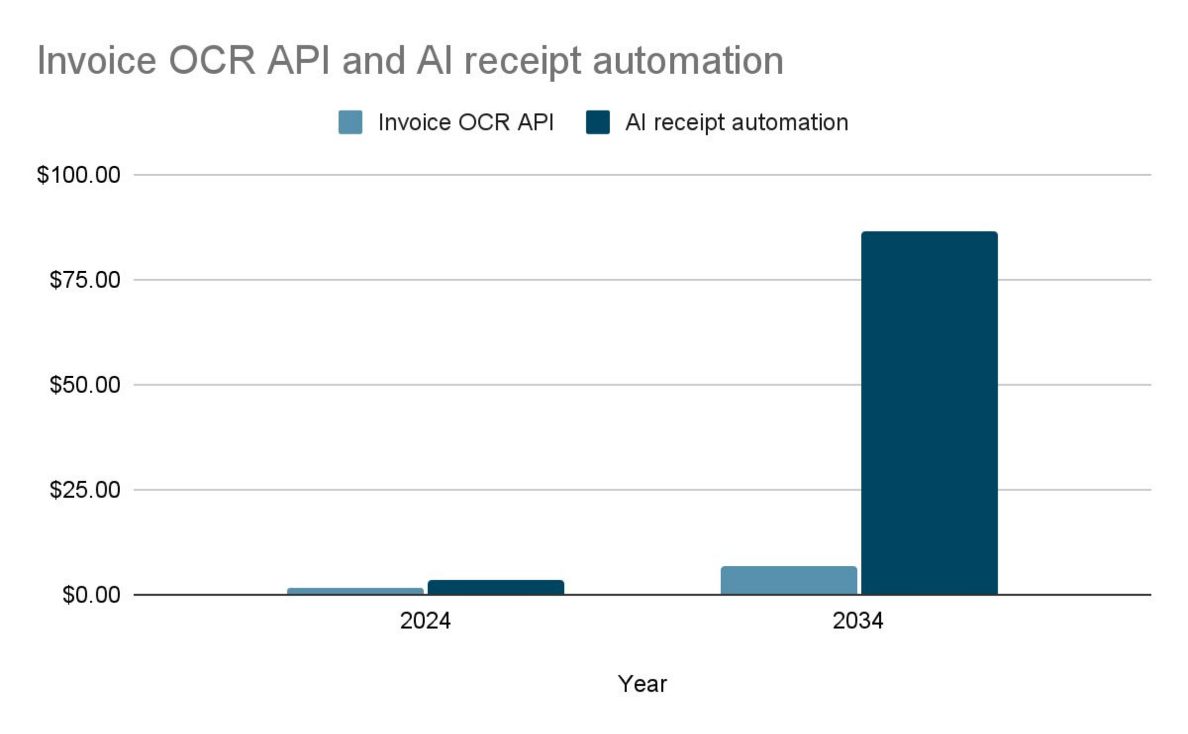
Receipt and Invoice Analyzer

Problem 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.

Market Research

OCR market valued at $17.06 billion in 2025, expanding to $38.32 billion by 2030. Invoice OCR API segment at $1.5 billion in 2024, reaching $5.8 billion by 2033 (CAGR 16.8%). AI-powered receipt automation at $3.5 billion in 2024, surging to $86.4 billion by 2034 (CAGR 37.8%).



Existing Solutions and Limitations

Leading solutions include Veryfi, Google Cloud Vision, Mindee, Rossum, Nanonets, Koncile, ABBYY, and Instabase, focusing on OCR for invoices and receipts. Veryfi leads in speed (2.8s average) and accuracy (98.7%), while Google Cloud Vision prioritizes cost efficiency ($320/1000 docs). Rossum and Nanonets excel in AI/ML-based extraction for unstructured data, with Koncile claiming near-100% field accuracy via computer vision and LLMs

Accuracy Challenges

Existing systems misread 1 in 10 characters, especially on low-quality scans, faded prints, handwritten notes, or crumpled paper.

Handling Variability

Layouts vary wildly by vendor, country, and format, causing keyword search failures and field misidentification.

Manual Intervention and Costs

Even top tools like standalone OCR demand regular manual corrections for 10-15% of documents, inflating operational costs and slowing workflows.

Objectives

- Automated OCR workflow.
- Multi-format file upload.
- History Vault for storage of structured results(on individual Basis.)
- Display Analytics.
- Visually appealing UI design

Project Description

The Receipt and Invoice Digitizer is an AI-driven system that automates the digitization and processing of physical and digital receipts and invoices using Optical Character Recognition (OCR) and Natural Language Processing (NLP), reducing manual effort and improving accuracy in financial operations. The system combines computer vision (OpenCV), OCR (Tesseract), and lightweight NLP to transform unstructured documents into structured, actionable insights — all within a user-friendly web interface.

What it does ?

The application allows users to upload invoices in various formats (PNG, JPG, PDF), automatically extracts key fields such as vendor name, invoice ID, date, tax, and total amount, and validates the extracted data for consistency.

It detects potential duplicates, stores all processed invoices in a personal history vault, and provides interactive analytics including spending trends, vendor comparisons, and category breakdowns.

Users can review raw text, view preprocessed images, manage their invoice history, and even ask natural language questions about their bills via an integrated AI chatbot.

How it Does ?

The pipeline begins by converting uploaded files—whether images or PDFs—into high-quality grayscale images using pdf2image (for PDFs) and OpenCV-based preprocessing (denoising, adaptive thresholding). Tesseract OCR then extracts raw text, which is parsed using regex-based rules to identify structured fields.

All results are saved in a SQLite database scoped per user, enabling persistent storage and history management.

The frontend, built with Streamlit, orchestrates file handling, visualization (Matplotlib/Seaborn), and user interactions, while a Gemini-powered chat interface provides conversational support.

The entire system is modular: core logic resides in ocr_pipeline.py, data persistence in database.py, and UI in app.py—ensuring maintainability and extensibility.

The OCR Engine

Our system uses a robust, multi-stage OCR pipeline optimized for real-world invoice documents. First, uploaded files—whether images or PDFs—are converted into high-quality grayscale images. For PDFs, we use pdf2image at 200 DPI to balance speed and accuracy. Each image then undergoes OpenCV-based preprocessing: noise is reduced with Gaussian blur, and adaptive thresholding creates a clean black-and-white version ideal for text recognition. Tesseract OCR (with PSM mode 1 for automatic layout detection) extracts raw text, which is then parsed using regex patterns to identify structured fields like vendor, invoice ID, date, tax, and total amount. This approach handles diverse layouts while maintaining high accuracy—even on skewed or low-contrast documents.

Data Visualization

Once invoices are processed, the system transforms raw data into actionable insights through interactive visualizations. Using Pandas, Matplotlib, and Seaborn, we generate five key chart types:

- Bar charts compare spending across vendors,
- Line charts reveal monthly cost trends,
- Pie charts show category-wise expenditure,
- Scatter plots (with simulated quality scores) explore price-performance relationships,
- Heatmaps visualize vendor performance matrices.

All charts are dynamically generated from the user’s personal invoice history, enabling quick financial oversight without manual analysis.

Database

Data persistence is handled via a lightweight yet secure SQLite database, ensuring each user’s invoices remain private and accessible across sessions. Every processed invoice—along with its raw text, extracted fields, validation status, and original file bytes—is stored in a structured schema. The system supports full CRUD operations: new invoices are saved automatically after upload, the history vault displays all entries in a searchable table, and users can delete one or multiple records with a single click. Crucially, the database design includes resilience features like short-lived connections and error handling to prevent “database locked” issues in concurrent environments like Streamlit.

UI Design

The frontend is built entirely in Streamlit, offering a clean, intuitive, and responsive interface. The app is organized into logical tabs: Upload, History Vault, Analytics Dashboard, and AI Chat Support. Users can drag-and-drop mixed file types (PDF + images), instantly see original vs. preprocessed previews, and review extracted text. The history vault provides tabular browsing with bulk-delete functionality, while the analytics tab delivers publication-ready visualizations. A sidebar login ensures data isolation, and session state management maintains context during interactions. The entire UI prioritizes clarity, feedback (success/error messages), and zero-config usability—making advanced document intelligence accessible to non-technical users.

Features:

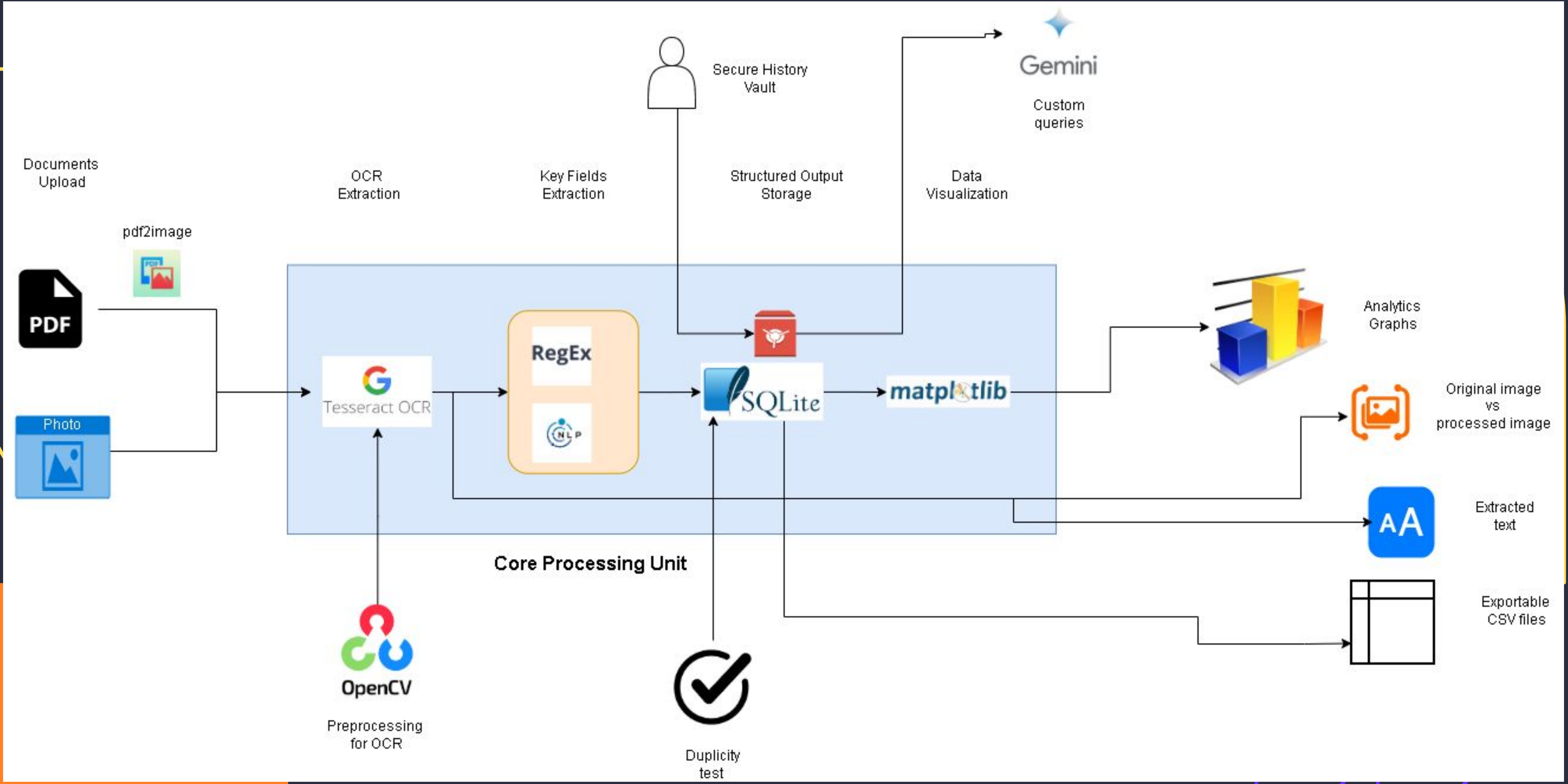
01. Multi-format Invoice Upload

02. Fully Automated OCR Pipeline

03. Personal History Vault

04. Modular Architecture

Architecture Diagram



Tools and resources

Frontend

Streamlit – Web app framework for rapid UI development

Core Libraries and Frameworks

- OpenCV
- Tesseract OCR + pytesseract
- Pdf2image
- Pillow (PIL)
- Matplotlib, Seaborn
- Regex

Database and data management

- SQLite
- Python Standard Library (json, io, datetime)

External APIs

Google Gemini API

Future Works:

01.

Improved Accuracy for >90% accuracy (data Extraction)

02.

Dynamic Data visualization pipeline

03.

Handling User Queries in context to database

04.

Offline Functionality

THANK
YOU

Any questions? Ask away!