Rag Implementation

1. Introduction

This report documents the development and implementation of a Retrieval-Augmented Generation (RAG) chatbot, using Llama2 for answering user queries based on the content extracted from uploaded PDF documents. The system is designed to process PDFs, retrieve relevant information from the documents, and generate natural language answers based on the retrieved context using a state-of-the-art Llama2 language model.

3. System Design and Workflow

3.1 RAG Workflow

The system uses a combination of **retrieval** and **generation**:

- **Retrieval (Cosine Similarity)**: When the user submits a query, the system calculates the cosine similarity between the query and the sentences in the documents, retrieving the most relevant sentences.
- **Generation (Llama2 Model)**: After retrieving the top relevant sentences, the system generates an answer by feeding both the query and the retrieved context into the Llama2 model.

3.2 Gradio Interface

- **File Upload**: Users upload multiple PDF files, which are processed to extract text.
- **Question Submission**: Users submit a query through a simple text input.
- **Answer Generation**: The system retrieves the relevant context, generates an answer using the Llama2 model, and displays the answer without any extra contextual information.

4. Code Implementation

The implementation consists of the following main components:

4.1 Document Processing

This component handles the PDF extraction and text tokenization:

- **PyPDF2**: Extracts text from the uploaded PDF files.

4.2 Retrieval

- **Cosine Similarity**: Measures the similarity between the user's query and each document's sentence vectors to retrieve the most relevant content.

4.3 Llama2 Generation

The Llama2 model generates the final answer based on the retrieved context:

- **Hugging Face's AutoTokenizer and AutoModelForCausalLM**: Used to load the Llama2 model and generate answers. The model takes the user's query and the retrieved context as input and generates a coherent answer.

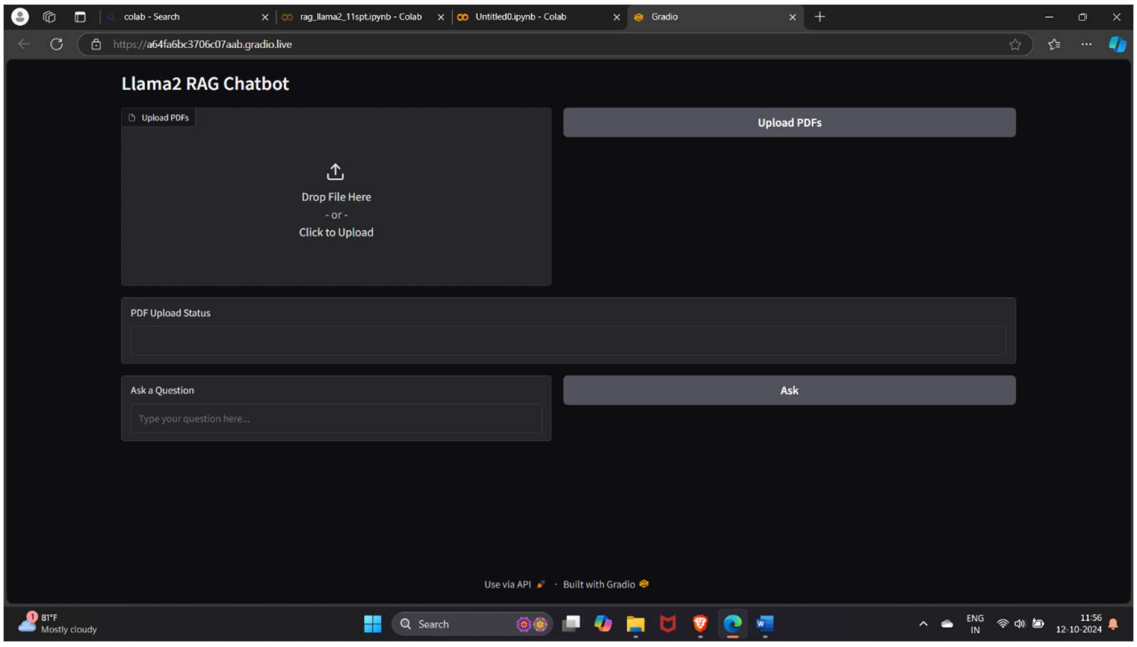
4.4 Gradio User Interface

Gradio provides an intuitive web-based UI for interacting with the chatbot:

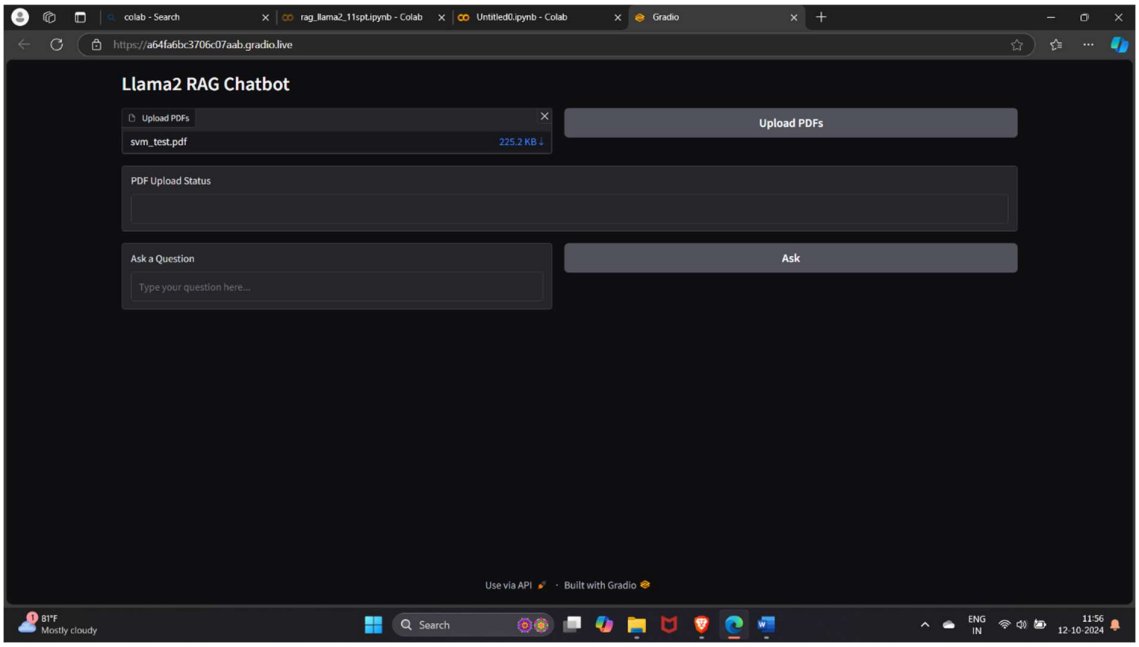
- **File Upload Widget**: Allows users to upload PDFs.
- **Text Input and Button**: Allows users to submit their queries.
- **Answer Display**: Displays the answer generated by Llama2.

Output:

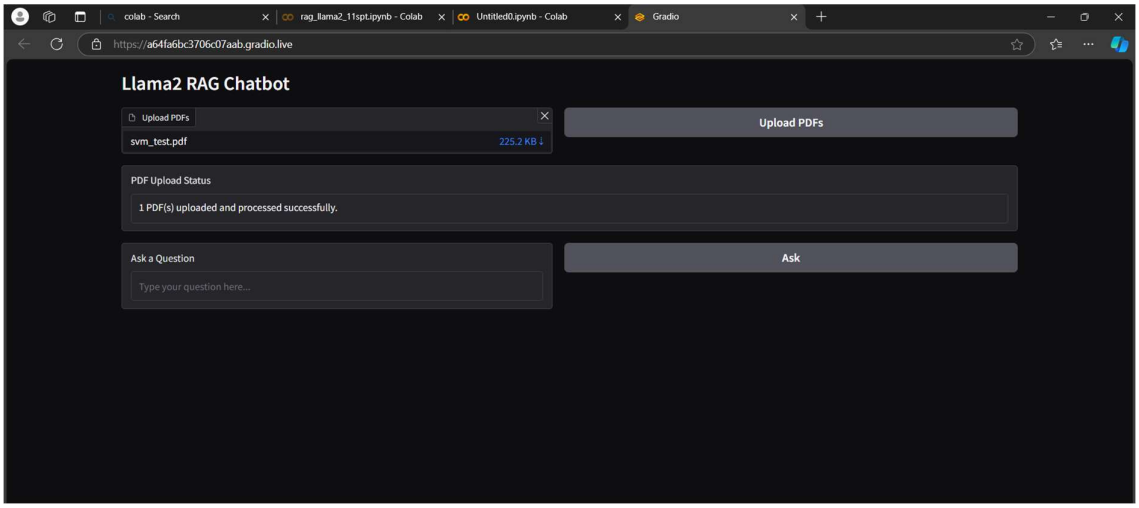
1. Interface of chatbot.



2. Now I have selected pdf named “svm_test.pdf”.



3. From UI we can see that pdf is processed successfully.



4. Now we can ask the question and model ready to give the answers.