In [1]:

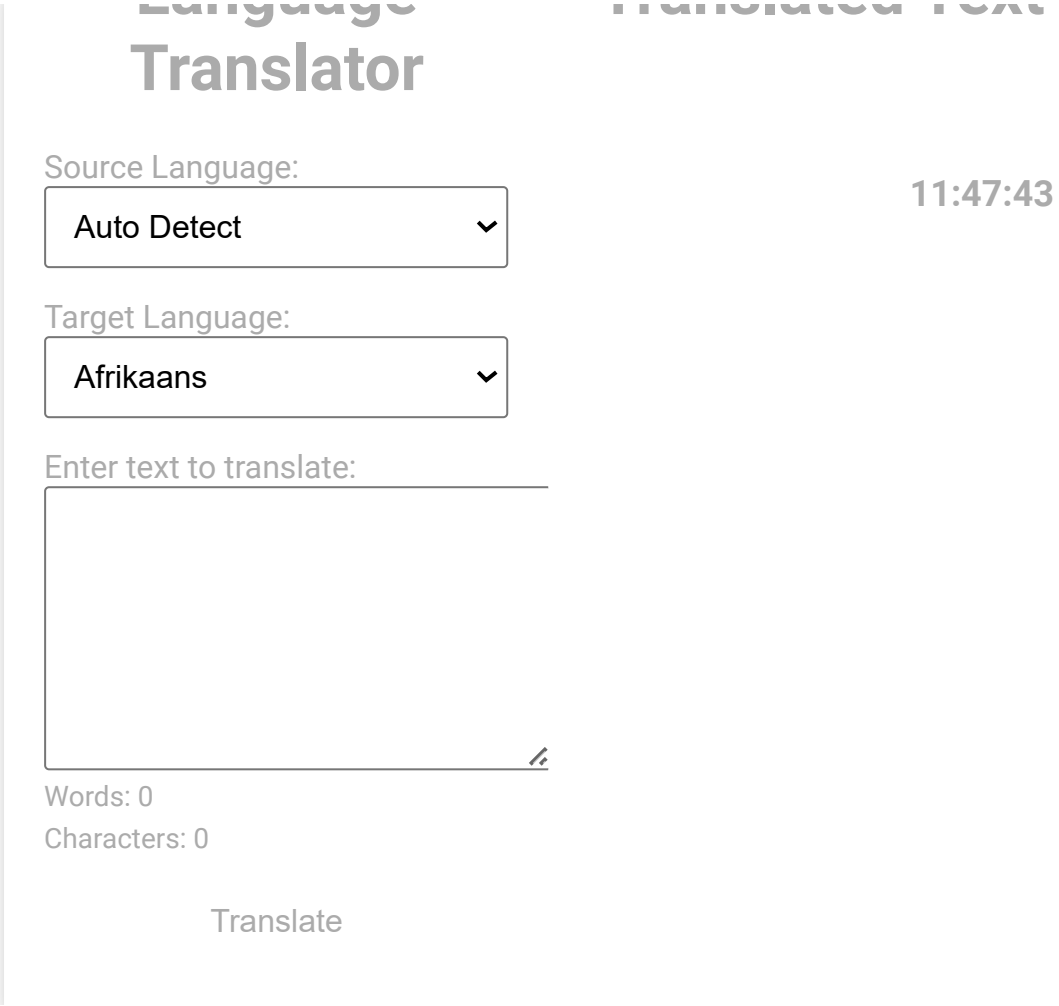
```
from flask import Flask, render_template_string, request, jsonify
from googletrans import Translator
from IPython.display import display, IFrame
from flask_ngrok import run_with_ngrok
import threading

# 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>
<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&displa
    <style>
        body {
            font-family: 'Roboto', sans-serif;
            background-color: #f5f5f5;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            position: relative;
        }
        .container {
            display: flex;
            width: 80%;
            background-color: #ffffff;
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0,0,0,0.2);
        }
        .input-section, .output-section {
            padding: 20px;
            width: 50%;
        }
        .input-section {
            background-color: #4CAF50;
            color: white;
            border-radius: 10px 0 0 10px;
        }
```

```
        .output-section {
            background-color: #ff9800;
            color: white;
            border-radius: 0 10px 10px 0;
        }
        h1 {
            text-align: center;
        }
        select, textarea, button {
            display: block;
            width: 100%;
            margin-bottom: 15px;
            padding: 10px;
            font-size: 16px;
        }
        textarea {
            height: 120px;
        }
        button {
            background-color: #ff5722;
            color: white;
            border: none;
            cursor: pointer;
            border-radius: 5px;
            display: flex;
            justify-content: center;
            align-items: center;
            font-size: 16px;
        }
        button:hover {
            background-color: #e64a19;
        }
        #translated_text_box {
            padding: 20px;
            background-color: rgba(255, 255, 255, 0.2);
            border-radius: 5px;
        }
        #translated_text {
            margin: 0;
        }
        #clock {
            margin-top: 10px;
            color: white;
            font-size: 18px;
            text-align: right;
            font-weight: bold;
        }
        #word_count, #char_count {
            font-size: 14px;
            margin-top: -10px;
            color: white;
        }
    </style>
</head>
<body>

    <div class="container">
        <div class="input-section">
            <h1>Language Translator</h1>
            <label for="source_lang">Source Language:</label>
            <select id="source_lang">
                <option value="">Auto Detect</option>
                {% for lang, code in languages.items() %}
                    <option value="{{ code }}">{{ lang }}</option>
```

```
            {% endfor %}
        </select>

        <label for="target_lang">Target Language:</label>
        <select id="target_lang">
            {% for lang, code in languages.items() %}
                <option value="{{ code }}">{{ lang }}</option>
            {% endfor %}
        </select>

        <label for="text">Enter text to translate:</label>
        <textarea id="text" oninput="updateCharWordCount()"></textarea>
        <p id="word_count">Words: 0</p>
        <p id="char_count">Characters: 0</p>

        <button onclick="translateText()">Translate</button>
        <button onclick="clearText()">Clear</button>
    </div>
    <div class="output-section">
        <h1>Translated Text</h1>
        <div id="translated_text_box">
            <p id="translated_text"></p>
        </div>
        <div id="clock"></div>
    </div>
</div>

<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 + ':
    }
    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
    }

    async function translateText() {
        const sourceLang = document.getElementById('source_lang').value;
        const targetLang = document.getElementById('target_lang').value;
        const text = document.getElementById('text').value;

        const response = await fetch('/translate', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json'
            },
            body: JSON.stringify({
                source_lang: sourceLang,
                target_lang: targetLang,
                text: text
            })
        });

        const result = await response.json();
        document.getElementById('translated_text').innerText = result.translate
```

```
        }

        function clearText() {
            document.getElementById('text').value = '';
            document.getElementById('translated_text').innerText = '';
            document.getElementById('word_count').innerText = 'Words: 0';
            document.getElementById('char_count').innerText = 'Characters: 0';
        }
    </script>
</body>
</html>
"""

@app.route('/')
def index():
    return render_template_string(index_html, languages=lang_codes)

@app.route('/translate', methods=['POST'])
def translate():
    data = request.json
    source_lang = data['source_lang'] or None  # Auto-detect if None
    target_lang = data['target_lang']
    text = data['text']

    translator = Translator()
    translated = translator.translate(text, src=source_lang, dest=target_lang)
    return jsonify({'translated_text': translated.text})

def run_app():
    app.run()

# Start Flask app in a new thread
threading.Thread(target=run_app).start()

# Display the app in Jupyter Notebook
display(IFrame(src="http://127.0.0.1:5000", width=700,height=500))
```

```
 * Serving Flask app "__main__" (lazy loading)
```




```
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deploymen
t.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [14/Oct/2024 10:57:24] "GET / HTTP/1.1" 200 -
Exception in thread Thread-6:
Traceback (most recent call last):
  File "C:\Users\Anwar Alam\anaconda3\lib\threading.py", line 980, in _bootstrap_i
nner
    self.run()
  File "C:\Users\Anwar Alam\anaconda3\lib\threading.py", line 1306, in run
    self.function(*self.args, **self.kwargs)
  File "C:\Users\Anwar Alam\anaconda3\lib\site-packages\flask_ngrok.py", line 70,
in start_ngrok
    ngrok_address = _run_ngrok()
  File "C:\Users\Anwar Alam\anaconda3\lib\site-packages\flask_ngrok.py", line 31,
in _run_ngrok
    ngrok = subprocess.Popen([executable, 'http', '5000'])
  File "C:\Users\Anwar Alam\anaconda3\lib\subprocess.py", line 951, in __init__
    self._execute_child(args, executable, preexec_fn, close_fds,
  File "C:\Users\Anwar Alam\anaconda3\lib\subprocess.py", line 1420, in _execute_c
hild
    hp, ht, pid, tid = _winapi.CreateProcess(executable, args,
OSError: [WinError 193] %1 is not a valid Win32 application
127.0.0.1 - - [14/Oct/2024 10:57:28] "GET / HTTP/1.1" 200 -
```

In [ ]: