Portfolio Advisor: Your Investment Analysis Tool

Analysis

Portfolio Advisor is an AI-powered chatbot that provides investment portfolio analysis, financial insights, and asset allocation recommendations using Google's **Gemini API**. The chatbot operates within a **Streamlit** web application, allowing users to interact with it in real-time.

Aim

The primary aim of **Portfolio Advisor** is to create an **interactive investment assistant** that helps users with:

- Understanding investment concepts.
- Providing portfolio optimization strategies.
- Assessing risk-return trade-offs.
- Offering insights on asset allocation and diversification.
- Ensuring an easy-to-use and accessible chatbot interface for portfolio management.
- **Portfolio Analysis and Optimization:** Helping users analyze their investment portfolios to maximize returns while managing risk.

Objectives

1. **AI-Powered Investment Guidance:** Use **Google Gemini AI** to generate responses based on portfolio queries.
2. **Dynamic Model Selection:** Allow users to choose from available **Gemini models**.
3. **User-Friendly Interface:** Implement an easy-to-use UI with **Streamlit**.
4. **Interactive Chat Experience:** Maintain chat history within the session for continuity.
5. **Error Handling & Stability:** Provide proper error messages for API-related issues.
6. **Secure API Usage:** Ensure safe and authorized access to the **Gemini API**.
7. **Portfolio Analysis:** Implement **risk-return analysis, Sharpe ratio calculation, and visualization tools** to help users evaluate their investment portfolios.

---

4. Requirements

Software & Libraries

- **Python 3.13**

- **Streamlit** (pip install streamlit)
- **Google Generative AI SDK** (pip install google-generativeai)
- **Pandas & NumPy** for data analysis (pip install pandas numpy)
- **Matplotlib & Seaborn** for data visualization (pip install matplotlib seaborn)

API Requirements

- A valid **Google Gemini API Key**
- Proper billing setup for API usage

System Requirements

- A stable internet connection for API requests.
- A local or cloud-based environment for running Streamlit applications.

---

5. Portfolio Analysis

Key Features

- **Portfolio Return Calculation:** Computes the expected return based on asset weights and returns.
- **Portfolio Risk Estimation:** Uses standard deviation to measure portfolio volatility.
- **Sharpe Ratio Calculation:** Evaluates risk-adjusted returns to determine investment efficiency.
- **Data Visualization:** Pie charts for portfolio allocation and scatter plots for risk vs. return analysis.

Portfolio Evaluation Example

The bot calculates the following metrics for a given investment portfolio:

- **Portfolio Return:** Measures overall expected gains from the portfolio.
- **Portfolio Risk:** Assesses potential volatility and fluctuations in returns.
- **Sharpe Ratio:** Evaluates the risk-adjusted return using a predefined risk-free rate.

Using **Streamlit**, the chatbot visually represents portfolio allocation and risk-return tradeoffs, allowing users to make informed investment decisions.

---

6. Results

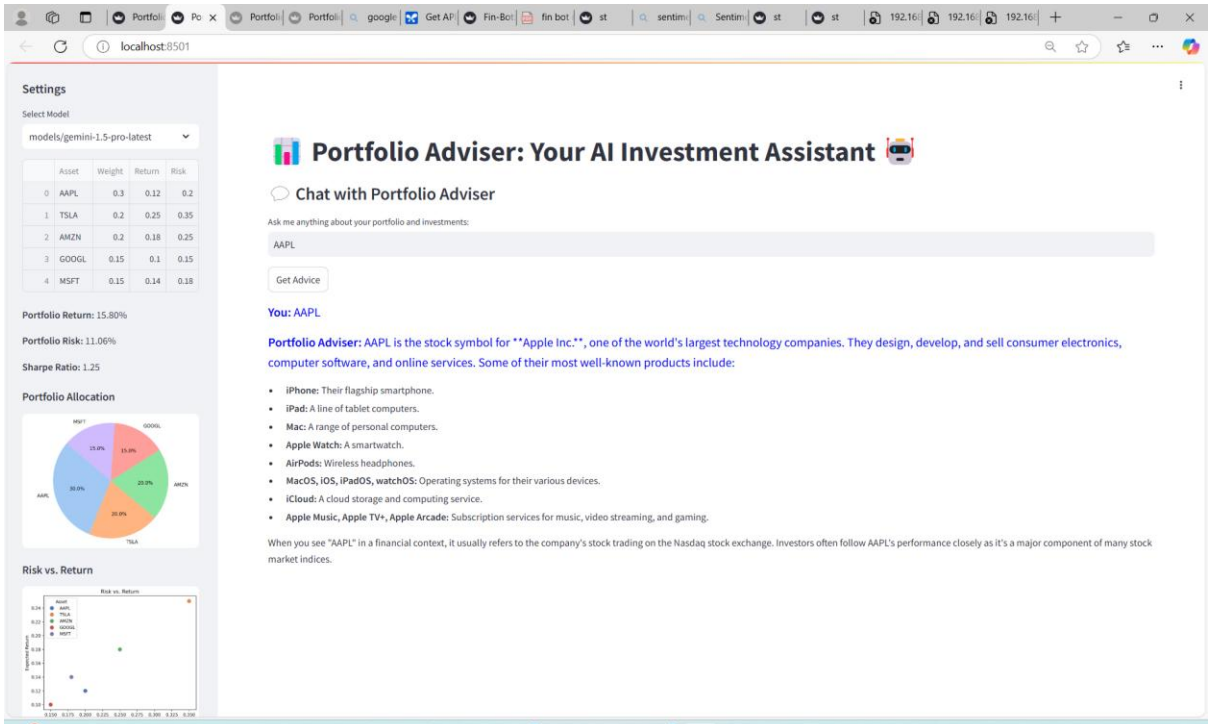
Expected Outcomes

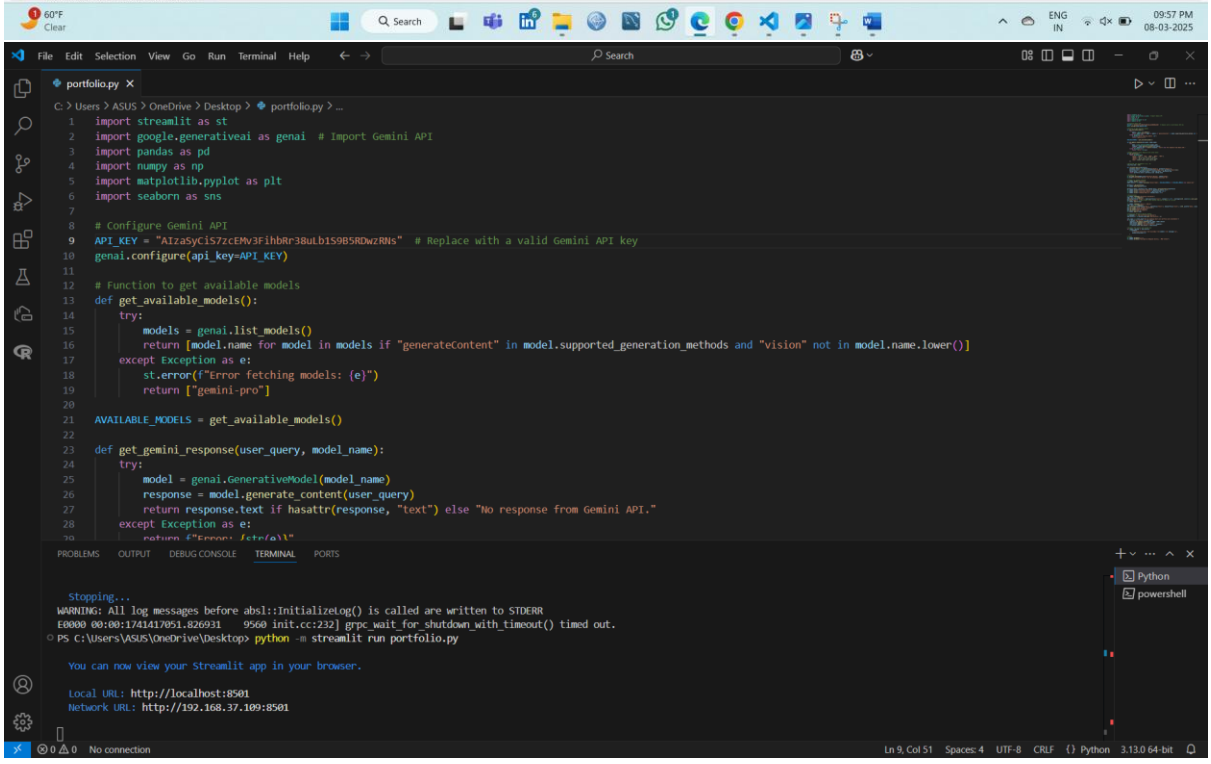
- Users can **ask investment-related questions** and receive AI-generated insights.
- Chat history is maintained for better interaction and continuity.
- Users can select different **AI models** dynamically based on availability.

- The chatbot UI is simple, effective, and visually engaging.
- **Portfolio analysis tools provide valuable investment insights.**
- Proper **error handling mechanisms** ensure smooth operation.
- **Scalability & Future Enhancements:** The bot can be extended with more financial APIs, data visualization tools, and predictive analytics.

Conclusion

Portfolio Advisor successfully demonstrates an AI-powered investment analysis tool, making financial knowledge accessible and interactive. Using **Google Gemini AI**, the chatbot provides accurate responses in real-time while maintaining a structured chat experience. With the added **portfolio analysis** feature, users gain deeper insights into their investments.


localhost:8501
Settings
Select Model
models/gemini-1.5-pro-latest
Asset Weight Return Risk
0 AAPL 0.3 0.12 0.2
1 TSLA 0.2 0.25 0.35
2 AMZN 0.2 0.18 0.25
3 GOOGL 0.15 0.1 0.15
4 MSFT 0.15 0.14 0.18
Portfolio Return: 15.80%
Portfolio Risk: 11.06%
Sharpe Ratio: 1.25
Portfolio Allocation
Risk vs. Return
Portfolio Adviser: Your AI Investment Assistant
Chat with Portfolio Adviser
Ask me anything about your portfolio and investments:
AAPL
Get Advice
You: AAPL
Portfolio Adviser: AAPL is the stock symbol for **Apple Inc.**, one of the world's largest technology companies. They design, develop, and sell consumer electronics, computer software, and online services. Some of their most well-known products include:
iPhone: Their flagship smartphone.
iPad: A line of tablet computers.
Mac: A range of personal computers.
Apple Watch: A smartwatch.
AirPods: Wireless headphones.
MacOS, iOS, iPadOS, watchOS: Operating systems for their various devices.
iCloud: A cloud storage and computing service.
Apple Music, Apple TV+, Apple Arcade: Subscription services for music, video streaming, and gaming.
When you see "AAPL" in a financial context, it usually refers to the company's stock trading on the Nasdaq stock exchange. Investors often follow AAPL's performance closely as it's a major component of many stock market indices.


portfolio.py
C: > Users > ASUS > OneDrive > Desktop > portfolio.py > ...
import streamlit as st
import google.generativeai as genai # Import Gemini API
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# Configure Gemini API
API_KEY = "AIzaSyCiS7zcEMv3FihbRr3BuLb1S9B5RDwzRNs" # Replace with a valid Gemini API key
genai.configure(api_key=API_KEY)
# Function to get available models
def get_available_models():
try:
models = genai.list_models()
return [model.name for model in models if "generateContent" in model.supported_generation_methods and "vision" not in model.name.lower()]
except Exception as e:
st.error(f"Error fetching models: {e}")
return ["gemini-pro"]
AVAILABLE_MODELS = get_available_models()
def get_gemini_response(user_query, model_name):
try:
model = genai.GenerativeModel(model_name)
response = model.generate_content(user_query)
return response.text if hasattr(response, "text") else "No response from Gemini API."
except Exception as e:
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Stopping...
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1741417051.826931 9560 init.cc:232] grpc_wait_for_shutdown_with_timeout() timed out.
PS C:\Users\ASUS\OneDrive\Desktop> python -m streamlit run portfolio.py
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://192.168.37.109:8501
Python
powershell
No connection
Ln 9, Col 51 Spaces: 4 UTF-8 CRLF Python 3.13.0 64-bit