Receipt & Invoice Digitizer (MydigiBill)

Milestone-1 Documentation

---

1. Introduction

In today's digital economy, businesses and individuals generate large volumes of receipts and invoices. Manual handling of these documents is time-consuming, error-prone, and inefficient. Automating the digitization process helps in expense tracking, financial analytics, and record management.

The **Receipt & Invoice Digitizer (MydigiBill)** project aims to automate the ingestion, extraction, and intelligent processing of receipt data.
Milestone-1 focuses on building the *foundation layer* of the system: **Document Ingestion and Optical Character Recognition (OCR).**

This milestone ensures that receipt documents can be uploaded, pre-processed, and converted into machine-readable text reliably.

---

2. Scope

It covers only the **input and OCR pipeline**, not intelligent field extraction or analytics.

Included:

- File upload system
- Image preprocessing
- OCR text extraction
- Raw text validation
- OCR readiness for AI processing

Excluded:

- AI field mapping (fully implemented in later milestones)
- Database analytics
- Expense insights

---

3. Objectives

The primary objectives of Milestone-1 are:

1. Enable users to upload receipt and invoice files
2. Support multiple file formats (images and PDFs)
3. Improve OCR accuracy using preprocessing

4. Extract clean, readable raw text from receipts
5. Prepare extracted text for AI-based parsing

---

4. System Overview

High-Level Flow

User Upload

↓

File Validation

↓

Image Conversion (PDF → Image)

↓

Image Preprocessing

↓

OCR Engine (Tesseract)

↓

Raw Text Output

This modular design ensures flexibility and scalability.

---

5. Technologies Used

Category	**Technology**
Programming Language	Python
Frontend	Streamlit
OCR Engine	Tesseract OCR
Image Processing	OpenCV
PDF Handling	pdf2image
AI Engine (Introduced)	Ollama (Local LLM)
Database (Later use)	SQLite
Platform	Windows

6. File Upload Module

6.1 Purpose

The file upload module allows users to submit receipt documents for OCR processing.

6.2 Supported File Types

- .jpg
- .jpeg
- .png
- .pdf

6.3 Implementation

- Built using Streamlit file uploader
- File saved locally in an uploads/ directory
- Duplicate file detection prevents re-uploading the same receipt

6.4 Advantages

- Simple user experience
- Fast ingestion
- Prevents redundant processing

---

7. PDF to Image Conversion

7.1 Why Conversion Is Needed

Tesseract OCR works primarily on images, not PDFs.

7.2 Conversion Process

- pdf2image library used
- First page of PDF converted into a JPEG image
- Converted image passed to preprocessing pipeline

7.3 Benefits

- Enables OCR on scanned invoices
- Maintains compatibility with image-based OCR

8. Image Preprocessing for OCR

8.1 Importance of Preprocessing

Raw images often contain:

- Noise
- Shadows
- Skew
- Poor contrast

Without preprocessing, OCR accuracy drops significantly.

---

9. Preprocessing Techniques Used

9.1 Image Resizing

- Enlarges text for better OCR detection
- Improves recognition of small fonts

9.2 Grayscale Conversion

- Removes color complexity
- Simplifies image structure

9.3 Contrast Enhancement (CLAHE)

- Enhances faded text
- Improves edge detection

9.4 Noise Removal

- Removes background dots and grain
- Improves clarity

9.5 Adaptive Thresholding

- Converts image to black and white
- Helps detect text under uneven lighting

9.6 Deskewing

- Corrects tilted receipts
- Aligns text horizontally

10. OCR Text Extraction

10.1 OCR Engine

- **Tesseract OCR** is used
- Open-source and widely adopted
- Supports multiple OCR modes

10.2 OCR Configuration

- Optimized page segmentation modes
- Multi-pass extraction
- Confidence-based filtering

10.3 Output

- Extracted raw text includes:
  - Store names
  - Dates
  - Item descriptions
  - Prices
  - Totals
  - Payment info (if present)

---

11. Raw OCR Output Handling

11.1 Cleaning OCR Text

- Remove extra whitespace
- Fix common OCR errors
- Normalize characters

11.2 Deduplication

- Removes repeated lines
- Prevents noise accumulation

---

12. Introduction to Ollama (Local AI Engine)

Although full AI parsing is implemented in later milestones, **Ollama is introduced in Milestone-1 as the intelligence backbone**.

12.1 What Is Ollama?

Ollama is a **local Large Language Model (LLM) runtime** that allows running AI models entirely on the local machine.

Key Features

- No internet required
- No API key
- No usage limits
- Full data privacy

---

13. Why Ollama Was Chosen

Reason	**Benefit**
Local execution	No cloud dependency
Privacy	Sensitive receipts stay on device
Cost	Completely free
Performance	Low latency
Control	Full prompt control

---

14. Ollama Architecture

OCR Text

↓

Prompt Engineering

↓

Local LLM (LLaMA 3 / Mistral)

↓

Structured JSON Output

---

15. Ollama Working Procedure (Detailed)

Step 1: Model Download

- Model pulled once using:
- ollama pull llama3

Step 2: Model Hosting

- Ollama runs a local REST API at:
- http://localhost:11434

Step 3: Prompt Creation

- OCR text embedded inside structured prompts
- Strict JSON output enforced

Step 4: Inference

- Model processes receipt context
- Maps text into structured format

Step 5: Output Parsing

- JSON extracted
- Validated
- Corrected if required

---

16. Role of Ollama in Milestone-1

In Milestone-1:

- Ollama is **introduced conceptually**
- OCR output is validated for AI readiness
- Preprocessing ensures AI-friendly text

---

17. Challenges Faced

17.1 OCR Noise

- Solved via preprocessing

17.2 Receipt Layout Variability

- Solved via flexible parsing

17.3 Skewed Images

- Solved via deskewing

---

18. Results & Observations

- OCR accuracy improved significantly after preprocessing

- System successfully extracts readable text from:
    - Restaurant receipts
    - Retail invoices
    - Supermarket bills
- OCR pipeline is stable and scalable

---

19. Milestone-1 Completion Status

✓ File upload implemented
✓ Image preprocessing completed
✓ OCR extraction successful
✓ PDF and image support
✓ AI-ready OCR output

---

20. Conclusion

Milestone-1 successfully establishes the foundation of the Receipt & Invoice Digitizer system. The implemented OCR pipeline ensures reliable document ingestion and prepares the system for AI-driven data extraction and analytics in future milestones.