Sakshi Vilas Awachare

Government White Paper Voice Assistant - Documentation

1. Setup Instructions

To run the Government White Paper Voice Assistant locally, follow these steps:

1. Install Python 3.8 or higher.
2. Set up a virtual environment (optional but recommended):
   python -m venv venv
   source venv/bin/activate (Linux/macOS)
   venv\Scripts\activate (Windows)
3. Install required Python packages:
   pip install flask spacy PyMuPDF pyttsx3 googletrans==4.0.0-rc1 gTTS
4. Download the English spaCy model:
   python -m spacy download en_core_web_sm
5. Place the target PDF file under a `data/` directory and ensure its path is correctly set in `app.py`.
6. Run the Flask application:
   python app.py
7. Open the browser and go to http://127.0.0.1:5000 to use the assistant.

2. Usage Instructions

- Open the web page and select your preferred language using the language dropdown.
- Type or speak your question related to the government white paper.
- The assistant processes your query and provides a relevant answer from the PDF document.
- The answer is also converted to speech and played in your selected language.
- Audio playback controls are provided for playing, pausing, and stopping the response.

3. System Architecture

The system is composed of the following components:

1. Frontend:
   - Built using HTML, CSS (Bootstrap, custom styles), and JavaScript.

  - Provides multilingual support and microphone integration for voice input.
  - Sends queries to the Flask backend via POST requests.

2. Backend (Flask - Python):
  - Loads and processes a white paper PDF using PyMuPDF.
  - Uses spaCy for keyword extraction and sentence matching.
  - Translates queries and responses using Google Translate.
  - Converts text responses to audio using gTTS (or pyttsx3 fallback).
  - Serves audio files and HTML templates.

3. Languages Supported:
  - English, Hindi, Marathi, Spanish, French.
  - Language code and voice config are used for translation and TTS.

4. File Structure

- app.py: Main backend application (Flask server and logic)
- index.html: Frontend UI and interaction logic
- static/audio/: Folder where generated audio responses are saved
- templates/index.html: Template rendered by Flask
- data/: Contains the white paper PDF