PDF Extractor to JSON

This Python script extracts structured data from PDF files and converts it into a JSON format. It uses the pdfplumber library to extract text and tables from PDF pages and the PyPDF2 library to retrieve document metadata. The output JSON file contains metadata, text, and tables for each page in the PDF.

Features

- **Text Extraction**: Extracts and cleans the text from each page of the PDF.
- **Table Extraction**: Extracts tables from the PDF (if any) and structures them in a dictionary format.
- **Metadata Extraction**: Retrieves the document's metadata (such as title, author, and creation date).
- **Interactive File Selection**: Uses tkinter for file selection dialogs to choose the PDF file and specify where to save the resulting JSON output.

Requirements

- Python 3.x
- Dependencies:
  - pdfplumber (for extracting text and tables from PDFs)
  - PyPDF2 (for reading PDF metadata)
  - tkinter (for file dialog)
  - re (standard library for text manipulation)
  - datetime (standard library for date handling)

Coding Challenge: Extract Structured Data from PDF and Generate JSON

- **Sample PDF Content:**

- Assume we have a PDF that contains invoice data, and the extracted text looks like this:
- Invoice Number: INV-2024-001
  Date: 2024-11-26
  Billing Address: 123 Example St.
  Items:
    Item 1, 2 units, $10.50 each
    Item 2, 1 unit, $5.00 each

  Total: $26.00
- From this PDF, the expected JSON output should be:

```
•  json
•  {
     "document_title": "Invoice INV-2024-001",
     "author": "John Doe",
     "date": "2024-11-26",
     "pages": [
       {
         "page_number": 1,
         "text": "Invoice Number: INV-2024-001\nDate: 2024-11-26\nBilling Address: 123
   Example St.\nItems:\n  Item 1, 2 units, $10.50 each\n  Item 2, 1 unit, $5.00 each\nTotal:
   $26.00",
•        "title" : "Invoice Data"
         "tables": [
           {
             "table_name": "Invoice Data",
             "columns": ["Item", "Quantity", "Price"],
             "rows": [
               ["Item 1", 2, 10.5],
               ["Item 2", 1, 5.0]
             ]
           }
         ]
       }
```

Here are the 5 key points that summarize the functionality of this code:

- **PDF Extraction**: The `PDFExtractor` class handles the extraction of text and tables from each page of a PDF using `pdfplumber`. It also extracts metadata such as the document title, author, and creation date using `PyPDF2`.
- **Text and Table Processing**: The code extracts text from PDF pages and cleans it by removing extra spaces and newlines. It also extracts tables from the PDF, structures them into rows and columns, and includes them in the output.
- **Metadata Extraction**: The script retrieves the metadata (title, author, creation date) of the PDF using `PyPDF2`. If metadata is unavailable, it falls back to default values.
- **JSON Generation**: The script combines the extracted text, tables, and metadata into a structured JSON format. This JSON includes page-by-page details, with tables and text data for each page.

- **File Selection & Output**: Using `tkinter`, the script opens a graphical interface for the user to select a PDF file and specify where to save the resulting JSON file. The JSON data is saved in the specified location.