Jodhpur Institute of Engineering and Technology

PROJECT NAME: TRANSLATOR AND DICTIONARY

GUI (TKINTER) PROJECT REPORT
B.TECH I YEAR (COMPUTER SCIENCE & ENGINEERING)

SUBMITTED TO:	SUBMITTED BY:
DR. UMESH AGARWAL	DIVYANSHI ARORA

ACKNOWLEDGEMENT

We would like to acknowledge the contributions of the following people without whose help and guidance this project would not have been completed. I respectfully thank Dr. Umesh Agarwal, for providing us an opportunity to do this project work and giving all support and guidance, which made us complete the project up to very extent.

I am also thankful to Ms. Mamta Garg, (HOD Mentor) and Ms. Anju Jangid (HOD Academics) of Computer Science and Engineering Department, Jodhpur Institute of Engineering and Technology, for her constant encouragement, valuable suggestions and moral support and blessings.

This acknowledgement will remain incomplete if we fail to express our deep sense of obligation to our parents for their consistent blessings and encouragement.

INTRODUCTION

This tkinter project is a software program that offers a translator and an English dictionary to its users. The translator is capable of translating text from various languages to English, and vice versa. The English dictionary, on the other hand, provides the user with the meaning, synonyms, and antonyms of English words. Additionally, it also has a button that allows the user to hear the output out loud in case they have difficulty reading the text. This project aims to provide users with a simple and efficient tool for language translation and learning. It can be particularly useful for individuals who are learning a new language or those who need to communicate with people from different parts of the world. With its user-friendly interface and diverse features, the tkinter project is a valuable resource for anyone seeking to improve their language skills.

TECHNOLOGY USED IN PROJECT

1. Python programming language.
2. IDLE (Python 3.6 32-bit)
3. Internet

FUNCTION / MODULES USED

- ❖ **Tkinter**- It is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

- **Label**: The **Label** function is used to create a text label that displays text or images. In the project, labels are used to display text on the GUI, such as the title and headings.
- **Scrollbar**: The **Scrollbar** function is used to create a scrollbar widget that allows scrolling of text or other widgets that exceed the available space. In the project, scrollbars are used with the text widgets to enable scrolling when the content exceeds the visible area.
- **Button**: The **Button** function is used to create a button widget that triggers a specific action when clicked. In the project, buttons are used to perform actions such as translation or dictionary lookup when clicked.
- **PhotoImage**: The **PhotoImage** function is used to create an image object from an image file. It supports formats such as GIF, PGM, and PPM. In the project, **PhotoImage** is used to load images that are displayed on the GUI, such as icons or arrow images.
- **iconphoto**: The **iconphoto** function is used to set the icon of the tkinter window. It takes a **PhotoImage** object as an argument. In the project, **iconphoto** is used to set the window icon using the loaded image.
- **Frame**: The **Frame** function is used to create a container widget that holds and organizes other widgets. It provides a way to group related widgets together. In the project, frames are used to create separate areas or sections within the GUI layout.
- **Combobox**: The **Combobox** function is used to create a drop-down combo box widget that allows the user to select an option from a list. It provides a convenient way to present a selection of choices. In the project, combo boxes are used to select the source and destination languages for translation.

❖ **googletrans**: This open-source Python library allows for easy translation of text from one language to another. It uses the Google Translate API to provide quick and accurate translations. In our project, the googletrans module is used to translate text entered by the user from one language to another.

- **translate**: This Python package allows for translation of text using various machine translation services. It supports a wide range of languages and offers advanced features like language detection. In our project, the translate module is used as a backup translator in case the Google Translate API is unavailable.
- **PyMultiDictionary**: This Python package provides easy access to multiple online dictionaries. It supports various dictionaries like Cambridge, Oxford, and Merriam-Webster. In our project, the PyMultiDictionary module is used to provide the meaning, synonyms, and antonyms of English words.
- **pyttsx3**: This Python library is used for text-to-speech conversion. It supports multiple languages and voices and can be used to provide audio feedback to users. In our project, it is used to read out the translated text for the user.

Project Code

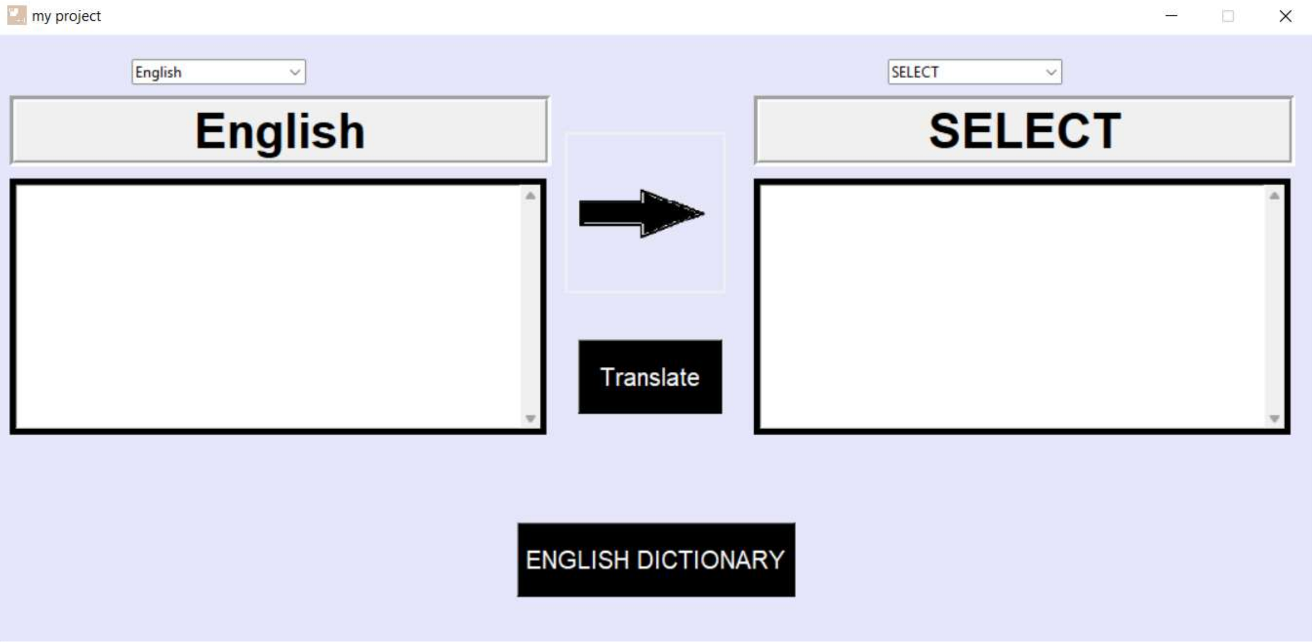
```
import tkinter as tk
from tkinter import ttk,messagebox,PhotoImage
import googletrans as gtr
from translate import Translator
from PyMultiDictionary import MultiDictionary
import pyttsx3

win=tk.Tk()
win.title("my project")
win.geometry("1080x500")
win.resizable(0,0)
win.configure(background="Lavender")



def open_dictionary_window():
    def dict1():
        dictionary=MultiDictionary()
        meaning.config(text=dictionary.meaning('en', word.get())[1])
        synonym.config(text=dictionary.synonym('en', word.get()))
        antonym.config(text=dictionary.antonym('en', word.get()))

    def speaknow():
        engine = pyttsx3.init()
        engine.say(meaning.cget("text")+"  "+synonym.cget("text")+"  "+antonym.cget("text"))
        engine.runAndWait()

    root=tk.Toplevel(win)
    root.title("Dictionary")
    root.geometry("1250x750")
    root.configure(bg="lavender")

    tk.Label(root, text="Search Here", font=("Arial 36 bold"), fg="darkslateblue").pack(pady=10)

    frame =tk.Frame(root)
    tk.Label(frame, text="Type Word:", font=("Arial 28 bold")).pack(side=tk.LEFT)
    word = tk.Entry(frame, font=("Arial 23 bold"))
    word.pack()
    frame.pack(pady=10)


    frame1 =tk.Frame(root)
    tk.Label(frame1, text="Meaning: ", font=("Arial 18 bold")).pack(side=tk.LEFT)
    meaning =tk.Label(frame1, text="", font=("Arial 18"), wraplength=1000)
    meaning.pack()
    frame1.pack(pady=15)


    frame2 =tk.Frame(root)
    tk.Label(frame2, text="Synonym: ", font=("Arial 18 bold")).pack(side=tk.LEFT)
    synonym =tk.Label(frame2, text="", font=("Arial 18"), wraplength=1000)
    synonym.pack()
    frame2.pack(pady=15)


    frame3 =tk.Frame(root)
    tk.Label(frame3, text="Antonym: ", font=("Arial 18 bold")).pack(side=tk.LEFT)
    antonym=tk.Label(frame3, text="", font=("Arial 18"), wraplength=1000)
    antonym.pack(side=tk.LEFT)
    frame3.pack(pady=20)


    tk.Button(root, text="Submit", font=("Arial 18 bold"), command=dict1).pack()
    tk.Button(root, text="Speak", font=("Arial 18 bold"), command=speaknow).pack()
    root.mainloop()


def label_change():
    c1=combo1.get()
    c2=combo2.get()
    lbl1.configure(text=c1)
    lbl2.configure(text=c2)
    win.after(10,label_change)



def translate_fn():
    txt=text1.get(1.0, tk.END)
    src_lang=combo1.get()
    dest_lang=combo2.get()

    translator=Translator(from_lang=src_lang, to_lang=dest_lang)
    trans_text=translator.translate(txt)

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

#icon

image_icon=tk.PhotoImage(file="icon.png")
win.iconphoto(False,image_icon)
#arrow
arrow_image=tk.PhotoImage(file="arrow.png")
image_lbl=tk.Label(win,image=arrow_image,width=128)
image_lbl.place(x=465,y=80)

language=gtr.LANGUAGES
languageV=list(language.values())
lang1=language.keys()



#combobox1
combo1=ttk.Combobox(win,values=languageV,state="r")
combo1.place(x=110,y=20)
combo1.set("English")

lbl1=tk.Label(win,text="ENGLISH",width=18,bd=5,font="seoge 30 bold",relief=tk.GROOVE)
lbl1.place(x=10,y=50)
```

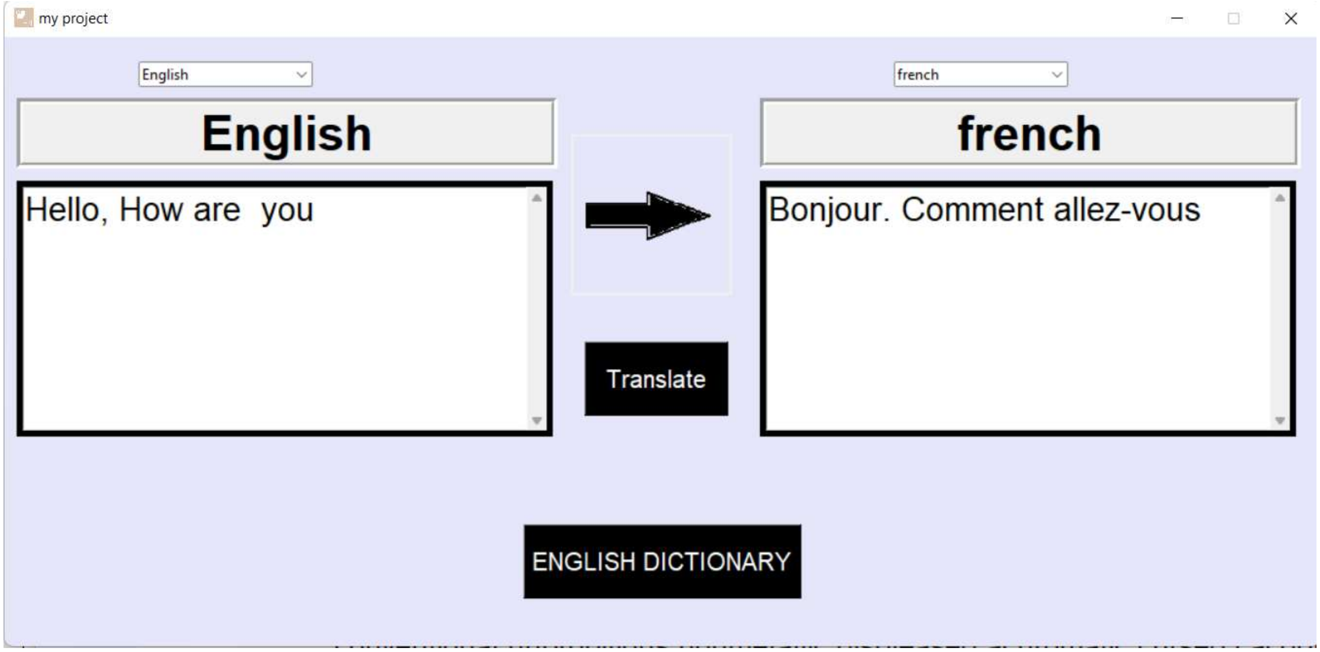
```
#combobox2
combo2=ttk.Combobox(win,values=languageV,state="r")
combo2.place(x=730,y=20)
combo2.set("SELECT")

lbl2=tk.Label(win,text="SELECT",width=18,bd=5,font="seoge 30 bold", relief=tk.GROOVE)
lbl2.place(x=620,y=50)

#frame1
f1=tk.Frame(win,bg="Black",bd=5)
f1.place(x=10,y=118,width=440,height=210)

text1=tk.Text(f1,font="Robote 20",bg="white",relief=tk.GROOVE,wrap=tk.WORD)
text1.place(x=0,y=0,width=430,height=200)

scrollbar1=tk.Scrollbar(f1)
scrollbar1.pack(side="right",fill='y')

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

#frame2
f2=tk.Frame(win,bg="Black",bd=5)
f2.place(x=620,y=118,width=440,height=210)
```

```
text2=tk.Text(f2,font="Robote
20",bg="white",relief=tk.GROOVE,wrap=tk.WORD)
text2.place(x=0,y=0,width=430,height=200)


scrollbar2=tk.Scrollbar(f2)
scrollbar2.pack(side="right",fill='y')


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


#translate button
tr_button=tk.Button(win,text="Translate",font=("Roboto",15),activebac
kground="white",cursor="hand2",


bd=1,width=10,height=2,bg="black",fg="white",command=translate_fn
)


tr_button.place(x=476,y=250)





button3 =tk.Button(win, text="ENGLISH DICTIONARY",
font=("Roboto", 15), activebackground="white", cursor="hand2",
                bd=1, width=20, height=2, bg="black", fg="white",
command=open_dictionary_window)
button3.place(x=426, y=400)
```

```
label_change()

win.mainloop()
```

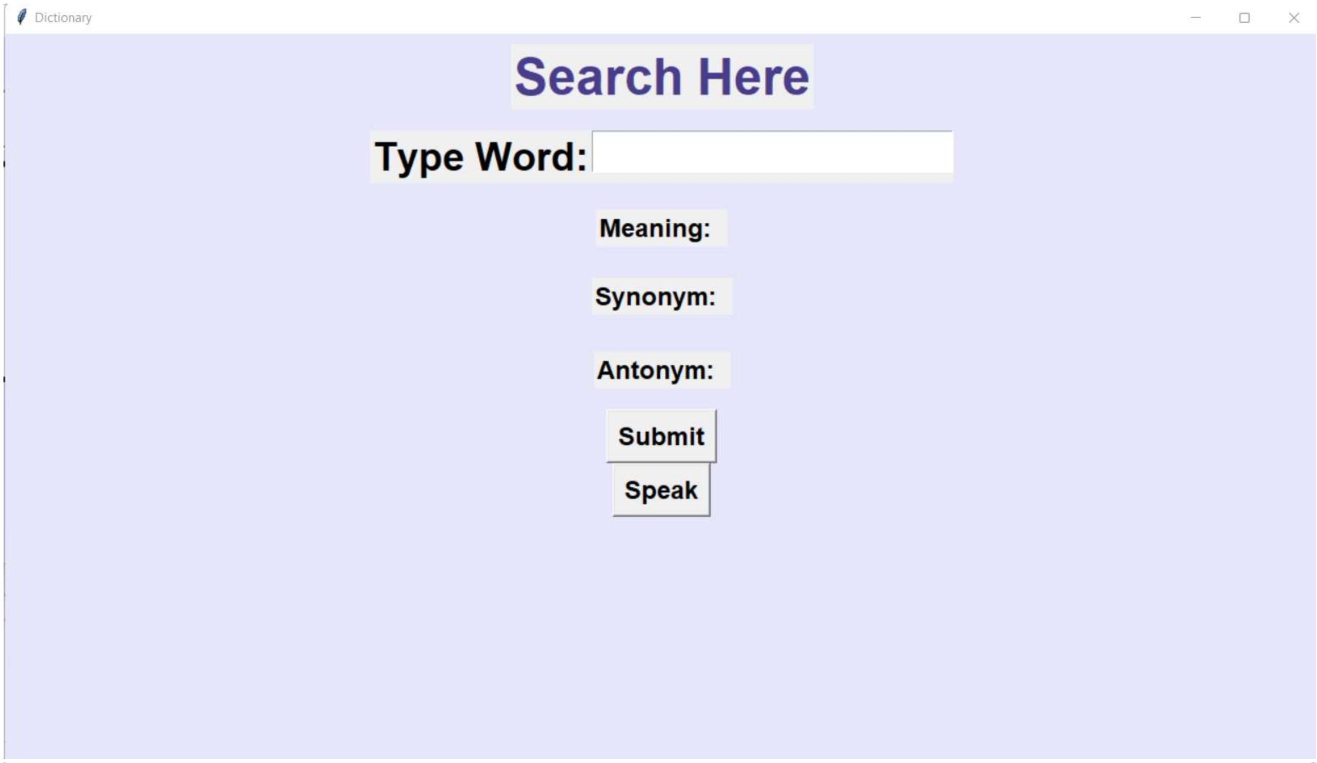
Project Output

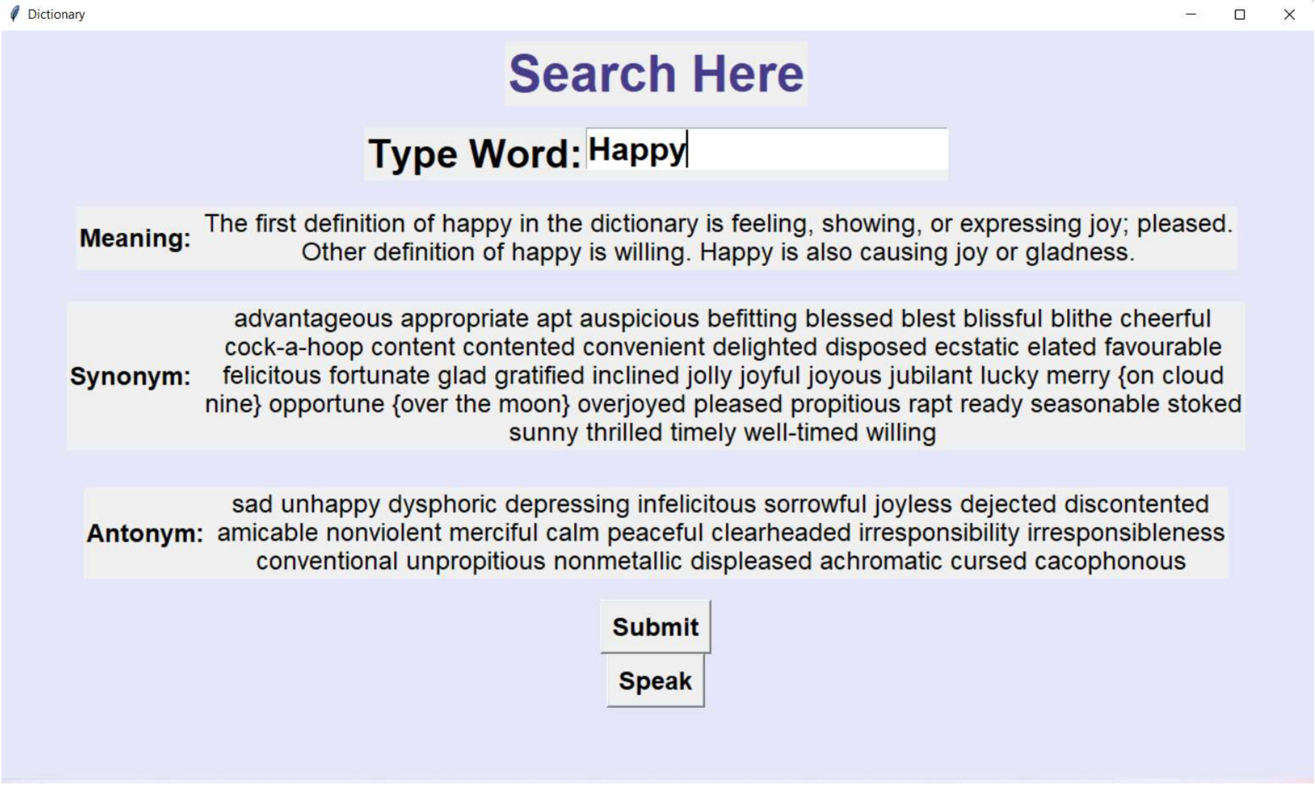




Hello, How are you

Bonjour. Comment allez-vous





Type Word: Happy

Meaning: The first definition of happy in the dictionary is feeling, showing, or expressing joy; pleased. Other definition of happy is willing. Happy is also causing joy or gladness.

Synonym: advantageous appropriate apt auspicious befitting blessed blest blissful blithe cheerful cock-a-hoop content contented convenient delighted disposed ecstatic elated favourable felicitous fortunate glad gratified inclined jolly joyful joyous jubilant lucky merry {on cloud nine} opportune {over the moon} overjoyed pleased propitious rapt ready seasonable stoked sunny thrilled timely well-timed willing

Antonym: sad unhappy dysphoric depressing infelicitous sorrowful joyless dejected discontented amicable nonviolent merciful calm peaceful clearheaded irresponsibility irresponsibleness conventional unpropitious nonmetallic displeased achromatic cursed cacophonous

APPLICATION

The tkinter project is an excellent resource for individuals who are learning a new language or need to communicate with people from different parts of the world. Its user-friendly interface and diverse features make it a valuable tool for anyone seeking to improve their language skills

FUTURE ENHANCEMENTS

- ❖ Dictionary can be made multi-lingual
- ❖ Text to Voice buttons can be added in translator
- ❖ Search tab can be given for selecting languages
- ❖ Search History can be saved

SOURCES

1. geeksforgeeks.com
2. stackoverflow.com
3. github.com