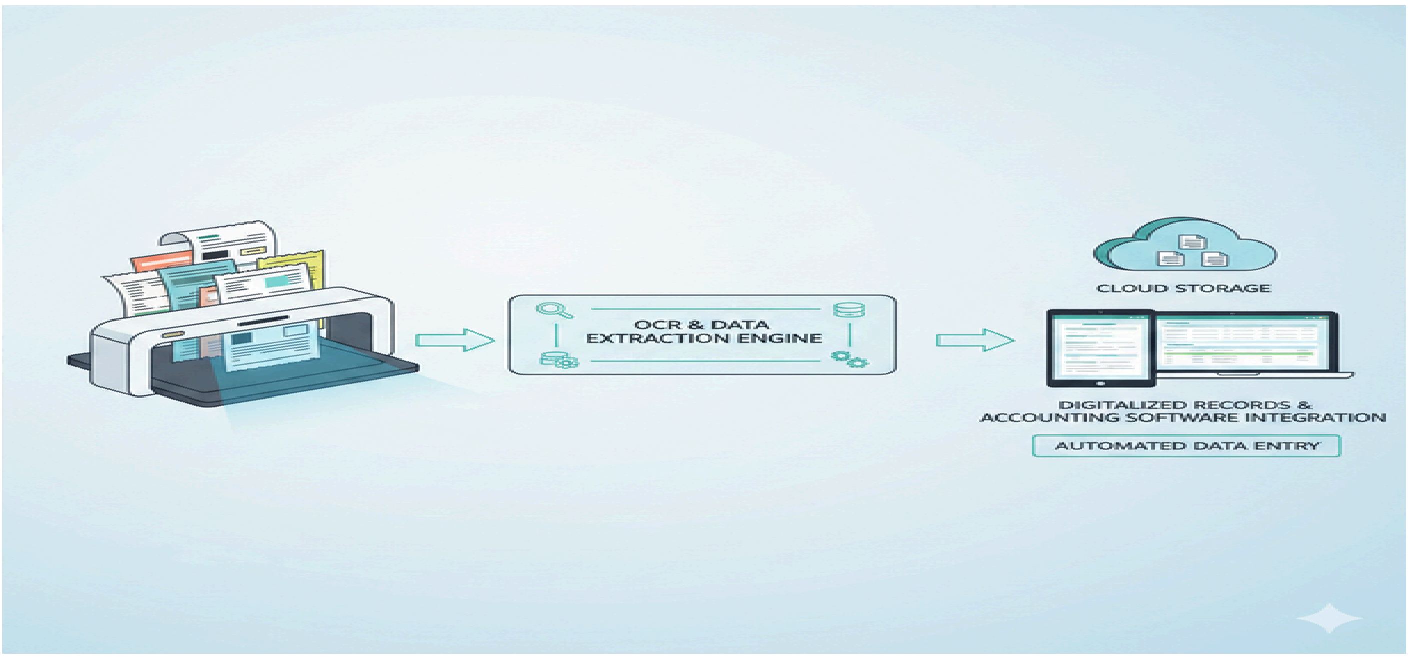

CLOUD STORAGE
OCR & DATA EXTRACTION ENGINE
DIGITALIZED RECORDS & ACCOUNTING SOFTWARE INTEGRATION
AUTOMATED DATA ENTRY

Receipt and Invoice Digitizer

1. Introduction

In the modern digital era, organizations and individuals handle a large volume of financial documents such as receipts and invoices on a daily basis. These documents play a crucial role in expense tracking, accounting, auditing, and financial planning. However, manual management of physical receipts or unstructured digital documents is time-consuming, error-prone, and inefficient. Traditional data entry methods often result in inaccuracies, data duplication, loss of records, and difficulties in retrieving historical financial information.

The **Receipt and Invoice Digitizer** project addresses these challenges by providing an **AI-powered automated solution** for extracting, validating, and storing information from receipts and invoices. The system leverages **Optical Character Recognition (OCR)** to extract raw text from uploaded documents and uses **Artificial Intelligence (AI)** to intelligently parse and structure critical financial data such as vendor details, transaction dates, totals, taxes, and line items.

The extracted data is further validated, normalized, and securely stored in a structured database, ensuring accuracy, consistency, and duplicate-free record management. Developed using **Python, Streamlit, Tesseract OCR, Gemini AI, and SQLite**, this project demonstrates the practical application of modern AI and web technologies to solve a real-world financial document management problem in an efficient and scalable manner.

2. **Problem Statement**

Traditional methods of managing receipts and invoices rely heavily on manual collection, physical storage, and manual data entry into spreadsheets or accounting systems. These approaches are inefficient and present several challenges:

- Manual data entry is time-consuming and labor-intensive
- High likelihood of human errors and data inconsistencies
- Difficulty in organizing, searching, and retrieving historical records
- Increased risk of duplicate or redundant entries
- Poor scalability when handling large volumes of financial documents

Due to these limitations, there is a strong need for an **automated and intelligent system** that can accurately extract relevant information from receipts and invoices, **detect and prevent duplicate records**, and store the data in a **structured, secure, and reliable format** for efficient financial management.

---

3. **Objectives of the Project**

The main objectives of the **Receipt and Invoice Digitizer** project are:

- To automate the digitization of receipts and invoices through a web-based system
- To extract key financial information using Optical Character Recognition (OCR) and Artificial Intelligence (AI)
- To transform unstructured receipt text into structured and machine-readable data
- To detect and prevent duplicate receipt and invoice records
- To minimize manual effort and reduce overall processing time
- To provide an intuitive and user-friendly web interface for document upload and review
- To securely store extracted receipt data for future access and analysis
- To optimize AI API usage while ensuring system reliability and stability

---

4. **Modules to be Implemented**

1. Document Ingestion Module

This module handles the secure upload of receipt and invoice documents into the system. It supports multiple file formats such as **PDF, JPG, JPEG, and PNG**. Multi-page PDF files are automatically converted into individual images to ensure consistent processing. Basic image preprocessing techniques such as resizing, noise reduction, and binarization are applied to enhance text readability before OCR.

2. OCR & Text Extraction Module

This module is responsible for extracting raw textual content from uploaded documents. It utilizes **Tesseract OCR** to convert receipt and invoice images into machine-readable text. The module is designed to handle common challenges such as skewed text, varying fonts, and tabular layouts typically found in financial documents.

3. AI-Based Field Extraction & Validation Module

This module processes the cleaned OCR text using **Gemini AI** to extract structured financial information such as vendor name, transaction date, total amount, tax, and line items. Strict JSON prompting ensures consistent output. The extracted data undergoes validation checks including mandatory field verification, numeric validation, and duplicate detection using OCR text to prevent redundant records.

4. Database Storage Module

This module manages the persistent storage of extracted receipt and invoice data using a **SQLite database**. It stores receipt-level information and related line-item details in a normalized structure. The module ensures data integrity, supports efficient retrieval, and enables users to view previously stored records within the application.

5. User Interface & Visualization Module

This module provides a **Streamlit-based web interface** for interacting with the system. It enables users to upload documents, view extracted receipt details, and browse stored records through organized tabs. The interface ensures smooth navigation, real-time feedback, and secure access through API key validation.

---

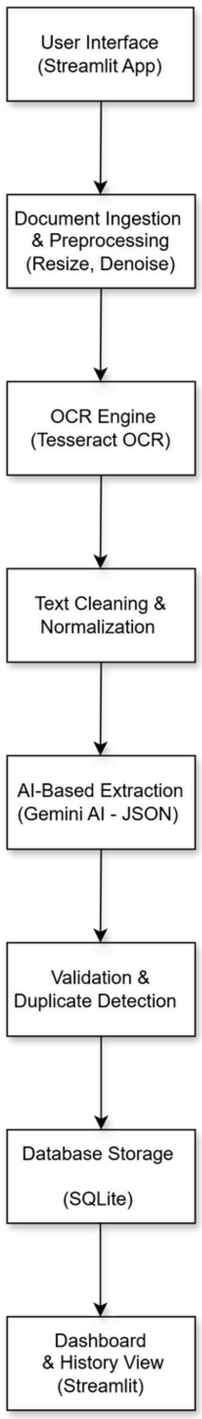
5. System Overview

The **Receipt and Invoice Digitizer** is a web-based application designed to automate the extraction, validation, and storage of financial data from receipts and invoices. The system integrates **OCR, AI-based parsing, validation logic, and database storage** to transform unstructured documents into structured, reliable digital records.

Users upload receipt or invoice files through a **Streamlit-based interface**. The uploaded documents are preprocessed and passed through **Tesseract OCR** to extract raw text. This text is cleaned and forwarded to **Gemini AI**, which extracts structured financial fields using strict JSON prompting. The extracted data is validated to ensure correctness and checked for duplication before being securely stored in a **SQLite database**. Stored records can then be viewed and managed through the application dashboard.

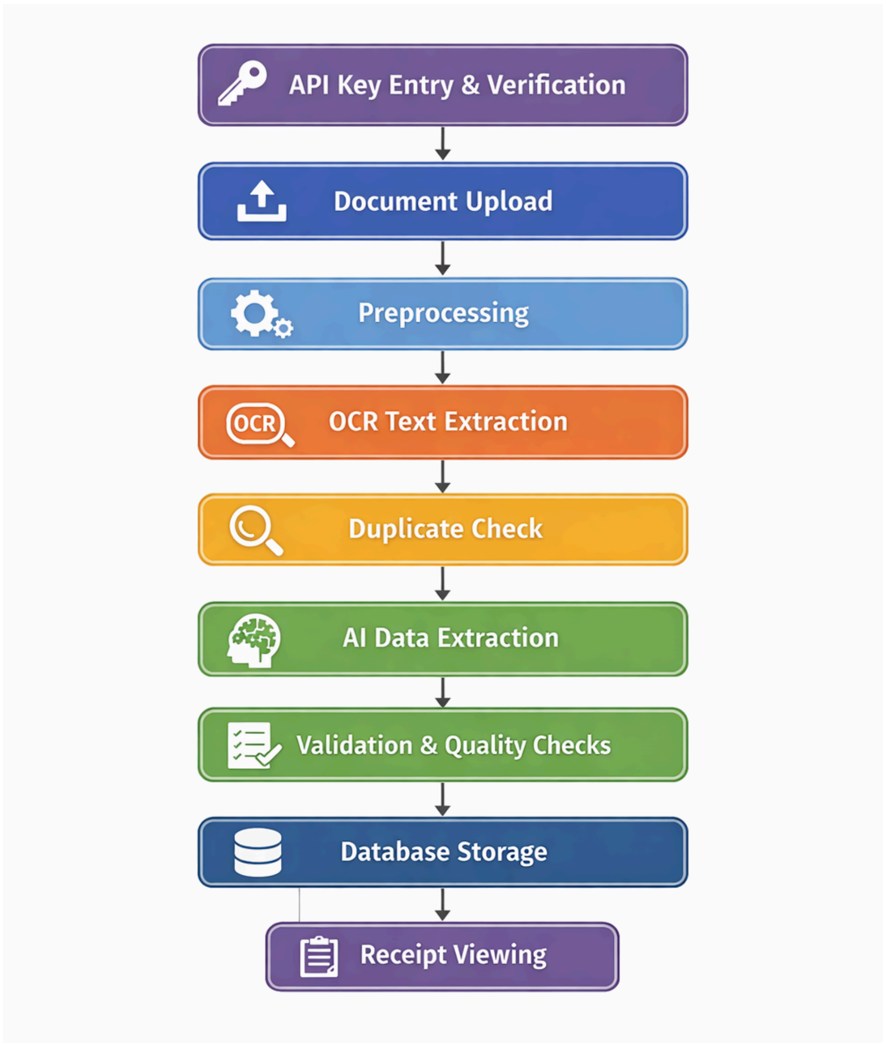
The system ensures **accuracy, consistency, duplicate prevention, and persistent storage**, making it suitable for expense tracking and financial record management.

5. **System Architecture**


User Interface (Streamlit App)
Document Ingestion & Preprocessing (Resize, Denoise)
OCR Engine (Tesseract OCR)
Text Cleaning & Normalization
AI-Based Extraction (Gemini AI - JSON)
Validation & Duplicate Detection
Database Storage (SQLite)
Dashboard & History View (Streamlit)

6. **Detailed Workflow of the Receipt & Invoice Digitizer**

- **API Key Entry & Verification**
  The user enters a Gemini API key, which is validated and stored in the session to control access to AI-based extraction features.
- **Document Upload**
  The user uploads a receipt or invoice in supported formats (PDF, JPG, JPEG, PNG) through the web interface.
- **Document Preprocessing**
  Uploaded documents are converted into images (for PDFs) and preprocessed using resizing, noise reduction, and thresholding to enhance OCR accuracy.
- **OCR Text Extraction**
  Tesseract OCR is applied to extract raw textual content from the processed images.
- **Text Cleaning & Normalization**
  The raw OCR output is cleaned to remove noise, normalize spacing, and prepare the text for AI parsing.
- **Early Duplicate Detection**
  The cleaned OCR text is checked against previously stored records to identify duplicate receipts and prevent redundant processing.
- **AI-Based Data Extraction**
  Unique receipt text is forwarded to Gemini AI using strict JSON prompting to extract structured financial data.
- **Data Validation & Quality Checks**
  Extracted fields are validated through numeric checks, mandatory field verification, and line-item consistency rules.
- **Database Storage**
  Validated receipt and line-item data are stored securely in a normalized SQLite database.
- **Receipt Viewing & Management**
  Users can browse stored receipts, view detailed bill items, and review historical records through the application dashboard.



7. Database Description

The Receipt and Invoice Digitizer uses a **SQLite relational database** to store structured receipt and invoice data extracted from uploaded documents. The database is designed to ensure **data integrity, efficient retrieval, and duplicate prevention**. It separates receipt-level information from individual line-item details using a normalized schema.

Each receipt record stores metadata such as vendor name, transaction date, total amount, tax, and the associated raw OCR text. Line-item details are stored in a separate table and linked to the corresponding receipt using a foreign key relationship. This design avoids data redundancy and supports scalable storage of detailed billing information.

Duplicate detection is implemented at the database level using unique constraints on receipt content, ensuring that the same receipt cannot be stored multiple times. The database enables persistent local storage and supports efficient querying for displaying historical records in the application.

Database Tables & Fields

1. Receipts Table

Stores high-level receipt or invoice information.

- id – Primary key (auto-increment)
- vendor – Merchant or vendor name
- date – Transaction date
- total – Total payable amount
- tax – Tax amount
- raw_text – Cleaned OCR text (used for duplicate detection)
- created_at – Timestamp of record creation

2. Line_Items Table

Stores individual item details linked to a receipt.

- id – Primary key (auto-increment)
- receipt_id – Foreign key referencing receipts.id
- item_name – Name/description of the item
- quantity – Quantity purchased
- price – Item price

Table	Column	Type	Notes
receipts	id	INTEGER	Primary Key (Autoincrement)
	merchant	TEXT	Name of the store
	date	TEXT	Date of transaction
	total	TEXT	Total amount
	tax	TEXT	Tax amount
	raw_text	TEXT	Full OCR/scanned text
items	id	INTEGER	Primary Key (Autoincrement)
	receipt_id	INTEGER	**Foreign Key** (Links to receipts.id)
	name	TEXT	Item description
	quantity	INTEGER	Number of items
	price	REAL	Cost per item

---

8. **Technologies Used**

Layer	Technology	Key Functionality
Frontend	Streamlit	Handles the Web UI, sidebar configuration, and session state.
Document Processing(Milestone 1)	OpenCV / pdf2image	Image preprocessing, noise reduction, and PDF-to-image conversion.
OCR Engine(Milestone 1)	Tesseract OCR	Primary engine for raw text extraction from processed images.
AI Parsing(Milestone 2)	Google Gemini (LLM)	Performs NLP-based semantic parsing to extract structured fields such as vendor name, date, line items, tax, and total using prompt engineering.
Pattern Matching & Validation (Milestone 2)	Python Regex, Datetime	Applies regex-based field extraction, date normalization, financial validation, and consistency checks to ensure correctness of extracted data.
Duplicate Detection (Milestone 2)	Text normalization, Hash/Comparison Logic	Detects duplicate receipts by comparing cleaned OCR text to prevent redundant storage.
Storage	SQLite	Relational database for storing receipts and line items with data integrity.

---

9. **Validation Strategy**

Validation in the Receipt and Invoice Digitizer is implemented at multiple levels to ensure data accuracy, consistency, and reliability throughout the processing pipeline.

- **AI Schema Enforcement**
  Structured JSON prompting is used while interacting with Gemini AI to enforce a fixed output schema, ensuring consistent and predictable field extraction.
- **Application-Level Validation**
  Python-based checks validate mandatory fields, numeric data types, and value correctness before further processing.
- **Duplicate Detection Validation**
  Early duplicate detection is performed using cleaned OCR text to prevent redundant processing and storage of identical receipts.
- **Database-Level Integrity Constraints**
  Unique and relational constraints in the SQLite database ensure data integrity and prevent duplicate or inconsistent records.
- **Failure Handling & Safe Exits**
  Invalid, incomplete, or low-quality extractions are safely rejected to maintain overall system reliability.

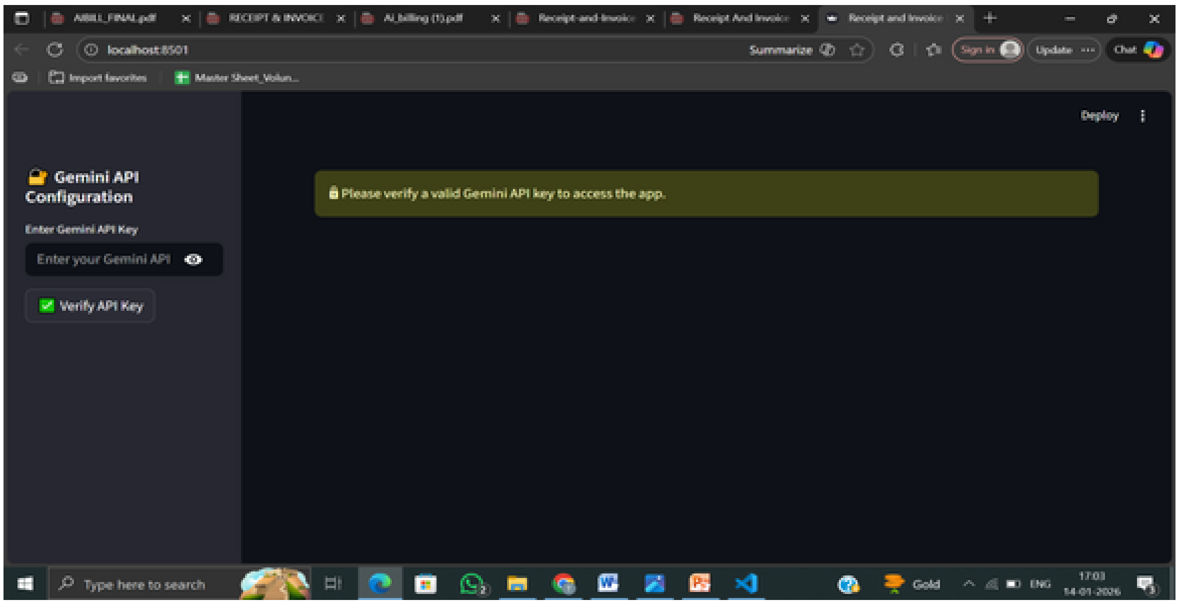
---

10. **Error Handling and System Stability**

The Receipt and Invoice Digitizer incorporates multiple error-handling mechanisms to ensure stable operation and a smooth user experience.

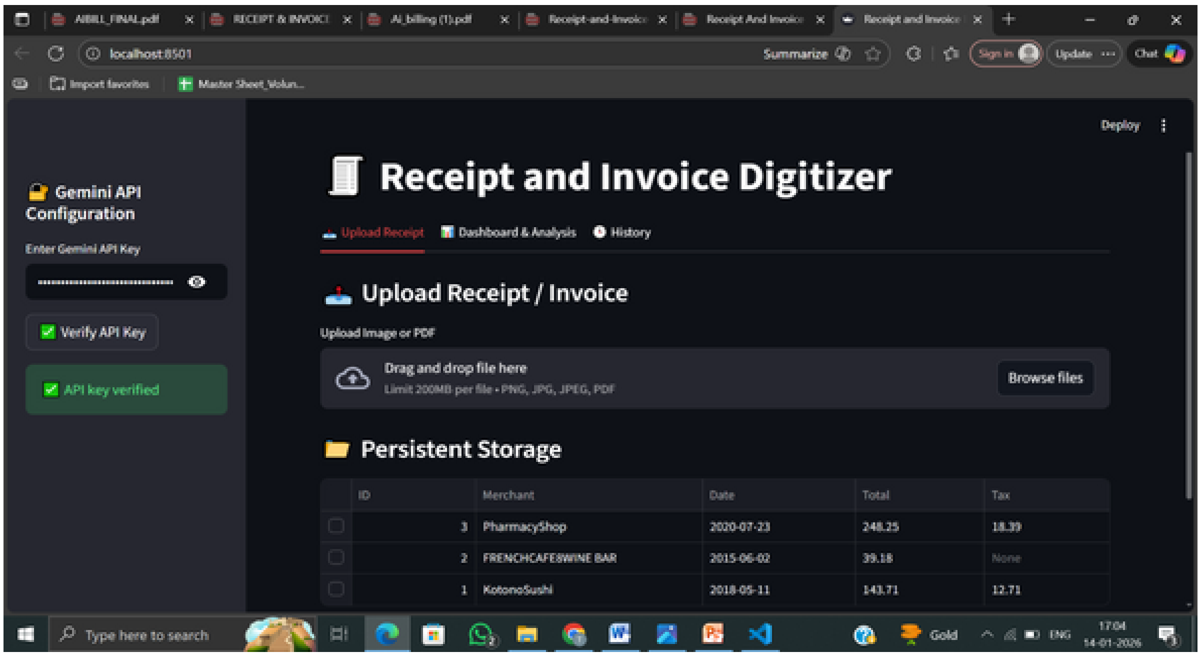
- **API Key Validation & Access Control**
  Detects invalid or missing Gemini API keys and restricts access to AI-based extraction until a valid key is provided.
- **AI API Rate-Limit Handling**
  Handles API rate-limit and service errors gracefully by displaying informative warnings without crashing the application.
- **OCR & Parsing Failure Handling**
  Safely manages OCR or AI parsing failures by stopping further processing and notifying the user.
- **Duplicate Upload Handling**
  Prevents application crashes caused by duplicate uploads through early duplicate detection and database constraints.
- **Graceful User Feedback**
  Displays clear warning and error messages using the interface instead of terminating execution.
- **Crash Prevention & Safe Execution Flow**
  Uses controlled execution paths and exception handling to prevent unexpected application termination.

11. **Outputs**



Please verify a valid Gemini API key to access the app.

1.1 Landing Page



Receipt and Invoice Digitizer

Upload Receipt / Invoice

Upload Image or PDF

Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG, PDF

Persistent Storage

ID	Merchant	Date	Total	Tax
3	PharmacyShop	2020-07-23	248.25	18.39
2	FRENCHCAFE8WINE BAR	2015-06-02	39.18	None
1	KotonoSushi	2018-05-11	143.71	12.71

1.2 Uploading screen

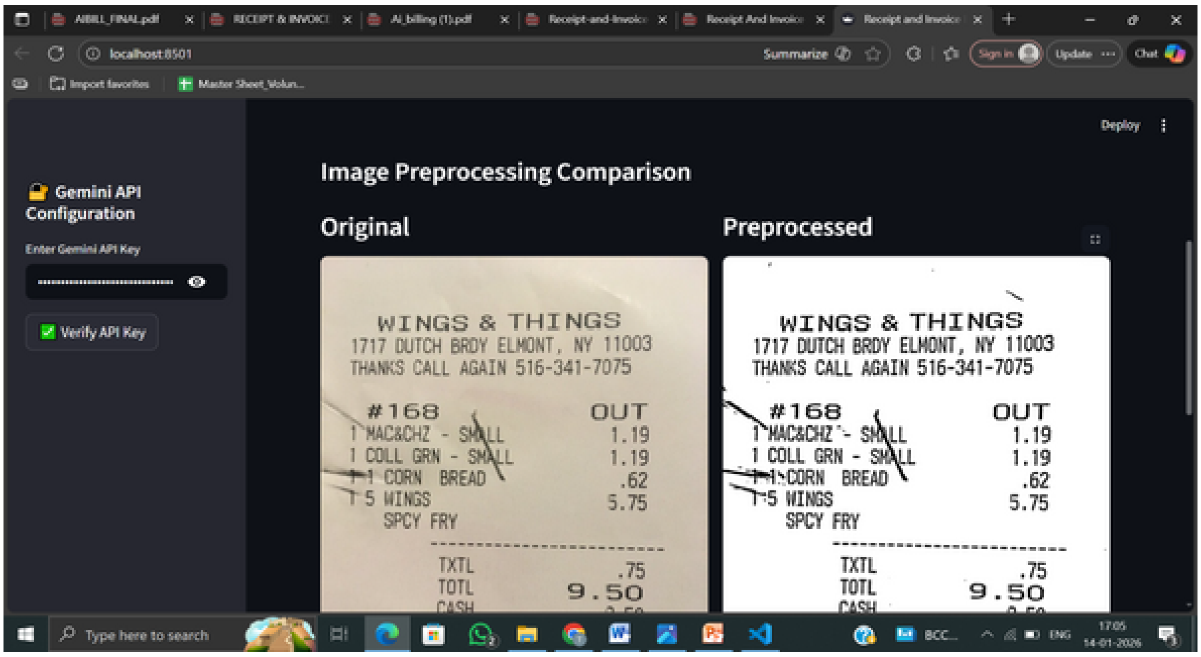


Image Preprocessing Comparison

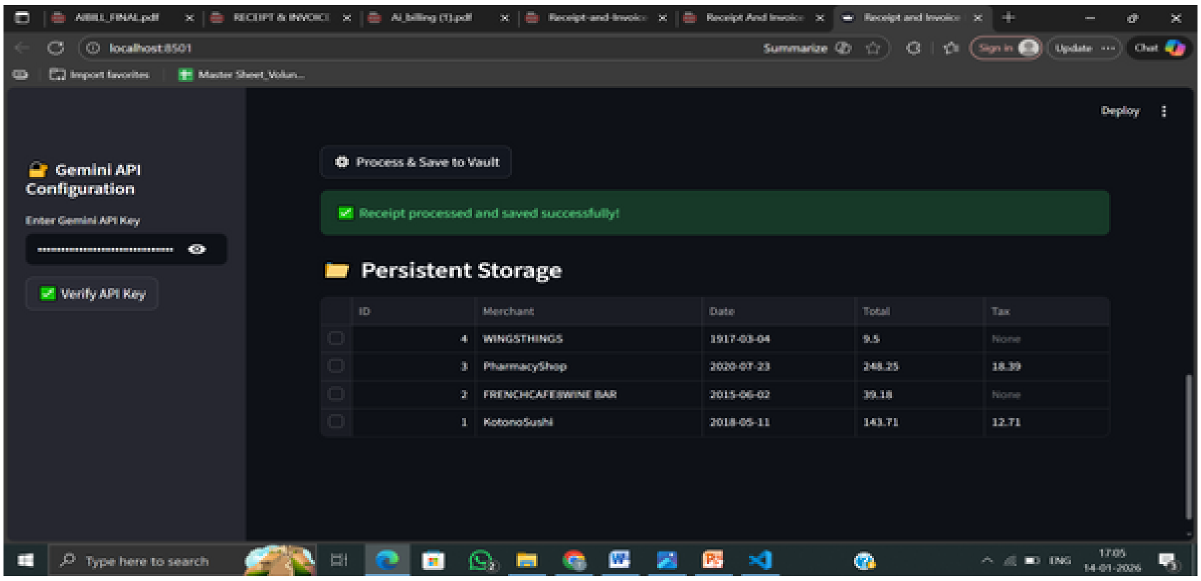
Original

```
WINGS & THINGS
1717 DUTCH BRDY ELMONT, NY 11003
THANKS CALL AGAIN 516-341-7075

#168                         OUT
1 MAC&CHZ - SMALL           1.19
1 COLL GRN - SMALL          1.19
1 CORN BREAD                 .62
1 5 WINGS                   5.75
  SPCY FRY
-------------------------------
     TXTL                    .75
     TOTL                   9.50
```

Preprocessed

```
WINGS & THINGS
1717 DUTCH BRDY ELMONT, NY 11003
THANKS CALL AGAIN 516-341-7075

#168                         OUT
1 MAC&CHZ - SMALL           1.19
1 COLL GRN - SMALL          1.19
1 CORN BREAD                 .62
1 5 WINGS                   5.75
  SPCY FRY
-------------------------------
     TXTL                    .75
     TOTL                   9.50
```

1.3 Image Preprocessing Comparison

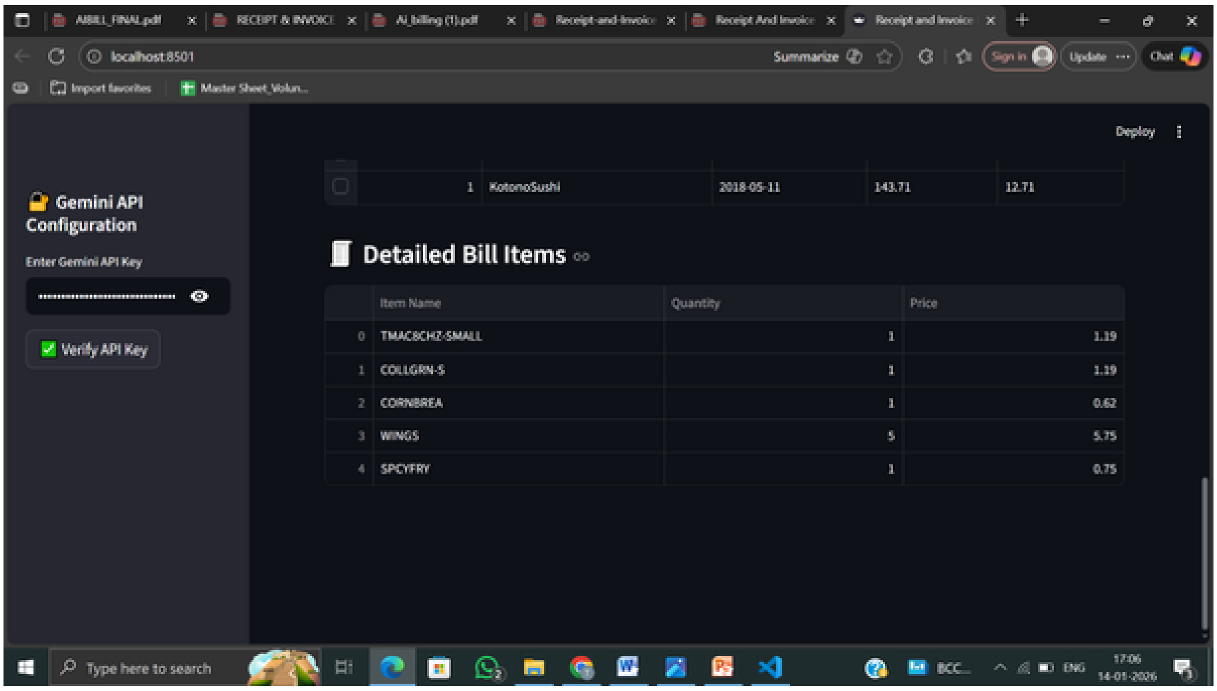


Receipt processed and saved successfully!

Persistent Storage

ID	Merchant	Date	Total	Tax
4	WINGSTHINGS	1917-03-04	9.5	None
3	PharmacyShop	2020-07-23	248.25	18.39
2	FRENCHCAFE&WINE BAR	2015-06-02	39.18	None
1	KotonoSushi	2018-05-11	143.71	12.71

1.4 Persistent Storage



	1	KotonoSushi	2018-05-11	143.71	12.71

Detailed Bill Items

	Item Name	Quantity	Price
0	TMAC8CHZ-SMALL	1	1.19
1	COLLGRN-S	1	1.19
2	CORNBREA	1	0.62
3	WINGS	5	5.75
4	SPCYFRY	1	0.75

1.5 Detailed Bill Items Table

---

12. **Conclusion**

The **Receipt and Invoice Digitizer** project presents an AI-powered solution for automating the extraction, validation, and storage of financial data from receipts and invoices. By integrating **Tesseract OCR**, **Gemini AI**, and a **SQLite database** within a **Streamlit-based web application**, the system successfully converts unstructured documents into structured, reliable digital records.

The project reduces manual data entry, improves accuracy through multi-layer validation, and prevents duplicate records, ensuring consistent and secure data management. Overall, this project demonstrates the effective application of OCR, artificial intelligence, and modern web technologies to solve a real-world financial document management problem and provides a strong foundation for future scalability and enhancements.

Milestone 2: Field Extraction & Validation

- ## Objective

The objective of this milestone is to transform unstructured OCR text from receipts and invoices into **reliable, accurate, and structured digital records**. This milestone focuses on extracting key fields such as vendor details, transaction date, line items, quantities, subtotal, tax, and total amount using a **hybrid approach of Regex-based pattern matching and NLP-driven contextual understanding**.

Another key objective is to **validate the extracted financial data** by cross-verifying subtotal, tax, and total values to identify inconsistencies or extraction errors. The system also aims to **detect duplicate**

receipts by comparing cleaned OCR content, ensuring data integrity and preventing redundant storage.

Finally, this milestone ensures that all validated information is **stored in a structured and persistent database**, enabling efficient retrieval, historical tracking, analytics, and seamless integration with future dashboards or accounting systems.

• Submodules Of Milestone-2

• Apply regex + NLP to parse key fields.

• Validate totals and detect duplicates.

• Store structured results in DB.

1. Apply Regex + NLP to Parse Key Fields

After OCR converts receipt images into raw text, the extracted text often contains noise, formatting inconsistencies, and OCR errors. To handle this, a **hybrid extraction approach** is implemented using **Natural Language Processing (NLP)** and **Regular Expressions (Regex)**.

NLP-Based Extraction

- An NLP model (Gemini LLM) is used to understand the **semantic structure** of receipts.
- It identifies fields such as:
  - Vendor name
  - Date
  - Line items (item name, quantity, price)
  - Subtotal, tax, and total
- NLP provides flexibility to handle different receipt layouts and formats.

Regex-Based Extraction

- Regex patterns are applied as a **deterministic fallback** to extract critical numerical fields.
- Regex is used to:
  - Extract dates in common formats
  - Identify monetary values near keywords like *Total*, *Tax*, *Subtotal*
  - Capture invoice or receipt IDs
- Regex ensures consistent extraction when NLP output is incomplete or uncertain.

Hybrid Strategy

- NLP handles structure and semantics.
- Regex ensures accuracy and consistency.
- This hybrid approach improves robustness across varied receipt formats.

2. Validate Totals and Detect Duplicates

Raw extracted values may contain inconsistencies due to OCR or parsing errors. Therefore, a validation layer is applied before storing data.

Total and Tax Validation

- The system verifies financial correctness using arithmetic rules:
  - **Subtotal + Tax ≈ Total**
- A tolerance threshold is applied to account for rounding errors.
- Printed receipt totals are prioritized over computed totals to prevent OCR-induced inaccuracies.
- Tax values are validated to ensure they fall within reasonable percentage ranges.

Date Validation

- Dates are normalized into a standard format (YYYY-MM-DD).
- The system distinguishes between:
  - Missing dates
  - Invalid date formats
  - Valid dates

Required Field Validation

- Ensures mandatory fields (vendor, date, total) are present.
- Missing fields are flagged for review instead of silently failing.

Duplicate Detection

- Duplicate receipts are detected by comparing cleaned OCR text against stored records.
- If a receipt with the same textual fingerprint already exists:
  - The system prevents re-insertion
  - A warning is displayed to the user
- This avoids duplicate financial records and maintains data integrity.

3. Store Structured Results in Database

Once extraction and validation are complete, the verified data is stored in **persistent storage** using SQLite.

Database Structure

- Receipts are stored in a structured format:
  - Receipt ID
  - Vendor name
  - Date

    - Subtotal, tax, total
- Line items are stored separately and linked using a foreign key relationship.

Benefits of Structured Storage

- Enables fast searching and filtering
- Supports analytics and reporting
- Allows historical tracking of receipts
- Prevents data loss and duplication

Persistence Strategy

- Only validated and non-duplicate receipts are stored.
- Structured storage ensures the system can scale to handle large numbers of receipts reliably.

- ## Detailed Workflow

Upload → Preprocess → OCR → Clean Text

↓

Duplicate Check

↓

Gemini NLP Parsing

↓

Regex Refinement

↓

Field Validation

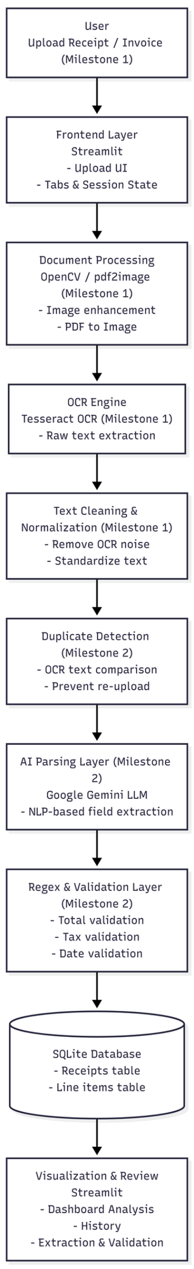
↓

SQLite Storage

↓

Dashboard / History / Validation View

- ## System Architecture



- ## Outcome – Milestone 2:

By the successful completion of this milestone, the system achieves a reliable pipeline for transforming raw receipt data into meaningful, structured information. The outcomes of this milestone are as follows:

1. Structured Data Generation

- Unstructured text obtained from OCR is accurately converted into **well-defined structured records**, including vendor name, date, line items, quantities, subtotal, tax, and total.
- This transformation makes receipt data machine-readable and suitable for storage, querying, and further processing.

2. Improved Accuracy through Validation

- Financial values such as subtotal, tax, and total are **validated for arithmetic consistency**, reducing errors caused by OCR noise or misinterpretation.

- Date formats and mandatory fields are verified, ensuring the extracted data is both complete and standardized.

3. Duplicate-Free Data Storage

- The system successfully **detects and prevents duplicate receipt uploads**, ensuring that each receipt or invoice is stored only once.
- This improves data integrity and avoids redundancy in persistent storage.

4. Reliable Persistent Storage

- Only validated and verified data is stored in the database, resulting in a **clean, trustworthy dataset**.
- Structured storage enables efficient retrieval, historical tracking, and future extensions such as analytics, dashboards, or accounting integration.

5. Readiness for Advanced Processing

- With accurate, validated, and duplicate-free data, the system becomes **reliable for downstream applications** such as reporting, visualization, trend analysis, and decision support.
- This milestone establishes a strong foundation for further milestones involving analytics and insights.

• Conclusion – Milestone 2:

Milestone 2 successfully establishes a robust mechanism for extracting and validating key information from receipts and invoices. By combining **NLP-based semantic parsing with regex-driven pattern matching**, the system is able to accurately identify essential fields such as vendor details, transaction date, line items, tax, and total amount across varied receipt formats.

The inclusion of **validation checks and duplicate detection** ensures the correctness, consistency, and integrity of the extracted data before it is stored. Storing only validated and structured records in a persistent database significantly improves data reliability and prepares the system for efficient retrieval, analytics, and future integration with financial or reporting platforms.

Overall, this milestone transforms noisy OCR output into **trustworthy, structured data**, forming a strong foundation for subsequent milestones involving analysis, visualization, and intelligent insights.