ACT (Agentic Corporate Trader)

Over the past two decades, the internet has revolutionised all aspects of commerce. In particular, the world of stock market investment has been transformed. Retail investors and companies alike are no longer tied to expensive brokers to manage their portfolios and they can now choose to invest directly at a fraction of the cost. Furthermore a recent surge in AI technology has spawned a paradigm shift where investors don't even need the advice of financial consultants and may now choose to let automated robotic applications make investment decisions on their behalf. This technological convergence has created an opportunity for your firm to develop applications to address a gap in the market for the management of corporate treasuries. Your company will develop a bespoke ***prototype*** system which enables governments and large corporates to manage their funds by investing directly in technology stocks and crypto assets through agentic robotic automation. Some companies may still want a brokerage firm to manage their assets so there will be an option within this system to invest on behalf of other companies.

In addition to a Systems Administrator, there are two types of user:

- Fund Administrators (users who only manages their own assets directly i.e. governments or large corporates)
- Fund Managers (users who use the application to manage the assets of multiple companies i.e. large brokerages)

The system consists of the following key features:

FEATURES:

To develop the ACT system, several applications are required:

- The ACT-Mobile app enables a Fund Administrator to purchase up to 10 technology stocks and 3 crypto assets.
- For Fund Managers, a portal website, ACT-Web, stores all client records. This website will hold the details of each client company who the fund manager acts for. It will also allow for the same functionality as ACT-Mobile so the fund manager can manage the portfolio for each client.
- Agentic AI will harness the power of a variety of LLMs to create buy, sell and hold signals on a basket of stocks. The backend AI (i.e. ACT-AI Engine) would most likely be developed as a Flask app. Initially the app could work on localhost but ultimately it should be cloud based to facilitate integration with other components of the project.

ACT-Mobile app basic features should include:

- **Register/login of the client**: email and social media login option of the fund administrator
- **Payment of ACT-AI premium (if Agentic AI is being used used)**
- **Purchase assets:** ability to select a portfolio of 10 technology shares and 3 crypto assets
- **View financial information**: view relevant financial for chosen shares and crypto assets
- **Price Alerts**: notifications for large movements in prices of chosen assets
- **ACT-AI Engine Interface** view predicted future prices of 10 technology shares and 3 crypto assets
- **Request help** ability to call a support number, from within the app
- **Support** forms to contact support
- **Ratings & reviews.** capability of rating the app and leaving reviews of the app's performance

ACT-Web basic features should include:

- **Registration / Login** of fund managers using email and social media login
- **Record details of the fund manager** i.e. who is dealing with the fund manager
- **Access Restrictions**: fund managers should only be able to see the details of their own clients
- **ACT-AI Engine** allows fund managers to view predicted future prices of 10 technology shares and 3 crypto assets for each of their clients
- **Reports:** fund managers may run financial reports on their chosen technology and crypto set
- **Price Alerts:** notifications for large movements in prices of chosen assets

ACT-AI Engine basic features should include:

- Simple API linking ACT AI Engine backend with the client ACT-Mobile app and ACT-Web applications.
- Using either CrewAI or AutoGen, develop an agentic AI application that automatically makes buy and sell recommendations of the basket of stocks identified by the fund manager or corporate client.
- There will be 4 agents within the application:
    - Researcher (to research a particular stock)
    - Accountant (to calculate various accounting ratios)
    - Recommender (to make a buy, sell or hold decision)
    - Blogger (to format the output nicely)
- In programming these agents, you will have to decide on the task completed by each agent. To assist in this regard you may use ask other LLMs such as Claude AI to find out what should be incorporated within the task e.g. with the Accountant agent, what ratios should you be included e.g. liquidity ratios, profitability ratios, growth metrics etc.
- Within at least **one** of these agents, you must incorporate an external tool which helps the agent perform its work. Possible agents could include the WebsiteSearchTool or the YahooFinanceNewsTool which are very easily incorporated with simple crewAI applications.
- In order to introduce diversity into the application, it will integrate with 2 LLM APIs namely: ChatGPT, Grog (not to be confused with Elon Musk's Grok) and 2 locally-hosted models of your choosing. To locally host a model, it is recommended to use the Ollama application.

Each project group should also incorporate a custom feature which is unique to the group. This custom feature should be implemented in one (or more) of the key components of the project i.e. the ACT-Mobile, ACT-Web or ACT-AI Engine.

Technology:

The first application, the ACT-Mobile app, involves the development of an Android app to interface with the ACT-AI engine. Development may be carried out using Android studio or a similar development environment.

The second application, the website ACT-Web, involves developing a web site for fund managers. ACT-Web may be developed using React, HTML/JavaScript or another language of your choice. Both ACT-Mobile and ACT-Web will store their data in a centralised database such as Firebase. To facilitate rapid development of the site, a Firebase Web App may be developed. In-app payments via credit cards may use PayPal or Strip and you may prototype this functionality using their Sandbox environment.

The agentic AI application will be developed using CrewAI, AutoGen or similar AI framework.

Github must be used for code management as must an appropriate project management tool such as Jira. Marks will be allocated for their use.

References:

It's important to go through these resources since some of this project involves self-directed learning. The links below outline resources which should help with AI components of the project.

1. CrewAI Tutorial - Next Generation AI Agent Teams (Fully Local):

   https://youtu.be/tnejrr-0a94?si=yAk1zozP7CSlz7a8

2. The RIGHT WAY To Build AI Agents with CrewAI (BONUS: 100% Local):

   https://youtu.be/iJjSjmZnNlI?si=zZ7P8zy53lDCbj5L

3. Groq LLM: https:// groq.com/ and https://console.groq.com/

4. Using Ollama To Build a FULLY LOCAL "ChatGPT Clone"

   https://youtu.be/rIRkxZSn-A8?si=F_ywCM-QwfvgAvBc

5. The following uses a "tooling solution" called Composio (you don't necessarily need this tool but the tutorial is useful in that it showcases how agents in AI can be used:

   https://composio.dev/blog/ai-investment-analyst/

6. Using LangChain and CrewAI:

   https://blog.langchain.dev/crewai-unleashed-future-of-ai-agent-teams/