Project-Documentation

Version: 1.0
Author: Teachnik Nest Team
Date: 3-Feb-2025

---

1. Overview

This project extracts structured **table data** from multiple PDFs using **Python, pdfplumber, and Llama3-70B (via Groq API)**. It processes text from PDFs, extracts tables in **JSON format**, and converts them into an **Excel spreadsheet** for easy readability.

---

2. Features

 Extract structured tables from multiple PDFs
 Preserve original table structure in **JSON format**
 Convert extracted data into **Excel (.xlsx)**
 Automatically add a **total row** at the end of each table
 Fully automated process—just add PDFs and run the script

---

3. Requirements

Ensure you have the following installed before running the project:

- **Python 3.8+**
- Required Python libraries (listed in `requirements.txt`)
- **Groq API Key** (You need an API key from Groq)

---

4. File Structure

```
/pdf_extraction_project
├── pdf_files/              # Folder containing input PDF files
├── extracted_data/         # Folder for output JSON & Excel files
├── .env                    # Environment variables (Groq API Key)
├── requirements.txt        # Dependencies
├── main.py                 # Main script to execute the project
├── process.py              # PDF processing functions
├── README.pdf              # This documentation
```

---

5. Installation & Setup

Step 1: Create and Activate a Virtual Environment

It is recommended to run the project inside a **virtual environment** to avoid dependency conflicts.

Windows

```
python -m venv venv
venv\Scripts\activate
```

Mac/Linux

```
python3 -m venv venv
source venv/bin/activate
```

Step 2: Install Dependencies

Run the following command inside the activated virtual environment:

```
pip install -r requirements.txt
```

Step 3: Set Up API Key

1. Open the `.env` file in a text editor.
2. Replace `your_groq_api_key_here` with your actual **Groq API Key**:

```
GROQ_API_KEY=your_groq_api_key_here
PDF_FOLDER=pdf_files
OUTPUT_FOLDER=extracted_data
```

---

6. How to Run

Step 1: Add PDFs

Place all the **PDF files** you want to process inside the `pdf_files/` folder.

Step 2: Run the Script

Ensure the virtual environment is activated, then run:

```
python main.py
```

Step 3: View Extracted Data

- JSON output files will be saved in `extracted_data/`
- An Excel file (`extracted_data.xlsx`) will also be generated with all extracted tables

---

7. Understanding the Code

main.py

- **Handles the main execution flow**
- Calls the `process.py` functions to:
  - Extract text from PDFs
  - Process text with **Llama3-70B**
  - Convert structured JSON into **Excel**

process.py

- **extract_text_from_pdf(pdf_path)** → Extracts text from a given PDF.

- **get_structured_json(pdf_text)** → Sends extracted text to LLM (Llama3-70B) to generate structured JSON.
- **extract_json_from_response(response_text)** → Extracts valid JSON output from LLM response.
- **save_json(data, filename)** → Saves extracted data as JSON.
- **json_to_dataframe(json_data)** → Converts JSON into a Pandas DataFrame and adds a total row.

---

8. Expected Output

Example JSON Output

```
[
  {
    "Month": "January",
    "Room Occupied": 120,
    "Vacancies": 30,
    "Occupancy Rate": 80
  },
  {
    "Month": "February",
    "Room Occupied": 110,
    "Vacancies": 40,
    "Occupancy Rate": 75
  },
  {
    "Month": "Total",
    "Room Occupied": 230,
    "Vacancies": 70,
    "Occupancy Rate": 77.5
  }
]
```

Excel Output

Month	Room Occupied	Vacancies	Occupancy Rate
January	120	30	80
February	110	40	75
Total	230	70	77.5

---

9. Troubleshooting

Issue	Possible Solution
No PDFs found error	Ensure PDF files are placed inside the `pdf_files/` folder.
Empty output files	Check if the PDFs contain selectable text. If they contain scanned images, use OCR tools like Tesseract.
Invalid JSON format error	Ensure the API key is correct and that the LLM is returning proper JSON.

---

10. Customization

Modify JSON Output

To customize the JSON format, edit the **prompt** in `process.py`:

```
prompt = """
Extract tables from the given PDF text...
"""
```

Change Excel Sheet Naming

Modify:

```
sheet_name = os.path.splitext(pdf_file)[0][:30]
```

Adjust Output Folder

Change `OUTPUT_FOLDER` in `.env`:

```
OUTPUT_FOLDER=custom_folder
```

Conclusion

This project automates PDF table extraction, converts it into structured JSON, and generates Excel reports. **Simply place your PDFs in the folder, activate the virtual environment, and run the script!**