Assignment: Build a TED Talk RAG Assistant

Goal

Create a **knowledgeable AI assistant specialized in TED talks** using a **Retrieval-Augmented Generation (RAG)** system.
Your assistant must answer a set of questions **only from the provided TED dataset**, without relying on any external information.

Dataset

You will work with a TED dataset in English.

Schema (CSV columns):

```
talk_id,title,speaker_1,all_speakers,occupations,about_speakers,views,recorded_date,published_date,event,native_lang,available_lang,comments,duration,topics,related_talks,url,description,transcript
```

Downloads:

- English dataset: ted_talks_en.csv

Functional Requirements: Query Capabilities

Your Retrieval-Augmented Generation (RAG) system should be able to accurately answer several distinct categories of questions **using only the dataset (metadata + transcripts)**.

1. **Precise Fact Retrieval**

Goal: The model must locate a single, specific entity or fact based on semantic criteria within the corpus.

Example: ***"****Find a TED talk that discusses overcoming fear or anxiety. Provide the title and speaker."*

Explanation: The assistant should locate a concrete fact or detail and return a response according to prompt.

2. **Multi-Result Topic Listing (Up to 3 Results)**

Goal: Return multiple talks titles that match a theme or topic.

Example: *"Which TED talk focuses on education or learning? Return a list of exactly 3 talk titles."*

Explanation: The assistant must retrieve multiple distinct talks titles, not multiple chunks of the same talk. No need to support lists larger than 3.

3. **Key Idea Summary Extraction**

Goal: Identify a relevant talk and generate a concise summary of its main idea.

Example: *"Find a TED talk where the speaker talks about technology improving people's lives. Provide the title and a short summary of the key idea."*

Explanation: The assistant should provide the key idea based on transcript chunk evidence (not necessarily the whole talk).

4. **Recommendation with Evidence-Based Justification**

Goal: Recommend one relevant talk and justify the choice.

Example: *"I'm looking for a TED talk about climate change and what individuals can do in their daily lives. Which talk would you recommend?"*

Explanation: The assistant should choose a talk and provide a **justification grounded in the retrieved data**.

Your system must be able to answer these questions **without relying on model common knowledge**.

Tools, Budget & Constraints

Available Models:

- `RPRTHPB-text-embedding-3-small` (default dimensions **1536)**
- `RPRTHPB-gpt-5-mini`

Budget Constraint:

- Your **total budget for this assignment is 5 USD** (including all development & testing).
- You must design efficiently:
    - Avoid embedding the same data repeatedly.
    - Start with a **smaller subset**, validate your approach, then scale up.
    - Overstepping the budget will **reduce your final score**.

RAG Hyperparameters (you must choose & report):

- **Chunk size**: Maximum: **2048 tokens**
- **Overlap**: Maximum: **30%** (0.3)
- **Top-k** (number of retrieved chunks): Maximum: **30**

Note, Pushing too much unnecessary data into the model context will be considered suboptimal (inefficient).

System Prompt for `RPRTHPB-gpt-5-mini` (Required Section)

You must include the following (or extremely similar) **system prompt section** when calling `RPRTHPB-gpt-5-mini`:

> **System Prompt Section (use as system / instructions role):**
> *You are a TED Talk assistant that answers questions strictly and only based on the TED dataset context provided to you (metadata and transcript passages). You **must not** use any external knowledge, the open internet, or information that is not explicitly contained in the retrieved context. If the answer cannot be determined from the provided context, respond: "I don't know based on the provided TED data." Always explain your answer using the given context, quoting or paraphrasing the relevant transcript or metadata when helpful.*

You may add additional clarifications (e.g., response style), but you **must keep the above constraints**.

Vector Database & Deployment

- Use **Pinecone** as your vector database:
    - Pinecone: https://www.pinecone.io
    - Make sure your Pinecone index remains **active until you receive a grade or are otherwise instructed**.
    - Make sure to set dimensions according to the embedding model.

- Deploy your app to **Vercel**:
    - Vercel: https://vercel.com
    - a **public Live URL** must be submitted, instructions to follow.

API Requirements

Your system must expose the following HTTP endpoints:

1.POST {your-url}/api/prompt

Used to query your system with questions (similar to the above questions).

Input format (JSON):

```
{
  "question": "Your natural language question here"
}
```

Output format (JSON):

```
{
  "response": "Final natural language answer from the
model.",
  "context": [
    {
      "talk_id": "1234",
      "title": "Sample TED Talk",
      "chunk": "transcript chunk retrieved",
      "score": 0.1234
    }
  ],
  “Augmented_prompt”: {
     “System”: “the system prompt used to query the chat
model”
     “User”: “the user prompt used to query the chat model”
  }
}
```

- `response`: what `gpt-5-mini` returns after using the retrieved context.
- `context`: **array** of context chunks retrieved.
- `Augmented_prompt`: prompt to the gpt-5-mini model (system and user).

2.GET {your-url}/api/stats

Returns the **configuration you chose** for your RAG system.

Strict JSON format (must match exactly these field names):

```
{
  "chunk_size": 1024,
  "overlap_ratio": 0.2,
  "top_k": 5
}
```

Where:

- `chunk_size`: integer (tokens or approximate tokens/chars as you define in code)
- `overlap_ratio`: number between 0 and 0.3
- `top_k`: integer between 1 and 30

If you later change your hyperparameters, this endpoint **must always** reflect the **current values**.

Deliverable & Deadline

Submit your URL by **21.12.2025 (End of day)**.

Notes

- Practical advice:
    - Start with a small subset of talks, verify your RAG pipeline, then scale.
    - Don’t re-embed the whole dataset every time you tweak a parameter, design your workflow to find the right parameters with minimal cost.

Good luck, and have fun building your TED Talk RAG assistant!