Task 1

Install Libraries

```
[1] !pip install --quiet --upgrade google-cloud-logging google_cloud_firestore google_cloud_aiplatform langchain langchain-google-vertexai langchain_community langchain_exp
```

```
                                        229.5/229.5 kB 6.9 MB/s eta 0:00:00
                                        368.8/368.8 kB 13.9 MB/s eta 0:00:00
                                        7.7/7.7 MB 74.8 MB/s eta 0:00:00
                                        100.2/100.2 kB 8.0 MB/s eta 0:00:00
                                        2.5/2.5 MB 63.7 MB/s eta 0:00:00
                                        209.2/209.2 kB 13.2 MB/s eta 0:00:00
                                        24.1/24.1 MB 66.5 MB/s eta 0:00:00
                                        65.8/65.8 kB 3.1 MB/s eta 0:00:00
                                        42.1/42.1 MB 37.0 MB/s eta 0:00:00
                                        44.4/44.4 kB 2.1 MB/s eta 0:00:00
                                        44.7/44.7 kB 2.3 MB/s eta 0:00:00
                                        50.9/50.9 kB 4.2 MB/s eta 0:00:00
```

```
[2] import vertexai
    import logging
    import google.cloud.logging
    from vertexai.language_models import TextEmbeddingModel
    from vertexai.generative_models import GenerativeModel

    import pickle
    from IPython.display import display, Markdown

    from langchain_google_vertexai import VertexAIEmbeddings
    from langchain_community.document_loaders import PyMuPDFLoader
    from langchain_experimental.text_splitter import SemanticChunker

    from google.cloud import firestore
    from google.cloud.firestore_v1.vector import Vector
    from google.cloud.firestore_v1.base_vector_query import DistanceMeasure
```

Set Variables and VertaxAI.init()

```
[3] PROJECT_ID="qwiklabs-gcp-01-4db5c0a69bf4"
    LOCATION="us-central1"
    vertexai.init(project=PROJECT_ID, location=LOCATION)
```

Set embedding Model

```
[4] embedding_model = VertexAIEmbeddings(model_name="text-embedding-005")
```

Task 2 Download, process and chunk data semantically

In this section, you will prepare the NYC Food Safety Manual for Retrieval-Augmented Generation (RAG). Clean the PDF content and split it into meaningful chunks based on semantic similarity using sentence embeddings and generate numerical representations (embeddings) for each identified text chunk.

Download the New York City Department of Health and Mental Hygiene's Food Protection Training Manual. This document will serve as your Retrieval-Augmented Generation source content.

```
[5] !gcloud storage cp gs://partner-genai-bucket/genai069/nyc_food_safety_manual.pdf .
```

```
Copying gs://partner-genai-bucket/genai069/nyc_food_safety_manual.pdf to file://./nyc_food_safety_manual.pdf

Average throughput: 112.9MiB/s
```

Use the LangChain class PyMuPDFLoader to load the contents of the PDF to a variable named data.

The following function is provided to do some basic cleaning on artifacts found in this particular document. Create a variable called cleaned_pages that is a list of strings, with each string being a page of content cleaned by this function.

```
[6] from langchain_community.document_loaders import PyMuPDFLoader

    loader = PyMuPDFLoader("nyc_food_safety_manual.pdf")
    data = loader.load()
```

```
[7] def clean_page(page):
      return page.page_content.replace("-\n","")\
                              .replace("\n"," ")\
                              .replace("\x02","")\
                              .replace("\x03","")\
                              .replace("fo d P R O T E C T I O N  T R A I N I N G  M A N U A L","")\
                              .replace("N E W  Y O R K  C I T Y  D E P A R T M E N T  O F  H E A L T H  &  M E N T A L  H Y G I E N E","")

    # Clean pages into list of strings
    cleaned_pages = [clean_page(page) for page in data]
```

Use LangChain's SemanticChunker with the embedding_model you created earlier to split the first five pages of cleaned_pages into text

Use LangChain's SemanticChunker with the embedding_model you created earlier to split the first five pages of cleaned_pages into text chunks. The SemanticChunker determines when to start a new chunk when it encounters a larger distance between sentence embeddings. Save the strings of page content from the resulting documents into a list of strings called chunked_content. Take a look at a few of the chunks to get familiar with the content.

Use the embedding_model to generate embeddings of the text chunks, saving them to a list called chunked_embeddings. To do so, pass your list of chunks to the VertexAIEmbeddings class's embed_documents() method.

You should have successfully chunked & embedded a short section of the document. To get the chunks & corresponding embeddings for the full document, run the following code:

```
[8] from langchain_experimental.text_splitter import SemanticChunker

    # Use embedding_model from Task 1
    splitter = SemanticChunker(embedding_model)

    # Chunk the first 5 cleaned pages
    docs = splitter.create_documents(cleaned_pages[:5])

    # Extract only the chunk text into a list
    chunked_content = [doc.page_content for doc in docs]

    # Preview
    for chunk in chunked_content[:3]:
        print("🧩 Chunk:\n", chunk[:300], "\n---")
```

```
🧩 Chunk:
 The Health Code These are regulations that were formulated to allow the  Department to effectively protect the health of the population. Among the rules embodied in the H
---
🧩 Chunk:
 Registration is done on-line. The link is: nyc.gov/foodprotectioncourse Register for Health Academy Classes On-Line You may now register and pay online for courses offer
---
🧩 Chunk:
 If you don’t see a date that is convenient, check back as new course dates are added frequently. 1   INTRODUCTION T he New York City Department of Health and Mental Hygi
---
```

```
[9] chunked_embeddings = embedding_model.embed_documents(chunked_content)
```

```
[10] !gcloud storage cp gs://partner-genai-bucket/genai069/chunked_content.pkl .
     !gcloud storage cp gs://partner-genai-bucket/genai069/chunked_embeddings.pkl .
```

```
Copying gs://partner-genai-bucket/genai069/chunked_content.pkl to file://./chunked_content.pkl
Copying gs://partner-genai-bucket/genai069/chunked_embeddings.pkl to file://./chunked_embeddings.pkl

Average throughput: 144.1MiB/s
```

```
[11] import pickle

     chunked_content = pickle.load(open("chunked_content.pkl", "rb"))
     chunked_embeddings = pickle.load(open("chunked_embeddings.pkl", "rb"))
```

```
[12] import google.cloud.logging
     import logging

     client = google.cloud.logging.Client()
     client.setup_logging()

     log_message = f"chunked contents are: {chunked_content[0][:20]}"
     logging.info(log_message)
```

```
INFO:root:chunked contents are: The Health Code Thes
```

Task 3 Prepare your vector database

In this section, you will set up a Firestore database to store the processed NYC Food Safety Manual chunks and their embeddings for efficient retrieval. You'll then build a search function to find relevant information based on a user query.

Create a Firestore database with the default name of (default) in Native Mode and leave the other settings to default.

Next, in your Colab Enterprise Notebook populate a db variable with a Firestore Client.

Use a variable called collection to create a reference to a collection named food-safety.

Using a combination of your lists chunked_content and chunked_embeddings, add a document to your collection for each of your chunked documents. Each document can be assigned a random ID, but it should have a field called content to store the chunk text and a field called embedding to store a Firestore Vector of the associated embedding.

Create a vector index for your collection using your embedding field.

Note: A find_nearest() operation cannot be executed on a collection without an index. When attempted, the system will return an error message including instructions to create the index using a gcloud command.

Complete the function below to receive a query, get its embedding, and compile a context consisting of the text from the 5 documents with the most similar embeddings. This time, use the embed_query() method of the LangChain VertexAIEmbeddings embedding_model to embed the user's query.

```
[21] # Populate a db variable with a Firestore Client.
     db = firestore.Client(project=PROJECT_ID)

     # Use a variable called collection to create a reference to a collection named food-safety.
     collection = db.collection('food-safety')

     # Using a combination of our lists chunked_content and chunked_embeddings,
     # add a document to your collection for each of your chunked documents.
     for i, (content, embedding) in enumerate(zip(chunked_content, chunked_embeddings)):
         doc_ref = collection.document(f"doc_{i}")
         doc_ref.set({
             "content": content,
             "embedding": Vector(embedding)
```

```
        embedding : Vector(embedding)
    })
```

```
[22] !gcloud firestore indexes composite create \
  --collection-group=food-safety \
  --query-scope=COLLECTION \
  --field-config field-path=embedding,vector-config='{"dimension":"768", "flat": "{}"}' \
  --project="qwiklabs-gcp-01-4db5c0a69bf4"
```

```
Create request issued
Created index [CICAgOjXh4EK].
```

```
[ ] def search_vector_database(query: str):

      context = ""

      # 1. Generate the embedding of the query

      # 2. Get the 5 nearest neighbors from your collection.
      # Call the get() method on the result of your call to
      # find_nearest to retrieve document snapshots.

      # 3. Call to_dict() on each snapshot to load its data.
      # Combine the snapshots into a single string named context


      return context
```

```
def search_vector_database(query: str):
  context = ""
  query_embedding = embedding_model.embed_query(query)
  vector_query = collection.find_nearest(
    vector_field="embedding",
    query_vector=Vector(query_embedding),
    distance_measure=DistanceMeasure.EUCLIDEAN,
    limit=5,
  )
  docs = vector_query.stream()
  context = [result.to_dict()['content'] for result in docs]
  return context
```

Next, call the function with the query How should I store food? to confirm it's functionality.

```
[24] search_vector_database("How should I store food?")
```

```
[' Store foods away from dripping condensate , at least six inches above the floor and with enough space between items to encourage air circulation. Freezer Storage Freezing is an excellent method for prolonging the shelf life of foods. By keeping foods frozen solid, the bacterial growth is minimal at best. However, if frozen foods are thawed and then refrozen, then harmful bacteria can reproduce to dangerous levels when thawed for the second time. In addition to that, the quality of the food is also affected. Never refreeze thawed foods, instead use them immediately. Keep the following rules in mind for freezer storage:  Use First In First Out method of stock rotation. All frozen foods should be frozen solid with temperature at 0°F or lower. Always use clean containers that are clearly labeled and marked, and have proper and secure lids. Allow adequate spacing between food containers to allow for proper air circulation. Never use the freezer for cooling hot foods. * * Tip: When receiving multiple items, always store the frozen foods first, then foods that are to be refrigerated, and finally the non perishable dry goods. Dry Storage Proper storage of dry foods such as cereals, flour, rice, starches, spices, canned goods, packaged foods and vegetables that do not require refrigeration ensures that these foods will still be usable when needed. Adequate storage space as well as low humidity (50% or less), and low temperatures (70 °F or less) are strongly recommended.',
 'Only use food containers that are clean, non-absorbent and are made from food-grade material intended for such use. Containers made from metal may react with certain type of high acid foods such as sauerkraut, citrus juices, tomato sauce, etc. Plastic food-grade containers are the best choice for these types of foods. Containers made of copper, brass, tin and galvanized metal should not be used. The use of such products is prohibited. Re-using cardboard containers to store cooked foods is also a source of contamination. Lining containers with newspapers, menus or other publication before placing foods is also prohibited as chemical dyes from these can easily leach into foods. Storage Areas Foods should only be stored in designated areas. Storing foods in passageways, rest rooms, garbage areas, utility rooms, etc. would subject these to contamination. Raw foods must always be stored below and away from cooked foods to avoid cross contamination. Refrigerated Storage This type of storage is typically used for holding potentially hazardous foods as well as perishable foods for short periods of time—a few hours to a few days. An adequate number of efficient refrigerated units are required to store potentially hazardous cold foods. By keeping cold foods cold, the microorganisms that are found naturally on these foods are kept to a minimum. Cold temperature does not kill microorganisms, however, it slows down their growth. Pre-packaged cold foods must be stored at temperatures recommended by the manufacturer. This is especially important when dealing with vacuum packed foods, modified atmosphere packages and sous vide foods. Smoked fish is required by the Health Code to be stored at 38°F or below. Fresh meat, poultry and other potentially hazardous foods must be stored at 41°F or below, while frozen foods must be stored at 0°F or below. For foods to be maintained at these temperatures, refrigerators and freezers must be operating at temperatures lower than 41°F and 0°F., respectively. Thermometers placed in the warmest part of a refrigerated unit are necessary to monitor the temperature of each unit. The rule of storage, First In First Out (FIFO) ensures that older deliveries are used up before newer ones. In practicing FIFO, the very first step would be to date all products as they are received. The next step is to store the newer products behind the older ones. The following rules are important in making sure that foods are safe during refrigerated storage:  Store cooked foods above raw foods to avoid cross-contamination. Keep cooked food items covered unless they are in the process of cooling, in which case they must be covered after being cooled to 41°F. Avoid placing large pots of hot foods in a refrigerator.',
 'l Store food in vermin-proof containers – metal or glass  containers, with tightly fitted lids. l Remove dented, leaking, rusted, swollen or unlabeled canned goods. Cold Storage: l All PHFs must be stored at 41° F (Except smoked fish at 38° F and raw shell eggs at 45 ° F). l All cooked and ready-to-eat food must be stored away from and above raw food. l Do not store foods in quantities that exceed the storage unit's  capacity. l Place a refrigeration thermometer in the warmest spot in the unit to measure ambient air temperature of the unit l Check for condensation that may contaminate food. l Keep frozen foods frozen at 0° F or lower. STORAGE',
 'Furthermore, it is improper to store food in ice machines or ice that will be later used for human consumption. Food should be stored at least six inches off the floor, away from walls and dripping pipes. Keep all food, bulk or otherwise, covered and safe from contamination. Check food daily and throw away any spoiled or contaminated food. Store cleaning, disinfecting, and other chemicals away from foods, clearly marked and in their original containers. Keep food refrigerated at a temperature of 41°F or below. Monitor temperatures regularly with a thermometer placed in the warmest part of the refrigerator. Keep all cooling compartments closed except when you are using them. Store food in a refrigerator in such a way that the air inside can circulate freely. Keep all refrigerated foods covered, and use up stored leftovers quickly. When dishes and utensils are sparkling clean, keep them that way by proper storage.',
 'In addition to the above, avoid sunlight as it may affect the quality of some foods. Following are some of the guidelines:  Use First In First Out method of stock rotation. Keep foods at least 6 inches off the floor. This allows for proper cleaning and to detect vermin activity. Keep foods in containers with tightly fitted lids. Keep dry storage areas well lighted and ventilated. Install shades on windows to prevent exposure from sunlight. Do not store foods under overhead water lines that may drip due to leaks or condensation. Do not store garbage in dry food storage areas. Make sure that dry storage area is vermin proof by sealing walls and baseboards and by repairing holes and other openings. * * Safety Tip: Storage of harmful chemicals in the food storage areas can create hazardous situations and hence is prohibited by law. All chemicals must be labeled properly and used in accordance to the instructions on the label. Pesticide use is prohibited unless used by a licensed pest control officer. Storage in Ice Whenever food items are to be stored in ice, care must be taken to ensure that water from the melted ice is constantly being drained so that the food remains on ice and not immersed in iced water.']
```