**AAI-520 – Team Project Status Update Form**

**Team Number**: Gangadhar Singh Shiva, Akshobhya Rao BV, Nagarajan Mahalingam

**Team Leader/Representative:** Gangadhar Singh Shiva,

**Full Names of Team Members:**

1. Gangadhar Singh Shiva

2. Nagarajan Mahalingam

3. Akshobhya Rao BV

## Project Title

**Agentic AI for Investment Market Research (MCP Client–Server Framework with Yahoo Finance + LLM)**

**Short Description of Your Project and Objectives:**

This project proposes the design and implementation of an **autonomous investment research agent** that leverages **agentic AI workflows, Yahoo Finance data, and LLM-driven reasoning** to support informed decision-making in financial markets. The system will be built on a **modular MCP (Model Context Protocol) client–server framework**, providing extensibility, transparency, and adaptability. Core capabilities include stock data retrieval, news ingestion with sentiment analysis, dynamic task routing, prompt chaining, evaluator–optimizer loops, and persistent memory. This project contributes to advancing responsible and explainable AI applications in financial research.

## Introduction

The exponential growth of market data necessitates automated research agents that can ingest, process, analyze, and summarize financial information. Traditional financial research pipelines often require human analysts to retrieve stock data, monitor news, assess sentiment, and refine insights iteratively. This project introduces an Agentic AI system designed to automate these workflows using LLMs, structured tool APIs, and self-reflective learning mechanisms.

By adopting an MCP client–server design, the project ensures modularity:

* The MCP server provides financial tools (stock data retrieval via Yahoo Finance, news ingestion, and sentiment analysis).
* The MCP client hosts the agentic AI agent, orchestrating research tasks using LLM reasoning, planning, dynamic routing, and memory.

## Goals

1. Build a fully functional autonomous investment research agent.
2. Integrate financial data sources (Yahoo Finance, news feeds) for real-time market insights.
3. Implement agentic AI workflow patterns (prompt chaining, routing, evaluator–optimizer reflection).
4. Enable memory-driven learning across multiple runs for improved performance.
5. Deliver a transparent, modular MCP architecture to allow extensibility.

Are you using and practicing GitHub as a code hosting platform for version control and collaboration? If yes, provide the link here: https://github.com/gshiva1975/AAI-520

How many times have your members met in the last two weeks? 2 Times

## Project Objectives

### Build an Autonomous Research Agent

Enable planning, routing, prompt chaining, evaluation, and memory.

Automate financial research tasks with minimal human input.

### Implement Workflow Patterns

* **Prompt Chaining:** Ingest → Preprocess → Classify → Extract → Summarize.
* **Routing:** Dynamically route stock vs. news queries.  
  **Evaluator–Optimizer:** Generate → Evaluate → Refine insights.

### Integrate LLM + Memory

* Use LLMs for reasoning, summarization, and reflection.
* Maintain a persistent memory store (agent\_memory.json) to recall and improve across runs.

### MCP Architecture

* Build MCP Server (mcp\_server.py) with financial tools.
* Build MCP Client (mcp\_client.py) with the agentic AI orchestrator.
* Enable easy extension for additional datasets/APIs (e.g., SEC filings, macroeconomic indicators).

## System Design

### Architectural Overview

1. **MCP Server**

* Provides tools: Stock Data (Yahoo Finance), News + Sentiment.
* Uses asyncio for request handling.

1. **MCP Client (Agent)**

* LLM-powered planning + reasoning.
* Routes queries to the correct MCP tool.
* Performs evaluator–optimizer self-reflection.
* Maintains persistent memory.

1. **Memory Store**

* JSON-based persistent storage.
* Keeps logs of prior research outputs and insights.
* Enhances future decision-making.

1. **LLM Integration**

* llm\_utils.py provides structured prompts:
* Planning Prompt (decide research steps).
* Summarization Prompt (condense results).
* Reflection Prompt (evaluate and refine).

## Work Breakdown Structure (WBS)

**Level 1: Project Execution**

1. **Project Planning**
   * Define scope, objectives, deliverables.
   * Repo setup, environment config.
2. **MCP Server Development**
   * Stock Data Tool → integrate Yahoo Finance.
   * News Analysis Tool → sentiment + preprocessing.
   * Server orchestration

* **MCP Client (Agent) Development**
  + Core agent loop (planning, execution).
  + Prompt chaining: Ingest → Preprocess → Summarize.
  + Routing logic: stock vs. news analyzer.
  + Evaluator–Optimizer: reflection + refinement.

1. **LLM Integration**
   * Planning prompts.
   * Summarization prompts.
   * Reflection prompts.
2. **Memory Module**
   * JSON persistence.
   * Recall past insights.
   * Update memory across runs.
3. **Testing & Validation**
   * Validate LLM summaries
   * Multi-run simulations with memory.
4. **Documentation & Delivery**
   * Setup & usage guide.
   * Demo run examples.
   * Final report + architecture diagram.

List the specific contributions that each team member is providing for the Final Team Project in the table below.

* **NOTE:** ALL students on the team should contribute equally to the Final Team Project.

| Gangadhar Singh Shiva | Nagarajan Mahalingam | Akshobhya Rao BV |
| --- | --- | --- |
| Create Github Repo  Data Collection & Cleaning | Data Collection & Cleaning | Data Collection & Cleaning |
| Exploratory Data Analysis | Exploratory Data Analysis | Exploratory Data Analysis |
| MLP Server Implementation (Routing, NLP Query Processing, Performance Measurement) | Integration of Agentic AI, MCP Server, MCP Client | MCP Client (Routing, NLP Query Processing, Performance Measurement) |
| Model Optimization | Model Optimization | Model Optimization |
| Integration with Claude UI/ Claude MCP Client | Integration with Claude MCP Server | Integration with Claude MCP Server |
| Model Training & Testing, Documentation | Model Training & Testing, Documentation | Model Training & Testing, Documentation |

**Comments/ Roadblocks:**

This project balances AI innovation (LLM-driven reflection, memory) with practical utility (stock research).

The MCP framework ensures modularity, making the system adaptable for future financial APIs (Bloomberg, SEC filings, etc.).

Risks include LLM hallucination and data API limitations; mitigations include evaluator–optimizer loops, bias monitoring, and caching mechanisms.

Deliverables will include a working demo, source code, and documentation for reproducibility.