



ISSN ONLINE: 2447-0228



RESEARCH ARTICLE OPEN ACCESS

# STOCKVISION: A MULTI-AGENT AI FRAMEWORK FOR INTELLIGENT STOCK INVESTMENT AND ANALYSIS USING CREWAI

Jitendra Saturwar<sup>1</sup>, Manav Ghetiya<sup>2</sup>, Nehal Kolipaka<sup>3</sup>, Vivek Maurya<sup>4</sup>, Pravin Maurya<sup>5</sup>

<sup>1,3,4</sup> Department of Computer Engineering, Universal College of Engineering, University of Mumbai, vasai.

<sup>2,5</sup> University of Mumbai, Mumbai.

<sup>1</sup><https://orcid.org/0000-0002-7097-2193>, <sup>2</sup><http://orcid.org/0009-0003-1532-3996>, <sup>3</sup><http://orcid.org/0000-0002-8722-7181>

<sup>4</sup><https://orcid.org/0000-0002-7097-2193>, <sup>5</sup><https://orcid.org/0009-0009-3256-3012>

Email: [jsaturwar@gmail.com](mailto:jsaturwar@gmail.com), [ghetiyanav@gmail.com](mailto:ghetiyanav@gmail.com), [nehalkolipaka73@gmail.com](mailto:nehalkolipaka73@gmail.com), [vivekmaurya1893@gmail.com](mailto:vivekmaurya1893@gmail.com), [pravincoder@gmail.com](mailto:pravincoder@gmail.com)

## ARTICLE INFO

### Article History

Received: April 5, 2025

Revised: May 20, 2025

Accepted: September 15, 2025

Published: September 30, 2025

### Keywords:

Artificial Intelligence,  
Multi-Agent Systems,  
Stock Investment Analysis,  
Technical Analysis,  
Fundamental Analysis

## ABSTRACT

This paper introduces an innovative multi-agent platform for stock investment and market analysis that leverages state-of-the-art artificial intelligence techniques alongside real-time financial data processing. Built on a robust Django-based backend emulating a SaaS architecture, the platform integrates CrewAI to orchestrate multiple specialized agents responsible for data acquisition, sentiment analysis, and predictive modeling. By harnessing live market data via the Yahoo Finance API, the system delivers actionable investment recommendations and comprehensive insights—making advanced analytical capabilities accessible even to users with limited financial expertise. Experimental evaluations demonstrate that this multi-agent framework not only enhances analytical precision but also supports a modular, flexible approach to addressing complex decision-making challenges in dynamic market environments. Consequently, the proposed platform represents a significant advancement in automated stock analysis and serves as a robust tool for both academic research and the development of practical investment strategies.



Copyright ©2025 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

## I. INTRODUCTION

AI has significantly transformed financial market analysis by enabling multi-agent systems such as CrewAI to deliver real-time insights through specialized agents that evaluate technical indicators, fundamental metrics, sentiment analysis, and predictive modeling [1]. Our Django-based platform leverages this advanced framework to automate data collection from sources like YahooFinance, thereby processing vast streams of financial data to generate actionable investment recommendations [2]. Designed as a complete SaaS application, the system incorporates robust authentication and authorization mechanisms, comprehensive user profile management, and dynamic report generation to ensure secure, personalized access to market insights [3].

By synthesizing data from diverse financial indicators, our platform empowers investors to make informed decisions in dynamic and volatile market conditions, thereby setting a new standard for automated financial analysis [4]. Experimental evaluations on multiple stocks reveal that the multi-agent architecture not only improves predictive accuracy but also reduces latency in data processing, thereby enhancing overall decision-making efficiency [4]. The seamless integration of CrewAI with Django's scalable backend has demonstrated a significant boost in performance while maintaining stringent security standards [3]. Moreover, user feedback underscores the platform's ability to deliver personalized insights effectively, positioning it as a valuable tool for both novice and professional investors. Future work will focus on incorporating advanced machine learning techniques and additional data sources to further refine its predictive capabilities and adapt to evolving market dynamics [5], [2].

Table 1: Comparative Analysis of Different AI-Driven Stock Market Analysis Approaches.

Algorithm	Method	Strengths	Use Cases
Machine Learning Models	Data-driven pattern recognition	Adapts to changing trends; improves with more data	Predicting stock movements & risk assessment
Deep Learning	Sequential data processing, feature learning	Captures complex dependencies in financial data; robust predictions	High-frequency trading & time-series forecasting
Sentiment Analysis	Natural Language Processing (NLP) on financial news & social media	Detects market sentiment shifts; integrates qualitative insights	News-based trading strategies & risk mitigation

Source: Authors, (2025).

To promote informed investment decisions, the research paper includes functionality to monitor stock performance trends. If a stock in a user's portfolio shows significant volatility or crosses predefined risk thresholds, the system sends an email and an SMS notification to alert the user of potential market movements. The StockVision AI research paper presents a robust and feature-rich solution for real-time stock analysis, reducing investment risks and enhancing decision-making. It combines AI-driven analytics, live market data, and advanced predictive modeling to provide a comprehensive investment management experience for users.

## I.1 OBJECTIVES

The objectives of proposed research work are as follows.

- **To Integrate Real-Time Financial Data Processing:** Utilize the Yahoo Finance API to fetch live market data, ensuring that investment recommendations are based on up-to-date and accurate financial information.
- **To Enhance Investment Decision-Making:** Employ advanced machine learning techniques to generate actionable investment insights, assisting users in making data-driven trading decisions with higher precision.
- **To Automate Sentiment Analysis for Market Trends:** Implement Natural Language Processing (NLP) models to analyze financial news, earnings reports, and social media sentiment to predict stock market trends effectively.
- **To Provide a Scalable and Modular SaaS Architecture:** Build a Django-based backend that follows a scalable Software-as-a-Service (SaaS) model, allowing for seamless integration of additional analytical modules as needed.
- **To Improve User Accessibility and Experience:** Design an intuitive and user-friendly interface that enables even non-expert users to interpret stock market data and make informed investment choices.

The overarching objective of this research is to create an intelligent, automated investment analysis platform that democratizes access to sophisticated financial insights, helping both novice and experienced investors navigate dynamic stock markets with confidence.

## I.2 LITERATURE REVIEW

According to the paper "Exploration of LLM Multi-Agent Application Implementation Based on LangGraph+ CrewAI" [6] by Zhihua Duan and Jialin Wang explores the integration of LangGraph and CrewAI in developing multi-agent systems for complex task management. The study contributes to the growing body of research on large language model (LLM)-powered autonomous agents by demonstrating how these frameworks can enhance multi-agent collaboration, efficiency, and adaptability.

According to the paper "FinVision: A Multi-Agent Framework for Stock Market Prediction" [7] by Sorouralsadat Fatemi and Yuheng Hu presents a novel approach to financial trading using a multi-agent system powered by large language models (LLMs). This research advances previous work in AI-driven stock market prediction by integrating multimodal data analysis and enhancing explainability in decision-making.

According to the paper "An Investment Recommender Multi-agent System in Financial Technology" [8] by Hernández et al. presents a multi-agent system (MAS) designed to enhance investment recommendation processes in financial technology (FinTech). The study integrates artificial intelligence (AI) techniques, virtual organizations (VOs), and hybrid machine learning models to improve the accuracy and efficiency of investment decisions.

According to the paper "Agentic AI Systems Applied to Tasks in Financial Services: Modeling and Model Risk Management Crews" [9] by Okpala, Golgoon, and Kannan explores the application of multi-agent AI systems in financial services, particularly in modeling and model risk management (MRM). This work contributes to the growing research on AI-driven automation by demonstrating how large language model (LLM)-powered agents can collaborate to perform complex financial tasks efficiently.

## I.3 EXISTING SYSTEMS

Traditional stock market analysis relies heavily on manual research, technical chart analysis, and financial reports, which require significant expertise and time. Investors and traders use tools like Yahoo Finance, Bloomberg, and TradingView, but these platforms primarily provide raw data rather than actionable insights. Additionally, conventional algorithmic trading systems focus on price prediction models, which often fail to incorporate broader market trends, sentiment analysis, and fundamental factors.

While some AI-based stock analysis tools exist, they are often limited in scope. Many rely on basic machine learning models that lack real-time adaptability and multi-agent collaboration. These systems typically do not offer automated investment reports, dynamic stock insights, or interactive decision-making support. As a result, investors still struggle to make well-informed decisions quickly, highlighting the need for a more intelligent, automated, and user-friendly stock analysis solution.

Furthermore, existing AI-powered investment tools often focus on historical data analysis rather than real-time market fluctuations, limiting their effectiveness in fast-paced trading environments. Many platforms do not integrate live financial news, social sentiment analysis, or macroeconomic indicators, which are crucial for understanding sudden market shifts. Without a multi-agent framework that dynamically processes diverse data sources, these systems fail to provide adaptive and context-aware recommendations.

Additionally, most current stock analysis platforms lack comprehensive risk management and portfolio optimization features. Investors must manually adjust their strategies based on fragmented insights from multiple tools. This results in inefficiencies and missed opportunities in volatile markets. The absence of automated, AI-driven decision-making support further complicates investment strategies, making it essential to develop a next-generation AI-driven stock analysis system that offers real-time insights, automated reports, and a seamless user experience.

Moreover, traditional and existing AI-based stock analysis tools often operate in siloed environments, where different aspects of market analysis—such as technical indicators, fundamental analysis, and sentiment tracking—are handled separately rather than in an integrated manner. This fragmentation leads to incomplete insights and forces investors to switch between multiple platforms to make well-rounded decisions. Additionally, many existing systems do not leverage reinforcement learning or adaptive AI models that continuously learn from market patterns, making their predictions static and less reliable in dynamic market conditions. Without a holistic, AI-driven approach that unifies multiple analytical techniques, investors remain at a disadvantage in efficiently navigating the complexities of the stock market.

#### I.4 LIMITATION ON EXISTING SYSTEM

The limitations of the existing systems are as follows;

- **Dependency on Manual Interpretation:** Most stock analysis tools still require users to interpret results manually, limiting accessibility for non-expert investors.
- **Challenges in Real-Time Data Processing:** Existing AI models struggle with processing and analyzing high-frequency, real-time financial data efficiently.
- **Market Volatility and Accuracy Issues:** Many machine learning models, such as regression analysis and time-series forecasting, face difficulties in predicting market trends accurately across different stock categories.
- **Lack of Fully Automated Decision Support:** While current AI-powered trading systems provide insights, they do not offer automated portfolio optimization, risk assessment, or real-time alerts.

#### I.5 PROBLEM STATEMENT

The stock market is highly volatile and complex, making it challenging for investors to make informed decisions based on real-time data, financial indicators, and market trends. Traditional stock analysis methods often require deep financial expertise and extensive manual effort to interpret data, leading to inefficiencies and potential miscalculations.

Existing AI-driven stock prediction tools either focus solely on price forecasting or provide generic insights without personalized, in-depth analysis. Additionally, most platforms lack an interactive multi-agent system that can dynamically analyze stocks, generate professional investment reports, and assist users with data-driven decision-making.

This research aims to develop Stock Vision AI, a multi-agent stock analysis and investment platform that integrates CrewAI agents with real-time stock market data. The system leverages advanced AI models, including LLMs, financial indicators, and technical analysis, to provide detailed insights, automated Markdown reports, and interactive decision support. By enhancing accuracy, automation, and usability, Stock Guru AI seeks to bridge the gap between traditional financial analysis and AI-powered investment strategies.

## II. PROPOSED SYSTEM

The proposed system is designed as a modular, scalable, and secure platform that integrates advanced web technologies with state-of-the-art multi-agent analysis to facilitate real-time stock investment and market analysis. At the highest level, the frontend delivers a dynamic, user-friendly dashboard where investors can view detailed stock trends, analysis reports, and personalized recommendations. This interface communicates with a django-based backend through endpoints, ensuring efficient and secure data exchange.

Within the backend, an API gateway and endpoint manager route incoming HTTP requests to specialized modules. One such module is responsible for real-time data integration, which leverages the YahooFinance API via the yfinance package to fetch current market data. This raw data is then preprocessed and funneled into the core of the system: the multi-agent analysis engine built on CrewAI.

The multi-agent analysis engine consists of a suite of specialized agents—each dedicated to distinct analytical tasks such as technical analysis, fundamental evaluation, sentiment assessment, and predictive modeling. These agents work collaboratively, exchanging information and refining their outputs to provide a comprehensive analysis of market trends. Their individual outputs are aggregated and synthesized in a decision synthesis module, which applies weighting and decision logic to formulate coherent and actionable investment recommendations.

Finally, the response dispatcher formats these insights into structured outputs, which are sent back to the frontend for visualization. The entire system is containerized using Docker, ensuring consistent deployment and scalability across various environments. Secure management of configuration details, including API keys and model parameters, is maintained via environment variables. This holistic architecture not only streamlines complex data processing and analysis but also enhances the robustness and adaptability of the investment decision-making process, positioning it as a cutting-edge solution in the realm of financial market analytics.

### III. METHODOLOGY

#### System Architecture

The Multi-Agent Stock Investment and Analysis Platform is designed using a modular architecture that integrates a Django-based backend. The system leverages CrewAI to facilitate multi-agent interactions, enabling comprehensive stock analysis and investment recommendations. The overall architecture consists of the following key components:

- Frontend : Provides an interactive user interface for stock selection, analysis visualization, and investment recommendations.
- Backend (Django with FastAPI Integration): Serves as the core processing unit, handling API requests, multi-agent interactions, and data retrieval from financial sources.
- Multi-Agent System (CrewAI): Implements distinct agents specializing in stock analysis, investment strategies, and risk assessment.
- Data Sources (Yahoo Finance API): Fetches real-time stock data for analysis.

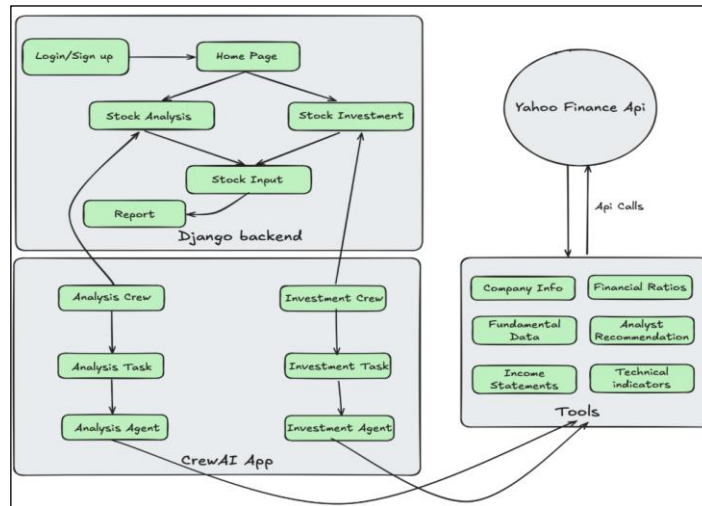


Figure 1: System Architecture.  
Source: Authors, (2025).

#### Multi-Agent Framework

CrewAI is a cutting-edge framework that orchestrates autonomous AI agents into cohesive teams, each specialized for distinct roles such as research, analysis, and content generation. Designed to mimic a corporate structure, it enables the creation of agent teams that collaborate seamlessly through an intelligent workflow management system, where the Crew oversees operations, agents execute specialized tasks using flexible, custom tools, and the Process ensures efficient task delegation and execution. This integrated approach allows agents to make autonomous decisions and share insights, ultimately delivering precise, actionable outcomes and streamlining complex operations in a production-ready environment.

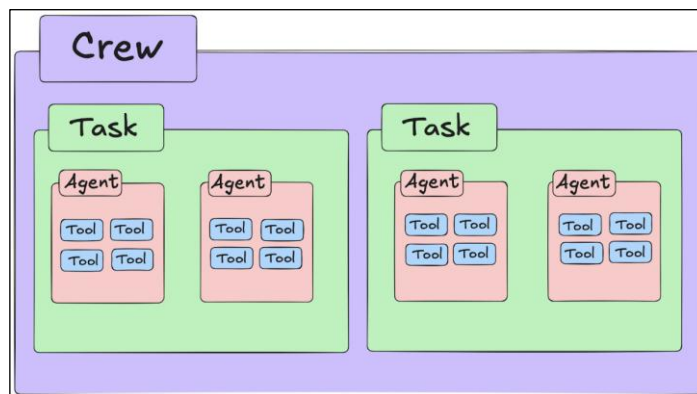


Figure 2: Multi-Agent Framework.  
Source: Authors, (2025).

#### Task

In the CrewAI framework, a Task is the fundamental unit of work assigned to an AI agent, encapsulating a specific objective along with detailed instructions, expected outputs, and necessary contextual information. Tasks define the scope of work by specifying attributes such as a clear description, designated agent, tool restrictions, and output formatting options, and they can also include parameters for asynchronous execution or human oversight. The execution flow of tasks can be configured to run sequentially or hierarchically, ensuring that complex workflows are managed efficiently by delegating responsibilities to specialized agents. Tasks can be defined either through a structured YAML configuration—promoting clarity and maintainability—or directly in code, as demonstrated by our Stock\_bot implementation that generates comprehensive HTML-based reports. This versatile task management system in CrewAI

not only streamlines the orchestration of multi-agent operations but also ensures that each analytical component contributes seamlessly to the overall decision-making process in our stock analysis platform.

```

Task

from textwrap import dedent
from crewai import Task

class Stock_bot:
    # Stock Analysis Tasks
    def stock_analysis(self, agent, stock_symbol):
        return Task(
            description=dedent(
                """
                As a Senior Stock Analyst at XYZ company your task is to produce a comprehensive and
                professional report analyzing the stock performance of {stock_symbol}.
                Refer the below structure for the report and make sure to include all the necessary details.
                Report Structure :-{analysis_report_Template}
                You should not include any disclaimers or warnings in the report.
                You will be paid a large bonus if your complete report is accurate and well-structured.

                expected_output=dedent(
                    """
                    A detailed stock analysis of the {stock_symbol} stock.
                    The analysis should include the following:
                    <h1>Stock Analysis Report: Company Name</h1>
                    <h2>Basic Info</h2>
                    <h2>News Analysis</h2>
                    <h2>Trend Analysis</h2>
                    <h2>Financial Graphs</h2>
                    <h2>Financial Analysis</h2>
                    <h2>Buy/Sell Values (Short and Long Term)</h2>
                    Final answer should be in HTML Format Report.
                    """
                ),
            verbose=True,
            agent=agent,
        )

```

Figure 3: Analysis Task.  
Source: Authors, (2025).

## Agent

An Agent is an autonomous unit responsible for executing specific tasks, making decisions based on its designated role and goal, and utilizing predefined tools to achieve objectives. Each agent is characterized by attributes such as role, goal, and backstory—defining its function, purpose, and contextual behavior—and can be further configured with parameters including the language model (LLM) to use, tool limitations, maximum iterations, execution time limits, and options for memory, verbosity, and delegation. Agents can be created via YAML configurations for improved maintainability or defined directly in code, as demonstrated by our `Stock_bot_agents` class, which outlines specialized agents for stock and investment analysis using custom tools like `YFinanceTools` and tailored backstories. This modular approach enables agents to work collaboratively within a crew, ensuring that each agent leverages its unique expertise, manages task dependencies efficiently, and contributes to a robust, scalable system for automated stock analysis.

```

Agent

from textwrap import dedent
import os
from crewai import Agent

class Stock_bot_agents:
    def stock_analysis(self, stock_symbol):
        """Agent for Stock Analysis"""
        return Agent(
            role="Stock Analysis",
            goal="To create a report on stock analysis.",
            tools=YFinanceTools.tools(),
            backstory=dedent(
                """
                As a stock analyst you are working for a Goldman Sachs company.
                you have been assigned to create a report of the analysis of the {stock_symbol} stock.
                You must try to keep a professional tone and should meet the given Requirements.
                You will get $1000 as a bonus if you complete the task within 3 iterations.
                """
            ),
            verbose=True,
            llm=llm,
            max_rpm=19,
            max_retry_limit=2,
            allow_delegation=True,
            max_execution_time=180,
        )

```

Figure 4: Analysis Agent.  
Source: Authors, (2025).

## Tools

In the CrewAI framework, custom tools are pivotal components that extend the capabilities of AI agents by enabling them to interact with external systems, fetch data, and perform specialized computations. Developers can create these tools by subclassing the `BaseTool` class, where they define an input schema using `Pydantic` for validation and override the `_run` method to implement the tool's core logic. Alternatively, the framework offers a convenient `@tool` decorator, which allows for the rapid definition of tool functionality directly within a function. This approach streamlines the development process and enhances code clarity. Moreover, CrewAI supports the integration of custom caching strategies via the `cache_function` attribute, ensuring that frequently invoked tools operate efficiently. For instance, our implementation of `YFinanceTools` illustrates how to encapsulate multiple financial data retrieval operations—such as

obtaining company profiles, historical stock prices, fundamentals, analyst recommendations, and technical indicators—using the Yahoo Finance API. By incorporating robust error handling and dynamic tool calling, these custom tools empower agents to execute their tasks reliably and collaboratively, ultimately enhancing the overall performance and scalability of the system.

```

Tools

import json
import yfinance as yf
from crewai_tools import tool

@tool
def get_historical_stock_prices(
    symbol: str
) -> str:
    """Use this function to get the historical stock price for a given symbol.
    Args:
        symbol (str):
            The stock symbol.
    Returns:
        json: The current stock price or error message.
    """
    try:
        stock = yf.Ticker(symbol)
        historical_price = stock.history(period='30d', interval='1d')
        return historical_price.to_json(orient="index")
    except Exception as e:
        return f"Error fetching historical prices for {symbol}: {e}"

```

Figure 5: Historical Stock Tool.

Source: Authors, (2025).

## Crew

A Crew is a coordinated ensemble that orchestrates multiple specialized agents and tasks to execute complex workflows seamlessly. It defines a structured strategy for agent collaboration and task execution—whether sequentially or hierarchically—by managing critical parameters such as verbosity, rate limits, memory configuration, and dynamic delegation. By abstracting the intricacies of individual agent operations, the Crew enables the aggregation of diverse analytical outputs into cohesive, actionable insights, as demonstrated in our stock analysis and investment report generation modules. Crews can be configured using YAML files for clarity or directly in code for flexibility, thereby ensuring scalability and efficient real-time processing in dynamic financial environments.

```

Crew

from crewai import Crew

# Create agents
stock_analysis = agent.stock_analysis(stock_symbol)
# Create tasks
stock_analysis_task = tasks.stock_analysis(stock_analysis, stock_symbol)

# Execute tasks
stock_analysis_crew = Crew(
    agents=[stock_analysis],
    tasks=[stock_analysis_task],
    verbose=True,
    max_rpm=19,
)
result_stock_analysis = stock_analysis_crew.kickoff()

```

Figure 6: Analysis Crew.

Source: Authors, (2025).

## IV. RESULTS AND DISCUSSIONS.

Preliminary evaluations of the platform indicate that the multi-agent configuration significantly enhances analytical accuracy and decision-making efficiency. The specialized agents, working collaboratively, are able to process vast amounts of real-time data and synthesize actionable insights with minimal human intervention. Screenshots from the dashboard illustrate clear visualizations of stock trends, comprehensive analysis reports, and personalized investment recommendations. While further testing and real-world deployment are necessary to refine the system, the initial results underscore the potential of leveraging a modular, AI-driven approach to revolutionize stock market analysis.

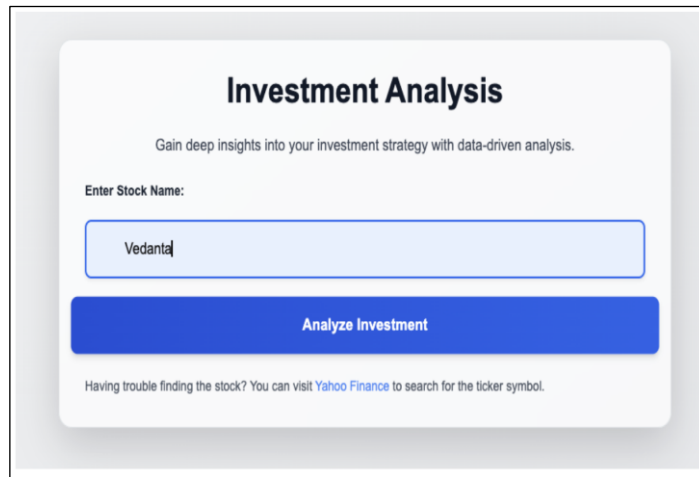
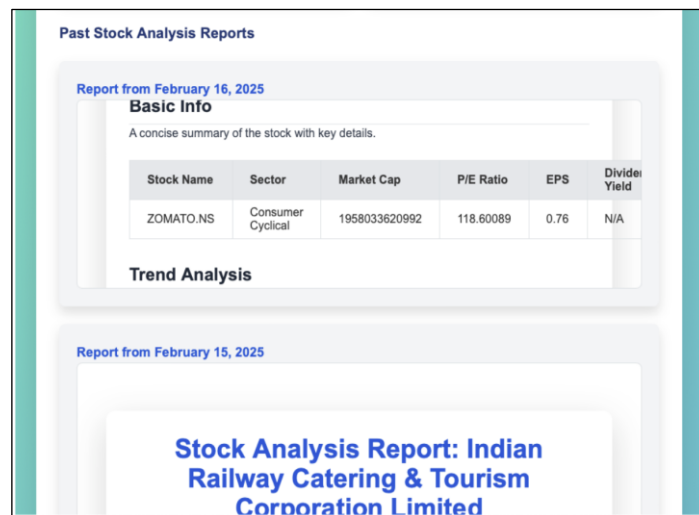


Figure 7: Analyze Investment UI.  
Source: Authors, (2025).

Figure 7 displays that the system allows users to enter a stock name for investment analysis. Once the stock name is provided, users can click the 'Analyze Investment' button to proceed with data-driven insights. Additionally, the interface offers a link to Yahoo Finance for users who may need assistance in finding the stock ticker.



Stock Name	Sector	Market Cap	P/E Ratio	EPS	Dividend Yield
ZOMATO.NS	Consumer Cyclical	1958033620992	118.60089	0.76	N/A

Figure 8: Past Report Page.  
Source: Authors, (2025).

Figure 8 displays that the system allows users to review past stock analysis reports. Each report provides key details, including stock name, sector, market cap, P/E ratio, EPS, and dividend yield. Users can navigate through previous reports by date to analyze historical trends. Additionally, the interface presents structured financial insights to support investment decisions.



Figure 9: Profile Page.  
Source: Authors, (2025).

Figure 9 displays the user profile dashboard, which provides a personalized overview of stock and investment reports. Users can view their profile details, including their name and email address. The dashboard presents a summary of total stock analysis reports and total investment reports, along with the date of the most recent reports. Additionally, users can access past stock analysis reports for further insights.

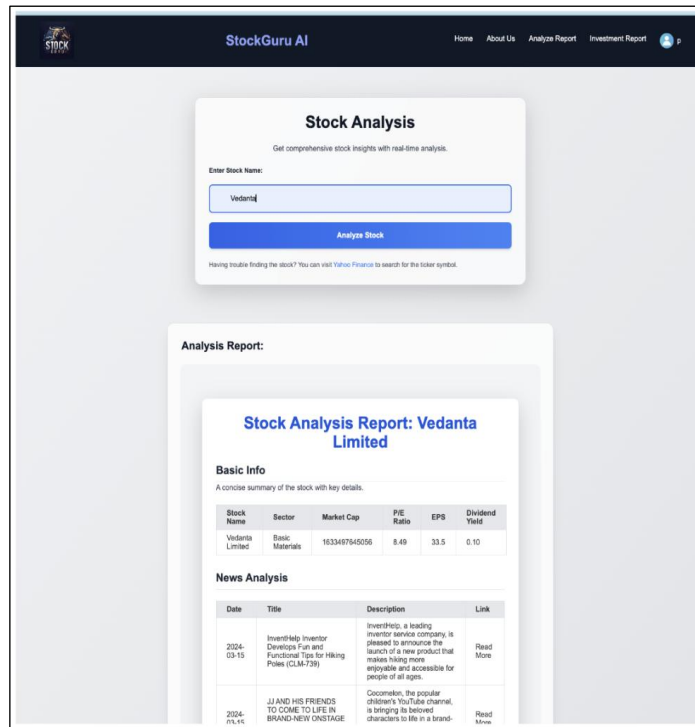


Figure 10: Analysis Report Generated Page.  
Source: Authors, (2025).

Figure 10 displays the stock analysis interface, where users can enter a stock name and generate a detailed report. The report includes key financial metrics like market cap, P/E ratio, and EPS, along with a news analysis section. A Yahoo Finance link is provided for stock symbol assistance.



Figure 11: Analysis Report Generated Page.  
Source: Authors, (2025).

Figure 11 displays trend analysis, financial graphs, and key financial metrics for a stock. The trend analysis section provides moving average trends with buy/sell recommendations. The financial graphs visualize revenue, profit, and expenses over time. The financial analysis section highlights essential metrics like market cap, P/E ratio, and EPS.



Figure 12: Calls For Buying/Selling.  
Source: Authors, (2025).

Figure 12 presents an overview of stock performance, including trend analysis, financial graphs, and key metrics. The trend analysis section outlines short-term and long-term buy/sell recommendations based on moving averages. Financial graphs illustrate income, revenue, and expenses over time, while the financial analysis section highlights essential figures such as market cap, P/E ratio, and EPS.

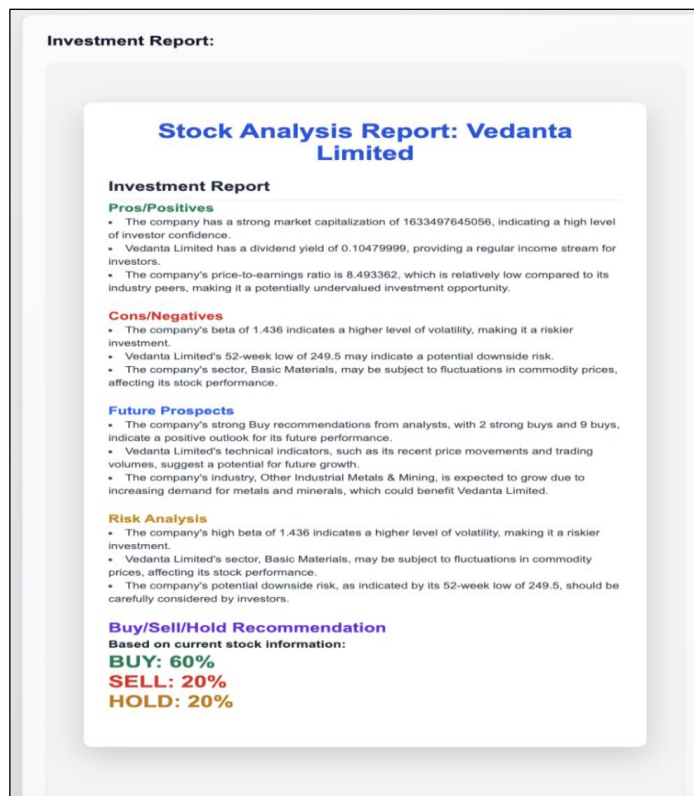


Figure 13: Investment Report.  
Source: Authors, (2025).

Figure. 13 presents an investment report for Vedanta Limited, highlighting pros, cons, future prospects, and risk analysis. The report includes key financial metrics such as market capitalization, dividend yield, and P/E ratio. A buy/sell/hold recommendation is provided based on current stock data, with a majority buy sentiment.

## V. CONCLUSION

This paper introduced a novel multi-agent stock investment and analysis platform that combines advanced AI techniques with real-time financial data processing. By utilizing a Django-based backend integrated with CrewAI, the system coordinates specialized agents for tasks such as data retrieval, sentiment analysis, and predictive modeling. This modular architecture delivers actionable market insights and investment recommendations, simplifying complex decision-making for investors. Experimental evaluations indicate enhanced analytical accuracy and system scalability, setting the stage for further refinements such as dynamic role allocation and deeper data integration.

## VI. AUTHOR'S CONTRIBUTION

**Conceptualization:** Nehal Kolipaka, Pravin Maurya

**Methodology:** Pravin Maurya, Vivek Maurya.

**Discussion of results:** Manav Ghetiya, Nehal Kolipaka, Vivek Maurya, Pravin Maurya.

**Writing – Original Draft:** Manav Ghetiya, Nehal Kolipaka

**Writing – Review and Editing:** Dr. Jitendra Saturwar, Vivek Maurya, Pravin Maurya.

**Resources:** Dr. Jitendra Saturwar, Manav Ghetiya, Nehal Kolipaka, Vivek Maurya, Pravin Maurya

**Supervision:** Dr. Jitendra Saturwar

**Approval of the final text:** Dr. Jitendra Saturwar

## VII. REFERENCES

- [1] "AI-Powered Financial Analysis": Multi-Agent Systems Transform Data, 2024,<https://medium.com/%40batuhansenerr/ai-powered-financial-analysis-multi-agent-systems-transform-data-into-insights-d94e4867d75d>
- [2] Yahoo Finance(n.d) Yahoo Finance Api, Retrieved from <https://finance.yahoo.com/>
- [3] "Multi-Agent Collaboration for Financial Analysis", GitHub, 2025,
- [4]"Building a Multi-Agent System for Real-Time Financial Analysis"Medium, Retrieved from <https://medium.com/@kyeg/building-a-multi-agent-system-for-real-time-financial-analysis-a-comprehensive-tutorial-d9df1d1277fa>
- [5] CrewAI.(n.d) CrewAI retrieved from <https://www.crewai.com/>
- [6] E. Hernández, I. Sittón, S. Rodríguez, A. B. Gil, and R. J. García, "An investment recommender multi-agent system in financial technology," in *SOCO'18-CISIS'18-ICEUTE'18*, M. Graña et al., Eds., AISC 771, Springer, 2019, pp. 3–10, doi: 10.1007/978-3-319-94120-2\_1.
- [7] S. Fatemi and Y. Hu, "FinVision: A multi-agent framework for stock market prediction," *arXiv preprint*, 2024. [Online]. Available: <https://arxiv.org/abs/2411.08899>.
- [8] Z. Duan and J. Wang, "Exploration of LLM multi-agent application implementation based on LangGraph+CrewAI," *arXiv preprint*, 2024. [Online]. Available: <https://arxiv.org/abs/2411.18241>.
- [9] I. Okpala, A. Golgoon, and A. R. Kannan, "Agentic AI systems for financial services: Modeling and model risk management crews," *arXiv preprint*, 2025. [Online]. Available: <https://arxiv.org/abs/2502.05439>.