

## ****RECEIPT & INVOICE DIGITIZER – WHAT IT ACTUALLY MEAN :****

The project title **“Receipt & Invoice Digitizer”** clearly represents the main goal of the system.

* **Receipt** refers to a proof of payment document usually generated after a purchase (billing slips, store receipts, etc.).
* **Invoice** refers to a formal billing document generated by sellers/service providers for transactions.
* **Digitizer** means converting physical or scanned documents (paper/image/PDF) into **digital structured data**.

So, **Receipt & Invoice Digitizer** is a system that takes receipt/invoice images or PDFs as input and automatically extracts key information (like date, amount, invoice number, vendor, items, tax, etc.), then converts it into a **machine-readable digital format** such as JSON/CSV/database records.

**INTRODUCTION :**

In today’s digital era, organizations and individuals handle a large number of receipts and invoices for expenses, billing, taxation, and financial record maintenance. However, most receipts and invoices are still generated in paper format or as image/PDF files, making manual data entry time-consuming, error-prone, and inefficient.

The **Receipt & Invoice Digitizer** project is designed to solve this problem by automatically extracting important information from receipts and invoices such as **invoice number, date, vendor name, item details, quantity, total amount, and tax information**.

This system converts unstructured receipt/invoice images into structured digital data that can be stored, searched, and analyzed easily.

By using modern technologies like **OCR (Optical Character Recognition)** and **AI-based text processing**, the proposed solution aims to improve speed, accuracy, and efficiency in document handling. It reduces human effort, enables faster financial processing, and supports better decision-making by maintaining clean and digitized records.

**PROJECT STATEMENT :**

Businesses and individuals handle numerous paper receipts and invoices, which are prone to loss, errors, and manual entry delays. This project builds a system that automatically scans, extracts, and digitizes information from receipts and invoices using OCR (Optical Character Recognition) and NLP-based field extraction. The digitized data is stored in a structured format, making it easy to search, analyze, and integrate with accounting or ERP systems.

**OBJECTIVES :**

1. **File Upload Module**

Enable users to upload receipts/invoices in formats like **JPG, PNG, JPEG, and PDF**.

1. **Document Ingestion Pipeline**

Store uploaded files safely for processing (local folder / cloud storage).

1. **Image Preprocessing for OCR**

Improve input quality using preprocessing techniques such as:

* + - resizing,
    - grayscale conversion,
    - noise removal,
    - thresholding / binarization,
    - sharpening / contrast enhancement.

1. **OCR Text Extraction**

Extract **raw text** from uploaded sample receipts/invoices using OCR engine.

1. **Initial Output Display**

Show extracted raw text to verify correctness and readability.

## ****SCOPE :****

* Upload receipt/invoice images and PDFs.
* Basic preprocessing to improve OCR accuracy.
* OCR extraction of raw text.
* Display extracted text result on UI (web app)

## ****SYSTEM ARCHITECHTURE :****

## 

## ****1) File Upload Module****

The File Upload Module is the first component of the Receipt & Invoice Digitizer system and is responsible for collecting input documents from the user. In this module, the user uploads receipts or invoices in supported formats such as JPG, PNG, JPEG (and PDF if applicable). The system validates the uploaded file to ensure it matches the accepted format and avoids corrupted or unsupported inputs.

After validation, the uploaded document is stored in a temporary/local storage location so it can be accessed by the preprocessing and OCR stages. This module ensures smooth document ingestion and acts as the foundation for the complete OCR pipeline.

## ****2) Image Preprocessing Module****

The Image Preprocessing Module improves the quality of the uploaded receipt or invoice image to achieve better OCR accuracy. Since receipts are often captured using mobile cameras or scanners, the images may contain noise, blur, low brightness, shadows, skew, or uneven contrast. This module applies preprocessing operations such as resizing, grayscale conversion, denoising, thresholding (binarization), and contrast enhancement to make the printed text clearer and more distinguishable from the background.

In some cases, additional processing like deskewing can also be used to correct tilted documents. As a result, the preprocessing module generates an OCR-ready image that significantly increases the reliability of text extraction.

## ****3) OCR Extraction Module****

The OCR Extraction Module is the core part of milestone-1, where the system converts the preprocessed receipt/invoice image into machine-readable text. After receiving the enhanced image from the preprocessing stage, an OCR engine such as Tesseract OCR or EasyOCR detects characters and words from the document and extracts them as raw text output.

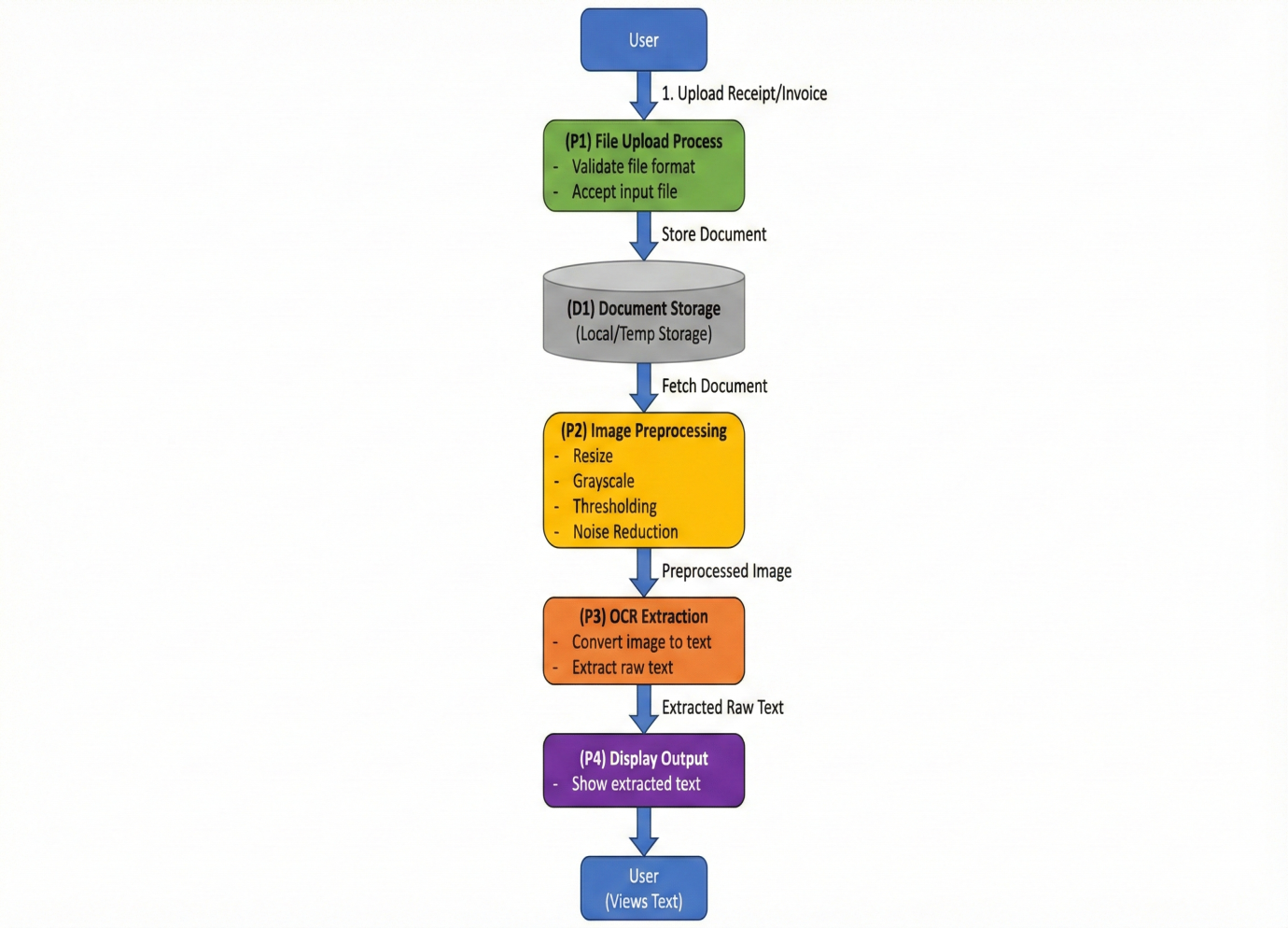
The extracted content may include vendor name, transaction details, item list, quantities, tax values, total amount, and other printed information present on the receipt/invoice. At this milestone stage, the extracted text is not yet structured into fields, but it provides the required raw text foundation for later milestones like key-value extraction and data digitization.

## ****4) Output Display Module****

The Output Display Module presents the extracted OCR text to the user for verification and evaluation. Once the OCR module generates raw text, this module displays the output in a readable format on the user interface, allowing the user to check the extracted content and confirm its accuracy.

This is useful in milestone-1 because it helps in testing the OCR pipeline with different sample receipts and invoices and ensures that preprocessing is improving recognition quality. By showing the extracted results clearly, the module supports performance validation and forms the basis for future steps such as structured data extraction and storage.

**DATA FLOW DIAGRAM :**

****

The **DFD diagram** explains the complete working flow of Milestone–1 in the Receipt & Invoice Digitizer system. It starts with the **User**, who uploads a receipt or invoice document into the application.

This input is first handled by **Process P1 (File Upload Process)**, where the system accepts the file and validates whether the format is correct and supported. After successful validation, the uploaded file is saved in **Data Store D1 (Document Storage)**, which represents local or temporary storage used during processing.

Once stored, the system retrieves the document from storage and sends it to **Process P2 (Image Preprocessing)**, where the receipt image is enhanced through steps like resizing, grayscale conversion, thresholding, and noise reduction to improve text clarity. The preprocessed output is then passed to **Process P3 (OCR Extraction)**, where an OCR engine converts the enhanced image into machine-readable **raw text**.

Finally, this extracted text is forwarded to **Process P4 (Display Output)**, which shows the OCR result on the interface so the user can view and verify the extracted content. This diagram clearly represents the milestone-1 pipeline from document upload to OCR text output.

**TECHNOLOGY STACK :**

| **Technology / Tool** | **Purpose in Milestone–1** | **Why Used** |
| --- | --- | --- |
| **Python** | Core programming language for the entire pipeline. | Easy integration for OCR and image processing libraries. |
| **Streamlit (or Flask UI)** | Frontend interface for uploading receipts and showing OCR output. | Simple, fast, and interactive web UI development. |
| **OpenCV (cv2)** | Image preprocessing (grayscale, resizing, noise removal, thresholding). | Improves OCR accuracy by enhancing image quality. |
| **Tesseract OCR (pytesseract)** | Extracts raw text from preprocessed images. | Converts scanned images into machine-readable text. |
| **NumPy** | Handles image array operations during preprocessing. | Efficient numerical processing for OpenCV image data. |
| **Local Storage** | Stores uploaded files temporarily before processing. | Easy and lightweight storage for initial milestone testing. |
| **VS Code / Jupyter** | Development and testing environment. | Helps in rapid debugging and experimenting with OCR outputs. |

## ****ERROR HANDLING AND VALIDATION :****

In Milestone–1, proper error handling and validation mechanisms are implemented to ensure smooth document ingestion and reliable OCR execution. During the file upload stage, the system validates the uploaded document by checking the file type and allowing only supported formats such as JPG, PNG, JPEG (and PDF if enabled).

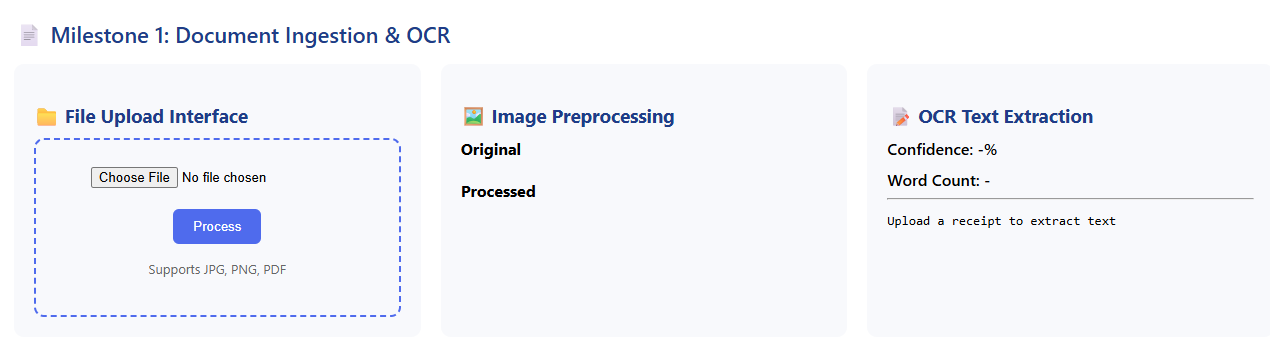
If an unsupported file type is uploaded, the application immediately rejects it and displays an appropriate error message to the user. File validation also includes checking whether a file is empty, corrupted, or unreadable, preventing the pipeline from crashing due to invalid inputs.

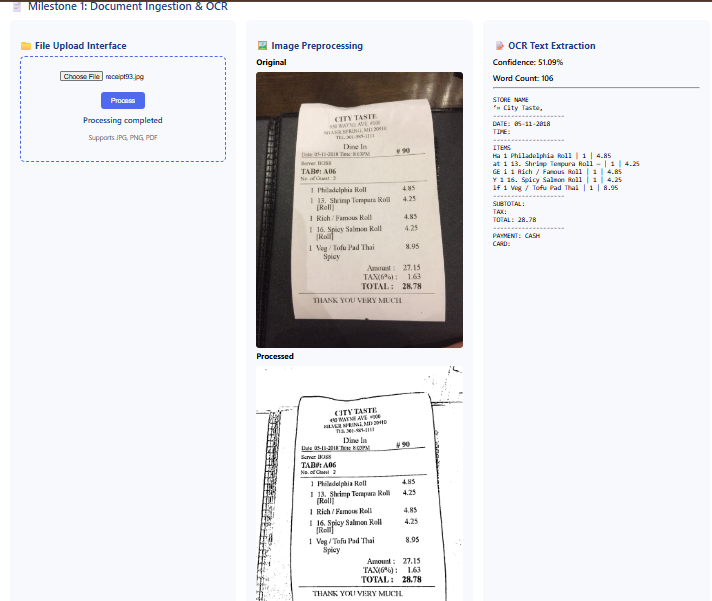
After upload, the system verifies successful storage of the document before forwarding it for preprocessing. In the preprocessing stage, validations are performed to confirm that the image is correctly loaded into memory and has valid dimensions; if image reading fails (for example, due to corrupted images or incomplete downloads), the system stops processing and notifies the user.

During OCR extraction, the system handles OCR failures such as unclear images, low resolution, heavy noise, and poor lighting by displaying a warning message when extracted text is insufficient or blank.

Additionally, exception handling is used throughout the pipeline to catch runtime errors, ensuring the system does not terminate unexpectedly and instead provides user-friendly feedback. These validation and error-handling practices improve system reliability and help ensure accurate OCR output during milestone testing.

**OUTPUT SCREENS :**

****

****

**CONCLUSION :**

In this milestone, the foundational stage of the **Receipt & Invoice Digitizer** project was successfully completed by implementing the end-to-end pipeline for **document ingestion and OCR processing**.

The system was able to accept receipt and invoice documents through a file upload interface, perform essential image preprocessing to enhance document quality, and extract raw text using an OCR engine.

This milestone validates that the application can reliably convert receipt/invoice images into machine-readable text, which serves as the base for future development. The outcomes of this milestone provide a strong foundation for upcoming phases such as structured key-value extraction, data validation, and storing digitized invoice/receipt details into databases for search and analytics.