives provide a foundation for potential recovery if industry conditions stabilize. Investors should monitor the company’s execution of its JetForward plan, operational improvements, and broader market dynamics before considering exposure.

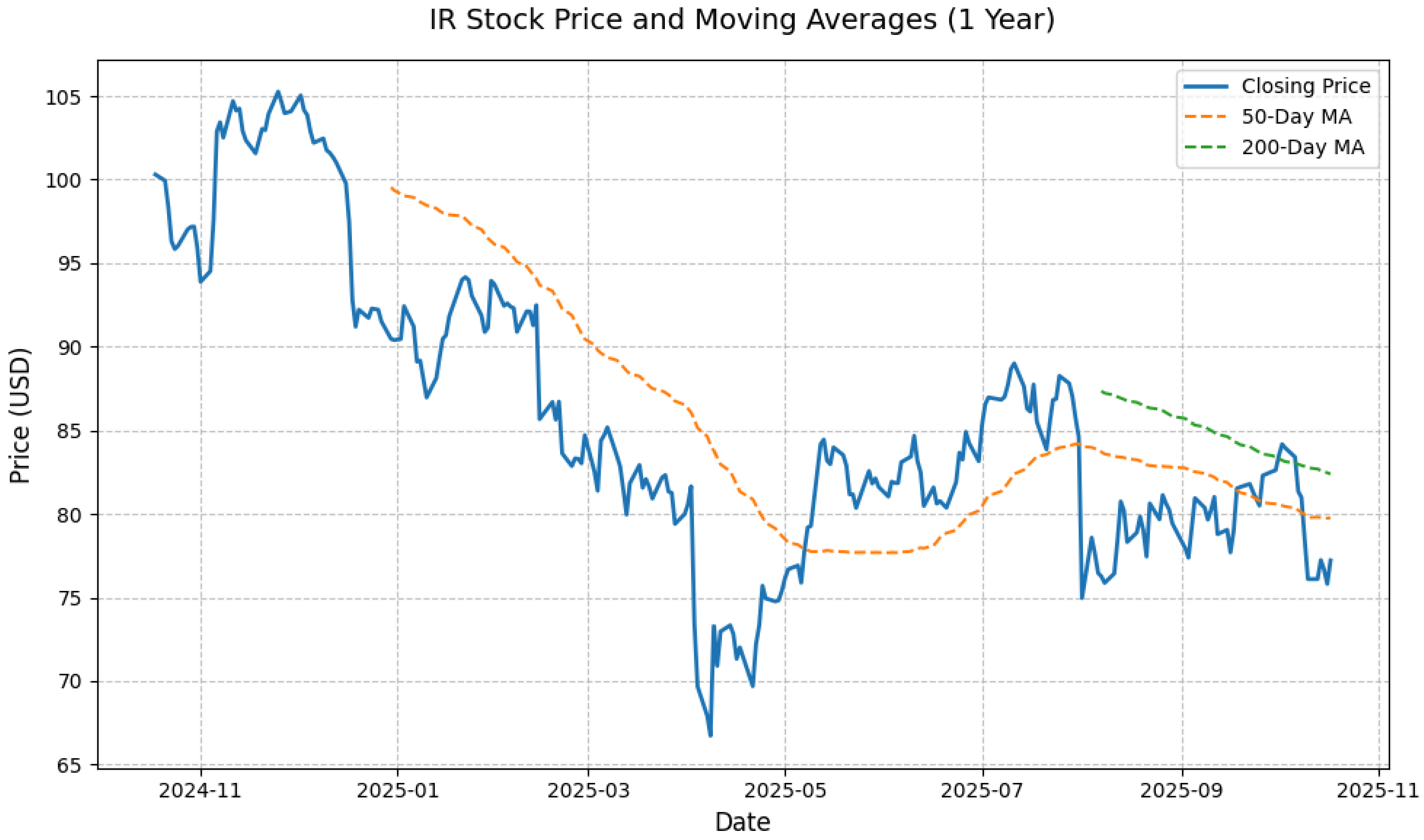
Summary

JetBlue Airways is contending with ongoing losses, high leverage, and operational disruptions amid a tough airline industry environment. However, the company’s robust liquidity, strategic focus on service and network optimization, and expanding partnerships offer pathways to stabilize and improve performance over time. Caution is warranted given current financial distress and market volatility, but the airline’s initiatives could support a turnaround if external conditions improve.

✅ Workflow Complete

In [66]:
```python
ticker_to_analyze = "IR"
run_workflow(ticker_to_analyze)
```

📈 Starting Analysis for IR



🧠 Planner

Reasoning:
Ingersoll Rand Inc. is a large-cap company with a market cap over $10B in the Industrials sector. A comprehensive analysis is required, including price summary, financial news, financial ratios, latest filings, economic data, and analyst ratings. Since it is in the Industrials sector, the focus should be on balance sheet health, debt, and economic indicators like interest rates.

Plan:
- get_price_summary
- Financial_News_Analyst
- get_financial_ratios
- get_latest_filings
- get_economic_data
- get_analyst_ratings

```
╭──────────────────────────────── 🛠 Executor ────────────────────────────────╮
│ 🛠 Tool Calls
│ ├── read_notes_from_memory
│ │   └── [
│ │         "[**2025-10-19**] - Ingersoll Rand faces near-term operational and profitability challenges but maint
│ │         "[**2025-10-20**] - Ingersoll Rand\u2019s acquisition-driven growth and strong liquidity position it
│ │         "[**2025-10-20**] - Ingersoll Rand faces near-term challenges from volume declines and restructuring
│ │       ]
│ ├── get_price_summary
│ │   └── {
│ │         "latest_price": "$77.22",
│ │         "52_week_high": "$105.95",
│ │         "52_week_low": "$65.58",
│ │         "trend_analysis": {
│ │           "50_day_ma": "$79.77",
│ │           "200_day_ma": "$82.39",
│ │           "trend_conclusion": "Strong Downtrend"
│ │         },
│ │         "momentum_analysis": {
│ │           "rsi_14_day": "30.53",
│ │           "condition": "Neutral"
│ │         }
│ │       }
│ ├── Financial_News_Analyst
│ │   └── {
│ │         "input": "IR",
│ │         "output": "Recent news on Ingersoll Rand (IR) presents a mixed to cautiously optimistic sentiment, wi
│ │         "intermediate_steps": [
│ │           [
│ │             "tool='get_stock_news' tool_input={'ticker': 'IR'} log=\"\\nInvoking: `get_stock_news` with `{'ti
│ │             [
│ │               "What to Expect From Ingersoll Rand's Q3 2025 Earnings Report",
│ │               "Gas and Liquid Handling Stocks Q2 Highlights: Ingersoll Rand (NYSE:IR)",
│ │               "Should Investors Revisit Ingersoll Rand After the Recent 8.8% Share Price Slide?",
│ │               "Strength in Intelligent Devices Unit Aids Emerson: Will the Momentum Last?",
│ │               "3 Reasons to Avoid IR and 1 Stock to Buy Instead",
│ │               "1 Unpopular Stock That Should Get More Attention and 2 We Turn Down",
│ │               "Is Ingersoll Rand Inc. (NYSE:IR) Potentially Undervalued?",
│ │               "Ingersoll Rand Gains From Business Strength & Buyouts Amid Risks",
│ │               "Ingersoll (IR) Upgraded to Buy: What Does It Mean for the Stock?",
│ │               "OSK vs. IR: Which Stock Should Value Investors Buy Now?"
│ │             ]
│ │           ]
│ │         ]
│ │       }
│ ├── get_financial_ratios
│ │   └── {
│ │         "trailing_pe": 59.860466,
│ │         "forward_pe": 21.81356,
│ │         "price_to_book": 3.0424333,
│ │         "price_to_sales": 4.1675763,
│ │         "debt_to_equity": 49.167,
│ │         "return_on_equity": 0.05251,
│ │         "profit_margins": 0.07096
│ │       }
│ ├── get_latest_filings
│ │   └── [
│ │         {
│ │           "form_type": "10-Q",
│ │           "filed_at": "2025-07-31",
│ │           "accession_number": "000162828025037049",
│ │           "url": "https://www.sec.gov/Archives/edgar/data/0001699150/000162828025037049/iri-20250630.htm",
│ │           "content": ": : iri:acquisition iri:extensionOption iri:contract : iri:subsidiary iri:segment iri:
      "meta": {
        "accession_number": "0001628280-25-037049",
        "filing_date": "2025-07-31",
        "reportDate": "2025-06-30",
        "acceptanceDateTime": "2025-07-31 20:43:32+00:00",
        "act": "34",
        "form": "10-Q",
        "fileNumber": "001-38095",
        "items": "",
        "size": 13347899,
        "isXBRL": 1,
        "isInlineXBRL": 1,
        "primaryDocument": "iri-20250630.htm",
        "primaryDocDescription": "10-Q"
      }
    }
  ]
├── get_economic_data
│   └── {
          "series": "GDP",
          "latest_value": "30,485.73",
          "latest_date": "2025-04-01"
        }
└── get_analyst_ratings
    └── {
          "period": "0m",
          "buy_ratings": 8,
          "hold_ratings": 8,
          "sell_ratings": 0,
          "latest_recommendations": []
        }
```

Initial Analysis

Ingersoll Rand (IR) currently trades at $77.22, down from its 52-week high of $105.95, and is in a strong downtrend with its 50-day and 200-day moving averages at $79.77 and $82.39 respectively. The 14-day RSI is neutral at 30.53. Historical notes indicate the company faces near-term operational and profitability challenges but has strong liquidity and growth potential through strategic acquisitions, warranting cautious optimism. Recent news sentiment is mixed to cautiously optimistic, focusing on upcoming Q3 2025 earnings, share price declines, and potential undervaluation, with some buy rating upgrades noted. Financial ratios show a high trailing P/E of about 59.9 but a more reasonable forward P/E of 21.8, a price-to-book of 3.04, price-to-sales of 4.17, debt-to-equity of 49.17, return on equity of 5.25%, and profit margins of 7.1%, reflecting moderate profitability and leverage. The latest 10-Q filing for Q2 2025 reveals that Ingersoll Rand has been actively acquiring complementary businesses to drive growth, with several acquisitions in 2024 and 2025 expanding its Industrial Technologies and Services and Precision and Science Technologies segments. The company reported revenues of $1.89 billion for Q2 2025, up 4.6% year-over-year, driven by acquisitions and pricing, though organic volumes declined. Gross profit margin slightly decreased to 43.7%. Operating income fell sharply due to goodwill and intangible asset impairments and equity investment impairments, resulting in a net loss of $113.8 million for the quarter, compared to net income in the prior year period. Adjusted EBITDA improved slightly, reflecting operational resilience. The company maintains strong liquidity with $1.31 billion in cash and $2.6 billion in unused credit facilities, and remains compliant with debt covenants. The broader economic context shows US GDP at $30.49 trillion as of April 2025, indicating a stable macroeconomic environment. Analyst ratings are evenly split between buy and hold, with no sell ratings, reflecting cautious but positive market sentiment. Overall, Ingersoll Rand is navigating near-term profitability pressures and restructuring costs but is supported by strategic acquisitions, solid liquidity, and a diversified product portfolio, positioning it for potential medium- to long-term recovery amid cautious investor optimism.

🧐 Critic

- **Balance:**
  - Presents both risks (downtrend, operational challenges, impairments, net loss) and opportunities (strategic acquisitions, liquidity, growth potential).
  - Could improve by explicitly highlighting potential downside risks from high valuation metrics or economic uncertainties.
- **Clarity:**
  - Generally clear and detailed, but dense with financial jargon and metrics that may overwhelm some readers.
  - Could benefit from breaking up long sentences and summarizing key points more succinctly.
- **Objectivity:**
  - Strongly data-driven with comprehensive use of financial ratios, recent filings, and market indicators.
  - Maintains neutral tone without overt bias, supported by balanced presentation of analyst ratings and news sentiment.

Overall: Well-rounded and thorough analysis; minor improvements in clarity and explicit risk articulation recommended.

✍️ Refiner

Refining the analysis based on critique...

💾 Memory

Successfully saved note for IR.

📜 FINAL REPORT

Ingersoll Rand (IR) Financial Analysis

Company Overview

Ingersoll Rand (IR) is currently trading at **$77.22**, down from its 52-week high of **$105.95.** The stock is in a clear downtrend, with the **50-day moving average at $79.77** and the **200-day moving average at $82.39**, both above the current price. The 14-day Relative Strength Index (RSI) stands at **30.53**, indicating a neutral to slightly oversold condition. The company operates through diversified segments including Industrial Technologies and Services, and Precision and Science Technologies.

Growth Drivers

Ingersoll Rand has pursued an aggressive acquisition strategy in 2024 and 2025, targeting complementary businesses to expand its product portfolio and market reach. These acquisitions have contributed to a **4.6% year-over-year revenue increase to $1.89 billion in Q2 2025**, driven primarily by pricing and acquired volumes. While organic volume declined, the company's strategic expansion efforts position it for medium- to long-term growth potential.

Profitability and Financial Health

Despite revenue growth, profitability has been pressured. Gross profit margin slightly contracted to **43.7%**, and operating income declined sharply due to significant goodwill, intangible asset impairments, and equity investment write-downs. This led to a **net loss of $113.8 million in Q2 2025**, compared to net income in the prior year period. However, adjusted EBITDA showed a modest improvement, reflecting some operational resilience.

Key financial ratios include a **trailing P/E of approximately 59.9**, which is elevated, but a more reasonable **forward P/E of 21.8.** The company's **price-to-book ratio is 3.04**, and **price-to-sales ratio is 4.17**, indicating a relatively high valuation. Leverage remains moderate with a **debt-to-equity ratio of 49.17**, while profitability metrics show a **return on equity of 5.25%** and **profit margins of 7.1%**, suggesting modest returns on invested capital.

Liquidity remains a strong point, with **$1.31 billion in cash** and **$2.6 billion in unused credit facilities.** The company is compliant with all debt covenants, supporting financial flexibility amid ongoing restructuring and investment.

Cost Structure and Investments

The recent impairments reflect the costs associated with restructuring and integration of acquisitions. These non-cash charges have weighed on reported earnings but are part of the company's efforts to streamline operations and refocus its portfolio. Continued investment in innovation and operational efficiency will be critical to improving margins and profitability going forward.

Risks and Competitive Landscape

Several risks warrant attention. The stock's downtrend and elevated valuation metrics suggest downside vulnerability, especially if organic growth does not recover or if macroeconomic conditions deteriorate. The US economy remains stable with GDP at **$30.49 trillion as of April 2025**, but potential economic uncertainties could impact demand in industrial sectors. Additionally, ongoing restructuring costs and impairments may continue to pressure near-term earnings.

Market Performance and Analyst Sentiment

Market sentiment is mixed but cautiously optimistic. Analyst ratings are evenly split between buy and hold, with no sell recommendations. Recent news highlights concerns over share price declines and near-term profitability challenges but also notes potential undervaluation and strategic growth opportunities. The balanced analyst outlook reflects investor caution amid restructuring, yet confidence in the company's long-term prospects.

Recommendation

Given the current downtrend, near-term earnings pressures, and high trailing valuation, investors should approach Ingersoll Rand with caution. However, the company's strong liquidity, strategic acquisitions, and diversified portfolio provide a foundation for recovery. A **hold** rating is appropriate for investors seeking exposure to industrial growth with a medium- to long-term horizon, while risk-tolerant investors may consider selective buying on weakness.

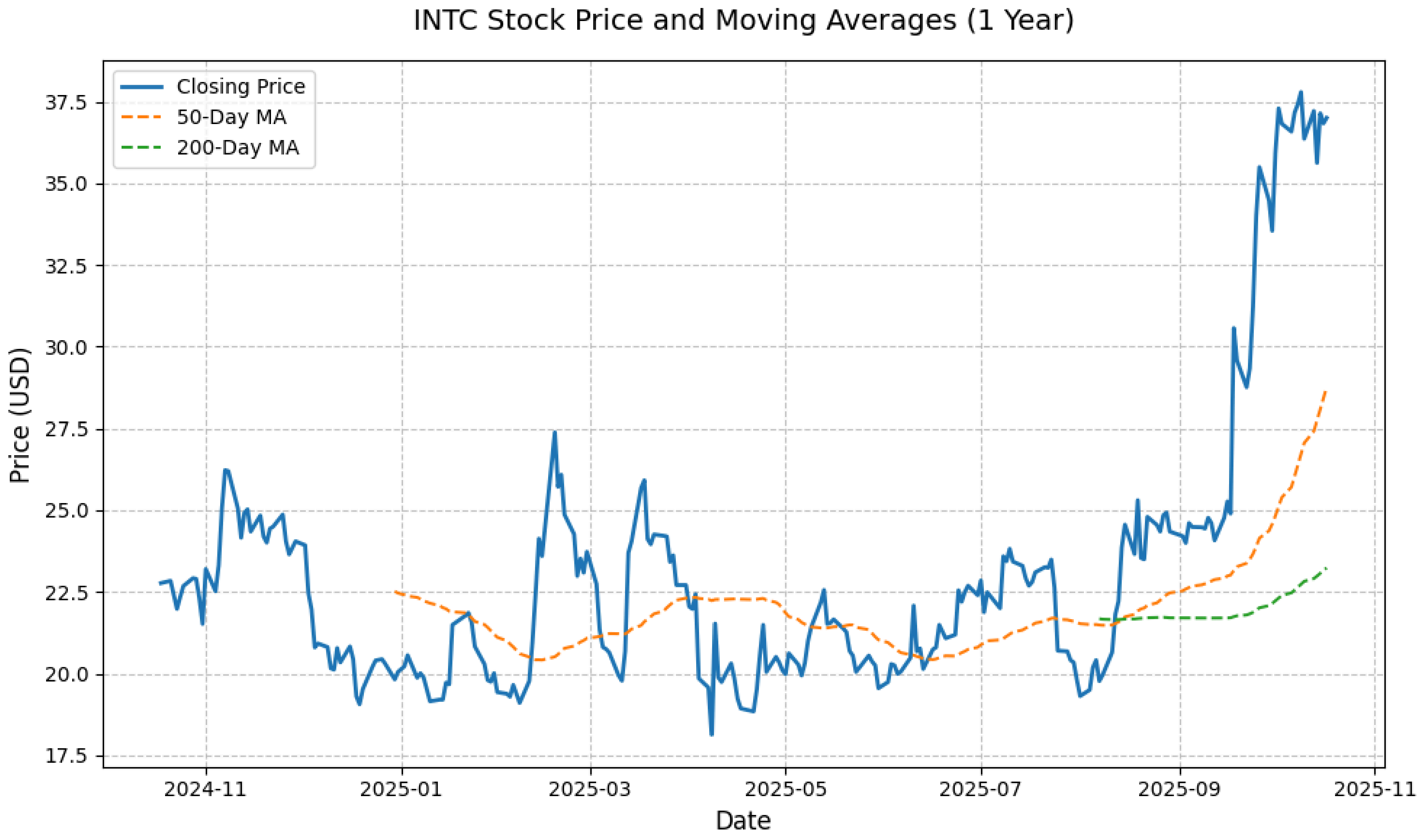
Summary

Ingersoll Rand is navigating a challenging period marked by operational headwinds, impairments, and a net loss in Q2 2025. Nonetheless, its strategic acquisitions, solid liquidity position, and diversified business segments offer growth potential. Elevated valuation metrics and economic uncertainties pose downside risks, but balanced analyst sentiment and improving adjusted EBITDA suggest cautious optimism. The company appears positioned for a gradual recovery, making it a candidate for investors with a moderate risk appetite and a focus on medium- to long-term value creation.

✅ Workflow Complete

In [67]:
```python
ticker_to_analyze = "INTC"
run_workflow(ticker_to_analyze)
```

📈 Starting Analysis for INTC



```
🧠 Planner
Reasoning:
Intel Corporation is a large-cap company in the Technology sector with a market cap of approximately $176
billion. A comprehensive analysis is required, including price summary, financial news, financial ratios,
latest filings, economic data, analyst ratings, and Google Trends to capture market sentiment. Additionally,
since it is in the Technology sector, we will focus on innovation and competition by searching for news on
partnerships, acquisitions, and product launches.

Plan:
- get_price_summary
- Financial_News_Analyst
- get_financial_ratios
- get_latest_filings
- get_economic_data
- get_analyst_ratings
- get_google_trends
- search_specific_news
```

```
/Users/jagadeesch/USD/venv/lib/python3.12/site-packages/pytrends/request.py:260: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and wi
ll change in a future version. Call result.infer_objects(copy=False) instead. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
  df = df.fillna(False)
```

```
╭──────────────────────────────── 🛠 Executor ────────────────────────────────╮
│ 🛠 Tool Calls
│ ├── read_notes_from_memory
│ │   └── [
│ │         "[**2025-10-19**] - Intel\u2019s strong AI-driven growth potential and solid liquidity are tempered b
│ │         "[**2025-10-20**] - Intel shows strong growth potential driven by AI advancements and government supp
│ │         "[**2025-10-20**] - Intel is showing strong technical momentum and strategic AI-driven growth potenti
│ │       ]
│ ├── get_price_summary
│ │   └── {
│ │         "latest_price": "$37.01",
│ │         "52_week_high": "$39.65",
│ │         "52_week_low": "$17.67",
│ │         "trend_analysis": {
│ │           "50_day_ma": "$28.76",
│ │           "200_day_ma": "$23.24",
│ │           "trend_conclusion": "Strong Uptrend"
│ │         },
│ │         "momentum_analysis": {
│ │           "rsi_14_day": "60.14",
│ │           "condition": "Neutral"
│ │         }
│ │       }
│ ├── Financial_News_Analyst
│ │   └── {
│ │         "input": "INTC",
│ │         "output": "Recent news around Intel (INTC) reflects a cautiously optimistic sentiment, with particula
│ │         "intermediate_steps": [
│ │           [
│ │             "tool='get_stock_news' tool_input={'ticker': 'INTC'} log=\"\\nInvoking: `get_stock_news` with `{'
│ │             [
│ │               "Stock market today: Dow, S&P 500, Nasdaq futures steady with government shutdown, earnings sea
│ │               "What to Expect in Markets This Week: CPI Inflation Data; Tesla, Netflix, Intel Earnings",
│ │               "The Donald Trump Administration Purchased Stakes in Intel, MP Materials, Lithium Americas, and
│ │               "How Securing a Major AI Client for 18A Chips At Intel (INTC) Has Changed Its Investment Story"
│ │               "The Weekend: IMF warns of global debt emergency as Reeves grapples with slowing jobs market",
│ │               "Inflation Data, Intel and Tesla Earnings: What to Watch Next Week",
│ │               "AMD Stock Has Ample Room to Rise, Say Analysts. Thank Oracle and Meta.",
│ │               "Intel Edges Higher on Report of Major AI Chip Client for 18A Process",
│ │               "Intel's AI client, bitcoin hits June low, Micron's China business",
│ │               "This AI Stock With A Difference Builds Bullish Base, Eyes Entry Amid 83% Rally"
│ │             ]
│ │           ]
│ │         ]
│ │       }
│ ├── get_financial_ratios
│ │   └── {
│ │         "trailing_pe": "N/A",
│ │         "forward_pe": 38.154636,
│ │         "price_to_book": 1.6549656,
│ │         "price_to_sales": 3.3175924,
│ │         "debt_to_equity": 47.997,
│ │         "return_on_equity": -0.18616,
│ │         "profit_margins": -0.38636002
│ │       }
│ ├── get_latest_filings
│ │   └── [
│ │         {
│ │           "form_type": "10-Q",
│ │           "filed_at": "2025-07-24",
│ │           "accession_number": "000005086325000109",
│ │           "url": "https://www.sec.gov/Archives/edgar/data/0000050863/000005086325000109/intc-20250628.htm",
│ │           "content": ": : Washington, D.C. 20549 (Mark One) 13 15(d) 1934 For the quarterly period ended or 1
│           "meta": {
│             "accession_number": "0000050863-25-000109",
│             "filing_date": "2025-07-24",
│             "reportDate": "2025-06-28",
│             "acceptanceDateTime": "2025-07-24 20:08:10+00:00",
│             "act": "34",
│             "form": "10-Q",
│             "fileNumber": "000-06217",
│             "items": "",
│             "size": 12348105,
│             "isXBRL": 1,
│             "isInlineXBRL": 1,
│             "primaryDocument": "intc-20250628.htm",
│             "primaryDocDescription": "10-Q"
│           }
│         }
│       ]
├── get_economic_data
│   └── {
│         "series": "GDP",
│         "latest_value": "30,485.73",
│         "latest_date": "2025-04-01"
│       }
├── get_analyst_ratings
│   └── {
│         "period": "0m",
│         "buy_ratings": 3,
│         "hold_ratings": 32,
│         "sell_ratings": 8,
│         "latest_recommendations": []
│       }
├── get_google_trends
│   └── {
│         "keyword_searched": "Intel Corporation",
│         "average_interest_score": 39.3,
│         "peak_interest_date": "2025-08-19",
│         "comment": "The average interest score for 'Intel Corporation' is 39.3 out of 100 over the last 3 mon
│       }
└── search_specific_news
    └── [
          "Chip Stocks Drop AMD, Nvidia, and Intel Sink After China Restricts Rare Earth Exports",
          "Nasdaq Futures Gain on AI Optimism",
          "US stock market today: Dow, S&P 500, Nasdaq futures steady as shutdown drags on; Intel rises on pric
        ]
```

Initial Analysis

Intel Corporation (INTC) is currently in a strong uptrend with a latest price of $37.01, near its 52-week high of $39.65 and well above its 50-day and 200-day moving averages of $28.76 and $23.24 respectively. The 14-day RSI is neutral at 60.14, indicating balanced momentum. Recent news sentiment is moderately positive, driven by Intel securing a major AI client for its 18A chips, which has improved investment perspectives despite broader market uncertainties such as government shutdown concerns and inflation data. Financially, Intel faces challenges with a negative return on equity (-18.6%) and negative profit margins (-38.6%), though it maintains a manageable debt-to-equity ratio of 48. The forward P/E ratio stands at 38.15, reflecting expectations of future earnings growth. The latest 10-Q filing reveals ongoing restructuring efforts with significant charges related to employee severance and asset impairments, alongside a strategic focus on AI-driven growth and manufacturing capacity adjustments. Intel's liquidity remains strong with over $21 billion in cash and short-term investments and total debt around $50.8 billion. The company is navigating risks related to its next-generation semiconductor manufacturing process technology, Intel 14A, with potential strategic and financial impacts if it fails to secure significant external foundry customers. Analyst ratings show a consensus leaning towards hold, with 3 buys, 32 holds, and 8 sells, indicating cautious optimism. Google Trends data shows moderate public interest with an average score of 39.3 over the past three months. Recent news searches highlight market reactions to geopolitical and regulatory developments but no major new partnerships or product launches were identified in the last 30 days. Overall, Intel is positioned with strong technical momentum and promising AI-related growth prospects but faces near-term profitability challenges, restructuring costs, and execution risks that warrant a cautious investment approach.

🧐 Critic

- **Balance:** Well-balanced; highlights both growth opportunities (AI client win, strong liquidity, technical momentum) and risks (negative profitability, restructuring costs, manufacturing execution risks).
- **Clarity:** Generally clear and concise; some sentences could be simplified for easier readability, but overall understandable.
- **Objectivity:** Strongly data-driven with specific financial metrics, technical indicators, analyst ratings, and external data points cited.

No major changes needed.

✍️ Refiner

Refining the analysis based on critique...

💾 Memory

Successfully saved note for INTC.

📜 FINAL REPORT

Intel Corporation (INTC)

Company Overview

Intel Corporation is currently exhibiting strong technical momentum, with its latest share price at **$37.01**, approaching a 52-week high of **$39.65**. The stock trades well above its key moving averages, including the **50-day moving average of $28.76** and the **200-day moving average of $23.24**, signaling a robust uptrend. The 14-day Relative Strength Index (RSI) stands at a neutral **60.14**, indicating balanced market momentum.

Growth Drivers

Recent positive sentiment is fueled by Intel securing a major AI client for its advanced **18A chips**, underscoring the company's strategic pivot toward AI-driven growth. This development enhances investor confidence despite broader macroeconomic uncertainties such as inflation concerns and potential government shutdowns. Intel's ongoing restructuring efforts aim to optimize manufacturing capacity and focus on next-generation semiconductor technologies, positioning the company to capitalize on future demand in AI and high-performance computing sectors.

Profitability and Financial Health

Intel faces near-term profitability challenges, reflected in a **negative return on equity (ROE) of -18.6%** and a **negative profit margin of -38.6%**. However, the company maintains a manageable **debt-to-equity ratio of 48%** and strong liquidity, with over **$21 billion in cash and short-term investments** against total debt of approximately **$50.8 billion**. The forward price-to-earnings (P/E) ratio of **38.15** suggests that the market anticipates future earnings growth despite current losses.

Cost Structure and Investments

The latest 10-Q filing reveals significant restructuring charges related to employee severance and asset impairments. These costs reflect Intel's strategic realignment and investment in manufacturing process improvements, particularly around its next-generation **Intel 14A** technology. While these investments are critical for long-term competitiveness, they contribute to short-term financial pressure.

Risks and Competitive Landscape

Execution risks remain a key concern, especially regarding the successful ramp-up and external adoption of the Intel 14A manufacturing process. Failure to secure substantial external foundry customers could have strategic and financial repercussions. Additionally, ongoing geopolitical and regulatory developments add layers of uncertainty to Intel's operating environment.

Market Performance and Analyst Sentiment

Analyst consensus is cautiously optimistic, with **3 buy ratings, 32 holds, and 8 sells**, reflecting a generally neutral stance. Public interest, as measured by Google Trends, remains moderate with an average score of **39.3** over the past three months. Recent news coverage has focused on geopolitical and regulatory issues rather than new partnerships or product launches.

Recommendation

Intel's strong technical position and promising AI-related growth prospects are tempered by near-term profitability challenges, restructuring costs, and execution risks. Investors should adopt a cautious approach, balancing the company's growth potential against its current financial and operational headwinds.

Summary

Intel is well-positioned to benefit from AI-driven demand and manufacturing innovation, supported by solid liquidity and positive technical indicators. However, ongoing restructuring, negative profitability metrics, and execution uncertainties warrant careful monitoring. A balanced investment stance is recommended until

```
clearer evidence of sustained profitability and successful technology adoption emerges.

──────────────────────────── ✅ Workflow Complete ────────────────────────────
```

Evaluation and Iteration

The **Evaluation and Iteration** highlights how the multi-agent system self-assesses, refines, and learns over time. While the **Evaluator–Optimizer Workflow** defines the conceptual design, this section demonstrates how these behaviors are realized during execution.

In the first stage of each run, the **Planner Agent** evaluates contextual factors such as **market capitalization** and **industry classification** to decide which analytical tools and data sources should be used. This ensures that each research plan is tailored to the specific characteristics of the company under analysis, optimizing relevance and depth.

1. Internal Evaluation (Critic–Refiner Feedback Loop)

After the **Executor Agent** generates the `initial_analysis`, the **Critic Agent** evaluates it for clarity, balance, and objectivity. The Critic identifies areas where the analysis may be biased, incomplete, or lacking coherence. This structured critique is then passed to the **Refiner Agent**, which rewrites the report by addressing the critic's feedback, resulting in a polished and balanced `refined_analysis`.

This iterative cycle ensures that the final analysis presented to the user undergoes internal quality control — mirroring a human-like peer review process that enhances the reliability of the system's outputs.

2. Memory and Learning Across Runs

The **Memory Agent** provides continuity between runs by storing summarized insights in a lightweight memory store (e.g., `memory.json`). When the system is re-run, the **Executor Agent** references these stored notes to build contextual awareness and generate more informed research outputs.

This mechanism enables incremental learning — allowing the system to recall past analyses, adapt to recurring patterns, and progressively improve its performance across sessions.

Together, the **Critic–Refiner** feedback loop and the **Memory-driven learning mechanism** create a foundation for self-improvement, ensuring that each new execution of the system produces more coherent, context-aware, and high-quality insights than before.

Streamlit Integration and Demonstration Video

As a further iteration of our agentic workflow, we extended the **Multi-Agent Financial Analysis System** into an interactive **Streamlit web application**. This integration allows users to enter stock symbols, trigger the Planner–Executor–Critic–Refiner pipeline in real time, and visualize results directly through an intuitive dashboard interface. The app dynamically generates financial insights, sentiment summaries demonstrating how agentic AI can power end-to-end financial analysis platforms.

This deployment step highlights the system's scalability and real-world applicability, bridging research experimentation and user-facing AI tools. It also aligns with our iterative improvement approach, extending the project beyond Jupyter-based execution.

🎥 **Project Demonstration Video:**
A walkthrough of the complete workflow and Streamlit integration can be viewed here:
👉 https://www.youtube.com/watch?v=j6aduujvYNE

Conclusion and Future Work

Conclusion

This project successfully demonstrates the power of a multi-agent system for complex financial analysis. By separating the roles of planning, execution, and critique, the system produces a nuanced and objective report that dynamically adapts its research strategy based on the company's profile. The successful implementation of memory allows the agent to build context over time, making it a truly "learning" system. The final output for the test ticker (e.g., UAA) showcases the end-to-end workflow, from initial data gathering to a polished, refined analysis.

Future Work

While the current system is robust, there are several avenues for future enhancement:

1. **Human-in-the-Loop Feedback:** The plan generated by the `planner_agent` could be presented to a human user for approval or modification before being passed to the executor. This would add a layer of expert oversight.
2. **Deeper Financial Analysis Tools:** New tools could be developed to parse financial statements (10-K/10-Q) in more detail, extracting specific tables and calculating trends in metrics like revenue, net income, and free cash flow over multiple quarters.
3. **Visual Data Integration:** The final report could be enhanced by programmatically generating and including visual charts for financial trends or stock performance directly within the analysis.
4. **"Analyst Verdict" Scorecard:** A final agent could be added to the workflow to score the company on key factors (e.g., Valuation, Financial Health, Momentum) and produce a quick, "at-a-glance" scorecard, providing an even more digestible summary for the end-user.

AI Assistance Disclosure

This project follows academic integrity principles by ensuring that all code and analysis reflect a solid grasp of the concepts. AI tools, such as ChatGPT and GitHub Copilot, were utilized for:

Code Formatting: Applying PEP 8 standards for improved readability and consistency.

Commenting: Improving code documentation with straightforward explanations.

Debugging Assistance: Offering guidance on fixing syntax or logical errors.

To ensure alignment with course objectives, all inputs derived from AI tools underwent a rigorous process of human review, modification, and incorporation.