PRESENTATION
ON
LANGUAGE TRANSLATION
MAJOR PROJECT STAGE I

SUBMITTED TO :-
PRIYANKA GUPTA
(ASSISTANT PROFESSOR)

SUBMITTED BY: -
NAME :- ANWAR ALAM
SID NO :- 100846
NAME :- ADITYA P.R
DEPARTMENT :- B.TECH (CSE) 7TH SEM

TABLE OF CONTENT

INTRODUCTION

Language Translation in Technology With advancements in artificial intelligence (AI) and machine learning (ML), automated translation systems have become more accurate and efficient. These systems, such as **Google Translate**, leverage powerful machine learning models, particularly in the field of **Natural Language Processing (NLP)**, to handle translations across multiple languagees.

Automated Translation:

- Machine Translation (MT)
- Multilingual support

WHAT IS LANGUAGE TRANSLATION IN PYTHON

Language Translation in Python refers to the process of converting text from one language into another using Python programming language tools and libraries. Python, being highly versatile and equipped with powerful libraries, enables developers to integrate machine translation features into applications easily.

Key Concepts in Language Translation with Python:

- Machine Translation (MT)
- Natural Language Processing (NLP)
- Libraries and Tools

ADVANTAGES & DISADVANTAGES

Advantages:

- **Ease of Integration:** Python's libraries like Googletrans make it simple to integrate translation features into web or desktop applications with minimal code.
- **Support for Multiple Languages:** Services like Google Translate support over 100 languages, making Python translation tools suitable for global applications.
- **Real-Time Translation:** Python can be combined with APIs to provide fast, real-time translation for text inputs, enhancing user experiences in dynamic applications.

Disadvantages of Language Translation in Python:

- **Dependency on External APIs:** Many translation libraries in Python depend on third-party APIs, which may limit functionality if there's an API outage or rate limitation.
- **Accuracy Issues:** Machine translations may struggle with idioms, regional phrases, and contextual accuracy, leading to errors in some cases.
- **Limited Customization:** The flexibility to fine-tune translations for specific use cases or industries is limited when relying on general-purpose services like Go
- ogle Translate.

IMPORTS AND SETUP

Flask: Flask is a lightweight web framework used to build web applications.

Googletrans: This is a Python library that uses Google Translate API for translating text.

IPython: It is used to display interactive elements like IFrame within Jupyter Notebooks.

Flask-Ngrok: This is used to expose the Flask app to the internet for external access, by running it through ngrok.

Threading: Allows the Flask app to run in a separate thread within a Jupyter Notebook.

```
from flask import Flask, render_template_string, request, jsonify
from googletrans import Translator
from IPython.display import display, IFrame
from flask_ngrok import run_with_ngrok
import threading
# Instantiate Flask app
app = Flask(__name__)
run_with_ngrok(app)  # Start ngrok to expose the app
```

LANGUAGE CODES DICTIONARY

The language codes dictionary in the translation app maps language names to their respective ISO 639-1 codes, which are used by Google Translate to identify the source and target languages. Here's a sample of the dictionary:

```
lang_codes = {
    'Afrikaans': 'af', 'Albanian': 'sq', 'Amharic': 'am', 'Arabic': 'ar',
    'Armenian': 'hy', 'Azerbaijani': 'az', 'Basque': 'eu', 'Belarusian': 'be',
    'Bengali': 'bn', 'Bosnian': 'bs', 'Bulgarian': 'bg', 'Catalan': 'ca',
    'Chinese (Simplified)': 'zh-cn', 'Chinese (Traditional)': 'zh-tw',
    'Croatian': 'hr', 'Czech': 'cs', 'Danish': 'da', 'Dutch': 'nl',
    'English': 'en', 'Finnish': 'fi', 'French': 'fr', 'German': 'de',
    'Greek': 'el', 'Gujarati': 'gu', 'Hindi': 'hi', 'Italian': 'it',
    'Japanese': 'ja', 'Korean': 'ko', 'Marathi': 'mr', 'Nepali': 'ne',
    'Polish': 'pl', 'Portuguese': 'pt', 'Punjabi': 'pa', 'Russian': 'ru',
    'Spanish': 'es', 'Tamil': 'ta', 'Telugu': 'te', 'Thai': 'th', 'Turkish': 'tr',
    'Ukrainian': 'uk', 'Urdu': 'ur', 'Vietnamese': 'vi', 'Zulu': 'zu'}
```

HTML TEMPLATE

The HTML template serves as the user interface for the language translation application. Below is the complete HTML code that defines the layout, styles, and interactivity for the translation app

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Language Translator</title>
    <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap"
      rel="stylesheet">
    <style>
        body {
            font-family: 'Roboto', sans-serif;
            background-color: #f5f5f5;
```

FLASK ROUTES

In a Flask application, routes define the endpoints that the application responds to. Below is an explanation and implementation of the routes used in the language translation app.

```
from flask import Flask, render_template_string, request, jsonify
from googletrans import Translator
from flask_ngrok import run_with_ngrok
import threading

# Instantiate Flask app
app = Flask(__name__)
run_with_ngrok(app)
```

HELPER FUNCTIONS (JAVASCRIPT IN HTML TEMPLATE)

In the HTML template of the language translation application, several JavaScript helper functions enhance user interactivity and functionality. Below are the key helper functions included in the template, along with their explanations.

```
<script>
  // Function to display the current time (Digital Clock)
  function updateClock() {
    const now = new Date();
    const hours = now.getHours().toString().padStart(2, '0');
    const minutes = now.getMinutes().toString().padStart(2, '0');
    const seconds = now.getSeconds().toString().padStart(2, '0');
    document.getElementById('clock').innerText = hours + ':' + minutes + ':' + seconds;
  }
  setInterval(updateClock, 1000);  // Update clock every second

  function updateCharWordCount() {
    const text = document.getElementById('text').value;
    const wordCount = text.trim().split(/\s+/).filter(word => word).length;
    document.getElementById('word_count').innerText = 'Words: ' + wordCount;
    document.getElementById('char_count').innerText = 'Characters: ' + text.length;
  }
```

CODE

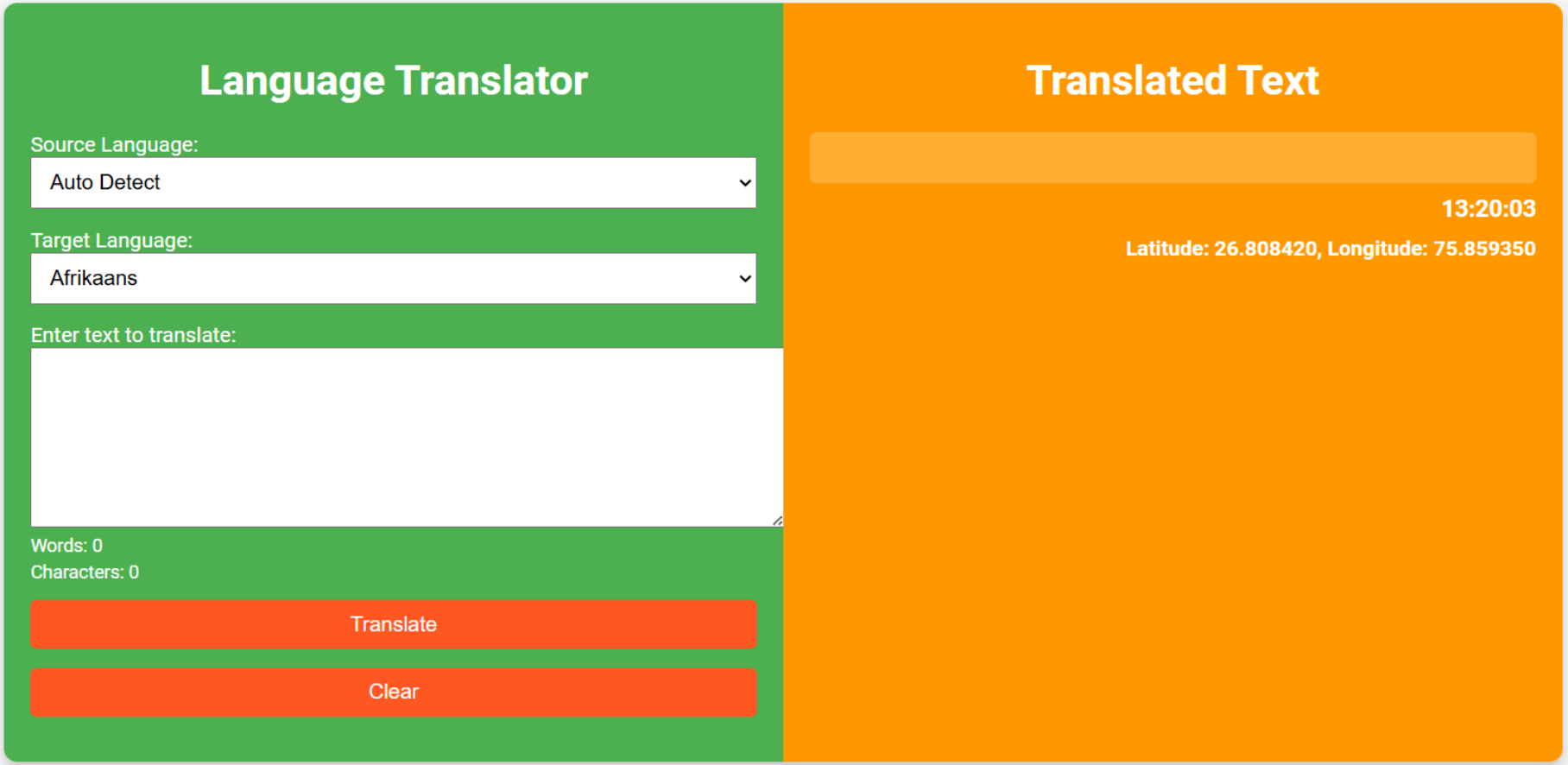
```
3]: from flask import Flask, render_template_string, request, jsonify
from googletrans import Translator
from IPython.display import display, IFrame
from flask_ngrok import run_with_ngrok
import threading

# Instantiate Flask app
app = Flask(__name__)
run_with_ngrok(app)  # Start ngrok to expose the app

# Language codes and names
lang_codes = {
    'Afrikaans': 'af', 'Albanian': 'sq', 'Amharic': 'am', 'Arabic': 'ar',
    'Armenian': 'hy', 'Azerbaijani': 'az', 'Basque': 'eu', 'Belarusian': 'be',
    'Bengali': 'bn', 'Bosnian': 'bs', 'Bulgarian': 'bg', 'Catalan': 'ca',
    'Chinese (Simplified)': 'zh-cn', 'Chinese (Traditional)': 'zh-tw',
    'Croatian': 'hr', 'Czech': 'cs', 'Danish': 'da', 'Dutch': 'nl',
    'English': 'en', 'Finnish': 'fi', 'French': 'fr', 'German': 'de',
    'Greek': 'el', 'Gujarati': 'gu', 'Hindi': 'hi', 'Italian': 'it',
    'Japanese': 'ja', 'Korean': 'ko', 'Marathi': 'mr', 'Nepali': 'ne',
    'Polish': 'pl', 'Portuguese': 'pt', 'Punjabi': 'pa', 'Russian': 'ru',
    'Spanish': 'es', 'Tamil': 'ta', 'Telugu': 'te', 'Thai': 'th', 'Turkish': 'tr',
    'Ukrainian': 'uk', 'Urdu': 'ur', 'Vietnamese': 'vi', 'Zulu': 'zu'
}

# Homepage template for Flask with updated features and digital clock
index_html = """
<!DOCTYPE html>
```

OUTPUT


Language Translator
Source Language:
Auto Detect
Target Language:
Afrikaans
Enter text to translate:
Words: 0
Characters: 0
Translate
Clear
Translated Text
13:20:03
Latitude: 26.808420, Longitude: 75.859350