Documentation for Translator web app

Submitted by: Himadri Goswami

Employee ID:46159312

**1.Introduction**

The Translation Application aims to provide a context-aware and domain-specific translation service with an active frontend, API-based connection, and backend processing. The application will primarily focus on translating text from English to German and Italian, with a domain emphasis on Technology. This document outlines the basic functionalities and requirements for developing this small-scale web-based service.

**2. Functional Requirements**

**2.1 Frontend Features:**

**Language Selection:**

From Language: English (default)

To Language: German, Italian

**Domain Selection:**

Generic and Technology domain

**Input Format:**

Supports input in word or digital PDF format

**User Interface:**

Simple and user-friendly design

**Submit Button:**

Initiates the translation process

**2.2 Backend Processing:**

**Translation Process:**

Utilizes API-based connection to perform translations

**Post Processing:**

Automatically downloads the translated document

Provides an option for the user to download the translated document

**3. Technology Stack**

**Frontend:**

Streamlit python

**Backend:**

Programming Language: Python

**API:**

Google Translate API

4. **Frontend code:**

import streamlit as st

import os

from googletrans import Translator

from PyPDF2 import PdfReader

from docx import Document

def translate\_text(text, target\_language):

    translator = Translator()

    translated\_text = translator.translate(text, dest=target\_language)

    return translated\_text.text

def translate\_file(input\_file\_path, output\_file\_paths, target\_language):

    # Read text from input file

    if input\_file\_path.endswith('.docx'):

        doc = Document(input\_file\_path)

        text = '\n'.join([paragraph.text for paragraph in doc.paragraphs])

    elif input\_file\_path.endswith('.pdf'):

        with open(input\_file\_path, 'rb') as file:

            pdf\_reader = PdfReader(file)

            text = ''

            for page in pdf\_reader.pages:

                text += page.extract\_text()

    else:

        raise ValueError("Unsupported file format")

    # Translate text to target language

    translated\_text = translate\_text(text, target\_language)

    # Write translated text to output file

    with open(output\_file\_paths, 'w', encoding='utf-8') as file:

        file.write(translated\_text)

    return output\_file\_paths

def main():

    st.title("Document Translator")

    # Language selection

    from\_language = "English"  # Fixed as English for now

    to\_language = st.selectbox("To Language", ['German', 'Italian'])

    # Display dropdown menu for domain selection (not used in translation)

    domain = st.selectbox("Domain", ['Generic', 'Technology'])

    # Mapping of language names to language codes

    language\_codes = {

        'German': 'de',

        'Italian': 'it'

    }

    # File upload

    uploaded\_file = st.file\_uploader("Upload a Word or PDF file", type=['docx', 'pdf'])

    # Translate file on submit

    if st.button("Translate"):

        if uploaded\_file is not None:

            output\_file\_paths = f"translated\_output\_{language\_codes[to\_language]}.txt"

            translated\_file\_path = translate\_file(uploaded\_file.name, output\_file\_paths, language\_codes[to\_language])

            st.success(f"Translation to {to\_language} completed.")

            with open(translated\_file\_path, "rb") as file:

                file\_contents = file.read()

            st.download\_button(label="Download Translated File", data=file\_contents, file\_name=output\_file\_paths)

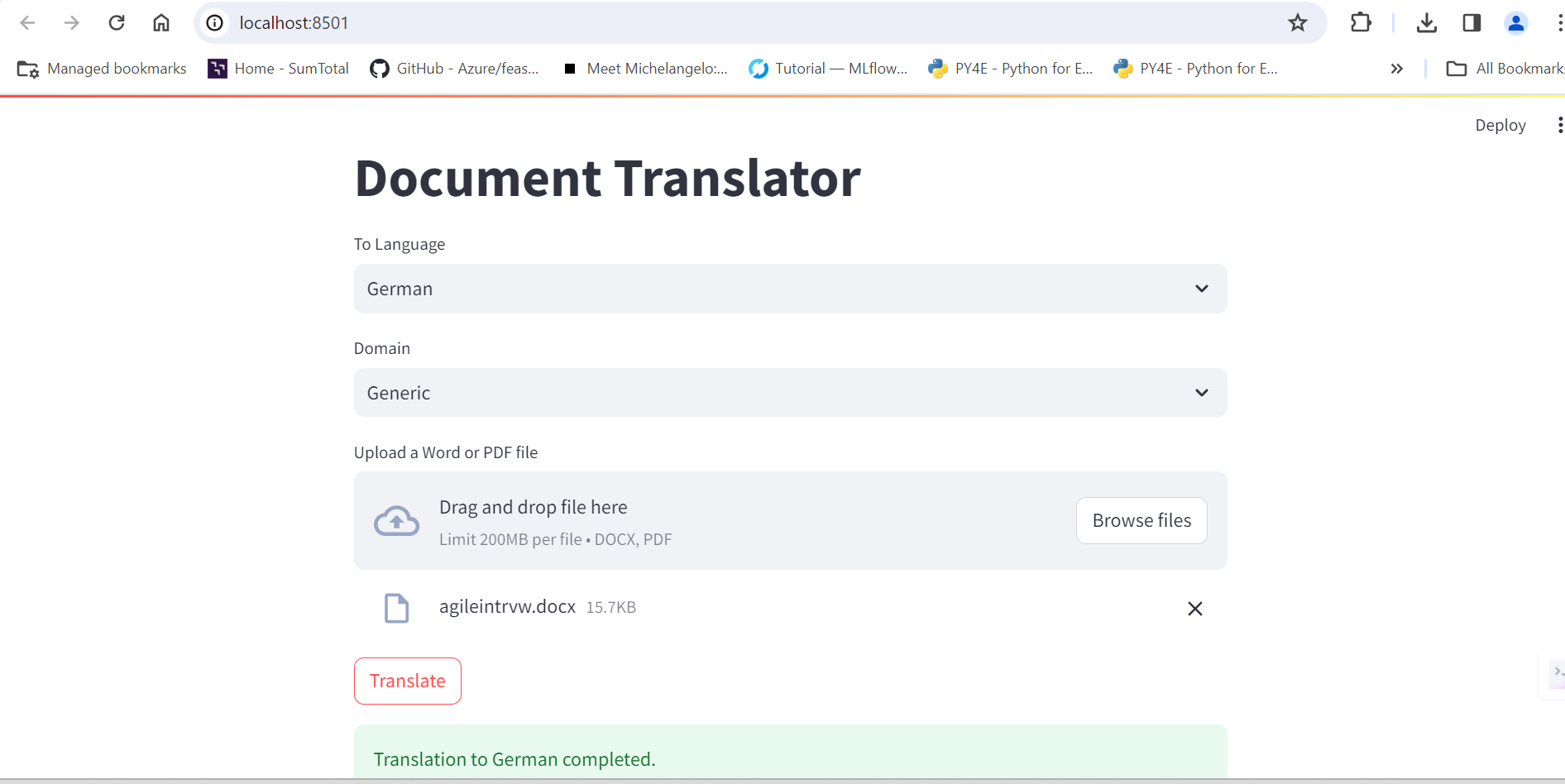
        else:

            st.error("Please upload a file first.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**5.Output Snapshot:**



**6. References:**

1. <https://lokalise.com/blog/how-to-translate-languages-in-python-with-google-translate-and-deepl-plus-more/>

2. https://stackabuse.com/text-translation-with-google-translate-api-in-python/