



AI-Powered Smart Inventory and Bill Management System: Use Cases & Architecture

1. Document and Important Bill Storage

Supported File Types:

- PDFs (`.pdf`)
- Images (`.jpeg` , `.png`) containing text

Flow:

```
[PDF/Image Upload] + [Metadata: Date, Keywords, Filename]
    ↓
[Text Extraction (OCR for Images)]
    ↓
[Chunked Text Data] → [Vector Embeddings with Metadata]
    ↓
[ChromaDB Storage]
```

Note: Each vector contains metadata pointing to the original file (filename, type, upload date, etc.)

2. Querying Documents and Bills Using RAG

2a) Info Retrieval from Documents

```
User Query
    ↓
Convert to Vector → Search Top-K Similar Vectors in ChromaDB
    ↓
Return Relevant Text Chunks
    ↓
Prompt = [User Query + Retrieved Text Chunks]
    ↓
→ [LLM] → Final Response
```

2b) File-Level Document Retrieval

```
User Query
    ↓
Convert to Vector → Search Top-K Similar Vectors
    ↓
Use Metadata to Retrieve File Names
```

↓
Display List of Relevant Files

3. Inventory Query using LLM + SQL

```
graph TD
    A[User Query] --> B["Prompt = [User Query + SQL Table Schema]"]
    B --> C["→ [LLM Generates SQL Query]"]
    C --> D["Execute Query on SQL DB"]
    D --> E["Return Result to User"]
```

4. LLM Query Panel for Specific Document

Users can:

- Select a document from the database or upload a new one
- Ask questions directly about that specific document

Flow:

```
graph TD
    A["Selected/Uploaded Doc + User Query"] --> B["Text Extraction → Embeddings"]
    B --> C["Query against Extracted Content"]
    C --> D["→ [LLM] → Answer"]
```

5. Custom Notification Generation

User-defined reminders based on:

- Custom events
- Inventory updates
- Scheduled tasks

Stored with dates & triggers for system alerts

6. System Notifications

Automated Notifications Based on System State:

- **Stock Alerts:**
 - Out of Stock
 - Overstock
 - **Udhar/Credit-Debit Alerts:**
 - Based on due dates
 - **Custom Reminders:**
 - Triggered by user-defined conditions
-

7. E-Generated Manual Bills (Sales or Purchase)

User initiates bill generation:

```
Type: Sales / Purchase
↓
Name & Phone Number
↓
Product List (auto-update product table if new)
↓
Quantity + Rate (buying & optional selling price)
↓
For Sales: Update Profit/Loss
↓
For Udhar: Set Payment Reminder Date
↓
Store all relevant information in tables
```

8. Udhar (Credit System)

- **Udhar Sales Table:** Links credit-based sales with payment reminders
- **Udhar Purchase Table:** Same for purchases

Each entry includes:

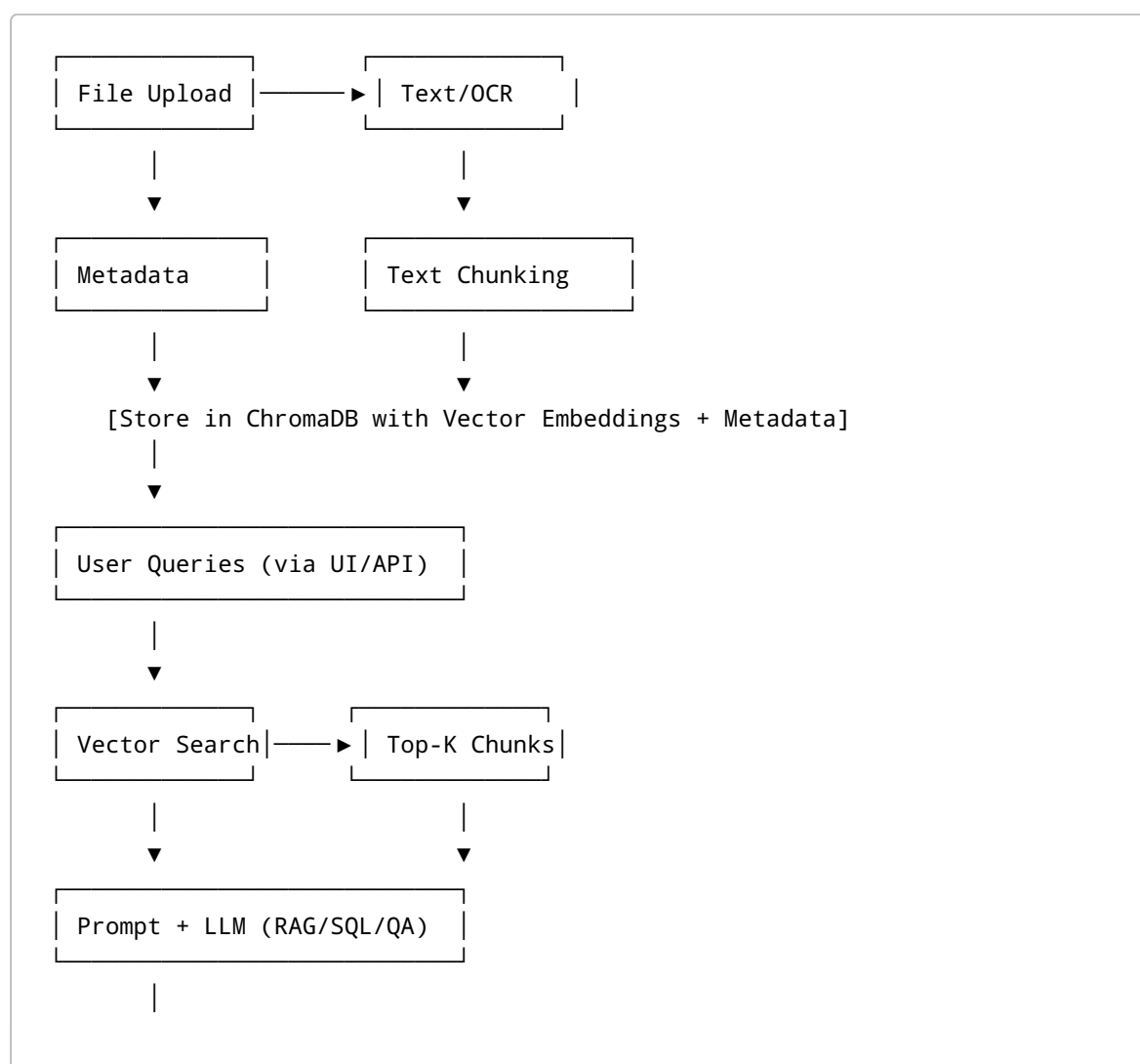
- Transaction ID
 - Entry Date
 - Expected Payment Date
-

9. Data Tables Overview

Table	Description
Sales_data	Records of sales: customer, amount, quantity

Table	Description
Purchase_data	Records of purchases: vendor, amount, quantity
Products	Product catalog with prices and available quantity
Profit_Loss	Tracks if each sale is profitable or not
Sale_X_Prod	Link table between sales and products sold
Purchase_X_Prod	Link table between purchases and products bought
Udhar_sales	Credit sales log with due dates
Udhar_purchase	Credit purchase log with due dates
Customers	Customer contact information
Vendors	Vendor contact information

System Architecture Overview Diagram



▼
Final Output to User

This system ensures robust inventory control, intelligent document retrieval, dynamic user interaction via LLM, and an adaptive credit/payment management system all working together locally or via private infrastructure.

Next Steps? We can now define APIs for each of these modules using FastAPI and a lightweight frontend (e.g., Streamlit or React). Want to proceed to the API design?