College Name: VIT-AP UNIVERSITY
Student Name: ANKIT KUMAR JHAJHARIA
Email Address : ankit.23bce8390@vitapstudent.ac.in

GEN AI PROJECT SUBMISSION DOCUMENT

1. Project Title:

Smart PDF-Based Question Answering System Using Semantic Search and Generative AI

2. Summary of work done.

Proposal and idea submission

To develop a system that allows users to upload a PDF document and ask natural language questions, receiving intelligent, relevant answers based on the content of the uploaded file. The goal is to improve comprehension and access to information using semantic search and generative AI.

Technologies Used:

Tool/Library	Purpose
Python	Programming language
PyPDF2	Reading and extracting text from PDF files
Sentence Transformers	Generating semantic vector embeddings (all-MiniLM-L6-v2)
FAISS	Fast vector similarity search for information retrieval
NumPy	Numerical array manipulation
Gemini 1.5 Flash	Used to generate fluent and context-aware answers
Google Generative AI SDK	Interface to access Gemini 1.5 Flash
FastAPI + ngrok	(Optional) API testing and web access in Colab
Google Colab	Development and experimentation environment

System Architecture:

1. **PDF Upload**:
   User uploads a PDF file.
   Text is extracted using PyPDF2.

2. **Text Chunking:**
   Text is broken into logical paragraph-sized chunks to retain context.

3. **Embedding with Sentence Transformer:**
   Each chunkis embedded using **all-MiniLM-L6-v2** to connect text to vectors.

4. **Vector Indexing with FAISS**
   All chunk vectors are stored in a FAISS index for efficient semantic search.

5. **Question Input**
   The user's question is also embedded and used to find the most similar chunks.

6. **Answer Generation using Gemini 1.5 Flash**
   Retrieved top chunks are passed as "context" to the Gemini 1.5 Flash model.
   Gemini generates fluent, context-aware answers based on that input.

Evaluation:

- **Accurately retrieves and answers questions from complex PDFs.**
- **Gemini-generated answers are natural, relevant, and fluent.**
- **Fast response time after document indexing.**
- **Works well for academic papers, exam papers, notes, etc.**

3. Limitations:

- Requires internet access to use Gemini 1.5 Flash.
- If Gemini API is slow/unavailable, system hangs (as seen in Colab).
- PDFs with poor formatting may result in weak chunking or missing text.
- Without Gemini, only context chunks are returned (no AI-generated answers).

4. Future Improvements:

- Replace Gemini with a local LLM like Mistral or LLaMA 3 for offline answering.
- Add OCR support for image-based PDFs using Tesseract.
- Build a GUI or React frontend for user-friendly interaction.
- Add a caching system to avoid re-embedding frequently used PDFs.

5. Sample Output:

```
fp = "C:/Users/ankit/Downloads/1680782682phpTz5Dpt.pdf"
```

```
query = "What is the answer to Physics question 7?"
```

```
The correct answer to Physics question 7 is **(c) 374 m/s**.

The speed of sound in a gas is proportional to the square root of its absolute temperature.  Therefore, we can set u
p a ratio:

v₂ / v₁ = √(T₂ / T₁)

Where:

* v₁ = initial speed of sound (340 m/s)
* T₁ = initial temperature (27°C = 300 K)
* v₂ = final speed of sound (what we want to find)
* T₂ = final temperature (90°C = 363 K)

Solving for v₂:

v₂ = v₁ * √(T₂ / T₁) = 340 m/s * √(363 K / 300 K) ≈ 374 m/s
```

6. Conclusion:

This project demonstrates how semantic embedding and generative AI (Gemini 1.5 Flash) can be used together to enable intelligent document understanding. By combining **FAISS** for semantic search with **Gemini** for natural language answers, this system bridges the gap between unstructured PDFs and structured Q&A interaction.