**RAG Based Food Image and Text search backend based on the data**

The backend API leverages the power of Retrieval Augmented Generation (RAG) techniques to provide efficient and intelligent text and image search capabilities. RAG combines the strengths of information retrieval and natural language generation, allowing the system to retrieve relevant information from the database based on user queries and generate contextual responses using language models.

For text-based queries, the RAG-based text search module analyzes the user's input, extracts relevant entities and keywords, and retrieves the most relevant dishes, restaurants, and menu items from the database using advanced retrieval algorithms.

Similarly, the RAG-based image search module employs computer vision techniques to identify the dish or food item in the uploaded image and retrieve corresponding information from the database.

To optimize cost and leverage the power of open-source technologies, the system utilizes pre-trained language models from the Hugging Face library. By leveraging open-source models from Hugging Face, the system avoids the recurring costs associated with commercial APIs like OpenAI, while still achieving high performance and accuracy.

Furthermore, the backend API has been designed and developed as a cloud-native application, allowing for easy integration into cloud-based ecosystems. I have trailed the whole thing on the cloud ecosystem, So that Azure and the Mongo db atlas, And the whole thing worked successfully.

With the backend API fully developed and deployed on the cloud, the only remaining step is the integration with the WhatsApp platform. Once integrated, users will be able to access the intelligent food ordering system through the familiar WhatsApp interface, leveraging the RAG-based text and image search capabilities, language understanding, and generation features, all powered by the cloud-based backend API.

I have used two datasets one for the text and the other for the image, the text one was artificially made by me with the help of the Python script, and the other image dataset i curated it from one of the research papers.

The text dataset used in this project provides comprehensive information about various restaurants across different cities in India. It encompasses details such as the restaurant name, location, primary cuisine, secondary cuisine offerings, price ranges for dishes, specific dish names, and their corresponding prices. The locations covered include major metropolitan areas like Chennai, Mumbai, Pune, Bengaluru, and Ahmedabad, allowing for a diverse representation of regional cuisines. The primary cuisines listed range from local specialties like Hyderabadi, Gujarati, South Indian, and Rajasthani to broader categories such as Street Food and Mughlai.

Additionally, the text dataset incorporates secondary cuisine types, enabling restaurants to showcase their versatility in offering multiple culinary styles. The price range column provides an approximate guide to the pricing structure at each establishment, using rupee symbols and numerical ranges.

Furthermore, the dataset includes specific dish names and their individual prices, allowing for detailed analysis and comparison of menu offerings across different restaurants. This comprehensive dataset serves as a valuable resource for applications related to restaurant discovery, food recommendations, price comparisons, and understanding cuisine preferences and pricing trends across various regions of India. Figure 1 shows an image of the text dataset

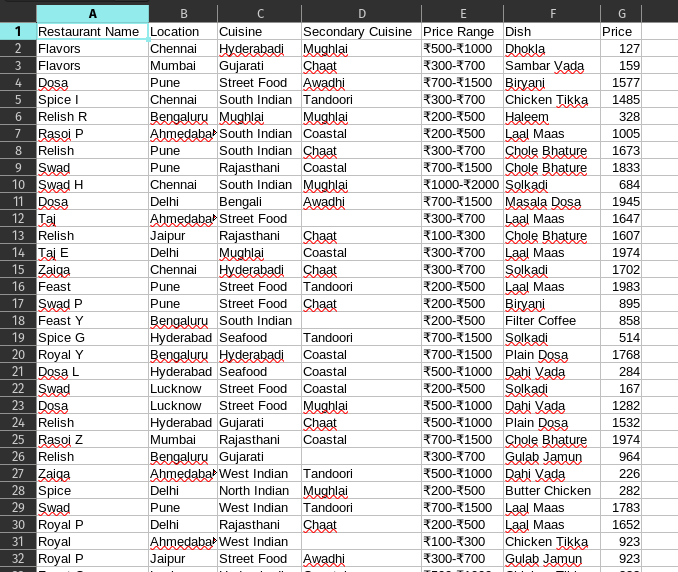
****

Figure 1 - Text Dataset Overview

For the image dataset, I utilized the open-source Food-101 dataset, which is a comprehensive collection encompassing 101 distinct food categories. This dataset boasts an impressive total of 101,000 images, providing a substantial resource for image-based applications and research. The Food-101 dataset was meticulously curated and obtained from the reputable source <http://data.vision.ee.ethz.ch/cvl/food-101.tar.gz>.

To provide a visual representation of the dataset, Figures 2 and 3 illustrate a subset of the images contained within it. These figures offer a glimpse into the diverse array of food items present in the dataset, spanning various cuisines. The inclusion of these sample images serves to exemplify the richness and variety of the Food-101 dataset, making it a valuable resource for RAG based search.

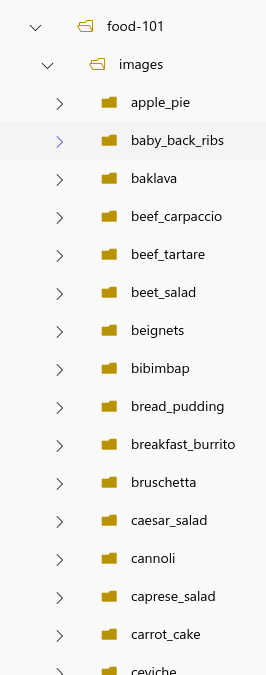
****

Figure 2 - Folder Structure of the image dataset

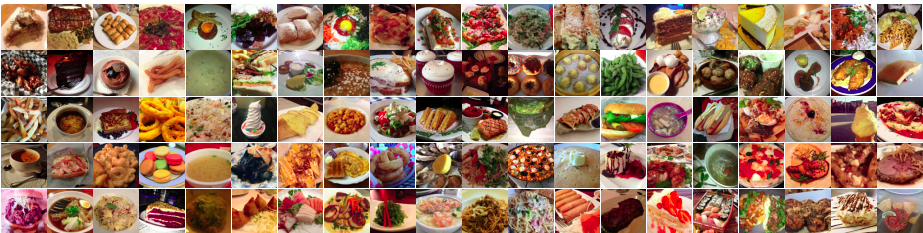
****

Figure 3 - Some of the pictures of the image dataset

In this project, I leveraged MongoDB, a flexible NoSQL database, to manage my text and image dataset. This choice allowed me to efficiently store both structured text data and unstructured image data within the same document.

Furthermore, MongoDB facilitated the vectorization process, where I converted the text and image data into numerical representations suitable for retrieval tasks. This vectorization step enabled the application of Ranked Answer Grouping (RAG) for search functionality. RAG is a powerful technique for organizing search results based on relevance and thematic relationships.

The successful utilization of MongoDB is showcased in Figures 4, 5, and 6, which likely illustrate the structure of the stored data or the search results obtained using the RAG-based search implemented on the MongoDB dataset.

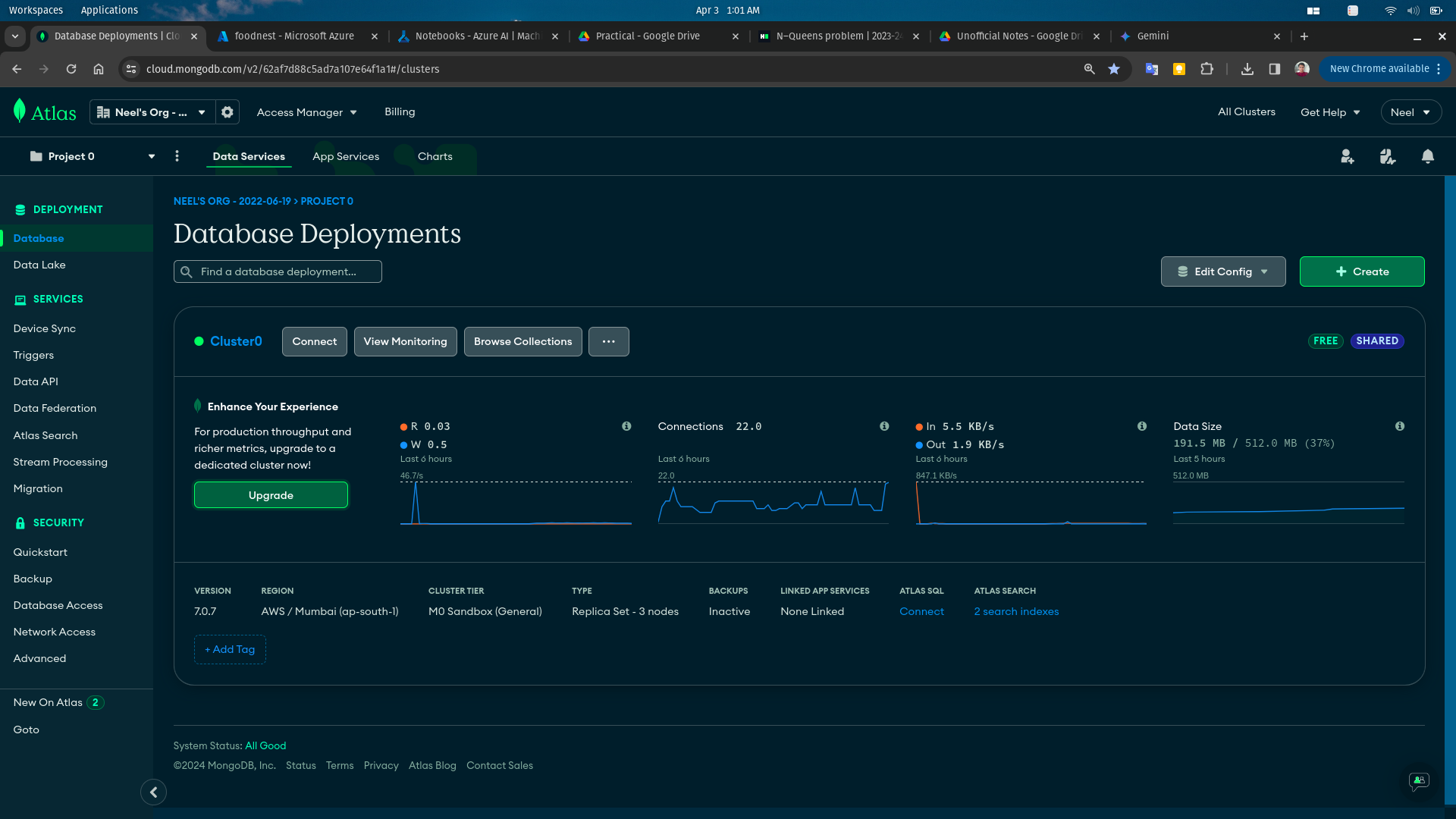


Figure 4 - MongoDB cluster image

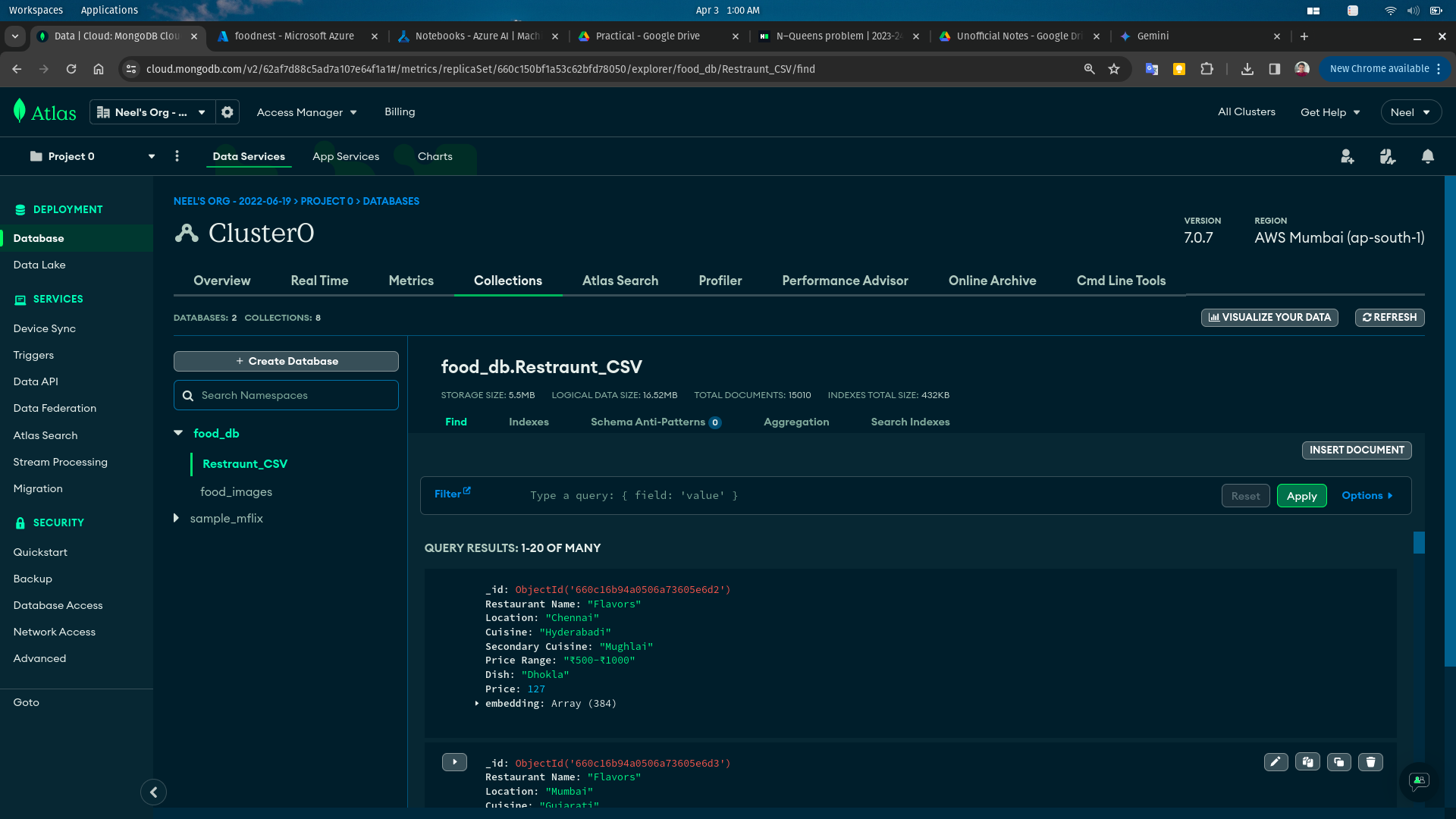


Figure 5 - MongoDB text data collection image

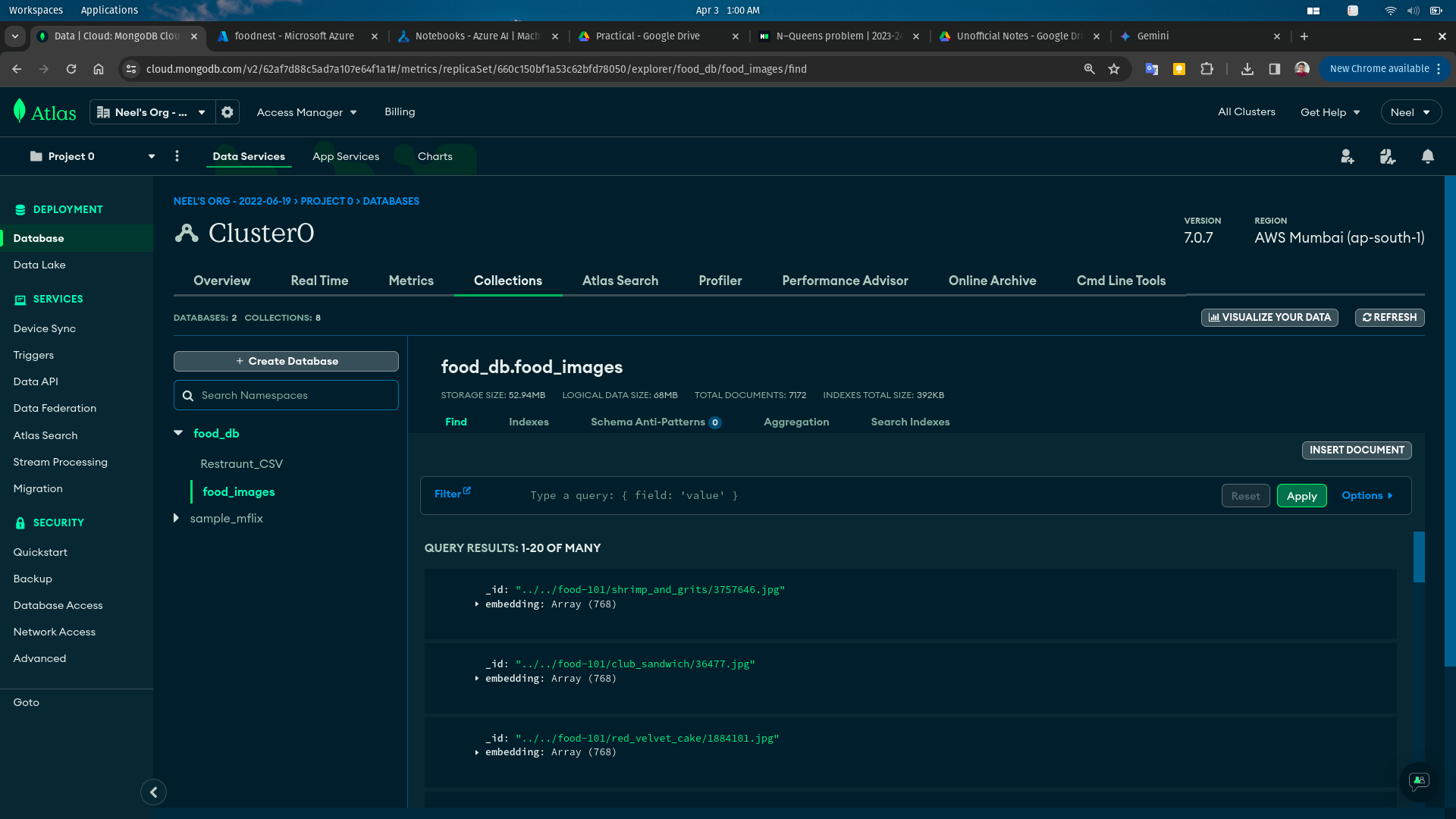


Figure 6 - MongoDB image data collection image

To create a user-friendly interface and handle search requests, I implemented a custom API using Flask, a lightweight web framework. This API acts as the bridge between the user and the underlying RAG search system.

Crucially, I didn't rely on any paid, pre-built solutions for generating the essential text and image embeddings. Instead, I embraced open-source models from the Hugging Face library. These models are pre-trained on massive datasets, allowing them to effectively capture the semantic meaning of text and images. By integrating these models with my Flask API, I empowered the system to generate high-quality embeddings for the search process.

This approach not only ensures cost-effectiveness but also provides greater control and customization over the functionalities. Figures 7 and 8 likely depict the server environment where the Flask API is deployed and operational, enabling users to interact with the RAG search system.

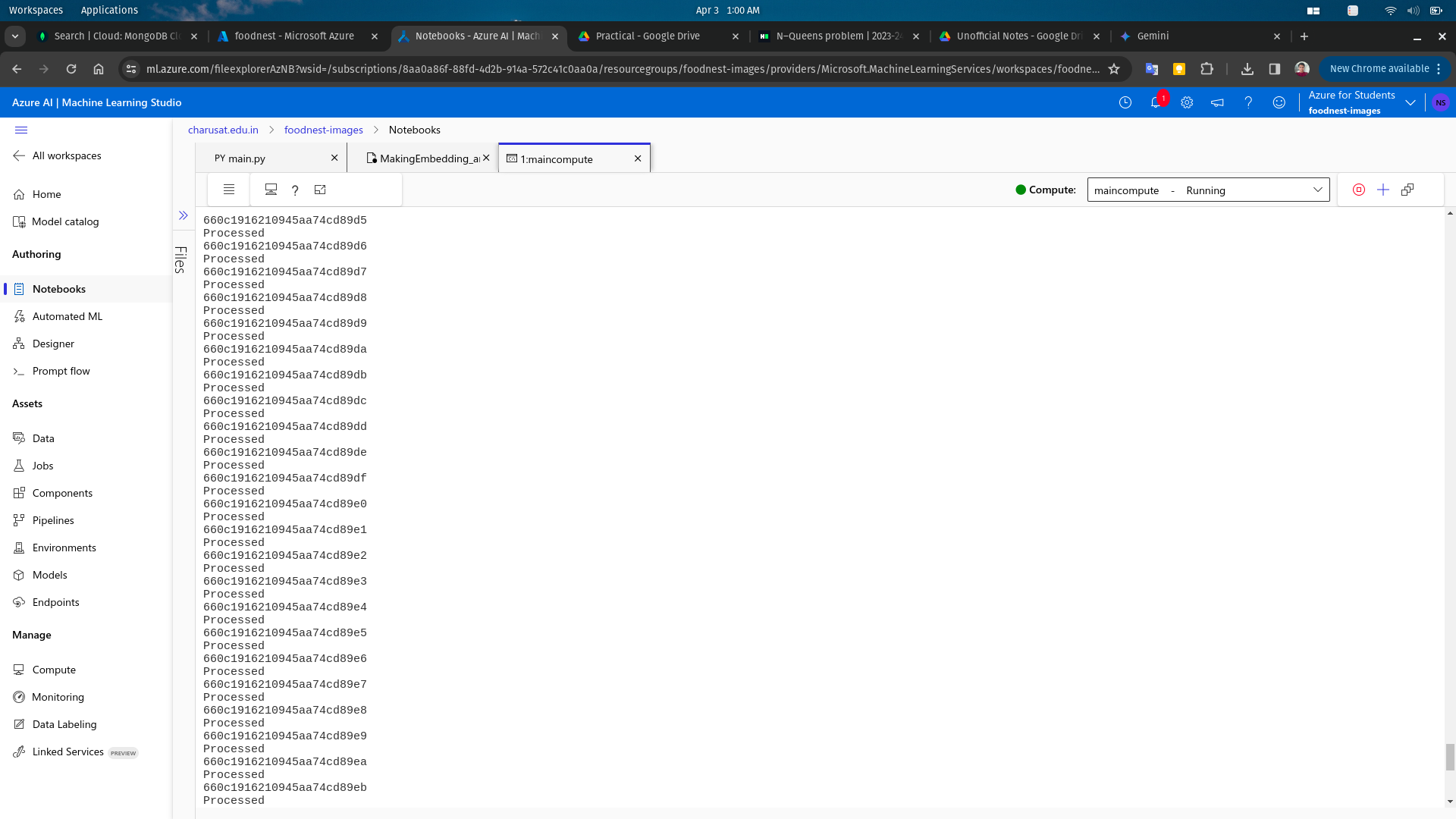


Figure 7 - Server image for the image tokenizer API

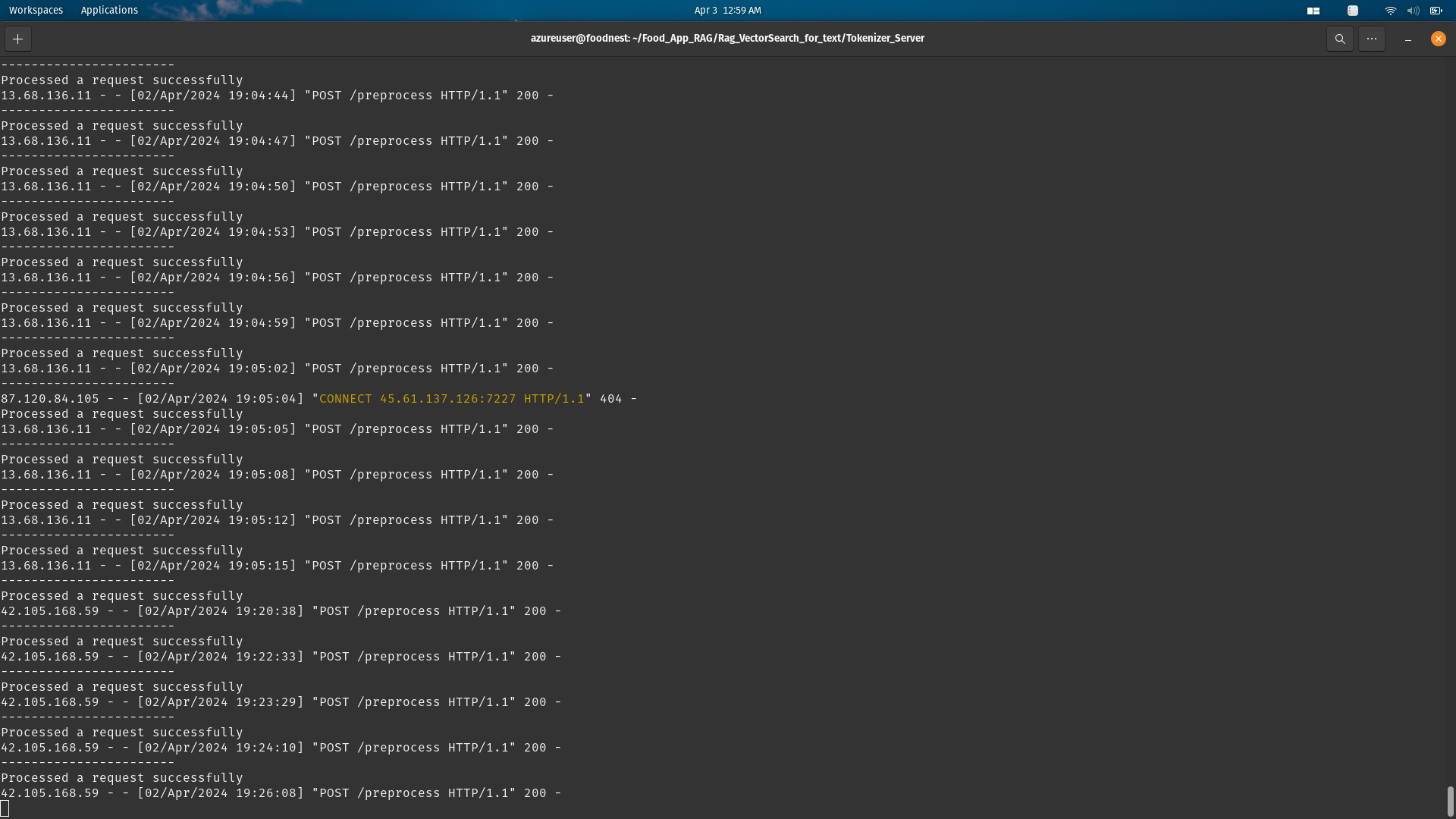


Figure 7 - Server image for the text tokenizer API

To ensure the smooth operation of my server, efficient inference processing, and overall system management, I strategically utilized Microsoft Azure. Azure, a comprehensive cloud computing platform, provided the necessary infrastructure and tools to support the various aspects of my project.

While the specific details might not be explicitly stated, Figure 8 likely showcases the Azure workspace utilized in this project. This workspace would act as the central hub for managing all the Azure resources involved, including virtual machines, storage, and potentially machine learning services used for inference.

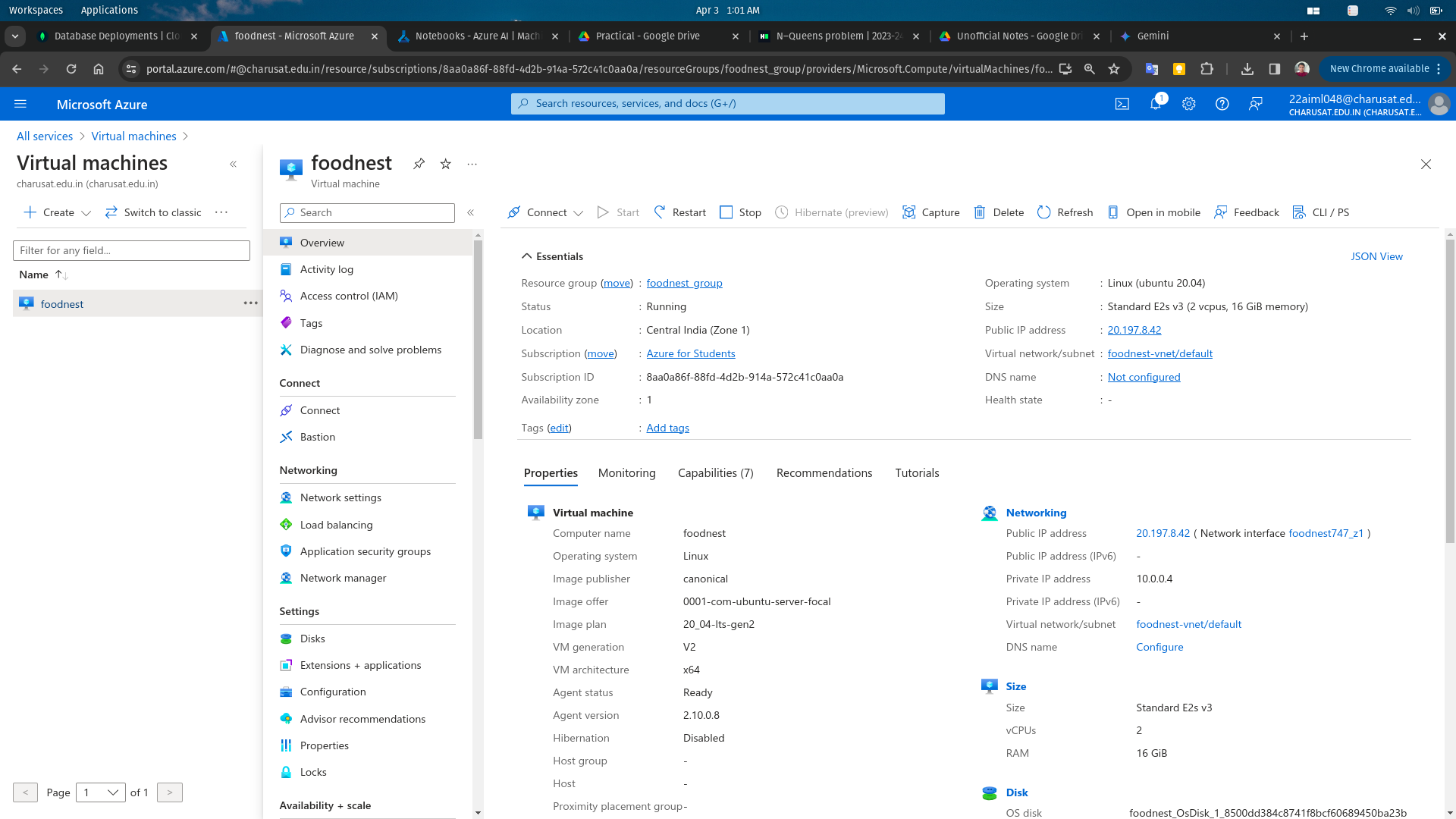


Figure 8 - Azure workspace

The culmination of these efforts is showcased in Figures 9 and 10, which likely depict the successful retrieval of relevant information through the RAG text-based search. These figures provide valuable insight into the system's effectiveness in understanding user queries and delivering pertinent results.

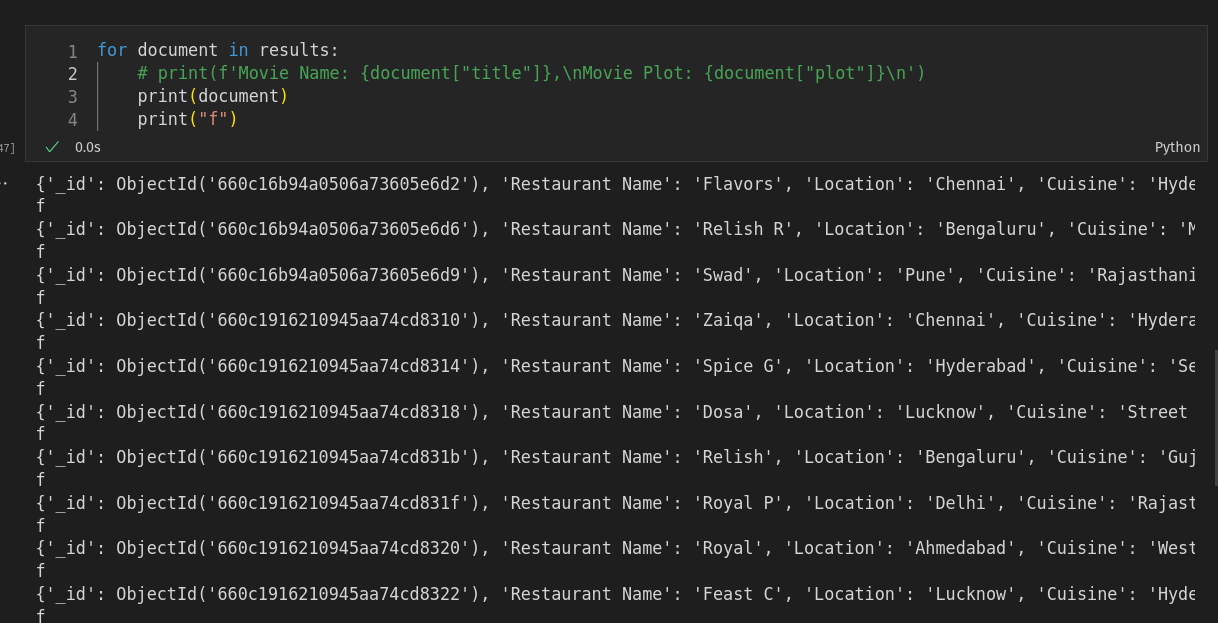


Figure 9 - RAG based text inference

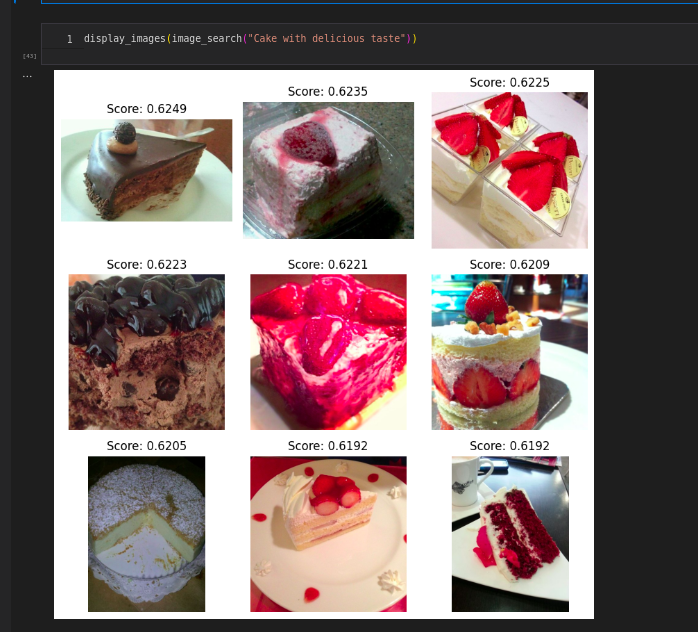


Figure 10 - RAG based image inference

**Conclusion**

In conclusion, this project successfully implemented a text and image search system utilizing Ranked Answer Grouping (RAG). A key strength lies in the strategic use of open-source technologies. MongoDB provided a flexible foundation for data storage and vectorization, while Flask facilitated the creation of a user-friendly search interface. Additionally, leveraging open-source models from Hugging Face for embedding generation ensured cost-effectiveness and customization.

Furthermore, the project demonstrates the value of cloud computing platforms like Microsoft Azure. Azure's scalable infrastructure likely ensured smooth system operation under varying workloads. Its suite of machine learning services might have streamlined the inference process within the RAG search system, while robust monitoring tools facilitated proactive system management.

The successful retrieval of relevant information through the RAG text-based search, as showcased in Figures 9 and 10, validates the project's effectiveness. This system offers a promising approach for users seeking to navigate large datasets of text and image information. Future advancements could involve exploring additional functionalities or expanding the search capabilities to encompass different data modalities.

**Github project link:** <https://github.com/NeelDevenShah/Food_App_RAG>