📘 Remoder Lab 2 · Project 1 — Financial Advisor AI Agent

Simple, visual, and student-friendly

Remoder.com

💼 Financial Advisor AI Agent — Lab 2 • Project 1

Build an AI agent that analyzes financial data, evaluates user risk profiles, and generates personalized investment recommendations.

👉 In this project, we combine AI models & APIs.

👉 This is a real-world setup showing how AI-Agents can be deployed in production.

What This AI Agent Does

📊 Evaluates user financial profile

🧠 Understands user risk tolerance

💡 Generates investment recommendations

⚙️ Runs on FastAPI + Python + Ollama

🐳 Fully Dockerized for portability & Docker Composed for full Container Orchestration testing before shipping it to your K8s Env.

Architecture

Client → API → Risk Engine → Portfolio Engine → LLM → Final Advice

- API receives user input
- System computes risk score
- Portfolio is built
- LLM turns raw numbers into natural-language advice

Components

The Agent is built from 4 main modules:

- 🔢 RiskModel — calculates risk score
- 💼 PortfolioBuilder — generates portfolio allocation
- 🤖 LLM Engine — uses Ollama to produce advice text
- 🌐 FastAPI App — connects everything

Risk Modeling

What is Risk Modeling?

- Measures how safely or aggressively someone wants to invest
- Based on:
  - Age
  - Investment goal
  - Savings
  - Timeline
  - Comfort with market volatility

Output:

👉 A single risk score (1–10)

Portfolio Building

Based on the risk score, the agent builds:

- 🟦 Stocks %
- 🟩 Bonds %
- 🟨 Cash %

Example:

- High risk → 90% stocks
- Medium risk → 60% stocks
- Low risk → 30% stocks

Using LLM (Ollama)

The AI doesn’t just calculate numbers —

It *explains* the portfolio like a human advisor:

✔️ Uses Mistral / Llama / Gemma models (in fact, any other models can be used and this input can be modified).
✔️ Generates a personalized explanation
✔️ Simple, natural-language output for users

FastAPI Backend

- 📲 Handles API routes like /advise
  🔌 Manages incoming requests and endpoints
- 📤 Passes user data into the agent
  🧠 Feeds user profile into AI decision engine
- 📦 Returns structured JSON response
  📄 Outputs clean, machine-readable financial advice
- 🚀 Designed to scale and support UI integrations
  🖥️ Easily connects to dashboards, apps, and web interfaces

Docker Architecture

Multi-container setup:

- 🐳 Container 1: FastAPI
- 🧠 Container 2: Ollama
- 📦 Both orchestrated with Docker Compose

Makes development + shipping easy.

Example Request - JSON


```
{
    "age": 32,
    "goal": "retirement",
    "savings": 45000,
    "investment_timeline": 10,
    "risk_tolerance": "medium"
}
```

Example Response - JSON


```
{
    "risk_score": 6,
    "portfolio": {"stocks": "70%", "bonds": "25%", "cash": "5%"},
    "advice": "Based on your 10-year timeline and medium tolerance..."
}
```


End-to-End Flow (Visual Walkthrough)

1. 💰 User enters financial details
   🧾 Input includes income, goals, timeline, etc.
2. 🔄 API receives request
   📡 Securely processes and forwards the data
3. 🧠 Risk model evaluates profile
   📊 Calculates risk score, investment preferences
4. 📈 Portfolio is generated
   🪙 Customized mix of stocks, bonds, cash, etc.
5. 📝 LLM writes natural-language advice
   🤖 Generates clear, friendly investment guidance
6. 📦 API sends final structured response
   📬 Delivered and added in JSON response in readable + machine-friendly format

Why This Project Matters

💼 Real-world finance use case

🔧 Shows modular AI system design

🧠 Teaches LLM + API + Python engineering

🚀 Ready to extend into production-grade advisor apps

What You Learned

🧩 How an AI financial advisor works

🧮 How risk modeling is computed

🗂️ How to build a portfolio engine

🔗 How to integrate Python modules

🧠 How to call and control LLMs via APIs

🐳 How to run everything in Docker

The Future Belongs to AI + DevOps + Systems Engineers

- You're now closer to becoming an AI + Systems Engineer.
- Why this project matters — and why your skillset now puts you ahead

🌐 The World is Changing — AI Needs More Than Just Prompting

Modern AI systems don’t live in notebooks —
they live in real infrastructure, deployed across containers, servers, networks, APIs, and monitoring systems.

That requires *engineers*, not just “AI users.”

🛠️ Why AI + DevOps + Systems Engineering = The Winning Combo

1. AI without DevOps = a model that can’t scale
2. DevOps without AI = outdated engineering
3. AI without Systems Engineering = fragile apps
4. Systems Engineering without AI = irrelevant future skills

But together:

- ⚙️ You can deploy AI anywhere (Docker, server, cloud)
- 📦 You understand containers, security, networking
- 🚦 You can monitor, log, troubleshoot and scale systems
- 🧠 You know how LLMs, vector search, and agents work
- 🤝 You can build complete AI products end-to-end

This is the exact profile top companies are hiring for.

💼 How This Financial AI Agent Demonstrates Real Skills

This project showed you how to:

🧮 Build decision engines (risk scoring + portfolio logic)

🔗 Chain multiple components into one unified AI system

🤖 Connect Python logic to real LLM inference engines

🌐 Expose AI via APIs (FastAPI best practices)

🐳 Containerize the whole pipeline using Docker

🔐 Follow security and engineering standards

🚀 Prepare for scaling, monitoring, and production workflows

This isn’t “toy AI.” This is the foundation of production-grade AI engineering.

🎉 You Didn’t Just Learn a Model — You Built an AI System

You’ve officially taken the first major step toward becoming a:

- AI Systems Engineer
- AI DevOps Engineer
- AI Platform Engineer
- AI Agent Architect

The future belongs to engineers who can build, deploy, and manage AI systems.

And you’re now on that path. 🔥

Questions? Reach out to us ٩(◕‿◕。)۶

📌 Website: https://remoder.com

📘 LinkedIn (Company): https://www.linkedin.com/company/remoder

👨‍💻 LinkedIn (Sanjars): https://www.linkedin.com/in/sanjars/

🎥 YouTube: https://www.youtube.com/@remoder-inc

🚀 Want the Full Walkthrough?

This entire project — step-by-step, production-ready, with diagrams, videos and code — is covered inside our Master AI Deployment course.

💬 Got questions or want to join the next cohort?

Reach out anytime — always happy to help engineers level up! 😊🔥