

Document Search Bot Documentation

Objective

This bot allows users to upload documents (PDF, Word, Excel, and PowerPoint files) and query information from them. It utilizes the **Gemini API (gemini-1.5-flash)** to process queries and deliver relevant answers based on the content of the uploaded documents.

Design & Approach

Architecture Overview

1. Frontend (React):

- a. Captures user input (file upload and query text).
- b. Sends data to the backend via REST API.
- c. Displays the query results or error messages.

2. Backend (Flask):

- a. Handles file uploads, text extraction, and query processing.
- b. Interacts with Gemini API for generating responses.
- c. Implements role-based access for Admin and Normal users.

3. Gemini API Integration:

- a. Uses the **gemini-1.5-flash model** for language understanding and query processing.
- b. Configured with the **GEMINI_API_KEY** environment variable.

System Design

Users

1. Admin:

- a. Upload and delete documents.
- b. Query the uploaded content.

2. Normal User:

- a. Query the uploaded content (read-only access).

File Handling

Supported File Formats:

- PDF
- DOCX (Word)
- XLSX (Excel)
- PPTX (PowerPoint)

File Constraints:

- **File Size Limit:** 2MB
- **Text Extraction Logic:**
 - **PDF:** Extracts text from all pages using PyPDF2==3.0.1.
 - **DOCX:** Reads all paragraphs using python-docx.
 - **XLSX:** Processes rows and cells using openpyxl==3.1.2.
 - **PPTX:** Extracts text from slides and shapes using python-pptx==0.6.21.

Input, Output, and Acceptance Criteria

Input:

- A file (PDF, Word, Excel, PowerPoint).
- A query text entered by the user.

Output:

- Relevant response to the query, derived from the uploaded document.
- Error messages for unsupported formats, missing input, or processing errors.

Acceptance Criteria:

1. **File Validation:**
 - a. File size must not exceed 2MB.
 - b. Only supported formats are processed.
2. **Response Quality:**
 - a. Accurate and concise.
 - b. Derived exclusively from the document's content.
3. **Error Handling:**

- a. Clear error messages for unsupported formats or missing inputs.

Environment Setup

Python Dependencies

The backend uses the following Python libraries:

- Flask==2.3.2 (Web Framework)
- python-dotenv==1.0.0 (Environment Variable Management)
- google-generativeai==0.3.0 (Gemini API Integration)
- Werkzeug==2.3.3 (File Handling Utilities)
- PyPDF2==3.0.1 (PDF Text Extraction)
- openpyxl==3.1.2 (Excel Text Extraction)
- python-pptx==0.6.21 (PowerPoint Text Extraction)

Environment Variable

To use the Gemini API, set the following environment variable in your .env file:

GEMINI_API_KEY=AIzaSyA6TrKxIv-ppEth9_62oaw-zw42lGdqqZw

Process Workflow

1. File Upload:

- a. User uploads a document (Admin or Normal User).
- b. The backend validates the file type and size.
- c. Extracts text using the appropriate library based on the file type.

2. Query Handling:

- a. User submits a query related to the uploaded content.
- b. Backend sends the extracted text and query to the Gemini API.
- c. Gemini API processes the input and returns a relevant response.

3. Response Display:

- a. The frontend displays the AI-generated response.
- b. Errors (e.g., unsupported formats, file size issues) are shown to the user.

Example Input/Output

Input:

- File: Report.pdf (1.5MB)
- Query: "What are the key findings from the report?"

Output:

- Response: "The key findings include increased revenue by 25% and a decrease in operational costs by 15%."

Error Example:

- Input File: Image.png
- Error: "Unsupported file type. Please upload a PDF, Word, Excel, or PowerPoint file."

Test Cases

File Upload Tests:

Test Case	Input	Expected Output
Upload valid PDF file	Report.pdf (1MB)	File successfully processed.
Upload unsupported file	Image.png	Error: Unsupported file type.
Upload file exceeding size limit	LargeFile.pdf (3MB)	Error: File size exceeds 2MB.

Query Processing Tests:

Test Case	Query	Expected Output
Valid query on uploaded content	"Key metrics?"	Relevant response from document content.
Query with no uploaded document	"What is the summary?"	Error: No document uploaded.

Role-Based Access Tests:

Test Case	User Role	Expected Behavior
Admin uploads a file	Admin	File successfully uploaded.
Normal user attempts file upload	Normal User	Error: Insufficient permissions.

