

AI-Powered Document Intelligence Automation Platform with Gradio Chatbot



Hello!

Demo & Reflection

Author: Lashawn Fofung, MBA, PMP, CSM, CSPO

Background

- **Challenge:**

- Critical information is often trapped in high-volume, unstructured formats—specifically large PDF reports and scanned documents.
- Manual retrieval is time-consuming, prone to human error, and creates a "data bottleneck" where decisions are delayed because the right facts cannot be found quickly.

- **Solution:**

- Intelligent RAG Architecture
- AI-powered Document Intelligence system built on a **Retrieval-Augmented Generation (RAG)** framework. Unlike standard search tools, this system:
 - **Sees with OCR:** Uses **Pytesseract** to convert scanned images into machine-readable text.
 - **Reasons with Structure:** Employs **Boundary Detection** to logically split documents (e.g., distinguishing between a "Financial Table" and a "Summary Report").
 - **Finds with Vectors:** Utilizes **FAISS** and **Semantic Embeddings** to find answers based on *meaning* rather than just keywords.

Pipeline Components

Each component handles a specific data challenge:

1. OCR Ingestion (Pytesseract)

- The entry point of the pipeline. It converts raw PDF images and scans into machine-readable text.
 - **Role:** Extracting text from unstructured visual data.
 - **Key Challenge:** Handling "noisy" scans that produce non-printable control characters.

2. Triple-Layer Sanitization

- The defensive core of the system. It scrubs text at three critical stages: before AI processing, during chunking, and before PDF export.
 - **Role:** Stripping null bytes (`\x00`) and non-ASCII characters.
 - **Key Challenge:** Preventing the "Zero-Char" JSON parsing error that crashes LLM logic.

Pipeline Components

Each component handles a specific data challenge:

3. Logical Boundary Detection

- A semantic layer that uses AI to "read" the document structure.
 - **Role:** Identifying where one logical document ends and another begins (e.g., separating a cover letter from a financial report).
 - **Key Challenge:** Maintaining accuracy when document layouts are inconsistent.

4. Sliding Window Chunking

- The process of breaking long text into searchable segments for the vector database.
 - **Role:** Creating 1,000-character segments with a 200-character overlap.
 - **Key Challenge:** Ensuring figures like "Total Assets" aren't cut in half, which would destroy retrieval accuracy.

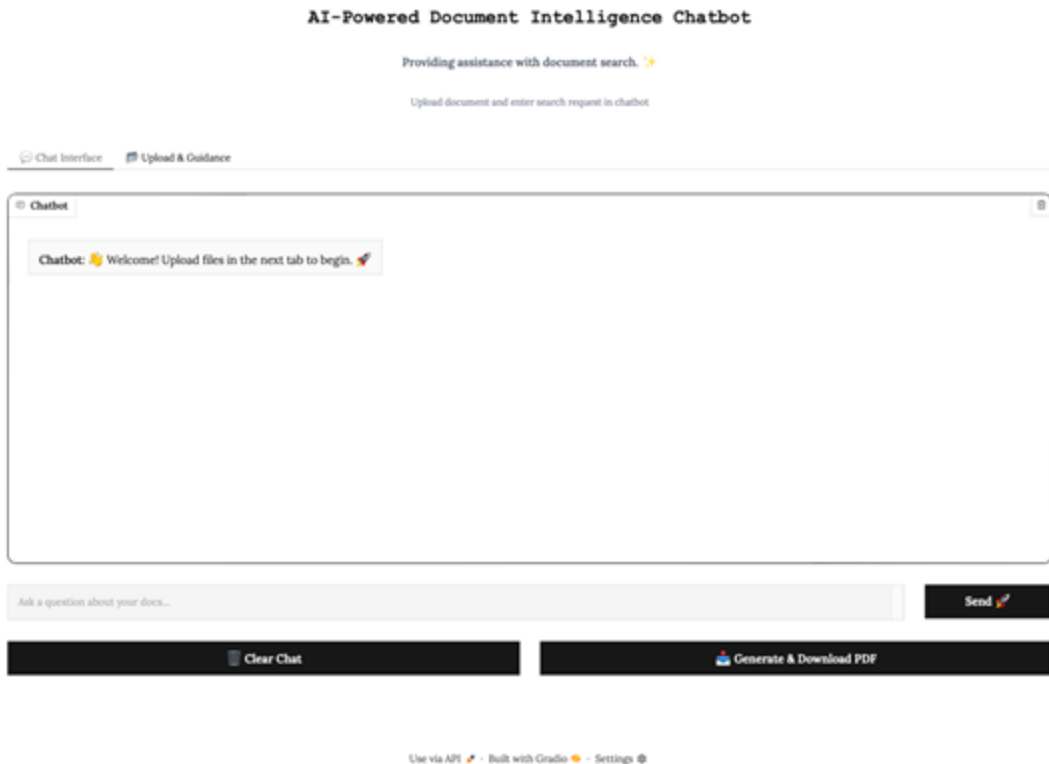
Pipeline Components

Each component handles a specific data challenge:

5. Vector Indexing & Retrieval (FAISS)

- The "brain" that allows the chatbot to find answers.
 - **Role:** Converting text into mathematical vectors and storing them for semantic search.
 - **Key Challenge:** Efficiently matching user queries to the most relevant document chunks.

Gradio Chat Interface Tab



- Basic UI Layout Deepsite Monochrome
- Active Buttons
 - Welcome message in chat window
 - “Ask Question” box for query
 - Send “Submit” button
 - Clear Chat History button
 - Generate & Download Chat History
 - PDF
- Labeling Button
 - Chat Message
 - You (input)
 - Chatbot (output)

Gradio Upload & Guidance Tab



- Basic UI Layout Deepsite Monochrome
- Upload & Guidance Tab
 - Upload PDF
 - “X” to delete file
 - Initialize AI Index
- Labeling Button
 - System Status & Metadata
 - Shows file metadata (pages, file size)
 - Instructions to return to Chat Interface tab once document indexed

Document Upload & Indexed

AI-Powered Document Intelligence Chatbot

Providing assistance with document search. ✨

Upload document and enter search request in chatbot

[Chat Interface](#) [Upload & Guidance](#)

🔧 Pro-Tips for Best Results

For best results, always upload clean, high-quality documents so the system can process them accurately. Use document type filters to narrow your search and avoid irrelevant results. If your query is broad, try breaking it into smaller, more specific questions – this often leads to more accurate answers. And remember, the magic happens when your data is well-prepared and your questions are clear, because even the smartest AI needs good input to give great output.

Upload your document 📎

Lashawn Fofung JQM2 TASK 2 PO Tread Bikes STOCKHOLDER REPORT.pdf728.7 KB ⓘ

Initialize AI Index ✨

System Status & Metadata

📎 File: Lashawn Fofung JQM2 TASK 2 PO Tread Bikes STOCKHOLDER REPORT.pdf

📏 Size: 0.71 MB

📄 Total Pages: 29

🔍 AI DETECTED: Stockholder Report (Pgs 1-29)

✅ Ready! AI Indexing Complete.

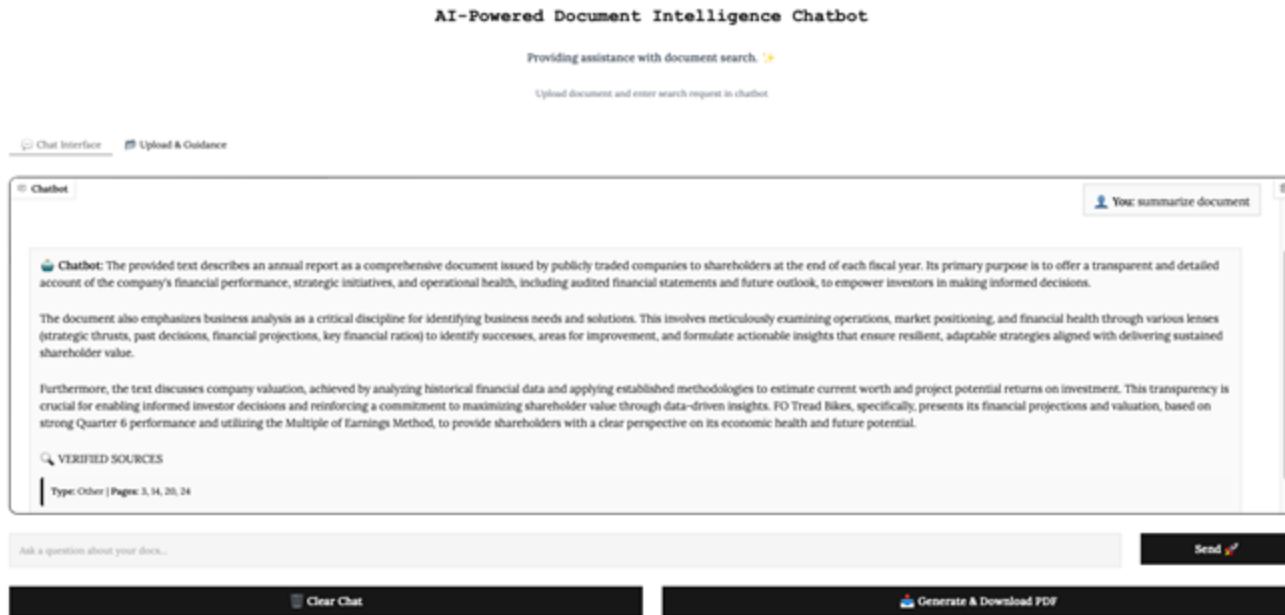
💡 You can now start chatting in the 'Chat Interface' tab.

🔥 System is optimized and ready for queries.

Use via API 🔗 · Built with Gradio 🍷 · Settings ⚙️

- File Uploaded: MBA Capstone PDF
- Before clicking Analyze & Index button
 - Filename
 - File size
- After clicking Analyze & Index
 - Processing time
 - System status shows number of pages
 - Instructions to ask questions about document in the “Chat Interface” tab

Query 1: Simple Question



- **Ask:** Summarize Document
- With each query, processing time is displayed
- **Chatbot Response:** Matches document information
- Source of information displayed in chat window (type of document, page number)

Query 2: Testing Memory & Calculations



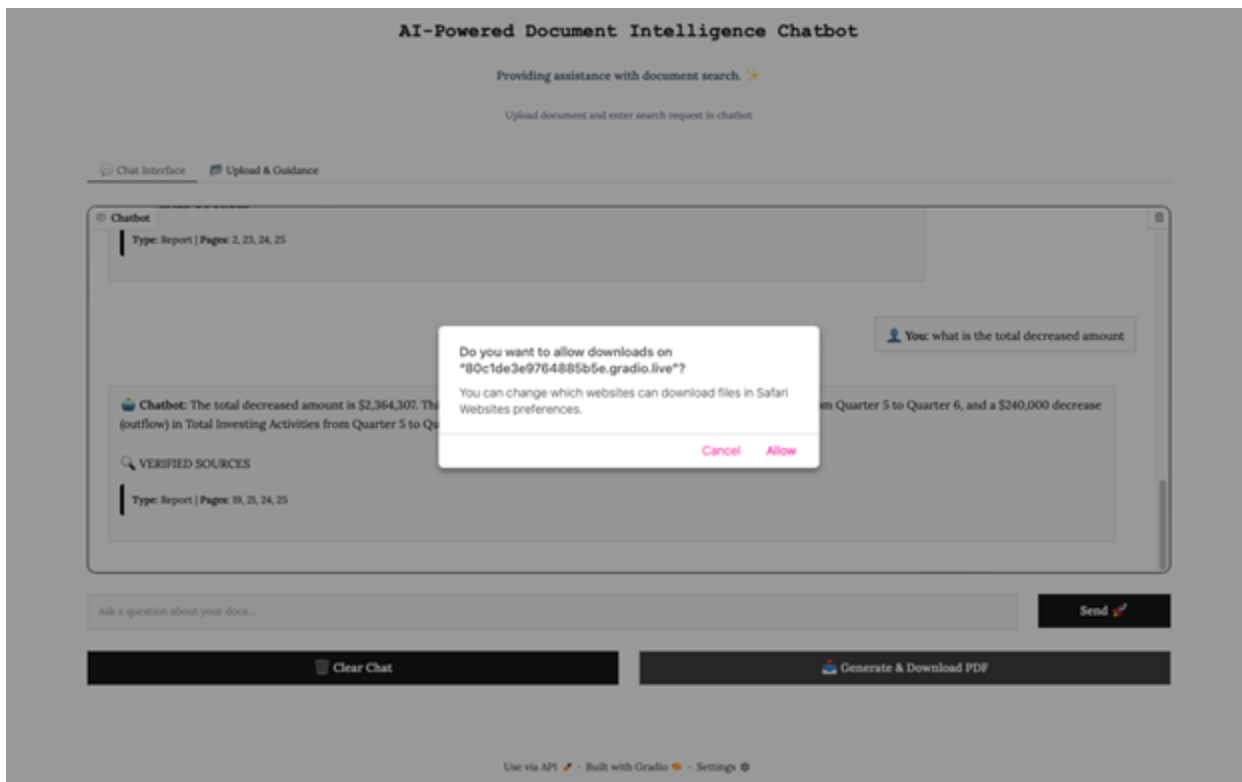
- **Ask:** Amount of Total assets
- **Chatbot Response:** detailed with information from document
- Will use this information to test memory and do calculations for new information not in document

Query 3: Testing Memory, Calculations, & New Information



- **Ask:** Follow up question taking previous response and asking to calculate a potential difference (new information not in document)
- **Chatbot Response:** Calculated difference

Generate & Download PDF



- Click on Generate & Download PDF button
- Downloads automatically as PDF file



Technical Challenges Overview



- Learning from technical issues transforms a system failure into a roadmap for robustness, ensuring that the same vulnerability never compromises the user experience twice.
- It shifts the development focus from reactive "firefighting" to proactive engineering, ultimately building a more resilient and trustworthy product.
- Technical Challenges Faced:
 - The "Zero-Char" & JSON Parsing Failure
 - Retaining Retrieval Accuracy
 - Engineering Solutions Implemented

The "Zero-Char" & JSON Parsing Failure

- **The Issue:** The system frequently encountered a **Splitting error: Expecting value: line 1 column 1 (char 0)**.
- **The Root Cause:** This was a "Data Poisoning" issue. Pytesseract would occasionally extract null bytes (\x00) or non-ASCII control characters from blurry scans. When this "dirty" text was sent to the AI Boundary Detector, the LLM returned an empty or malformed string, causing the JSON parser to crash at the first character (Char 0).
- **The Resolution:** Implemented a **Triple-Layer Sanitization Pipeline**.
 - **Input Layer:** Cleaning text before it hits the AI Boundary Detector.
 - **Processing Layer:** Scrubbing chunks before they enter the FAISS vector database.
 - **Output Layer:** Stripping incompatible symbols before PDF generation to prevent export crashes.

Retaining Retrieval Accuracy

- **The Issue:** Accuracy dropped when the AI could not find specific data points like "Total Assets," returning: *"The context does not contain information."* The AI often reported that "Information was not found" (e.g., Total Assets), even when the data existed in the PDF.
- **The Root Cause:** Inaccurate chunking. If a critical financial figure was split across two separate text chunks without overlap, the semantic meaning was lost, and the retriever could not "see" the full context.
- **The Resolution:** Transitioned from fixed-size splitting to Sliding Window Chunking. By using a 1,000-character window with a 200-character overlap, we ensured that every data point is captured in at least two overlapping contexts, preserving the semantic meaning of figures and tables.

Engineering Solutions Implemented

- **Sanitization Pipelines:** Implemented a pre-processing layer using `encode('ascii', 'ignore')` and regex filters to strip "poison" characters before they reached the AI logic.
- **Fail-Safe Redundancy:** Created a "Single Document Fallback." If the complex AI splitting logic fails, the system automatically treats the PDF as one continuous logical unit rather than crashing the UI.
- **Sliding Window Chunking:** Optimized retrieval by using a 1,000-character window with a 200-character overlap, ensuring that data points spanning page breaks are never "severed" from their context.
- **The Resolution:** Engineered a Fail-Safe Redundancy (Single-Doc Fallback). If the complex boundary detection fails, the system automatically defaults to a "Full Report" classification. This prioritizes system uptime and data accessibility over perfect categorization.

Reflections

- **Data Quality is the Foundation:** The effectiveness of RAG is entirely dependent on the quality of the "Ingestion Pipeline." I learned that sanitizing text (removing null bytes and non-ASCII characters) is as important as the AI model itself for maintaining accuracy.
- **The Importance of Fallbacks:** Building "crash-proof" AI requires anticipating failures in LLM logic. By implementing a fallback to "Single Document Mode" when boundary detection failed, I ensured the system remained functional and reliable under pressure.
- **Bridging the Human-AI Gap:** Success isn't just about the back-end logic; it's about the interface. Features like the "System Status Log" and "Verified Sources" box were critical in building user trust by providing transparency into how the AI reached its conclusions.

Reflections

- **Resilience Over Complexity:** The project proved that a complex AI feature (like Logical Document Splitting) is only as good as its fallback. Prioritizing system uptime over perfect categorization saved the user experience during edge-case failures.
- **The "Garbage In, Garbage Out" Reality:** Most technical bugs in RAG systems are actually data quality bugs. Investing time in the OCR cleaning phase yielded a higher return on accuracy than fine-tuning the LLM prompts.



Thank You!

View Repo: [AI-Powered-Document-Automation-Platform](#)