Home Assessment – OCR Endpoint Challenge (/ocr)

- 1 Overview
- 2 Functional Requirements
 - 2.1 Endpoint
 - 2.2 Supported document types
 - 2.3 Field extraction
 - 2.3.1 Referral Letter
 - 2.3.2 Medical Certificate
 - 2.3.3 Receipt
- **3 API Contract**
 - 3.1 Request
 - 3.2 Successful Response (HTTP 200)
 - 3.3 Error Responses
- 4 Deliverables
- 5 Appendix
 - 5.1 Sample Referral Letter Response (full)
 - 5.2 Suggested Tools & Libraries
 - 5.3 Contact
 - 5.4 Sample Document File

1 Overview @

Fullerton Health's engineering team processes thousands of medical documents every day. In this take-home assessment you will design and build a **single HTTP endpoint** that performs Optical Character Recognition (OCR), automatically detects the document type, and returns a structured JSON payload with the key data fields we need for downstream claim adjudication.

The exercise is intended to assess your skills in:

- Document classification & information extraction (classical / ML / deep-learning—your choice)
- API & micro-service design
- Clean, maintainable code and automated tests
- Written communication (README/comments)

2 Functional Requirements @

2.1 Endpoint *⊘*

Method	Path	Description
POST	/ocr	Accepts a document image/PDF and returns extraction results

2.2 Supported document types *∂*

Internal code	Human label
referral_letter	Referral Letters

medical_certificate	Medical Certificates
receipt	Receipts

The service **must** correctly identify which of the three types was supplied. If the document type is outside this list, return HTTP 422 with a message "unsupported_document_type".

2.3 Field extraction $\mathscr Q$

For each recognised type, extract the fields below **exactly as named**. Unless otherwise noted, missing or un-parsable values should be returned as null.

2.3.1 Referral Letter @

Key	Description
claimant_name	Patient Name
provider_name	Provider / Lab name (must not contain the literal string "Fullerton Health")
signature_presence	true if a handwritten signature is detected, else false
total_amount_paid	Integer with all currency symbols, separators and decimals removed
total_approved_amount	Same rules as above
total_requested_amount	Same rules as above

2.3.2 Medical Certificate 🖉

Key	Description
claimant_name	Claimant Name
claimant_address	Address
claimant_date_of_birth	DD/MM/YYYY
diagnosis_name	Diagnosis
discharge_date_time	DD/MM/YYYY
icd_code	ICD code
provider_name	Provider / Lab name (no "Fullerton Health")
submission_date_time	Admission datetime (DD/MM/YYYY)
date_of_mc	Date of MC (DD/MM/YYYY)
mc_days	Integer number of MC days

Key	Description	
claimant_name	Claimant Name	
claimant_address	Address	
claimant_date_of_birth	DD/MM/YYYY	
provider_name	Provider / Lab name (no "Fullerton Health")	
tax_amount	Integer, stripping all separators & decimals	
total_amount	Integer, stripping all separators & decimals	

3 API Contract @

3.1 Request ℰ

```
1 POST /ocr
2 Content-Type: multipart/form-data
3
4 file=<binary document>
```

Accept both PDF and common image formats (JPG/PNG).

3.2 Successful Response (HTTP 200) @

```
1 {
2
    "message": "Processing completed.",
   "result": {
4
      "document_type": "referral_letter",
5
     "total_time": 3.04,
     "finalJson": {
6
7
     "claimant_name": "...",
8
       "provider_name": "...",
9
      "signature_presence": true,
      "total_amount_paid": 3000000,
10
11
       "total_approved_amount": 3000000,
12
        "total_requested_amount": 3000000
    }
13
14 }
15 }
```

*_elapsed * timings are optional but nice to have.*

3.3 Error Responses $\mathscr O$

Status	Condition	Example body
400	No file / invalid MIME type	{ "error": "file_missing" }
422	Unsupported document type	<pre>{ "error": "unsupported_document_type" }</pre>
500	Unhandled exception	<pre>{ "error": "internal_server_error" }</pre>

4 Deliverables @

- Source code (public Git repository or zip)
- **README** with:
 - o Setup & run instructions
 - How to extend to new document types
- Sample curl commands and (optional) Postman collection

Tip: You do **not** need to train a full model; simple pattern rules or off-the-shelf LLM calls are acceptable as long as they meet the requirements.

5 Appendix *∂*

5.1 Sample Referral Letter Response (full) @

```
2 "date_of_birth": "1990-01-01",
3
     "gender": "Female",
     "location": "Tampines Polyclinic",
5
     "referral_no": "POLYS0000000003",
     "visit_date": "2022-09-05",
6
7
     "hospital": "Singapore National Eye Centre",
     "department": "SNEC@Bedok",
8
9
     "referral_type": "NORMAL",
     "appointment_date_and_time": null,
10
11
     "appointment_centre_tel": "62273333",
     "clinic": null,
     "block": null,
13
14
     "level": null,
15
     "allergies": "No",
     "adverse_reaction": "No",
16
17
     "indication_of_referral": "Others (pls specify): eye pain",
"referral_details": "",
19
     "provisional_diagnosis": "eye pain for evaluation"
20 }
```

5.2 Suggested Tools & Libraries @

- OCR: Tesseract (pytesseract), PaddleOCR, Google Vision API
- PDF handling: pdf2image, PyMuPDF, pdfplumber
- Web framework: FastAPI, Flask, Express (Node.js), or any framework you're proficient in
- ML / Prompting: scikit-learn, transformers, OpenAI GPT-40 (if you have API credits)

5.3 Contact @

For questions, please email john.bi@fullertonhealth.com. We typically reply within one business day.

5.4 Sample Document File $\,\mathscr{O}\,$

