PolicyNav Documentation

Overview

PolicyNav is a Streamlit-based web application designed to help users upload, extract, and interact with policy documents such as PDFs and images. The app uses OCR and PDF parsing to extract text, saves the extracted text to XML, and provides a chat interface for user interaction. It also supports working with sample datasets, such as those stored in the datasets folder.

Features

Upload PDF or image files (PNG, JPG, JPEG)
Extract text from scanned or digital PDFs and images
Save extracted text as XML files
View PDF metadata
Chat interface for interacting with extracted text
Use sample datasets for testing and demonstration

Pre-trained Datasets

Sample datasets are stored in the backend/datasets folder. For example:

finance_policy_sample.csv contains sample finance policy data, including policy names, ministries, years, and descriptions.

You can add your own public policy datasets (CSV, XML, PDF, images) to this folder for analysis and experimentation.

Usage

Start the Streamlit app:

```
streamlit run app.py
```

Upload a PDF or image file.
View extracted text and metadata.
Extracted text is automatically saved as an XML file in the temp_files directory.
Interact with the file using the chat interface.
Use sample datasets from the datasets folder for testing.

Use of Datasets

The datasets folder contains sample policy datasets such as finance_policy_sample.csv.
You can upload these datasets or your own documents through the app for extraction and analysis.
The app is flexible and works with any policy-related document or dataset relevant to your research.

Libraries and Tools Used

streamlit for building the web interface
os for file and directory operations
xml.etree.ElementTree for saving extracted text as XML
dotenv for loading environment variables
time for initialization delay

PyMuPDF for PDF text and metadata extraction
pytesseract for OCR text extraction from images and scanned PDFs
pdf2image for converting PDF pages to images
Pillow for image processing
opencv-python-headless for advanced image preprocessing
fastapi, watchdog, uvicorn, spacy included for possible API and NLP extensions

External Tool

Tesseract OCR required for OCR functionality

i