OCR/AI INVOICE SCANNING FEATURES

AIM:

To automate the extraction of invoice data from user-uploaded files (Image, PDF, DOC) with high accuracy and minimal manual effort.

OVERFLOW

User Uploads File

(JPG/PNG/JPEG, PDF, DOC/DOCX)

↓

Upload Validation & Security Checks

(PHP / CodeIgniter 3)

↓

Document Normalization

(PDF→images/text: Imagick / DOC/DOCX: PHPWord / LibreOffice)

↓

Image Preprocessing

(ImageMagick: grayscale, deskew, resize, contrast/noise enhancement)

↓

OCR Extraction

(Tesseract LSTM, optional PaddleOCR)

↓

Text Cleaning & Noise Removal

(PHP: regex/string functions)

↓

Structured Data Extraction

(Ollama: convert OCR text → structured invoice fields)

↓

Confidence Scoring

(PHP: combine OCR + Ollama confidence)

↓

Prefilled Editable Invoice Form

(PHP Views: show prefilled fields for high-confidence, empty fields for low-confidence; user can manually type/correct)

↓

Save to **Database**

(MySQL)

LIBRARIES /PLATFORMS USED:

Tool	Free / Paid	Accuracy (Typed)	Handwritten	Limits	Offline	Why Use
ImageMagick	Free	⬆️ Boost OCR	⬆️ Boost	No limits	✅	Improves OCR accuracy before recognition
Imagick (PHP)	Free	⬆️ Conversion	⬆️ Conversion	No limits	✅	Best PHP-friendly document handling
Tesseract LSTM	Free	⭐⭐⭐⭐☆	⭐⭐⭐☆☆	No limits	✅	Best free OCR for printed invoices
PaddleOCR	Free	⭐⭐⭐⭐☆	⭐⭐⭐⭐☆	No limits	✅	Better handwriting & complex layout support
Ollama (Local LLM)	Free	⭐⭐⭐⭐⭐ (Semantic)	Fixes OCR errors	No limits	✅	AI understanding & structured

PAID LIBRARIES (OPTIONAL):

Tool	Paid	Accuracy (Field Extraction)	Structured Invoice Output	Best For
Veryfi OCR API	🟡 Paid	⭐⭐⭐⭐⭐ (~98–99% field-level)	✅ Yes (robust fields + line-item extraction)	Fast, invoice-focused automation with high accuracy and sub-3 s processing on average
ABBYY FlexiCapture / FineReader	🟡 Paid	⭐⭐⭐⭐⭐ (enterprise-grade)	✅ Yes (excellent table & layout parsing)	Deep accuracy with configurable taxonomy, suited for complex global invoices
Google Cloud Vision / Document AI	🟡 Paid	⭐⭐⭐⭐☆ (~94–98%)	✅ Yes (key-value extraction	General OCR + structured extraction with broad language support

PROCEDURE I DONE SO FAR

(1) Imagick download

MY current:
(through test.php(localhost/test.php)
PHP 7.4.1 + Thread Safety enabled + x64
so installed Imagick: *PHP 7.4 TS x64 (vc15)* =extension

installed- *Ghostscript 10.06.0* for Windows (64 bit)

(2) Tesseract

tesseract-ocr-w64-setup-5.3.4.20240503.exe

(3)installed poppler **25.12.0-0**

Python executable:

C:\Program Files\Python311\python.exe

Python packages:
C:\Users\shaha\AppData\Roaming\Python\Python311\site-packages\

Python OCR project:
C:\ocr_service\app.py

PHP project:
C:\xampp\htdocs\your_php_project\

Result:

1 Why the OCR text looks messy (this is expected)

Your invoice has **all of these together**:

- Arabic **RTL** text (headers, totals)
- English **LTR** text (items, totals in words)
- **Tables** (Item / Qty / Unit / VAT)
- **Boxes + grid lines**
- Mixed font sizes
- Logos + QR code

Tesseract **does not understand tables**.
It only outputs **reading order text**, not structure.

So this happens:

What you see in PDF	**What Tesseract sees**
Table rows	Random text blocks
Arabic RTL	Fragmented words
Grid borders	Noise
Item columns	Merged text

That's why:

- Qty / Unit / Price are **not aligned**
- Arabic words look broken
- Some English words are misspelled (Neety, tight)

Correct decision matrix

Requirement	Tesseract	PaddleOCR
Read normal text	✅	✅
Read table text	⚠️ Partial	✅ Better
Detect rows/cols automatically	❌	❌
Rebuild rows via code	⚠️ Hard	✅ Easier
Offline	✅	✅
PHP-only	✅	❌ (Python)

1) Upload → save in invoice_documents (always)

When user uploads (image OR pdf), you must create **one row** in invoice_documents immediately.

Save these columns

- original_file_path → assets/uploads/invoices/{docId}/original/{file}
- original_file_name → uploaded file name
- file_ext → pdf/jpg/png/...
- mime_type → detected mime
- file_size → bytes
- page_count → initially 1 (update later for pdf)
- status → 'uploaded'
- created_at, updated_at

✅ Result: you get **document_id** (AUTO INCREMENT)

This document_id will be used in tables 2 and 3.

---

2) If PDF → normalize pages → save in invoice_document_pages

After upload, do normalization:

Case A: Uploaded file is IMAGE (jpg/png)

- Create **one normalized image**: normalized/page_1.png (or keep original extension)
- Insert **one row** into invoice_document_pages:

Fields

- document_id = document_id from step 1
- page_no = 1
- image_path = assets/uploads/invoices/{docId}/normalized/page_1.png
- ocr_text = NULL (for now)
- ocr_confidence = NULL
- created_at = now

Then update document:

- page_count = 1
- status = 'converted'

Case B: Uploaded file is PDF

- Convert PDF → multiple images:
  - normalized/page_1.png, page_2.png, ...
- Insert **multiple rows** into invoice_document_pages (one per page)

Example:

- page 1 row (page_no=1, image_path=..., ocr_text null)
- page 2 row (page_no=2, image_path=..., ocr_text null)
- …

Then update document:

- page_count = total pages
- status = 'converted'

✅ Your table already has a unique key (document_id, page_no) — perfect.

---

3) OCR stage → update invoice_document_pages

Now run OCR for each page row:

For each page:

- read image_path
- run tesseract → get text
- compute confidence (optional)
- update that same row:

Update these fields in invoice_document_pages

- ocr_text = extracted text
- ocr_confidence = confidence number (0–100 or 0–1, but your column is decimal(5,2))

After OCR completes, update document:

- status = 'ocr_done'

---

4) Prefill → user edits → final save to invoices

When user clicks **Save Invoice**, insert into invoices.

Must save

- document_id (foreign reference)
- all invoice fields: dates, order_no, reference_no, subject, totals, etc.
- set is_confirmed = 0 (or 1 if you want)
- created_at
  Then update document:
- status = 'saved'

Ethicfin Al Ghamdi Contracting Est. مؤسسة بن علي بن سعد ال محي الغامدي للمظاولاث
Al-Muzahimiyah, Wasila, 19658 =: VAIoA, العام»وسيلة»المزاحمية
CR : 798796645324 — ١٠١ NAAN : رقم السجل
VAT NO: 1224567456467 TYAN... Y : رقم الضريبية
TAX INVOICE / فاتورة ضريبية
Customer Name . ee ADIN SS اسم العميل
)' 0345 13D Al Manar Dist DAMMAM SAUDI ARABIA a.
SS ١# 7600 للتوزيع الدمام المملكة العربية السعودية all | eons! العنوان Due Date | 20-Jan-2026 | تاريخ الاستحقاق
Address للعميل
ZIP/Postal Code: 32273
1. 10 days returns policy. 2. Risk passes to the buyer
upon delivery. Ownership of goods remains with the
Payment Terms company until full payment is received. 3. All شروط الدفع Our Ref رقم المرجعى
applicable taxes, duties, and levies will be charged :
extra as per prevailing laws.
جموع امل قيمة الضريبة ين سعرالوحدة الوحدة الكمية الاسم والوصف رمز الصنف
#
_— . . Without VAT .
Item Code Item & Description Qty Unit Price VAT Amount Total inc. VAT
1} Plo1168 me lel 100 | Box | 70.00 | 5,999.70 899.96 6,899.66
Total Without VAT الاجمالي بدون ضريبة 7,000.00
VAT Total 15% ضريبة القيمة المضافة 899.96
Total With VAT in SAR المجموع شامل الضريبه 6,899.66
Saudi Riyals Six Thousand Eight Hundred Ninety-Nine and Sixty-Six Halalas
Bank Account Details | Sid! تفاصيل الحساب
ly ORs 0
ank Name سم الى F ral oF Sta
BAN We BAN46875 اتات