EVALUATION AI-HYBRID CHAT

PREPARED BY: BLUE ENIGMA TEAM

Task Overview

01

You are given a semi-functional hybrid AI assistant system.
Your goal is to **debug, complete, and improve*** it so that it can:

- Upload vector embeddings to Pinecone
- Query embeddings to retrieve relevant travel nodes
- Use Neo4j graph relations to enrich context
- Feed combined context into OpenAI's Chat model
- Return a final, intelligent answer
- Perform better than other submission.

Evaluation Rubric

02

Metric	Description	Points
Functionality	Program runs end-to-end without errors	20
Debugging Skills	Correctly fixes v2 SDK issues and OpenAI calls	15
Design & Readability	Code structure, modularity, comments	15
Prompt Engineering	Quality of generated answers	15
Neo4j Query Design	Quality of graph context retrieval	10
Bonus Innovation	Async, caching, extra tools	20
Documentation	Clear explanation of approach	5

Total : 100 points

Complete the Survey :-

03

Help us gain insight regarding the travel preference . The survey result will be shared to participants to gain greater insight in transforming their Travel-Hybrid AI .

Request to please fill the form -

Link : https://docs.google.com/forms/d/e/1FAIpQLSeN1oqy5t1GTT4RFrV4K_AFx9U2I8SBfW8anPaPrAyOY8zXkQ

/viewform?usp=sharing&ouid=111918697820140123890

Detailed Tasks

04

Task 1: Setup & Data Upload

- Run pinecone_upload.py to create the Pinecone index and upload embeddings.
- Fix any missing dependencies or environment issues.
- Confirm embeddings successfully appear in your Pinecone dashboard.

✅ Deliverable:
Screenshot of successful upsert batches and index details.

Task 2: Debug & Complete hybrid_chat.py

- Fix all deprecated / v2-incompatible Pinecone calls.
- Update OpenAI API calls to new OpenAI(api_key=...) client format.
- Ensure Neo4j queries return meaningful results.
- Implement the interactive CLI to handle real-time user queries.

✅ Deliverable:
Working python hybrid_chat.py session where a user can type:

"Enter your travel question: create a romantic 4 day itinerary for Vietnam"

and get a coherent, AI-generated response using both vector + graph context.

Task 3: Add Improvements (Bonus + Insight)

1. Encourage creative extensions:
- Add caching for embeddings.
- Integrate a simple search_summary() function to summarize top nodes.
- Use async (aiohttp) to parallelize embedding and graph fetch.
- Improve prompt clarity or add chain-of-thought style reasoning.

✅ Deliverable:
Brief writeup (improvements.md) describing what was changed and why.

Follow up Questions :- (to be answered in submission form)

05

Optimizing campaign techniques with better targeting, tailored landing pages, and intelligent email segmentation will greatly increase engagement, conversion rates, and campaign ROI. Here are the facts from the last campaign:

01

WHY USE BOTH PINECONE AND NEO4J INSTEAD OF ONLY ONE?

02

HOW WOULD YOU SCALE THIS TO 1M NODES?

03

WHAT ARE THE FAILURE MODES OF HYBRID RETRIEVAL?

04

IF PINECONE API CHANGES AGAIN, HOW WOULD YOU DESIGN FOR FORWARD COMPATIBILITY?

Submission Link : -

https://docs.google.com/forms/d/e/1FAIpQLSdJLO_EWapOMLJ7qWhZ131NhzlavFLkLyrlu46LVWWvecvknQ/viewform?usp=sharing&ouid=117876153235059378920

Provided Files

06

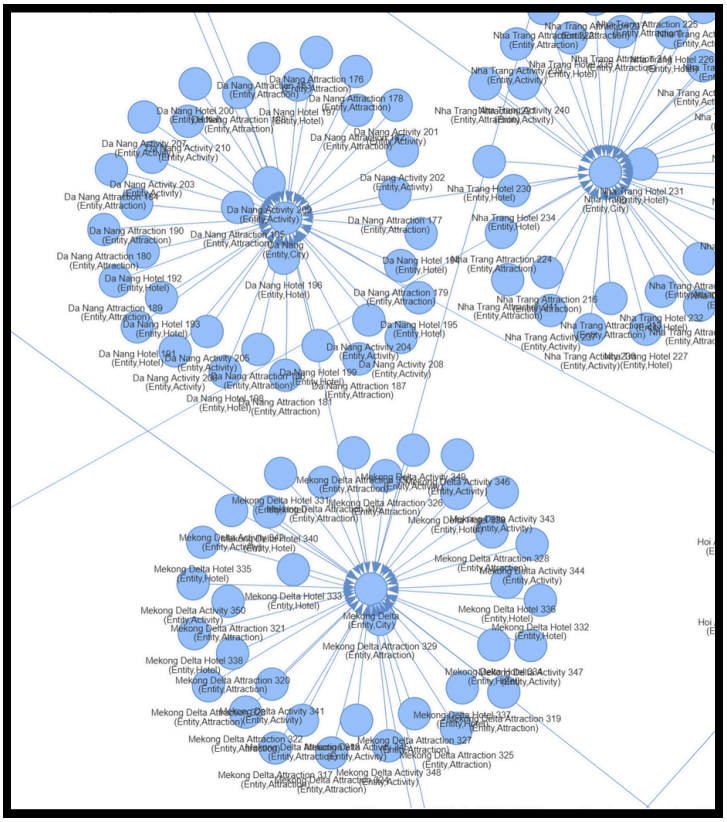
Candidates receive:

1. pinecone_upload.py → working, ready-to-run script (as you shared)
2. load_to_neo4j.py
3. visualize_graph.py
4. hybrid_chat.py → partial - (modify it to get better results than otheres , using agents , prompt engineering ,.etc)
5. config.py.sample → template for environment keys(modify yo insert your keys)
6. vietnam_travel_dataset.json → dataset for upload
7. **README.md** → instructions (see below)

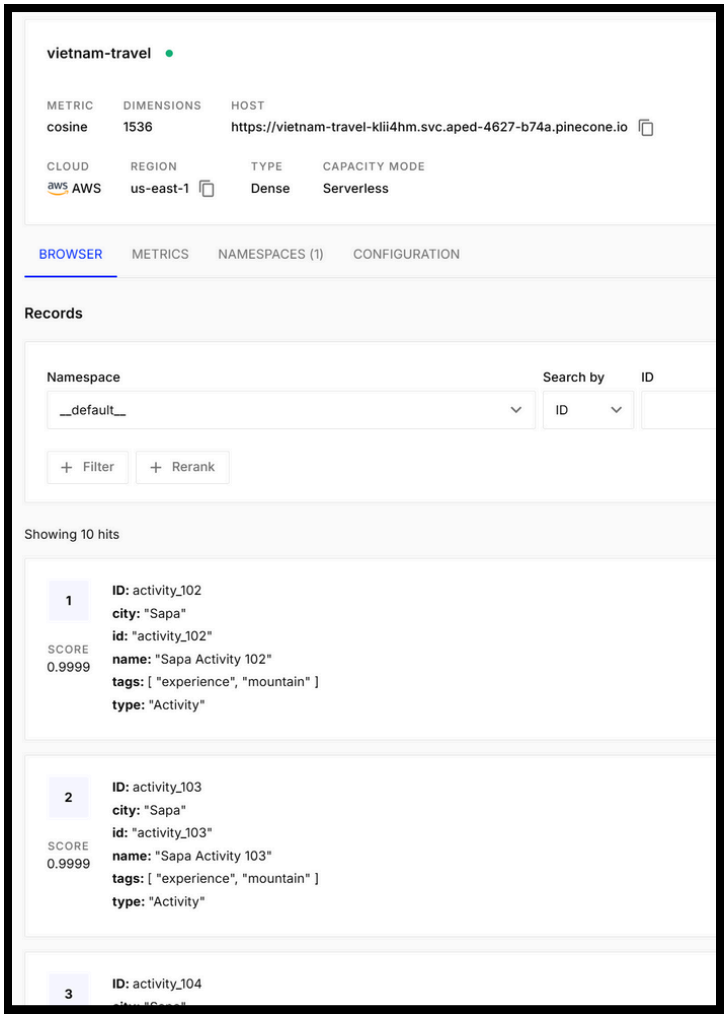
Link for download -

https://drive.google.com/drive/folders/1uCheVQcO08AQawk_eNu-GJfMQtdtS6O7?usp=sharing

visualizing the graph will look something as following :-



Your Pinecone index will look something like following -



README.md -

🧠 Hybrid AI Travel Assistant Challenge

Goal

Build and debug a hybrid AI assistant that answers travel queries using:

- Pinecone (semantic vector DB)

- Neo4j (graph context)

- OpenAI Chat Models

Steps

1.Set your API keys in `config.py`

2.create virtual environment & install all dependencies

3.Run ‘load_to_neo4j.py’

4.Run ‘visualize_graph.py’

5.Run `python pinecone_upload.py`

6.Run `python hybrid_chat.py`

7.Ask: `create a romantic 4 day itinerary for Vietnam`

8.**Modify “hybrid_chat.py” to improve the outcome.**

Deliverables

- Working scripts (`pinecone_upload.py`, `hybrid_chat.py`)

- **`improvements.md` (explain your fixes)**

- **Screenshot or log of working interaction**

Goal

07

The Goal of this test is to Assess candidates’ ability to build an intelligent retrieval-augmented chat system combining:

- Vector search (Pinecone)
- Knowledge graph (Neo4j)
- OpenAI embeddings + chat models
- Clean, modular Python engineering practices

And if the person in able to scale, modify and improve the system .

BEST OF LUCK

FURTHER ANALYSIS &
DISCUSSION ARE WELCOME
FOR OPTIMIZATION INSIGHTS.

BLUE ENIGMA

2025 EVALUATION TEST

+91-6377808345

WWW.BLUE-ENIGMA.COM