**Automatic Text Translator - Project Report & Documentation**

**1. Introduction**

This project involves developing a simple yet functional automatic text translation tool using Python and the googletrans library. The goal is to translate text from a source language to a target language reliably and efficiently.

**2. Objective**

* To create an automated language translation application.
* To facilitate easy translation between different languages.
* To provide a reusable codebase for integration into other applications.

**3. Tools and Technologies**

* **Programming Language:** Python 3.x
* **Library:** googletrans (Google Translate API wrapper)
* **Development Environment:** Any IDE supporting Python (e.g., VSCode, PyCharm)

**4. Code Implementation**

Below is the complete Python code for the translation program:

python

from googletrans import Translator

def translate\_text(text, src\_language, dest\_language):

translator = Translator()

try:

translation = translator.translate(text, src=src\_language, dest=dest\_language)

return translation.text

except Exception as e:

return f"Error during translation: {e}"

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

# Example usage

source\_text = "Hello, how are you?"

source\_lang = "en" # English

target\_lang = "es" # Spanish

translated\_text = translate\_text(source\_text, source\_lang, target\_lang)

print(f"Original ({source\_lang}): {source\_text}")

print(f"Translated ({target\_lang}): {translated\_text}")

**5. Output & Prototype Results**

**Sample Input:**

plaintext

Text: "Good morning!"

Source Language: English (en)

Target Language: German (de)

**Expected Output:**

plaintext

Original (en): Good morning!

Translated (de): Guten Morgen!

**Sample Run Output:**

plaintext

Original (en): Hello, how are you?

Translated (es): Hola, ¿cómo estás?

*Note:* Actual translations may vary slightly depending on the Google Translate service at runtime.

**6. Functional Overview**

* The core function translate\_text() accepts input text and language codes.
* It utilizes Google Translate API to perform translation.
* Errors during translation are caught and displayed as messages.

**7. Prototype & User Interaction**

This script is a command-line application, allowing users to modify variables source\_text, source\_lang, and target\_lang for different translation needs.

**Sample Prototype:**

Table

| **Input Text** | **Source Language** | **Target Language** |
| --- | --- | --- |
| "Bonjour" | "fr" | "en" |
| "Hallo" | "de" | "en" |
| "こんにちは" | "ja" | "en" |

*Outputs are displayed immediately in the console.*

**8. Limitations & Future Enhancements**

* **Limitations:**
  + Reliance on online connectivity; no offline mode available.
  + Rate limits may apply with high-volume usage.
  + Basic error handling, no GUI or advanced features.
* **Future scope:**
  + Develop a GUI interface for easier interaction.
  + Support for batch translation of multiple texts.
  + Incorporate language detection for input texts.
  + Support for additional translation services for redundancy.

**9. Conclusion**

This project successfully delivers an automatic text translation tool using Python. It is simple, extensible, and effective, suitable for integration or further development into larger applications such as multilingual chatbots, website localization, or language learning tools.