From Earnings Calls to Insights: How to Easily Build a Financial Analyst Assistant Using Azure OpenAI

# 🧭 Introduction

In financial services, extracting meaningful insights from lengthy earnings call transcripts is time-consuming.

What if you could ask questions and get accurate, contextual answers instantly?

In this blog post, I’ll walk you through how I built a lightweight, secure, and scalable financial analyst assistant using Azure OpenAI and Streamlit.

You’ll be able to upload a document, ask questions like "What drove Q2 revenue growth?", and get summarized answers in seconds.

# 🏗️ Solution Overview

Problem: Financial analysts need to rapidly extract and communicate insights from quarterly reports.

Solution: Build a web-based assistant that:

- Accepts .txt versions of earnings calls

- Uses Azure-hosted GPT-35-Turbo to analyze and summarize (a lightweight Retrieval-Augmented Generation pattern)

- Provides instant Q&A through a clean Streamlit frontend

Architecture overview:

- User uploads .txt file via Streamlit

- File content + question sent to Azure OpenAI model

- Model responds with concise, contextual answer

# ⚙️ Technologies Used

- Azure OpenAI Service (GPT-35 Turbo deployment)

- Streamlit (UI for file upload and question input)

- Python (OpenAI SDK 1.x)

- Optional: Azure Blob Storage, Azure Cognitive Search, Cosmos DB

# 🛠️ How I Built It (Step-by-Step)

Step 1: Set Up Azure OpenAI

- Created a resource in East US

- Deployed gpt-35-turbo model under name gpt35

- Grabbed the endpoint + key from Azure Portal

Step 2: Built the Frontend in Streamlit

- User uploads a .txt file (earnings call)

- Enters a natural-language question

st.file\_uploader("📄 Upload a .txt document")

st.text\_input("💬 What would you like to ask?")

Step 3: Call Azure OpenAI Securely

from openai import AzureOpenAI

client = AzureOpenAI(api\_key=..., azure\_endpoint=..., api\_version="2023-05-15")

response = client.chat.completions.create(

model="gpt35",

messages=[{"role": "user", "content": prompt}],

max\_tokens=300

)

# 💡 Real Use Cases

- Summarize earnings calls for exec briefings

- Answer risk/compliance-related questions

- Compare Q&A across multiple quarters

# 🚀 What’s Next: Expanding with Azure Services

This MVP is just the beginning. You can:

- Add Azure Cognitive Search for document indexing + retrieval (RAG)

- Use Azure Blob Storage for managing uploaded files

- Log interactions to Cosmos DB or SQL for auditing/QA

- Feed answers into Power BI dashboards

These services integrate seamlessly with Azure OpenAI for real enterprise value.

Agent-based orchestration (like Semantic Kernel or AutoGen) can come in a later phase.

# 🧠 Key Takeaways

- Azure OpenAI lets you securely deploy enterprise-grade GPT models

- Streamlit provides a fast way to build UI for AI apps

- This use case shows how LLMs can accelerate financial analysis today

# 📥 Try It Yourself

[Download the working app and demo files](#)

Questions or ideas? Reach out on [LinkedIn](#) or leave a comment.

Next Post Teaser: In the next post, I’ll walk through how to add Blob Storage and Cognitive Search to make this assistant enterprise-scale and production-ready.