

Offline Translator App with Voice Output and PDF Upload Support

=====

=====

1. Overview

This is a Flask web application that allows users to upload text or PDF files, or enter text directly to translate between languages offline. It supports voice output of the translated text.

2. Requirements

- Python packages:

- flask
- argostranslate
- langdetect
- pyttsx3
- PyPDF2
- fpdf2

Install them using:

```
pip install flask argostranslate langdetect pyttsx3 PyPDF2 fpdf2
```

3. File Structure

translator-app/

├─ app.py

├─ templates/

│ └─ index.html

└─ uploads/

4. app.py (Backend)

```
import os
import pyttsx3
from flask import Flask, render_template, request
from langdetect import detect
import argostranslate.package, argostranslate.translate
import PyPDF2
```

```
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'uploads'
os.makedirs('uploads', exist_ok=True)
```

```
engine = pyttsx3.init()
```

```
argostranslate.package.update_package_index()
argostranslate.package.get_installed_packages()
```

```
def extract_text_from_pdf(pdf_file_path):
    with open(pdf_file_path, 'rb') as f:
        reader = PyPDF2.PdfReader(f)
        text = ""
        for page in reader.pages:
            text += page.extract_text()
        return text
```

```
def translate_text(text, to_lang):
    try:
        from_lang = detect(text)
        translated = argostranslate.translate.translate(text, from_lang, to_lang)
        return translated, from_lang
    except Exception as e:
        return f"Translation error: {e}", "unknown"
```

```
def speak_text(text):
```

```
    engine.say(text)
```

```
    engine.runAndWait()
```

```
@app.route('/', methods=['GET', 'POST'])
```

```
def index():
```

```
    original_text = ""
```

```
    translated_text = ""
```

```
    detected_lang = ""
```

```
if request.method == 'POST':
```

```
    uploaded_file = request.files.get('file')
```

```
    typed_text = request.form.get('typed_text', "").strip()
```

```
    to_lang = request.form.get('to_lang', 'hi')
```

```
    play_voice = request.form.get('voice_output')
```

```
if uploaded_file:
```

```
    filename = uploaded_file.filename.lower()
```

```
    file_path = os.path.join(app.config['UPLOAD_FOLDER'], uploaded_file.filename)
```

```
    uploaded_file.save(file_path)
```

```
if filename.endswith('.txt'):
```

```
    with open(file_path, 'r', encoding='utf-8') as f:
```

```
        original_text = f.read()
```

```
elif filename.endswith('.pdf'):
```

```
    original_text = extract_text_from_pdf(file_path)
```

```
elif typed_text:
```

```
    original_text = typed_text
```

```
if original_text:
```

```
    translated_text, detected_lang = translate_text(original_text, to_lang)
```

```
if play_voice == 'on':  
    speak_text(translated_text)
```

```
return render_template('index.html',  
                        original_text=original_text,  
                        translated_text=translated_text,  
                        detected_lang=detected_lang)
```

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

5. templates/index.html (Frontend)

```
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="UTF-8">  
    <title>Offline Translator</title>  
    <style>  
        body {  
            font-family: Arial, sans-serif;  
            background: #eef2f3;  
            padding: 40px;  
        }  
        .container {  
            background: white;  
            padding: 30px;  
            max-width: 700px;  
            margin: auto;  
            border-radius: 10px;  
            box-shadow: 0 0 10px rgba(0,0,0,0.1);  
        }  
        textarea {
```

```

    width: 100%;
    height: 120px;
    margin: 10px 0;
}
select, input[type="file"], button {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
}
label {
    font-weight: bold;
}
.note {
    font-size: 0.9em;
    color: #555;
}
</style>
</head>
<body>
    <div class="container">
        <h2>Offline Translator App</h2>
        <form method="POST" enctype="multipart/form-data">
            <label>Upload a .txt or .pdf file (optional):</label>
            <input type="file" name="file" accept=".txt,.pdf">

            <label>OR Enter text:</label>
            <textarea name="typed_text" placeholder="Type your text here...">{{
request.form.typed_text }}</textarea>

            <label>Translate to:</label>
            <select name="to_lang">
                <option value="en">English</option>
                <option value="hi">Hindi</option>

```

```
<option value="kn">Kannada</option>
<option value="ta">Tamil</option>
<option value="te">Telugu</option>
</select>
```

```
<label>
```

```
<input type="checkbox" name="voice_output"> Speak translated text aloud
```

```
</label>
```

```
<button type="submit">Translate</button>
```

```
</form>
```

```
{% if original_text %}
```

```
<h3>Original Text:</h3>
```

```
<textarea readonly>{{ original_text }}</textarea>
```

```
<p class="note">Detected language: {{ detected_lang.upper() }}</p>
```

```
{% endif %}
```

```
{% if translated_text %}
```

```
<h3>Translated Text:</h3>
```

```
<textarea readonly>{{ translated_text }}</textarea>
```

```
{% endif %}
```

```
</div>
```

```
</body>
```

```
</html>
```

6. How to Run

- Place the files as per the folder structure.
- Run `python app.py`.
- Open the browser at <http://127.0.0.1:5000>.
- Upload .txt or .pdf file or type text directly.
- Choose target language and optionally enable voice output.

- Click Translate.