cloudstaff®

Invoice Validator Handover Document

Table of Contents

1. Project Overview

Project Name: Invoice Validator

Description:

The Invoice Validator is an AI-powered invoice validation system utilizing LangChain-based Retrieval-Augmented Generation (RAG). It supports multiple file formats, extracts and validates key invoice details, and detects discrepancies through a purchase order (PO) comparison.

Purpose:

- Automate invoice validation and data extraction.
- Prevent duplicate processing using ChromaDB.
- Enhance efficiency and accuracy in financial document processing.
- Identify discrepancies between invoices and purchase orders.

Tech Stack:

- **Backend:** Python
- **Frontend:** Streamlit
- **AI Processing:** LangChain, PyMuPDF, Tesseract OCR
- **Database:** ChromaDB
- **Deployment:** Localhost

---

2. System Architecture

Key Components:

- **Frontend (Streamlit UI):** Provides an intuitive user interface for invoice validation.
- **Backend (Python Services):** Manages invoice validation, PO comparison, and discrepancy detection.
- **AI Processing (LangChain + RAG):** Extracts structured invoice details and validates them against expected formats.
- **Database (ChromaDB):** Stores validated invoices and ensures no duplication.
- **OCR Processing (Tesseract):** Reads text from scanned invoices and images.
- **External APIs:** For future enhancements such as real-time invoice validation.

---

3. Installation & Setup

Prerequisites:

- Python 3.10+
- Virtual environment (`venv` or `conda`)
- Required dependencies (`pip install -r requirements.txt`)

Step-by-Step Setup:

```
# Clone the repository
git clone https://github.com/MFlores01/LangChain_Invoice_Validator.git
cd Langchain-Invoice-main

# Create and activate a virtual environment
python -m venv .venv
.venv\Scripts\activate  # (For Windows)
source .venv/bin/activate  # (For Mac/Linux)

# Install dependencies
pip install -r requirements.txt
```

Running the Application:

```
# Start the Streamlit frontend
streamlit run streamlit_app    .py
```

4. Code Structure

```
Invoice_Validator/
│── invoice_db/                # Database for validated invoices
│── po_db/                     # Database for purchase orders
│── src/                       # Source code directory
│   ├── app/
│   │   ├── __init__.py
│   │   ├── streamlit_app.py       # Streamlit UI for invoice validation
│   ├── core/
│   │   ├── chatbot.py            # AI-powered chatbot (if applicable)
│   │   ├── data_processor.py      # Handles invoice data extraction
│   │   ├── file_validator.py      # Validates file formats & structures
│   │   ├── po_comparator.py       # Compares invoices with purchase orders
│   │   ├── po_validator.py        # Purchase Order validation logic
│   │   ├── validation_engine.py   # Core validation logic for invoices
│   ├── utils/
│   │   ├── db.py                 # Database handling functions
│   │   ├── file_utils.py          # File handling utilities
│   │   ├── logger.py             # Logging setup and utilities
│   │   ├── vector_stores.py       # Handles vector database (ChromaDB)
│── .env                       # Environment variables (API keys, DB configs)
│── requirements.txt             # Required dependencies
│── setup.py                    # Installation script
```

---

5. User Interface Overview

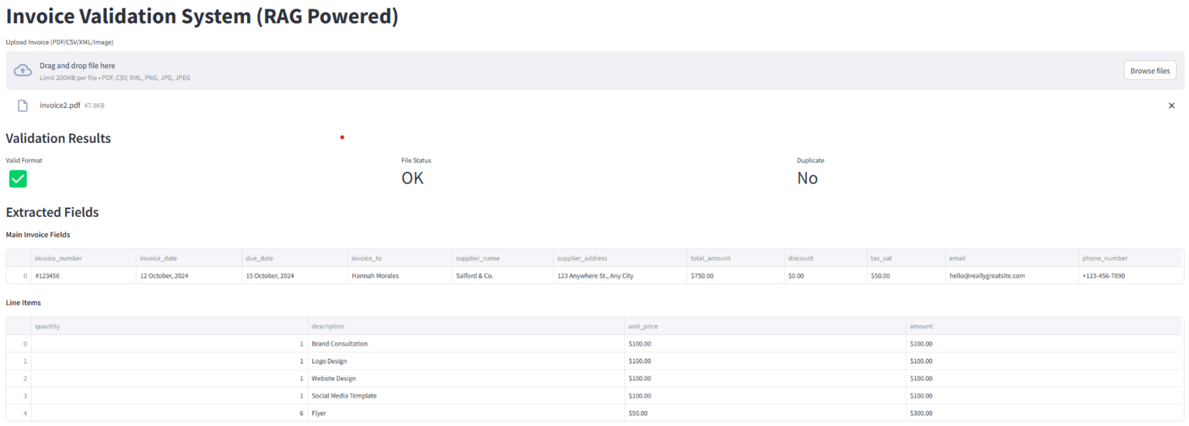
The AI Invoice Chatbot provides a simple, user-friendly Streamlit-based UI with the following features:

Main Features:

- **File Upload Section:** Supports invoices in PDF, CSV, XML, PNG, and JPG formats.
- **Validation Results:** Displays extracted invoice details and verification status.
- **PO Comparison:** Highlights discrepancies between invoices and purchase orders.
- **Duplicate Detection:** Prevents reprocessing of invoices already stored in the database.

Screenshots:

Complete Invoice Format



Invoice Validation System (RAG Powered)

Upload Invoice (PDF/CSV/XML/Image)

Validation Results

Valid Format ✓

File Status OK

Duplicate No

Extracted Fields

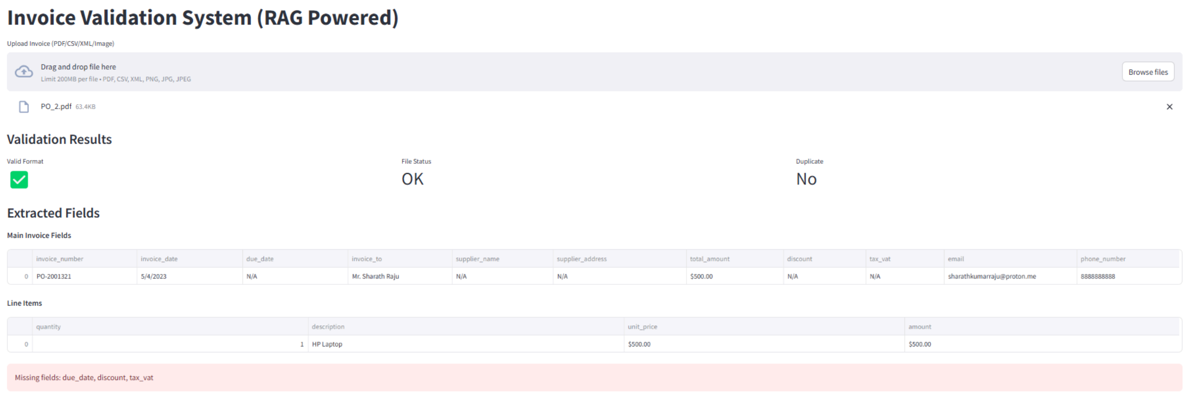
Main Invoice Fields

	invoice_number	invoice_date	due_date	invoice_to	supplier_name	supplier_address	total_amount	discount	tax_vat	email	phone_number
0	#123456	12 October, 2024	15 October, 2024	Hannah Morales	Salford & Co.	123 Anywhere St., Any City	$750.00	$0.00	$50.00	hello@reallygreatsite.com	+123-456-7890

Line Items

	quantity	description	unit_price	amount
0	1	Brand Consultation	$100.00	$100.00
1	1	Logo Design	$100.00	$100.00
2	1	Website Design	$100.00	$100.00
3	1	Social Media Template	$100.00	$100.00
4	6	Flyer	$50.00	$300.00

PO Format



Invoice Validation System (RAG Powered)

Upload Invoice (PDF/CSV/XML/Image)

Validation Results

Valid Format ✓

File Status OK

Duplicate No

Extracted Fields

Main Invoice Fields

	invoice_number	invoice_date	due_date	invoice_to	supplier_name	supplier_address	total_amount	discount	tax_vat	email	phone_number
0	PO-2001321	5/4/2023	N/A	Mr. Sharath Raju	N/A	N/A	$500.00	N/A	N/A	sharathkumarraju@proton.me	8888888888

Line Items

	quantity	description	unit_price	amount
0	1	HP Laptop	$500.00	$500.00

Missing fields: due_date, discount, tax_vat

---

6. Key Features & How They Work

- **Invoice Processing:** Extracts and validates key invoice details (invoice number, date, supplier, amount).

- **Multi-Format Support:** Handles PDFs, CSVs, XMLs, and images.
- **Duplicate & Integrity Check:** Ensures invoices are not corrupted or previously processed.
- **AI-Powered Validation:** Uses LangChain with RAG to validate invoice contents.
- **Purchase Order Comparison:** Matches invoice details with POs stored in the database.
- **Discrepancy Reporting:** Generates reports on mismatches and inconsistencies.

---

7. Deployment & Hosting

Deployment Platform:

- Currently deployed on localhost for development.
- Future deployment options include AWS/GCP or other cloud platforms.

Deployment Steps:

- **Option 1: Local Deployment (Current Setup)**
  - streamlit run try.py
- **Option 2: Future Cloud Deployment (AWS/GCP)**
  - Upload code to the cloud server.
  - Configure environment variables.
  - Start the API app.

---

8. Troubleshooting & FAQs

Common Issues & Fixes

Issue	Solution
ModuleNotFoundError	Ensure the correct file environment is used before running `streamlit run ....`
Database not connecting	Check database credentials and ensure ChromaDB is running.
Model not loading	Ensure LangChain dependencies are installed correctly.

---

9. Contact & Handover Notes

Previous Developers:

- *Miguel Flores* (Lead Developer)
  - **Support Email**: MiguelF@cloudstaff.com
  - **Github repository**: https://github.com/MFlores01/LangChain_Invoice_Validator.git

Next Steps:

- Use AWS/GCP or other cloud platforms for future deployment.
- Apply user authentication for platform security.
- Future enhancements for real-time validation.
- Improving OCR accuracy with pre-processing techniques.
- Expanding discrepancy reports with additional insights.