PDF Outline Extractor

A Python-based tool that scans all PDFs in `inputs/`, extracts headings by clustering font-sizes, filters out boilerplate and table text, and writes a structured JSON outline for each PDF into `outputs/`, validated against the provided schema.

Project Structure

```
Challenge_1a/
├── process_pdfs.py
├── requirements.txt
├── schema/
│   └── output_schema.json
└── README.md
```

- **process_pdfs.py**\ Main extraction script:
- Reads all `.pdf` files from `inputs/`
- Cleans and normalizes text lines
- Clusters font sizes into H1/H2/H3
- Promotes or filters headings according to heuristics
- Outputs `{pdfname}.json` files in `outputs/`
- **schema/output_schema.json**\ JSON-Schema defining the required `"title"` and `"outline"` structure.
- **requirements.txt**\ Python dependencies:

```
PyMuPDF>=1.23.0
numpy>=1.21.0
scikit-learn>=1.0.0
jsonschema>=4.0.0
```

Local Usage

1. Install dependencies:

   ```
   pip install --no-cache-dir -r requirements.txt
   ```

2. Create the input/output directories:

```
mkdir inputs outputs
```

3. Copy your PDF files into `inputs/`.
4. Run the extractor:

```
python process_pdfs.py
```

5. Find each generated `.json` in `outputs/`.

Docker Usage

1. **Build** the Docker image:

```
docker build --platform linux/amd64 -t pdf-outline:latest .
```

2. **Run** the container:

```
docker run --rm -v "${PWD}\input:/app/input:ro" -v "${PWD}\output:/app/
output" --network none pdf-outline:latest
```

After the container finishes, check `outputs/` for the JSON outlines matching each PDF.