# FinHelper

if we just run the llm in our local machine, this would be really low speed. In order to make the application more user friendly, we move all the llm in the remote machine.

In this project, we rent the GPU machine in autodl (A100,cuda 12.1), when there is need to use the llm, we would send the question to the remote machine, and then remote machine would do the inference work. The result would then send to local machine. The connection is established using ssh.

## Setup

### Move file to Remote server

- **Step 1**: move the `RAG/RAG_main` folder to the remote machine.
- **Step 2**: move all files under `remote_server` to remote machine

## Preparation in Remote Machine

- **Step 1** : follow the instruction in `RAG/RAG_main/readme.md`
- **Step 2**: download our finetune model

```
pip install -U huggingface_hub

huggingface-cli download liukarlie/forecast1 model_conv.gguf
```

- **Step 3**: load in the ollama, use the `ModelFile` **(remember to change the model path to your model located address in the `ModelFile`)**

```
ollama serve
ollama create example -f Modelfile
```

- **Step 4** : follow the below instruction

```
pip install litellm flask
```

- **Step 5** : run the application

```
export cuda_visible_devices=1

ollama serve

litellm --model ollama_chat/example

python RAG_main/server.py
```

## Run application in the local machine

- **Step 1**: download

```
pip install yfinance pyautogen langchain requests streamlit serpapi openai pandas
langchain langchain-community langchain-core langchain-experimental langchain-
groq langchain-openai flask duckduckgo_search
```

- **Step 2**: change the remote machine connection in the `Finance_Report_📈.py` page

```
host = "host"
port = 53208
username = "root"
password = "password"
```

- **Step 3**: run the application, get the user interface in the browser

```
streamlit run ./multi-agent/Finance_Report_📈.py
```

# Explanation for each component

> For more detail, please check the `readme.md` file for each component

- `data` folder is the data preparation for finetune
- `finetune` folder is all the work that run finetune for model training
- `RAG` folder for building the RAG, please check the `RAG_main` folder.
- `remote_server` folder for set up for the remote machine.
- `multi-agent`

you can check the `report_example.md` is our report output