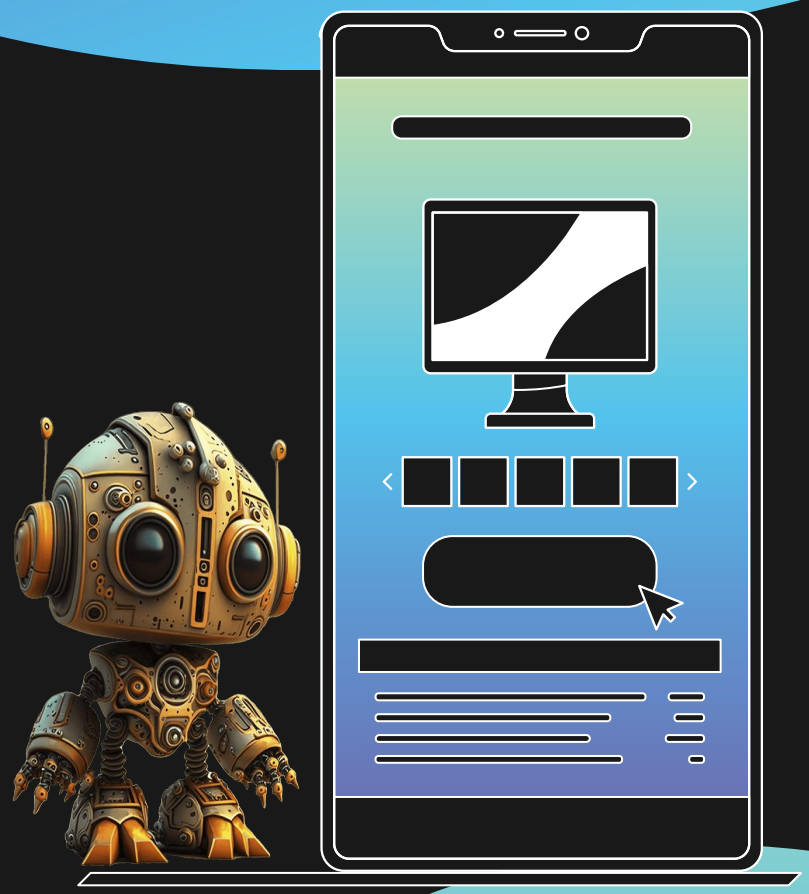
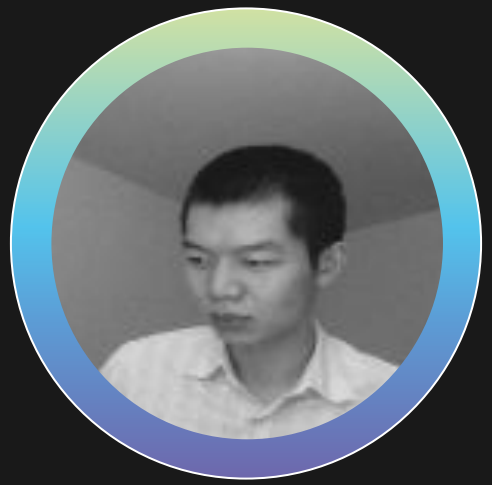


MONEY MENTOR

A Generative AI ChatBot for Customers in the Financial Services Industry

OUR TEAM

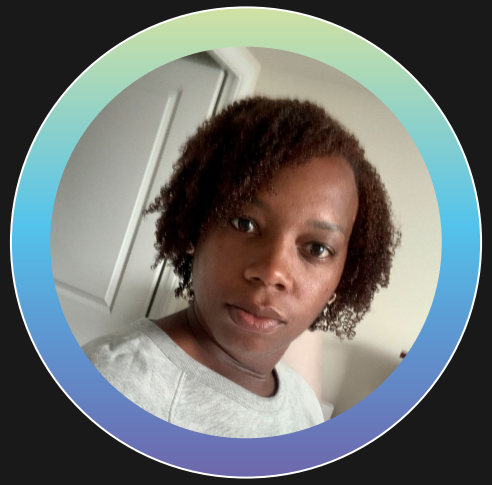


YINGQUAN LI

Engineer & Student
M.S. IT. (Virginia Tech 2023)

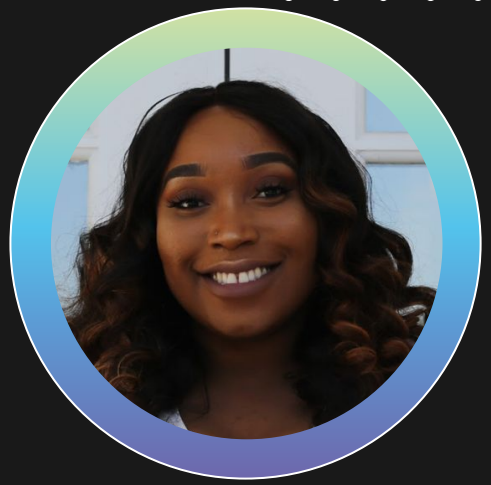
TOM ALEX

Summer Intern
(Restaurant Depot)
M.S. Computer Science
(Pace University 2023)



VERNELL GASPARD

Business Analyst
Consultant



AINA AHANMISI

Data Science Business
Manager
University of
Maryland CP

Agenda

01 WHY GENERATIVE AI

02 METHODOLOGY

03 DEMO

04 LIMITATIONS & CONSIDERATIONS

05 SCALABILITY

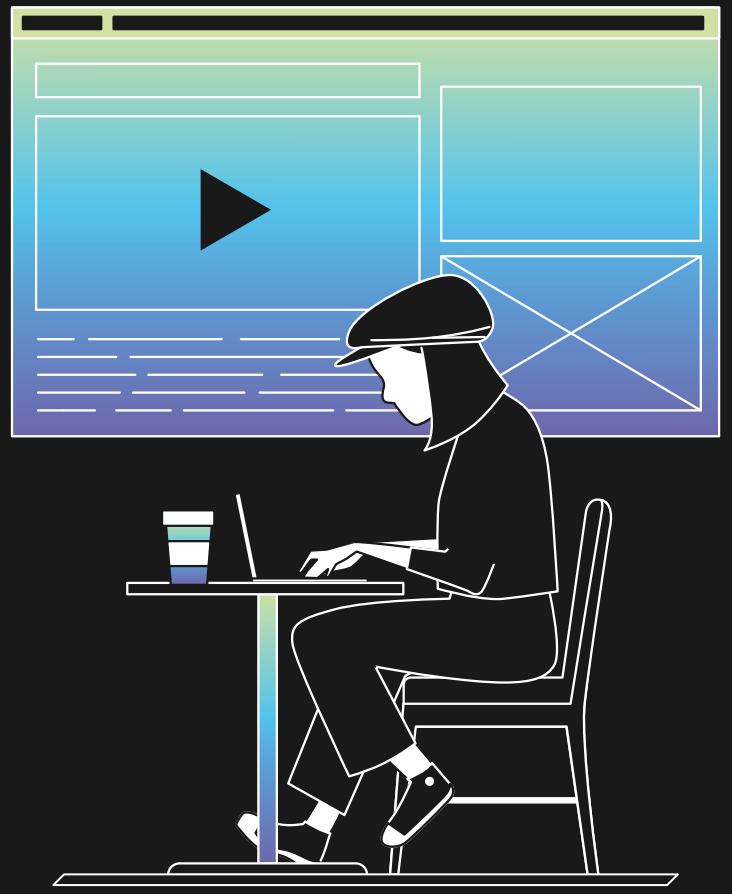
06 EVALUATION

07 BIAS & ALIGNMENT

08 FEEDBACK & QUESTIONS

01

WHY
GENERATIVE AI



Generative AI is a Disruptive Technology

- Generative AI in the form of Large Language Models (LLMs) is a disruptive technology ready to reshape all industries.

- According to a McKinsey study:

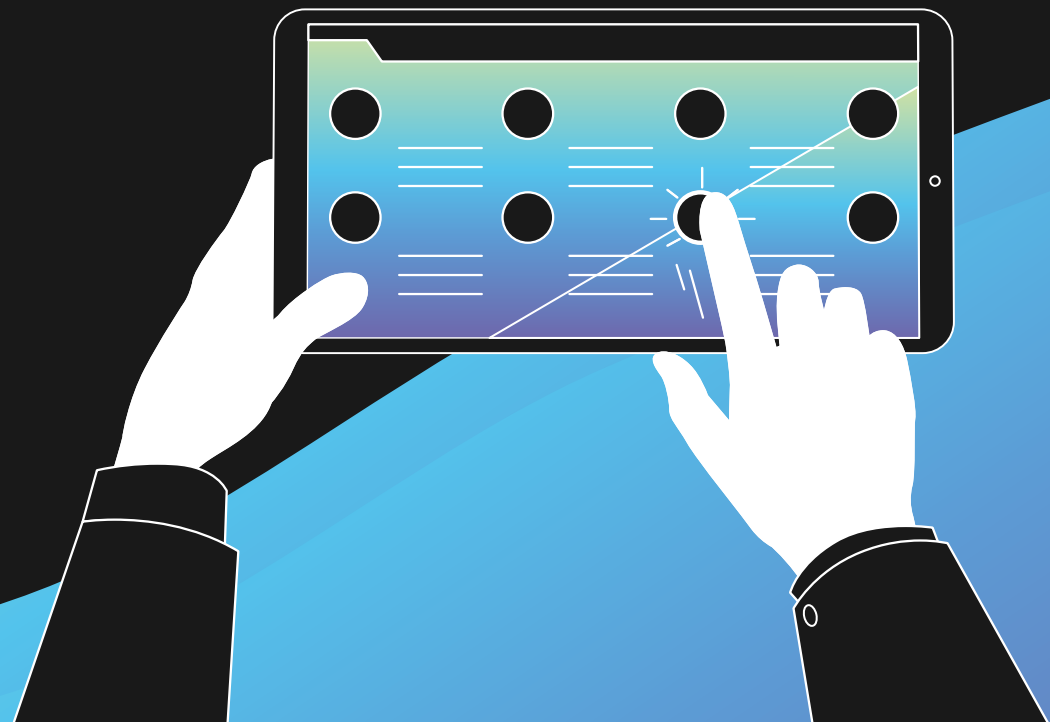
> ***Generative AI will have a significant impact across all industry sectors.*** Banking, high tech, and life sciences are among the industries that could see the biggest impact as a percentage of their revenues from generative AI. Across the banking industry, for example, the technology could deliver value equal to an additional $200 billion to $340 billion annually if the use cases were fully implemented. In retail and consumer packaged goods, the potential impact is also significant at $400 billion to $660 billion a year.

- **Objective:** With OpenAI opening ChatGPT APIs for public use, we are ready to leverage the power of prompt engineering to build a truly robust financial services tool that will "delight" customers.

02

METHODOLOGY

TOOLS & LIBRARIES



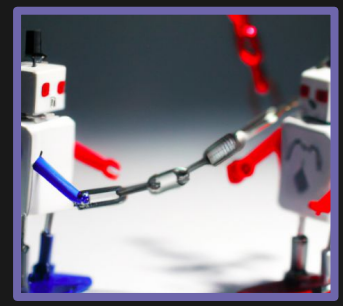
TOOLS & LIBRARIES

LangChain

- **OpenAI - Large Language Model (LLM)** accessed through *langchain.llms* module.
- **Api key** for accessing the LLM are stored in a separate file called *apikey.py*, which is imported through the *apikey* module.
- **STREAMLIT** Python library for building a user interface to input prompts and receive output from our LLM.
- **PROMPT ENGINEERING** involves structured user input to guide the language model and generate meaningful and contextually relevant responses via template-based prompts through *langchain.prompts.PromptTemplate* class.

Streamlit

TOOLS & LIBRARIES cont'd



LLM CHAINS used: 1) *LLMChain* and 2) *SequentialChain*.

CONVERSATION MEMORY: *ConversationBufferMemory* class from *langchain.memory* module to store and retrieve conversation history through two memory instances *topic_memory*, *advice_memory*, and *financial_institutions_memory*.

EXECUTION FLOW: The project follows a sequential execution flow.

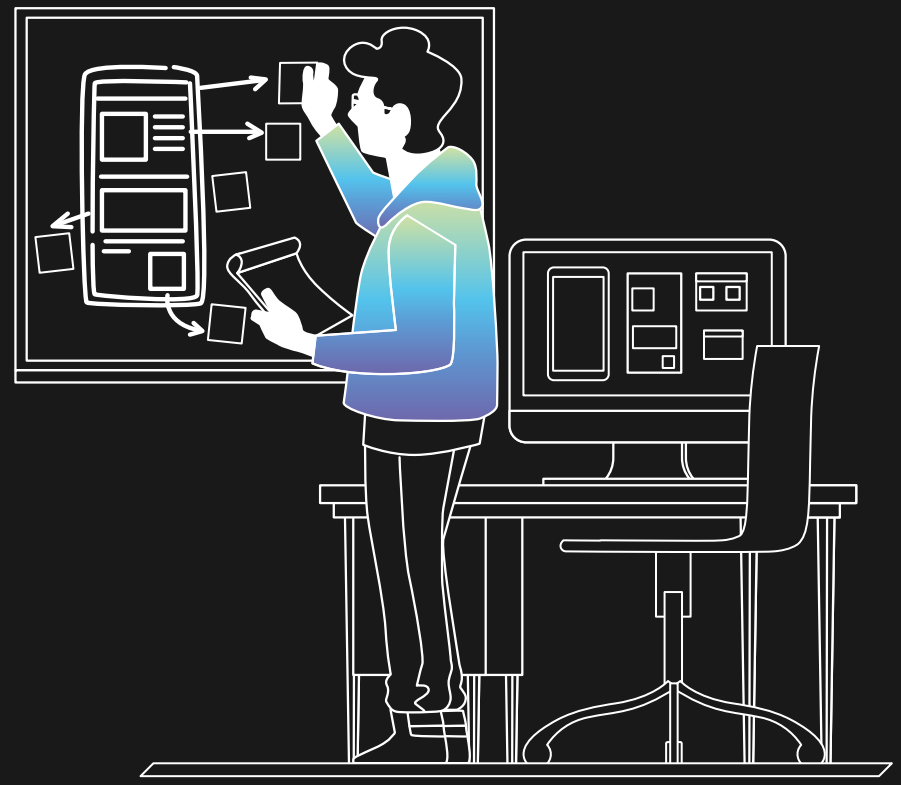
1. When the user enters a prompt, the topic generation chain (*topic_chain*) is invoked, which generates a topic based on the prompt and stores it in the *topic memory*.
2. The we invoke the advice generation chain (*advice_chain*), which offers what the chatbot thinks is the best advice give the topic. We update the *advice memory*.
3. The we invoke the financial institutions chain (*financial_institutions_chain*), which offers what the chatbot thinks is the best institutions that would have the investment vehicles of interest + potentially stock tickers of interest. We save the *financial institutions memory*.
4. This is all given to *SequentialChain* to execute.
5. The results are displayed on the UI (output + prompt histories stored in memory).

03

SOLUTION DEMO!

04

LIMITATIONS & CONSIDERATIONS



Addressing these issues can improve our solution and provide a better service:

Model Enhancements

- Model limitations, may need to update model to GPT-4
- Quality control and screen for hallucinations

Security & Privacy

- Privacy and security of user data and it's safeguarding
- Responsibility and ethics

Infrastructure

- System scalability
- Ongoing model updates
- User interface and user experience (UI/UX)

05

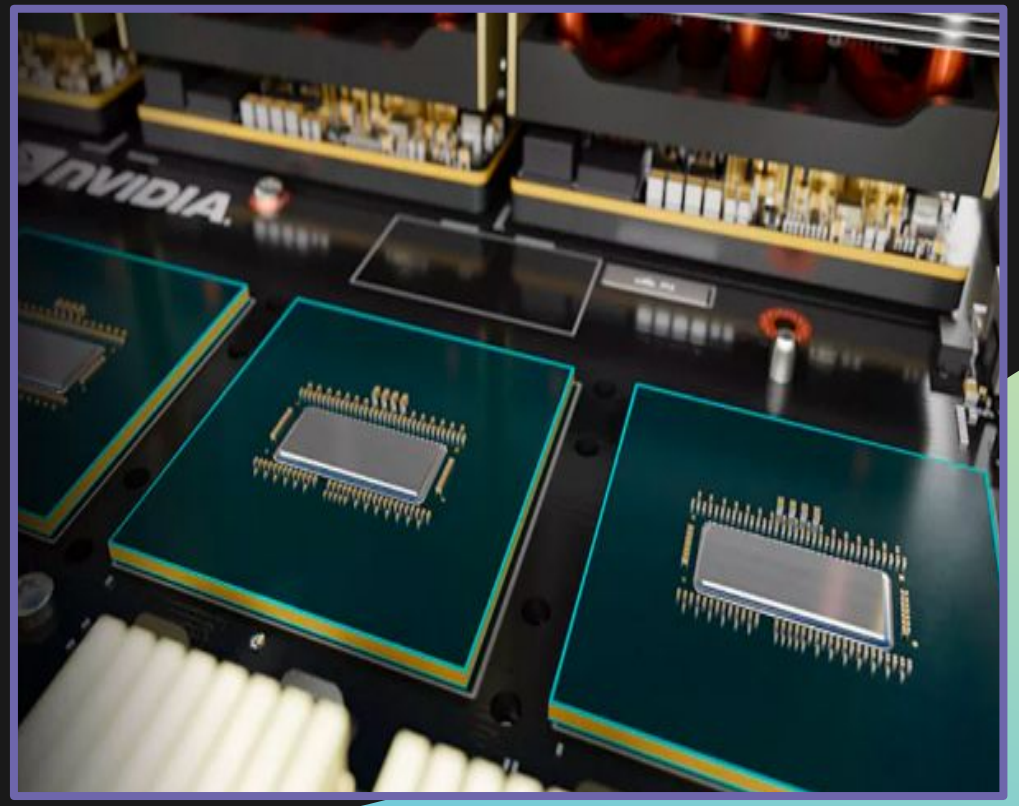
SCALABILITY

Scalability Solutions Include:

- prompt engineering techniques (automate the creation of prompts + create multiple prompt variations)
- cloud platforms (allow dynamic resource allocation)
- distributed systems (enable parallel processing and load balancing)

Scalability Challenges Include:

- high computational and data needs
- effective data management and distribution
- APIs and models are controlled and operated by OpenAI + +



06

EVALUATION

KEY PERFORMANCE INDICATORS

- Customer satisfaction
- Response time
- Customer retention

07

BIAS

& ALIGNMENT

How do we Counter Bias?

BIAS

- Quality Assurance and Monitoring output
- Prompt engineering Techniques

ALIGNMENT

- Elicitation
- Specification
- Verification

Thank you all for your Attention!

Any questions ❓