To build a Python app that uploads a PDF and returns its text content, we can use a simple web framework like Flask along with libraries like `pdfplumber` or `PyMuPDF` (or even `pdfminer.six`) for extracting the text from the PDF. Here's how you can set up such an app:

Steps:

1. **Install necessary libraries**:

  - Flask: for the web app

  - `pdfplumber`: for extracting text from PDFs

  - `werkzeug`: to handle file uploads if not included with Flask

```bash
pip install flask pdfplumber
```

2. **Create the Flask app**:

  - This app will allow the user to upload a PDF file, extract its content, and display it.

```python
from flask import Flask, request, render_template, redirect, url_for

import pdfplumber

import os

app = Flask(__name__)

# Configure the folder to store uploaded files

UPLOAD_FOLDER = 'uploads'

app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
```

```
# Ensure the upload folder exists
if not os.path.exists(UPLOAD_FOLDER):
    os.makedirs(UPLOAD_FOLDER)

# Route to upload PDF and extract text
@app.route('/', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        # Check if a file is uploaded
        if 'file' not in request.files:
            return 'No file part'

        file = request.files['file']
        # If no file is selected
        if file.filename == '':
            return 'No selected file'

        if file and file.filename.endswith('.pdf'):
            # Save the uploaded PDF file
            filepath = os.path.join(app.config['UPLOAD_FOLDER'], file.filename)
            file.save(filepath)

            # Extract text using pdfplumber
            with pdfplumber.open(filepath) as pdf:
                text = ''
                for page in pdf.pages:
                    text += page.extract_text() + '\n'

            return f'<h1>Extracted Text</h1><pre>{text}</pre>'
        return 'Invalid file type, please upload a PDF.'

    return '''
    <!doctype html>
    <title>Upload PDF</title>
    <h1>Upload a PDF to extract its text</h1>
    <form method=post enctype=multipart/form-data>
      <input type=file name=file>
      <input type=submit value=Upload>
    </form>
    '''

if __name__ == '__main__':
    app.run(debug=True)
```

How it works:

1. The Flask app presents a simple form for uploading a PDF file.

2. Once uploaded, the PDF is saved in a designated folder, and the app extracts the text from each page using `pdfplumber`.

3. The extracted text is then displayed on the webpage.

Running the App:

1. Save the code in a file, e.g., `app.py`.

2. Run the app:

   ```bash
   python app.py
   ```

3. Open your browser and go to `http://127.0.0.1:5000/` to upload a PDF and see the extracted text.

You can expand this further by adding file validation, multi-language support, or even additional features like downloading the extracted text.