# Final Presentation Report: Outstanding Issues & Next Steps

### 1. File Storage Inconsistency

**Issue:**

- Uploaded documents are stored in two different locations:
    - uploads/pdf (created as a sibling of the app directory)
    - data/documents (created inside the app directory)
- This inconsistency needs to be resolved to ensure all documents are stored in one location.

**Action Plan:**

- Decide on a single directory for all document uploads (preferably within the project structure).
- Update the file upload logic in routes.py to store all uploaded documents in the same location.
- Modify references in chunking.py and embeddings.py to ensure they access documents from the correct path.

---

### 2. LLM Responding with "I Don't Know"

**Issue:**

- For every uploaded document, when a user asks a question, the LLM fails to retrieve relevant information and responds with "I don't know."
- This indicates an issue in the flow of processing uploaded documents and storing their vector embeddings.

**Action Plan:**

- **Verify Embeddings Storage:**

    - Check whether extracted text is being properly chunked and stored in FAISS.
    - Ensure that each document's embeddings are indexed correctly.
- **Check Query Pipeline:**
    - Confirm that when a user asks a question, the correct vector search is being performed.
    - Ensure that retrieved context chunks are actually passed to the LLM for response generation.
- **Debug API Calls:**
    - Log responses at each step: chunking, embedding, vector retrieval, and LLM input.
    - If context retrieval is failing, investigate potential indexing or search issues.

---

**3. Final Flow Validation**

**Issue:**

- The entire RAG pipeline needs to be reviewed to ensure every component is functioning as expected.
- There may be issues in how data is stored and retrieved at different stages.

**Action Plan:**

- **End-to-End Testing:**
    - Upload a document and manually trace its journey through the pipeline.
    - Verify that embeddings are stored and retrieved properly.
- **Check Database Storage:**
    - Ensure uploaded documents are being recorded in MongoDB.

- Confirm vector embeddings are indexed in FAISS and accessible during retrieval.

- **Refactor Any Failing Components:**
  - If any storage or retrieval process is failing, debug and refactor the respective code.

    fin

---

**4. Hugging Face Inference Endpoint Integration**

**Issue:**

- **The .env file contains a Hugging Face token and endpoint URI for Llama3.2 3B, which needs to be correctly utilized for response generation.**

**Action Plan:**

- **Ensure that llm.py is correctly retrieving the API token and endpoint from the .env file.**
- **Verify that API calls to Hugging Face are correctly formatted and authenticated.**
- **Confirm that responses generated by Llama3.2 3B are integrated into the final output pipeline.**

---

**Conclusion & Next Steps**

**By resolving the file storage inconsistency and debugging the LLM's retrieval process, the project will be fully functional. The next step is to conduct a full test of document uploads, embedding storage, and query-response generation to ensure seamless functionality before the final presentation.**