AI-Powered Earnings Call Analysis System Documentation

Why this stack, Step by step guide, challenges faced, thought process, why this prompt.

# Table of Contents

1. Project Overview

2. System Architecture

3. Features

4. File Structure

5. Installation & Setup

6. Usage Guide

7. Technical Implementation

8. Challenges & Solutions

9. Future Improvements

# Project Overview

This project implements an advanced multi-agent AI system for analyzing earnings call transcripts and generating comprehensive investment reports. The system utilizes AutoGen's multi-agent framework to create a collaborative environment where different AI agents work together to produce accurate, detailed, and actionable investment insights.

# System Architecture

The system employs a multi-agent architecture with four specialized agents:

1. Client Agent: Acts as the project manager and quality controller

2. Writer Agent: Responsible for initial report drafting based on transcript analysis

3. Analyst Agent: Performs fact-checking and financial calculations using external data

4. Editor Agent: Ensures report quality, clarity, and completeness

Agent Workflow:

Client Agent -> Writer Agent -> Analyst Agent -> Editor Agent -> Writer Agent -> Client Agent

# Features

1. Automated Financial Analysis

2. Intelligent Report Generation

3. Data Validation & Quality Control

4. Flexible Input Processing

# File Structure

FYP\_Crewai/

├── app.py # Streamlit web application

├── autogenAI.py # Core AutoGen implementation

├── crewAI.py # CrewAI implementation

├── requirements.txt # Project dependencies

├── tools/

│ ├── research\_tools.py # Financial data retrieval tools

│ └── templates/ # Agent prompt templates

├── Earnings2Insights/ # Transcript storage

│ └── ECTsum/ # Organized by company and quarter

└── saved\_reports/ # Generated report storage

# Installation & Setup

1. Environment Setup

- Clone the repository

- Create and activate virtual environment

- Install dependencies

2. API Configuration

- Create .env file with API keys

3. Running the Application

- streamlit run app.py

# Usage Guide

1. Start the application using Streamlit

2. Upload transcript and select model

3. View generated structured report and access saved version

# Technical Implementation

1. Agent System using AutoGen

2. Financial Data Processing with Alpha Vantage API

3. Markdown-based Report Generation

4. Dynamic Content and Ratio Calculations

# Challenges & Solutions

1. Prompt Engineering

2. Financial Data Accuracy

3. Fiscal Year Handling

4. Tool Integration

5. Agent Communication

6. Data Consistency

7. Market Sentiment

# Future Improvements

1. Enhanced Data Sources

2. Advanced Analysis

3. User Interface Enhancements

4. Agent Capability Expansion

5. Performance Optimization