

Project Report: DocuMind AI

Intelligent Document Understanding & Validation Assistant

Made by:

Benkhouili Aya

Belkouri Wissal

Ahmed Louai Benkaddour

Ilyas Sandy

Oussama Boukhalifa

TPE-CYB

2025-2026

1. Introduction

Modern users manage large numbers of important documents such as passports, certificates, contracts, and academic records. Traditional document management systems focus mainly on storing and organizing files, but they do not understand document content, user goals, or task requirements. As a result, users must manually check their files, search for missing documents, and interpret complex requirements on their own.

DocuMind AI was created to address this limitation. Instead of acting as a simple storage platform, it functions as an intelligent document assistant. The system understands user intent, analyzes document content, and provides meaningful feedback. By combining Google Drive integration, semantic search, and large language models, DocuMind AI transforms raw documents into structured knowledge.

When users ask questions such as “*Do I have all the documents for a visa?*” or “*What is missing from my application?*”, the system reasons over the available files and gives clear, human-readable guidance. This makes the process faster, more reliable, and far less stressful.

2. Problem Statement

Traditional document management systems:

- Only store files
- Do not reason about content
- Do not understand user goals
- Do not validate completeness
- Do not provide intelligent feedback

Example problem:

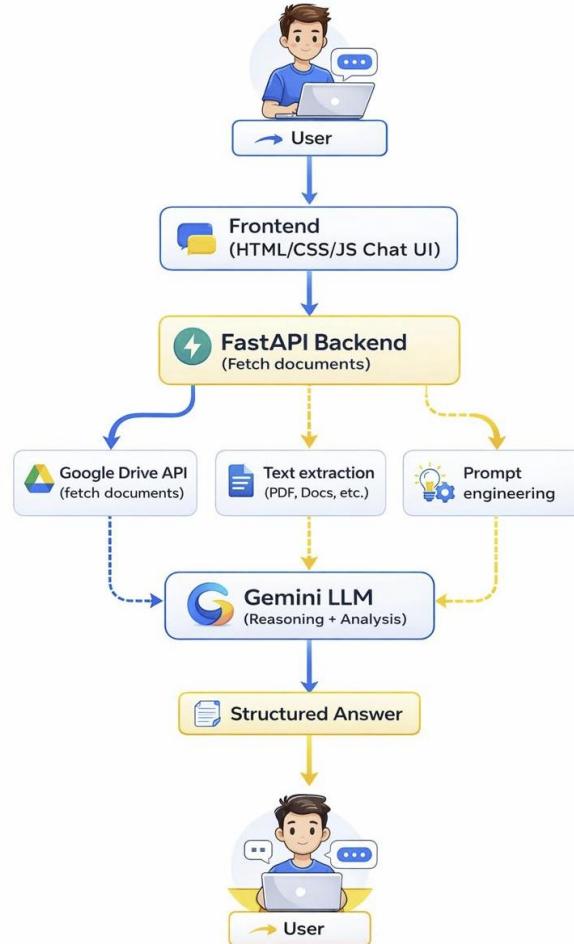
A user wants to apply for a visa, job, or university. They upload documents — but they don’t know if anything is missing.

DocuMind AI solves this by:

- ✓ Understanding the user’s request
- ✓ Analyzing their documents
- ✓ Comparing against inferred requirements
- ✓ Reporting what is present, missing, or unclear

3. System Overview

High-Level Architecture:



4. Technologies Used

Backend:

- **Python**
- **FastAPI** – API framework
- **Uvicorn** – ASGI server
- **Google Drive API** – file access
- **OAuth / Service Accounts** – authentication
- **dotenv** – environment variables

AI:

- **Google Gemini API**
- Prompt engineering
- Context injection
- Structured reasoning

Frontend:

- HTML
- CSS (modern UI)
- JavaScript (fetch API)
- Chat-based interface

5. Key Features

a) Intelligent Understanding

The assistant does not just summarize, it understands the **intent**:

- ✓ Validation
- ✓ Checklist
- ✓ Explanation
- ✓ Summary
- ✓ Comparison

b) Document Reasoning

The model is instructed to:

- Use ONLY provided documents as facts
- Clearly distinguish:
 - What is present
 - What is missing
 - What is unclear
- Infer external standards logically (e.g., visa rules)

c) Live Google Drive Integration

- Fetches files automatically
- Extracts text
- Feeds content to AI
- Adapts to newly uploaded files

d) Chat Interface

- Natural language interaction
- Real-time responses
- Modern UI
- No technical knowledge required

6. Backend Design

Document Processing

- PDF text extraction – Extract text from PDF files
- OCR (Optical Character Recognition) – Extract text from scanned images
- Automatic document type detection – PDF vs image handling
- Text cleaning & chunking – Prepare documents for AI analysis

Retrieval & Intelligence

- Vector Store (Embeddings) – Store document chunks for semantic search
- Similarity Search – Retrieve the most relevant document parts per question
- Context Injection – Provide AI only with relevant document content

RAG (Retrieval-Augmented Generation)

- Documents retrieved from Google Drive
- Relevant content injected into the LLM prompt
- Accurate, document-aware answers

Primary Dataset: Google Drive Documents

Our primary dataset consists of the user's personal documents stored in Google Drive, such as PDFs, scanned documents, and text files."

➤ **Important points:**

- Not static
- Not pre-downloaded
- Loaded dynamically at runtime
- Becomes the knowledge base

This is document-based RAG, not classical ML training.

Processed Dataset: Extracted & Chunked Text

After extraction, we create our own internal dataset:

- PDF text
- OCR text
- Text chunks
- Vector embeddings

which acts as a temporary vector database for semantic search

Hugging Face Embeddings Dataset

We used a pre-trained Hugging Face embedding model, trained on large public text corpora, to convert document chunks into vectors.

LLM Knowledge (Secondary / Reasoning Layer)

Gemini brings general world knowledge, not facts about the user.

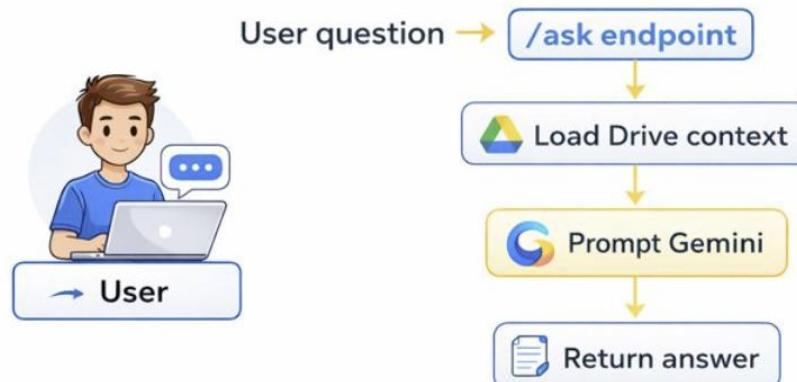
FAISS (Facebook AI Similarity Search) is used in our project as the core vector search engine that enables fast and accurate retrieval of relevant information from documents stored in Google Drive. After extracting and chunking document text, we convert it into numerical embeddings using a Hugging Face embedding model. FAISS indexes these embeddings and performs semantic similarity search to quickly identify the most relevant document chunks for a user's question.

app.py (FastAPI Server)

Responsibilities:

- ❖ Handle user requests (/ask)
- ❖ Load document context
- ❖ Call the LLM
- ❖ Return structured answers

Example Flow:



drive_client.py

Handles:

- ✓ Google Drive authentication
- ✓ Listing files
- ✓ Downloading files
- ✓ Exporting Google Docs

drive_loader.py

- ✓ Calls drive_client

- ✓ Extracts text
- ✓ Merges all files into a single context
- ✓ Sends it to the LLM

llm_client.py

This is the brain of the system.

- ✓ Builds the prompt
- ✓ Injects documents
- ✓ Injects user question
- ✓ Calls Gemini
- ✓ Returns structured reasoning

7. Prompt Engineering Strategy

Your system uses a **role-based prompt**:

“You are DocuMind AI, an intelligent document assistant.”

It enforces:

1. Intent detection
2. Task classification
3. Fact-only reasoning from docs
4. Logical inference
5. Structured output

Output Structure

The AI always tries to separate:

✓ Present

What exists in the documents

✗ Missing

What should exist but doesn't

✗ Unclear

What cannot be determined

8. Frontend Design

UI Components

- Chat window
- User message bubbles
- AI message bubbles
- Input box
- Send button

JavaScript Flow

User types → **sendMessage()**

- fetch(/ask?q=...)
- backend processes
- AI responds
- message rendered

Why Chat UI?

Because:

- Natural
- Intuitive
- No training required
- Feels like a real assistant

9. Google Drive Integration

How it Works

- Service account authentication
- File listing
- Download or export
- Text extraction
- Context merging

Problem Solved

Google Docs cannot be downloaded directly.

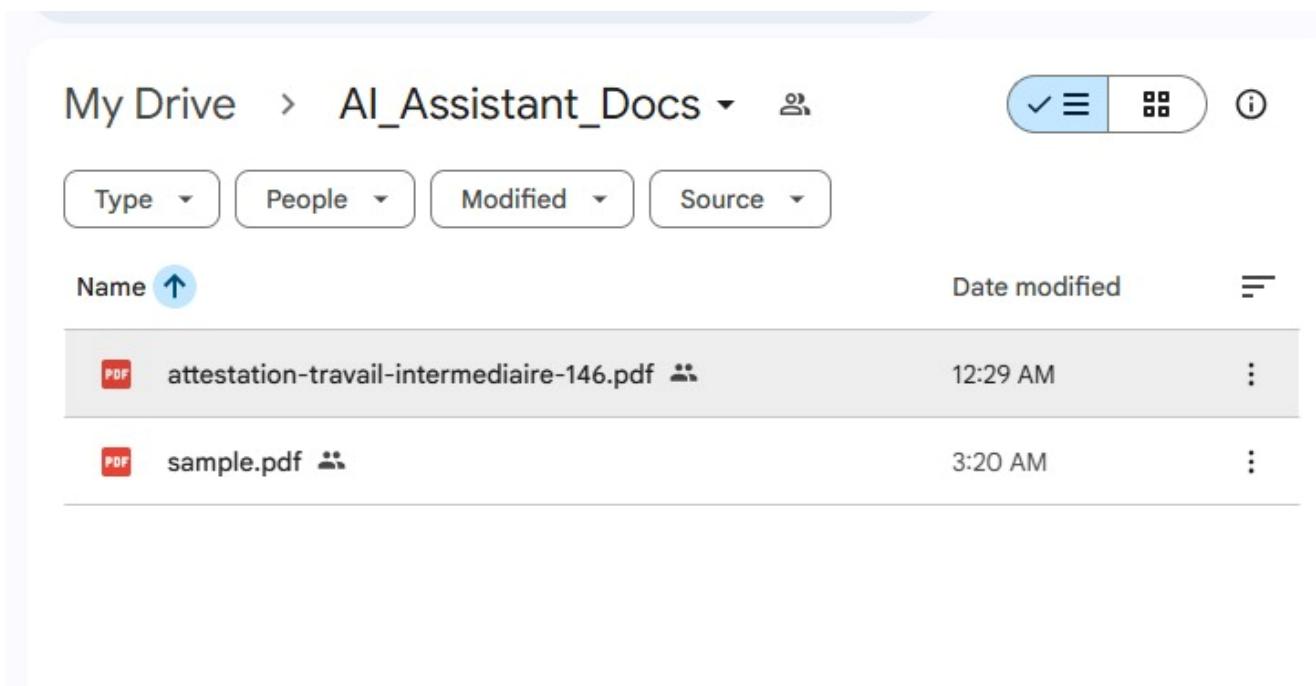
Solution:

- ✓ Use `files().export()`
- ✓ Convert to text
- ✓ Then analyze

10. Demo Flow

Step 1: User Uploads Files

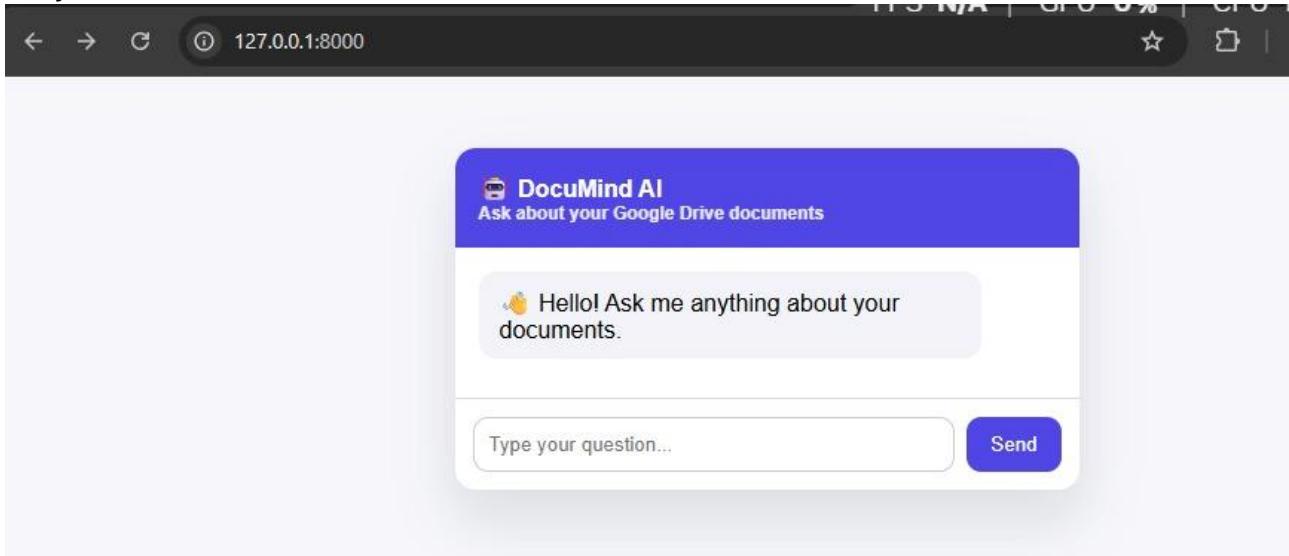
User adds documents to Google Drive.



The screenshot shows a Google Drive interface. At the top, it displays 'My Drive > AI_Assistant_Docs'. Below the navigation bar are four filter buttons: 'Type', 'People', 'Modified', and 'Source'. The main area lists two files: 'attestation-travail-intermediaire-146.pdf' (modified at 12:29 AM) and 'sample.pdf' (modified at 3:20 AM). The columns are labeled 'Name' (with an upward arrow), 'Date modified', and three vertical dots for more options.

Step 2: User Opens Web Interface

They see a chat interface.



The screenshot shows a web browser window with the URL '127.0.0.1:8000'. The page features a blue header bar with the 'DocuMind AI' logo and the text 'Ask about your Google Drive documents'. Below this is a white chat-like interface with a message bubble containing a yellow robot icon and the text 'Hello! Ask me anything about your documents.' At the bottom is a input field with the placeholder 'Type your question...' and a blue 'Send' button.

Step 3: User Asks a Question

Example:



Step 4: Backend Loads Drive Context

All files are read.

Step 5: AI Analyzes

- Reads documents
- Understands the goal
- Reasons
- Compares requirements

Step 6: Structured Answer

User receives:

- ✓ What they have
- ✓ What they lack
- ✓ What's unclear

 DocuMind AI
Ask about your Google Drive documents

 Hello! Ask me anything about your documents.

i wanna apply for spain visa

Of course, I can help you with that. Applying for a visa can seem complicated, but it's a straightforward process once you break it down. Your goal is to get a visa to visit Spain.

Let's look at the process and what you'll need.

Understanding the Spain Visa (Schengen Visa)

When you apply for a visa to Spain, you are typically applying for a **Schengen Visa**. This visa allows you to travel not only to Spain but also to the other 28 countries in the Schengen Area for up to 90 days.

To get this visa, you need to prove to the Spanish consulate:

- * The purpose of your trip (e.g., tourism, business).
- * That you have enough money to support yourself during your stay.
- * That you have strong reasons to return to your home country after your trip (like a job, family, or property).

Step 1: Reviewing Your Current Documents

Based on the files you provided, here is what we have. I must be direct: **these documents are not suitable for a new Spain visa application.**

- * **Flight Itinerary ('sample.pdf'):** This is a booking confirmation for a trip to **New York, USA in 2021**. For your Spain application, you will need a new, current flight reservation to and from Spain.
- * **Work Certificate ('attestation-travail-intermediaire-146.pdf'):** This is a proof of employment for a Mr. Salah Ayari that was issued in **2013**. Visa authorities require recent documents (usually issued within the last month). You will need a new, up-to-date letter from your current employer.

11. Error Handling

- Handled issues:
- Missing credentials
- Google Docs export errors
- API failures
- Empty responses
- Network errors

12. Security

- No hardcoded API keys
- .env file
- Service accounts
- Scoped permissions

13. Scalability

The system can be extended to:

- ✓ Legal documents
- ✓ HR onboarding
- ✓ University applications
- ✓ Job screening
- ✓ Insurance
- ✓ Compliance

15. Why This Project is Innovative

Unlike traditional systems, DocuMind AI:

Feature	Traditional Systems	DocuMind
Storage	✓	✓
Search	✓	✓
Summarize	✓	✓
Understand intent	✗	✓
Reason	✗	✓
Validate	✗	✓
Explain gaps	✗	✓

16. Conclusion

DocuMind AI demonstrates how artificial intelligence can move document systems beyond simple storage into intelligent assistance. The platform understands documents, reasons about user goals, and provides actionable feedback rather than raw file lists.

Through the integration of cloud storage, AI-based text understanding, and semantic retrieval, the system dynamically adapts as new documents are added. It identifies what is present, what is missing, and what is unclear—helping users make informed decisions.

This project highlights the potential of AI-driven document reasoning in real-world scenarios such as visa applications, job searches, and academic admissions. In the future, DocuMind AI can be expanded with multilingual support, OCR, deadline tracking, and automated compliance checks.

Overall, DocuMind AI transforms document management from a passive system into an intelligent, interactive assistant.