Chat with PDF using Gemini 🤖

This is a Streamlit-based application that allows users to interact with PDF files using Google's Gemini Generative AI model. The app enables PDF table and image extraction, question-answering, and exporting conversations in .txt or .docx formats.

Features

- PDF Interaction:
  - Extract tables using Camelot or Tabula.
  - Extract images embedded within PDFs.
- Question Answering:
  - Ask questions based on the content of uploaded PDF files.
  - Uses Google's Gemini Generative AI for intelligent responses.
- Export Capabilities:
  - Export conversation history as .txt or .docx files for future reference.

Installation

1. Clone the repository:
2. git clone <repository-url>
3. cd <repository-directory>
4. Install the required Python packages:
5. pip install -r requirements.txt
6. Set up your environment variables by creating a .env file:
7. GOOGLE_API_KEY = <your_google_api_key>
8. GOOGLE_PROJECT_ID = <your_project_id>
9. Run the application:
10. streamlit run app.py

Dependencies

The application uses the following Python libraries:

- streamlit – For building the web interface.
- PyPDF2 – For PDF processing.
- langchain – For splitting text and building conversational chains.
- google.generativeai – For integrating with Gemini AI.
- FAISS – For creating and managing vector stores.
- camelot and tabula – For table extraction from PDFs.
- fitz (PyMuPDF) – For image extraction from PDFs.
- Pillow – For processing images.
- python-docx – For exporting conversations to Word documents.

How to Use

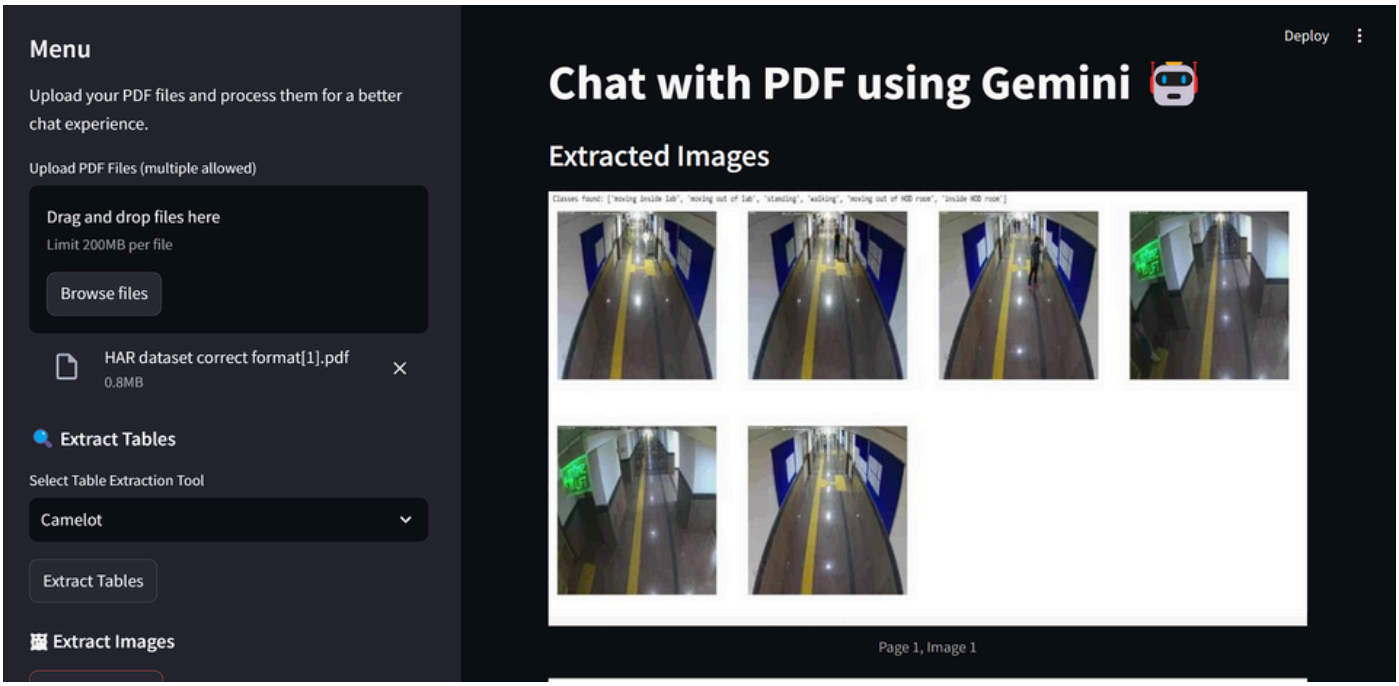
1. Upload PDFs: Drag and drop one or more PDF files into the file uploader in the sidebar.
2. Extract Tables or Images:
   - Select either Camelot or Tabula for table extraction.
   - Click on "Extract Images" to extract all embedded images.
3. Ask Questions:
   - Type your question into the input field and get AI-generated answers based on the uploaded PDF content.
4. Export Conversations:
   - Save the conversation history as .txt or .docx using the provided buttons.

Configuration

You can modify the following parameters:

- Model Temperature: Adjust the temperature in the Gemini AI model for controlling response creativity.
- Chunk Size: Update chunk_size and chunk_overlap in the RecursiveCharacterTextSplitter to suit your use case.

Screenshots



Limitations

- The app relies on the quality and content of the uploaded PDFs.
- Large PDFs may take longer to process.
- Some images or tables might not extract correctly depending on the structure of the PDF.