# LANGUAGE TRANSLATOR PROJECT

***A report of Virtual Internship (2021-22)***

EXPOSYS DATA LABS

**Submitted by**

Umang Sharma
B.Tech, IT 3rd year
umng01@gmail.com

**Supervised By:**

Mentor, Sir Vishnu Vardhan

# ABSTRACT

The objective of this Python project is to translate a piece of text into another language. You need to install, translate and import two modules: tkinter, googletrans. Also, little knowledge of tkinter is required along with the knowledge of functions in python.

The app interface must be able to convert all the text provided by the user from one end into another language that can also be chosen by the user. So basically, the user can write a text in any of the language and can receive the output in some other language. This enhances the user’s curiosity and interaction during the demonstration of the project.

# TABLE OF CONTENT

# INTRODUCTION

A translation app is an app which is capable of getting an input in some language, producing the same text converted into some other language of the choices selected by the user. This process of conversion of the same piece of text from one language to a variety of languages is known as Translation. Translation can be achieved by the use of Python programming language and for more readability and ease, we can use graphical user interfaces in Python to do so.

# EXISTING METHOD

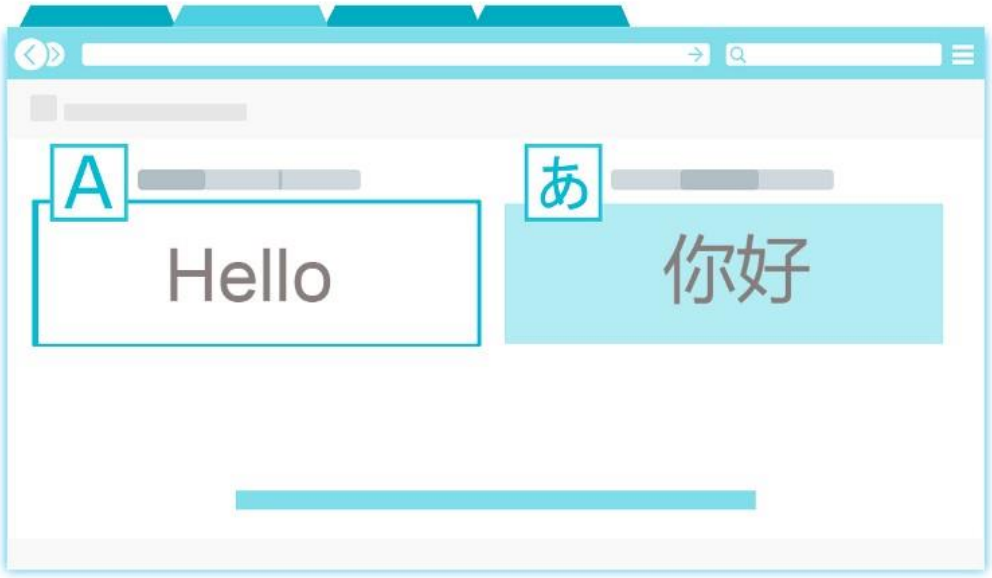
In the ancient methods of translation, people use a language translator in person, whose job is to listen to a speaker and convert the speech into a language understandable to everyone. With the advancement in the technologies, Google can perform the same task easily. You can track google translator here(https://translate.google.co.in/) . Since, the curious software developers have a high level of curiosity, they found some technologies and methodologies to perform the same task, and they excelled.

# PROPOSED METHOD

In the proposed method, we will make use of some python built-in modules, and we can perform the task. I used the two modules, tkinter and googletrans.

The objective of this Python project is to translate a piece of text into another language. You need to install, translate and import two modules: tkinter, googletrans. Basic knowledge of tkinter is required along with the knowledge of functions in python.

# METHODOLOGY




In order to achieve the output, I imported https://pypi.org/project/googletrans and https://docs.python.org/3/library/tkinter.html.

After that, I used the googletrans and created two combobox, one for input and other for output, and stored the text in some variable text1, and provided the output in some text2 variable in the combobox2.

I created the logic to translate via a function named as translate_now() and called it inside the Translate button, which the user will click.

| Output Code | Language Name | Script Name |
|---|---|---|
| af | Afrikaans | Latin |
| bn | Bangla | Bangla |
| bs | Bosnian | Latin |
| el-Latn | Greek | Latin |
| en | English | Latin |
| eo | Esperanto | Latin |
| pl | Polish | Latin |
| ps | Pashto | Arabic |
| pt | Portuguese | Latin |

Above are few languages offered by googletrans module and we can select any from the from a to z.

# IMPLEMENTATION

## 1. Import Modules:

```
from tkinter import *
from tkinter import ttk
from googletrans import Translator
```

Above are the modules imported.

## 2. Create a display window:

- Tk() initialized tkinter which means window is created
- Geometry() set the width and height of the window
- Resizable(False, False) means that we can shrink and grow the window anymore
- Bg: is used to set the background color
- **Label()** widget use to display one or more than one line of text that users aren’t able to modify.

3. Create an Input Frame:

```
f=Frame(root,bg="black",bd=5)
f.place(x=10,y=118,width=440,height=210)

text1=Text(f, font="Robote 20", bg="white",relief=GROOVE,wrap=WORD)
text1.place(x=0,y=0,width=430,height=200)

scrollbar1=Scrollbar(f)
scrollbar1.pack(side="right", fill='y')

scrollbar1.configure(command=text1.yview)
text1.configure(yscrollcommand=scrollbar1.set)
```

4. Create an Input combobox:

```
combo1=ttk.Combobox(root, values=languageV, font='Roboto 14',state='r')
combo1.place(x=110,y=20)
combo1.set("ENGLISH")


label1=Label(root, text="English", font="segoe 30 bold",
bg='white',width=18,bd=5,relief=GROOVE)
label1.place(x=10,y=50)
```

- **language** gets all the values from the 'LANGUAGES' dictionary in the form of a list.
- **ttk Combobox()** widget is a class of ttk modules. It is a drop-down list, which can hold multi-value and show one item at a time. Combobox is useful to select one option from many option.

## 5. Create an output frame:

```
f1=Frame(root,bg="black",bd=5)
f1.place(x=620,y=118,width=440,height=210)


text2=Text(f1, font="Robote 20", bg="white",relief=GROOVE,wrap=WORD)
text2.place(x=0,y=0,width=430,height=200)

scrollbar2=Scrollbar(f1)
scrollbar2.pack(side="right", fill='y')

scrollbar2.configure(command=text2.yview)
text2.configure(yscrollcommand=scrollbar2.set)
```

## 6. Create an output Combobox:

```
combo2=ttk.Combobox(root, values=languageV, font='Roboto 14',state='r')
combo2.place(x=730,y=20)
combo2.set("Select Language")


label2=Label(root, text="English", font="segoe 30 bold",
bg='white',width=18,bd=5,relief=GROOVE)
label2.place(x=620,y=50)
```

## 7. Logic to translate:

```
def translate_now():
    text_ = text1.get(1.0, END)
    t1=Translator()
    t1.raise_Exception = True
    trans_text=t1.translate(text_,src=combo1.get(),dest=combo2.get())
    trans_text=trans_text.text

    text2.delete(1.0, END)
    text2.insert(END, trans_text)
```

This function, translate_now() takes input from first combobox and translate after getting the instructions from user on the button click, and performs the translation.**text** gets the input text entered by the user.”1.0″ means that the input should be read from zero characters to line one.

- src gets the language selected as input text language
- dest gets the language select to translate
- The **END** part **means** to read the text until the end is reached
- translator = Translator() used to create a Translator class object
- **Output_text.delete(1.0, END)** delete all the text from line one to end
- **Output_text.insert (END, translated.text)** will insert the translated text in Output_text.

## 8. Creating a translate button:

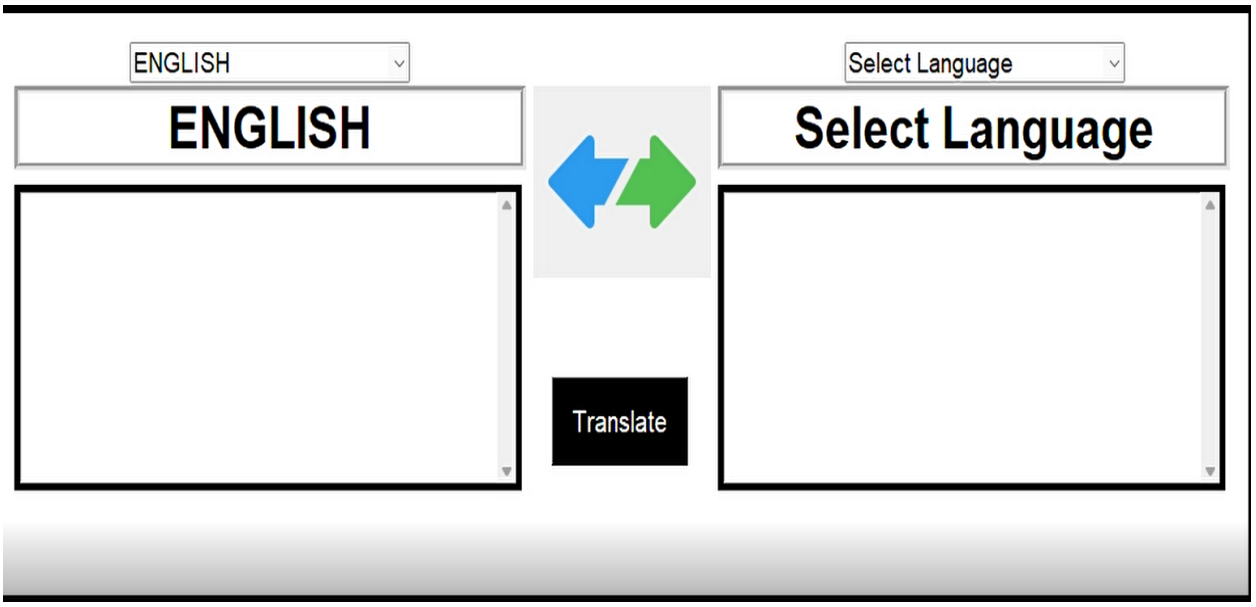
```
translate=Button(root,
text="Translate",font=("Roboto",15),activebackground='white',cursor="hand2",
    bd=1,width=10,height=2,bg='black',fg='white',command=translate_now)

translate.place(x=476,y=250)
```

The user will select the language in which he/she wants to get the text translated, and then he/she will click the translate button.

**Button()** widget used to display button on our window.

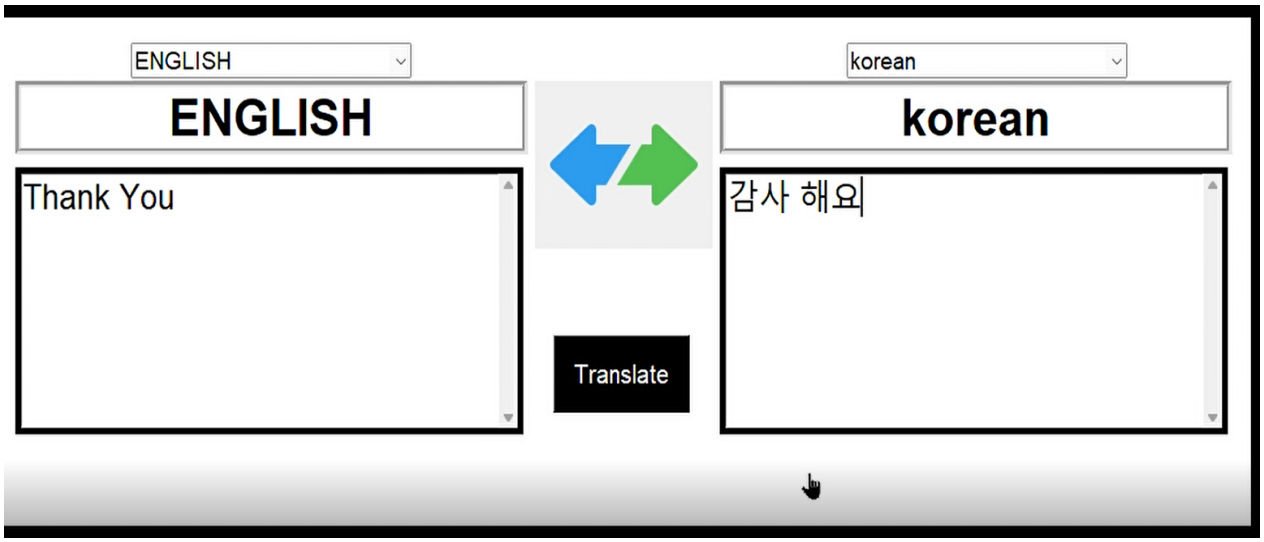
# OUTPUT

Initially, when the code loads, the window will look like this:
The default language is set to English and that you can change it depending on your convenience.



# OUTPUT

When we select the language Korean, and click the translate button, we achieve the provided output, thus we used the googletrans module very well and the goal is accomplished.



# CONCLUSION

At the end, I have successfully developed the Language Translator python project. I used the popular tkinter library for rendering graphics on a display window, googletrans library to translate text from one language to another.

I learned how to translate text, how to create Combobox, buttons widget, and pass the function to the button. In this way, we build a Language Translator.

Thus, the goal of creating the Language Translator App is achieved under the Software Development Internship, offered by Exposys Data Labs.