**Technical Documentation for app.py**

**Overview**

app.py is a Python utility to convert PDF documents into structured JSON. It extracts textual content (with heading detection), tables, images, and charts. For scanned or low-quality documents, it falls back to OCR using Tesseract. The output is saved as a JSON file, with references to extracted tables and temporary file paths for images.

**Features**

* **Text Extraction**
  + Uses pdfplumber to extract text blocks.
  + Groups words into paragraphs with heading/sub-heading detection.
  + Fallback OCR support via **pytesseract** for non-searchable text.
* **Tables Extraction**
  + Uses **Camelot** for table detection (lattice and stream modes).
  + Falls back to **pdfplumber** table extraction if Camelot fails.
* **Image/Chart Extraction**
  + Uses **PyMuPDF (fitz)** to extract images.
  + Uses **OpenCV** to determine if extracted images resemble charts, based on line density.
* **JSON Output Structure**  
  Each PDF page is represented as a JSON object with entries for:
  + Paragraphs (with hierarchy and headings)
  + Tables
  + Images and Charts (with temporary system file paths)

**Dependencies**

You need the following Python packages (install via requirements.txt):

* pymupdf (fitz) – PDF parsing and rendering
* pdfplumber – Text and table extraction
* camelot-py – Advanced table detection
* pytesseract – OCR engine (optional, requires Tesseract installation)
* opencv-python-headless & numpy – Image and chart detection
* pandas – Data handling for tables
* pillow – Image handling

System dependency: **Tesseract OCR** (if OCR fallback is needed).

**Installation**

1. Clone or copy the script app.py into a working directory.
2. Install Python requirements: pip install -r requirements.txt
3. If OCR support is needed, install Tesseract on your machine:Windows: Install from [Tesseract binaries] and add to PATH.

**Usage**

The script requires two arguments:

* **Input PDF file path**
* **Output JSON file path**

**Command**

bash

python app.py path/input.pdf path/output.json

**Example**

bash

python app.py ./docs/sample.pdf ./out/result.json

This will:

* Read docs/sample.pdf
* Extract structured information
* Save output to out/result.json

**JSON Output Format**

The generated JSON has the following structure:

Example :-

{

"pages": [

{

"page\_number": 1,

"content": [

{

"type": "paragraph",

"section": "Introduction",

"sub\_section": null,

"text": "This is a heading...",

"possible\_heading": true,

"source": "extracted"

},

{

"type": "table",

"section": null,

"description": null,

"table\_data": [["Column1", "Column2"], ["Val1", "Val2"]]

},

{

"type": "image",

"section": null,

"description": null,

"image\_path": "/tmp/tmpfile.png",

"detected\_chart": false

},

{

"type": "chart",

"section": null,

"description": null,

"image\_path": "/tmp/tmpchart.png",

"detected\_chart": true

}

]

}

]

}