

MINI PROJECT II

(2020 – 21)

iDCipher FINAL REPORT



Institute of Engineering & Technology

Submitted By:

Anya Gupta (181500128)
Jaya Shukla (181500294)
Kamlesh Yadav(181500304)
Nishtha Kapoor (181500431)

Supervised By:

Mr. Piyush Vashisth
(Assistant Professor)

Department of Computer Engineering & Applications

Acknowledgment

It is our pleasure to have the opportunity to extend our heartiest thank to everybody that has given a valuable contribution towards the successful completion of our project.

First of all, We would like to express our deep sense of gratitude to **Mr.Piyush Vashisth** sir so giving us the opportunity to work on the project “**iDCipher**” . We would also like to thank **GLA University Authorities** to give us the opportunity to work on such an innovative project further.

TABLE OF CONTENTS

Abstract

1. Introduction

1.1 General Introduction to the Topic

1.2 Hardware and Software Requirement

2. Technology used in the Project

3. Approaches used in the Project

4. Objective

5. Implementation and Some screenshots

6. Testing

7. Modular Description

8. Future Scope

9. References

ABSTRACT

The development of technology connects everyone from all around the worlds. The problem is, people cannot really mingle with one another because they have communication problems. Some of the problems are with other traveler, disabled peoples, Friends in social media, and International business partners. This device invented to solve this entire problem that faced by people in today's life.

Timely, this device can be used anywhere at any time and it can translate words with high speed. Solvable, it reduced misunderstanding between peoples. Importance, it will help people to travel safely and communicate smoothly. Profitable, it is needed almost by everyone because it helps them to communicate with different language speaking people. Content, this device is safe to use, and easy to carry.

This device invented to make people more knowledgeable, reduce miscommunication among people all around the world, connects people, get maximum profit and give job opportunity to people.

INTRODUCTION

1.1 General Introduction to the topic

iDCipher stands for 'Image Decipher'. In other words, it is an application that will be performing two functions. One of them will be to detect the language of the text passed as the input by the user and, its other functionality will be that it will extract the text in any language from an image passed as input or the text, words or phrases entered by the user and then convert it into any desired language of the world. It currently supports translation to 106 languages. For the purpose, we will be using technologies such as Python and Tkinter(GUI for Python), and various APIs such as Google Translate API, Speech Detection API and Google Language Detect API. Here, API stands for Application Programming Interface. It acts as an intermediate between two applications or software. Google API is developed by Google to allow communications with their servers and use their API keys to develop projects. Python offers multiple options for developing a GUI. Out of all the GUI methods, Tkinter is the most commonly used method. Python with Tkinter outputs the fastest and easiest way to create GUI applications.

PEOPLE'S languages are vital to them. Through language, people communicate, share meaning and experience their sense of individual and community identity . In 2012, the United Nations held a forum on "The Study on the role of languages and culture in the promotion and protection of the rights and identity of indigenous peoples". The importance of language is summed up in the following quote: "Language is an essential part of, and intrinsically linked to, indigenous peoples' ways of life, culture and identities. Languages embody many indigenous values and concepts and contain indigenous peoples' histories and development. They are fundamental markers of indigenous peoples' distinctiveness and

cohesiveness as peoples". Nigeria as a multi-lingual country is faced with challenges such as local language revitalisation, language preservation and sharing knowledge and information in pursuit of development goals in rural areas. The cost of human interpretation and translation is high. However, the improvement in computer-aided software engineering tools coupled with availability of cheap technologies such as mobile phones and personal computers has reduced the cost for machine translation. Computer technology can be a powerful tool for providing materials in local languages to foster participation and inclusion of minorities in national development. Technologies that offer speech-to-speech or text-to-text communication from one Language to another are ways that residents in multi-lingual societies can bridge communication gaps. For Nigeria to achieve her national goals, there needs to be effective communication among the diverse people, whilst her different local languages and cultures are preserved. This paper reports the development of an automated English-to-local-language translator, with the aim of providing solutions to language barriers and improving the understanding of how technology can be used to bridge the communication gap among residents in Nigeria.

1.3 Hardware and Software Requirements

The particular software and hardware requirements are:

1. Software Requirements:

- Python
 - Anaconda Navigator
 - Jupyter Notebook
 - Jupyter Lab
 - Google Colab
 - VS Code
- } anyone from these

2. Hardware Requirements:

- Windows10
 - Ubuntu
 - Mac
 - i3 or higher
 - 4GB RAM or higher
- } anyone from these

TECHNOLOGY USED IN THE PROJECT

- **Python:**

Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.

The biggest strength of Python is huge collection of standard library which can be used for the following:

- GUI Applications (like Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia
- Scientific computing
- Machine Learning
- Text processing and many more..

- **TKinter**

Tkinter is a **Python** binding to the Tk GUI toolkit. It is the standard **Python** interface to the Tk GUI toolkit, and is **Python's** de facto standard GUI. **Tkinter** is included with standard Linux, Microsoft Windows and Mac OS X installs of **Python**.

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 15 types of widgets in Tkinter.

This framework provides Python users with a simple way to create GUI elements using the widgets found in the Tk toolkit. Tk widgets can be **used** to construct buttons, menus, data fields, etc. in a Python application.

This is an open-source **framework** and is available on platforms like Unix and Windows. It is one of the simplest and most popular ways to build a GUI-based application in Python.

APPROACHES USED IN THE PROJECT

There are 7 Translation techniques to facilitate your work.

• BORROWING

Borrowing is a translation technique that involves using the same word or expression in original text in the target text. The word or expression borrowed is usually written in italics. This is about reproducing an expression in the original text as is. In this sense, it is a translation technique that does not actually translate...

Example: The gaucho was wearing a black *sombrero* and a worn *bombacha*.

• CALQUE

When a translator uses a calque, he or she is creating or using a neologism in the target language by adopting the structure of the source language.

Example: The German word *handball* is translated into Spanish as *balonmano*. Or the English term *skyscraper* is *gratte-ciel* in French or *rascacielos* in Spanish.

• LITERAL TRANSLATION

Usually this is called a literal translation or metaphrase. This means a word-for-word translation, achieving a text in the target language which is as correct as it is idiomatic. According to Vinay and Darbelnet, a literal translation can only be applied with languages which are extremely close in cultural terms. It is acceptable only if the translated text retains the same syntax, the same meaning and the same style as the original text.

Example: *Quelle heure est-il?* ⇒ *What time is it?*

• TRANSPOSITION

Transposition involves moving from one grammatical category to another without altering the meaning of the text. This translation technique introduces a change in grammatical structure.

Example: *The President thinks that* ⇒ *Selon le Président*

• MODULATION

Modulation is about changing the form of the text by introducing a semantic change or perspective.

Example: *Maybe you're right.* ⇒ *Tu n'as peut-être pas tort.*

• EQUIVALENCE OR REFORMULATION

This is a translation technique which uses a completely different expression to transmit the same reality. Through this technique, names of institutions, interjections, idioms or proverbs can be translated.

Example: *Chat échaudé craint l'eau froide.* ⇒ *Once burned, twice shy.*

• ADAPTATION

Adaptation, also called cultural substitution or cultural equivalent, is a cultural element which replaces the original text with one that is better suited to the culture of the target language. This achieves a more familiar and comprehensive text.

Example: *baseball* \Rightarrow *football*

Since the sixties, several authors (Michel Ballard, H  l  ne Chuquet, Michel Paillard, etc.) have established other methods of translation, such as explicitation (introducing specific details in the text of the target language), collocation (using a sequence of words that usually go together in the target language) and compensation (where an allusion or reference does not appear in one part of the text as in the source version, but later in the target text).

OBJECTIVE

The objective of this project is to translate text from one language to any other language in real-time with a button click. This project will be built using the Tkinter, googletrans libraries.

In this project, the user enters text in any language and get it translated in any other language by selecting the output language.

A good translation will render the original text in a tone, and style, appropriate to the market for which it is intended. Remember though, if it took five people half a day to agree on the phrasing of one sentence in the original text, this effect is multiplied in the translation **process**.

On the internet, nowadays we can see a lot of projects on Speech Recognitions, Speech to text, text to speech, etc. but here in this project we will be building something more advance than that. Let us assume a scenario, we are travelling to Malaysia and we don't know how to speak Malay or we are in any other country and we don't know their native language, then we can use this tool to overcome the problem. We can translate between all those languages which are present in google translator

An interpreter translates source code into object code one instruction at a time. It is similar to a human **translator translating** what a person says into another **language**, sentence by sentence, as they speak. The resulting object code is then executed immediately. The process is called interpretation .

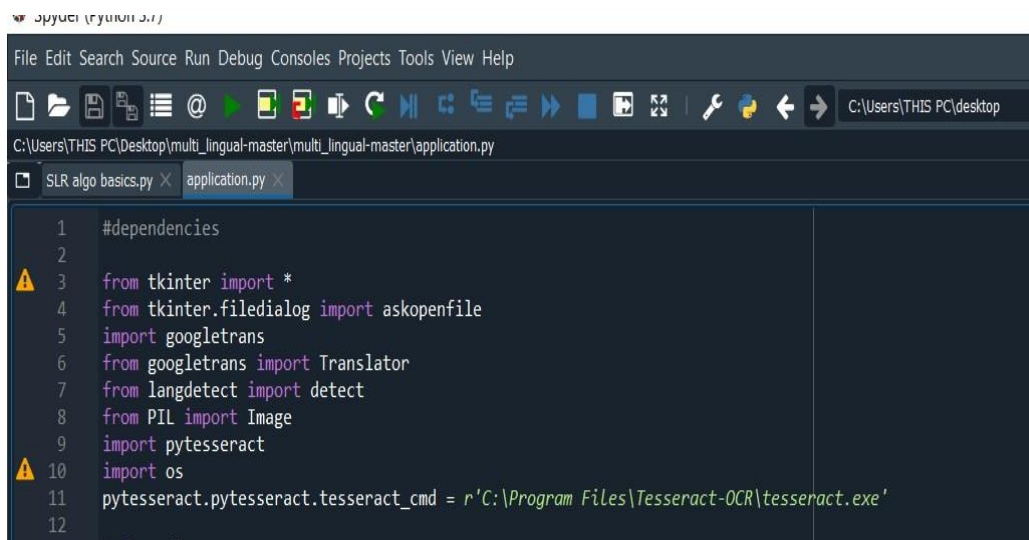
IMPLEMENTATION

To implement this project, we will use the basic concepts of Python, Tkinter, and googletrans libraries.

Tkinter is a standard GUI Python library. ttk module gives access to the Tk themed widget set.

googletrans is a module to translate text. We import the Translator from googletrans, which is used to do translations. We also import LANGUAGES from googletrans which lists all supported languages in a Python dictionary.

- **IMPORT MODULES**

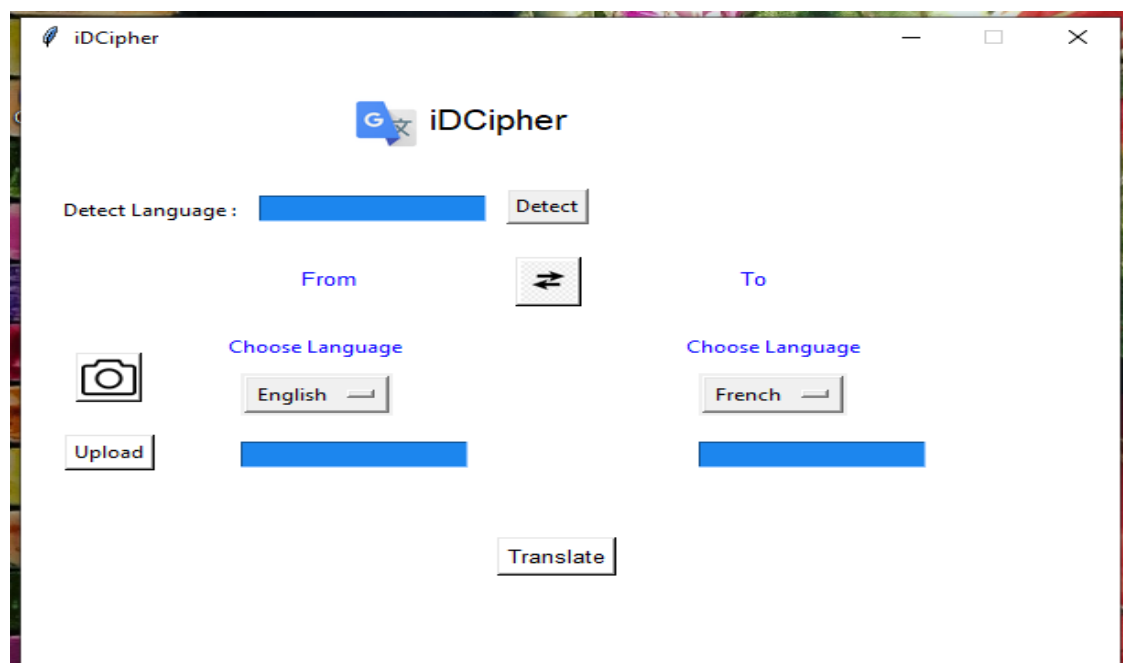


The screenshot shows a Python IDE window with a menu bar (File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help) and a toolbar. The file path is C:\Users\THIS PC\Desktop\multi_lingual-master\multi_lingual-master\application.py. The code editor displays the following Python code:

```
1 #dependencies
2
3 from tkinter import *
4 from tkinter.filedialog import askopenfile
5 import googletrans
6 from googletrans import Translator
7 from langdetect import detect
8 from PIL import Image
9 import pytesseract
10 import os
11 pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
12
```

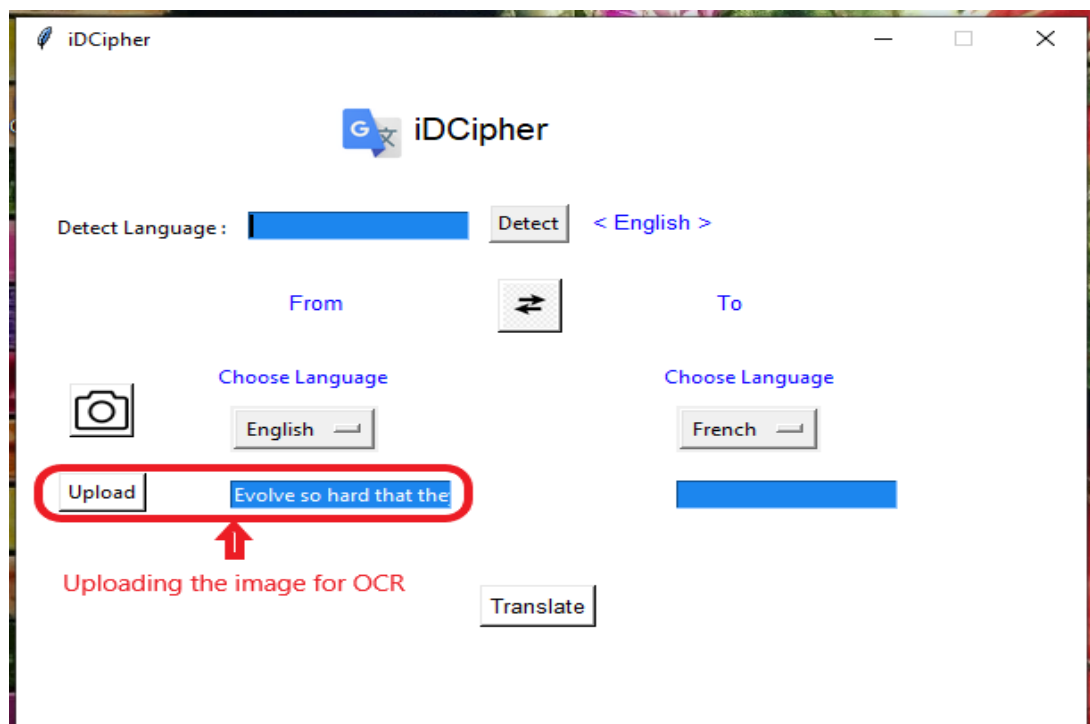
-
- **CREATE A DISPLAY WINDOW**

```
69
70 root = Tk()
71 root.geometry('600x500')
72 root.title('iDCipher')
73 root.configure(background='white')
74 root.resizable(0,0)
75
```



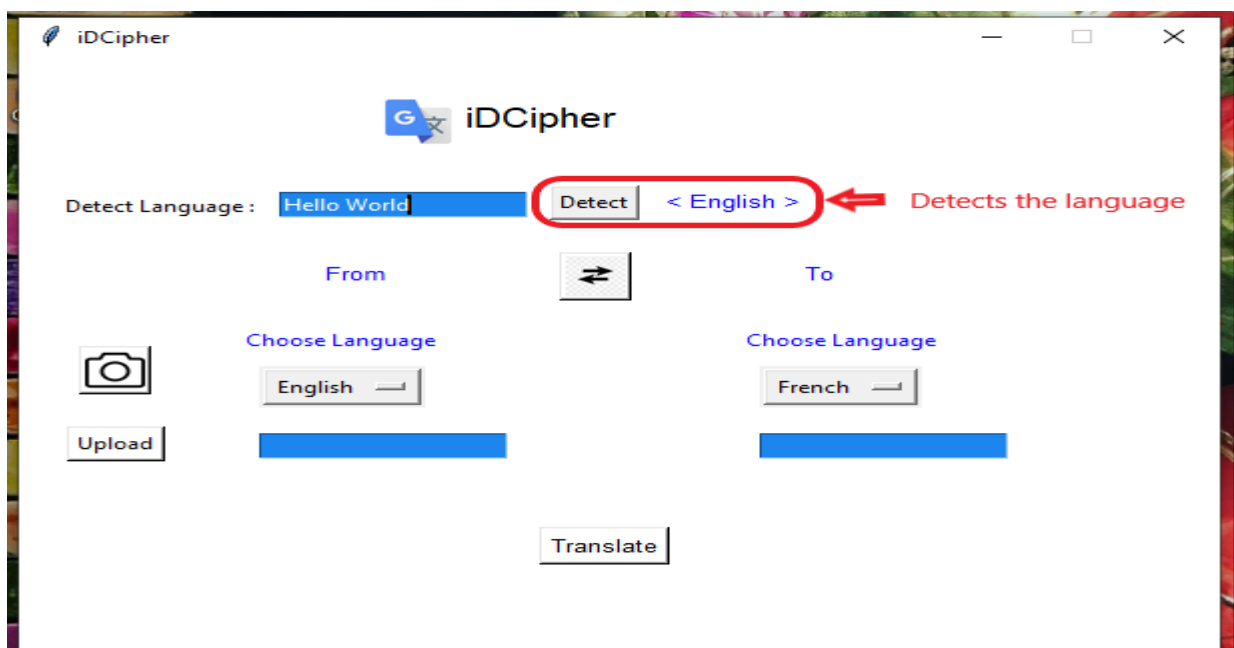
-
- **DEMONSTRATING THE WORKING OF OPTICAL CHARACTER RECOGNITION.**

```
12
13 ▼ def ocr():
14     global img_to_text
15     #print(file_path)
16     ocr_img = Image.open(file_path.name)
17     text = pytesseract.image_to_string(ocr_img, lang='eng')
18     #print(text)
19     img_to_text.set(text)
20
```



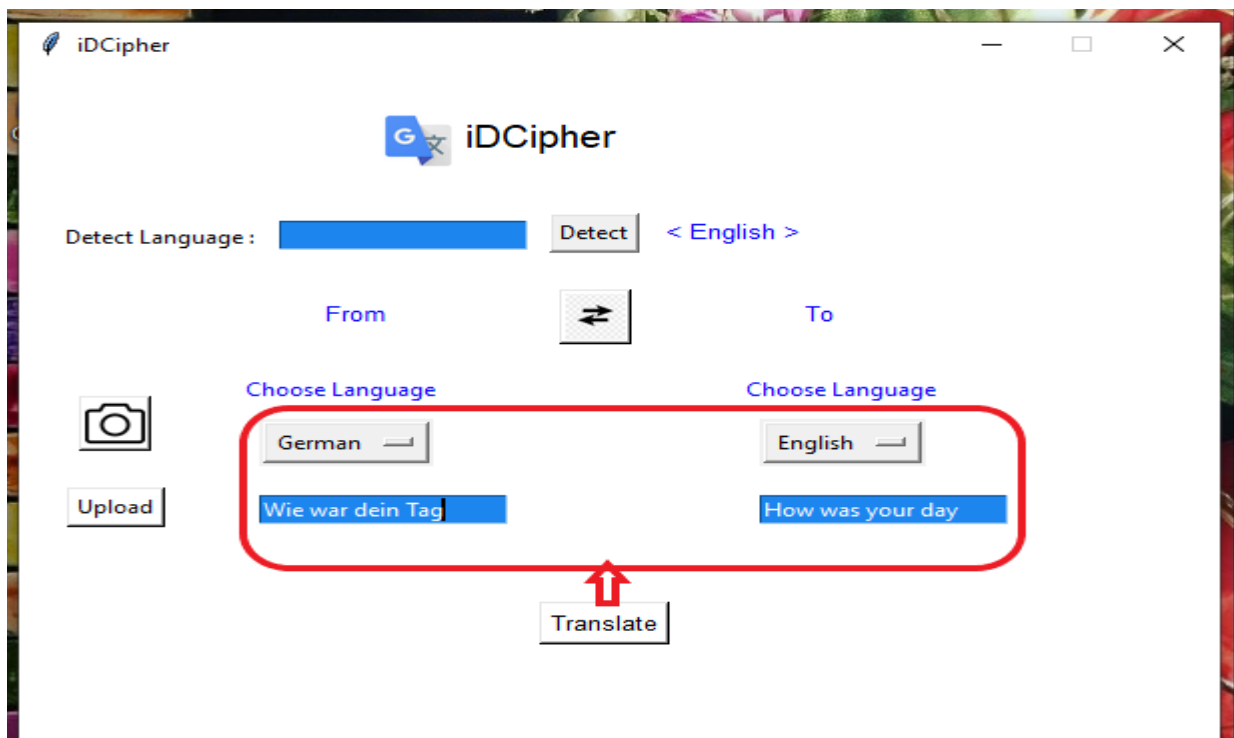
DEMONSTRATING THE WORKING OF LANGUAGE DETECTION.

```
42 ▼ No documentation available
43 #text = detect_entry.get() // the section has been jammed for the time being as it has an unsolved bug in it.
44 text = 'Hello World'
45 print(text)
46 #detect_lang_code = detect(text) #detects language and shows its lang code
47 #print(detect_lang_code)
48 #new_dict = {value: key for key, value in googletrans.LANGCODES.items()} #have swapped keys to values
49 #code_to_lang = new_dict[detect_lang_code] #now storing the name of the language from the lang code
50 code_to_lang = 'English'
51 detected_lang = Label(root,text='< '+code_to_lang+' >',bg='white',fg='blue',font=('bold',10))
52 detected_lang.place(x=320,y=96)
```

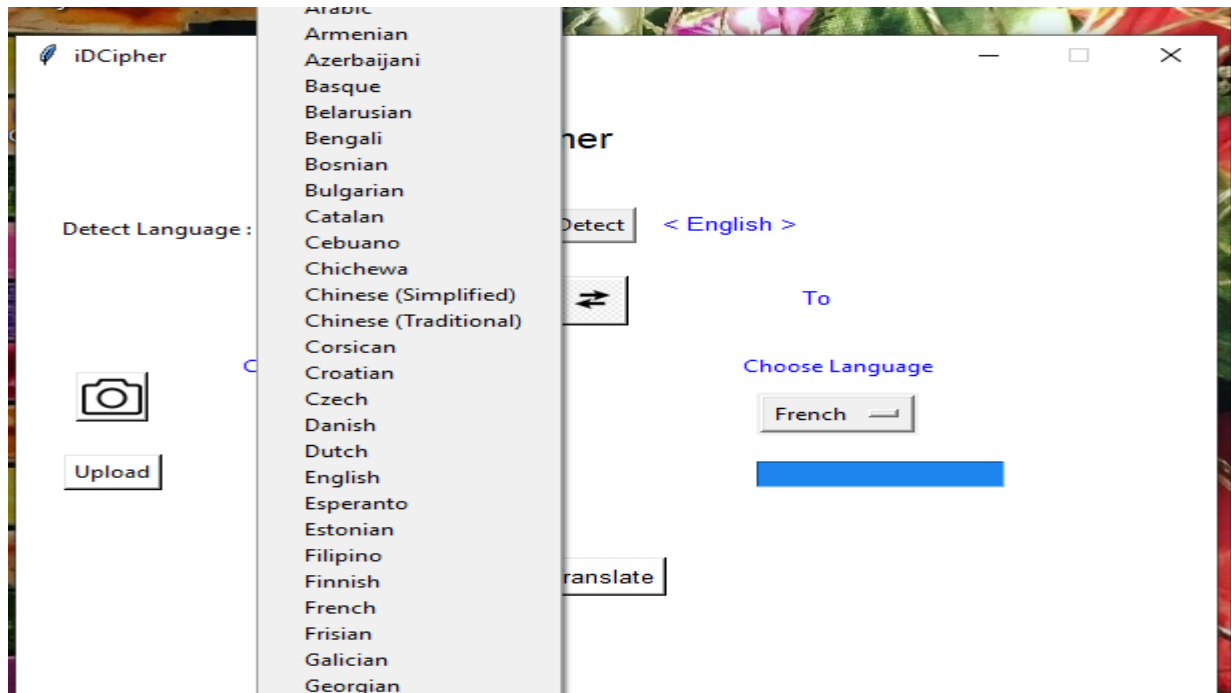


-
- **TRANSLATE THE MESSAGE AND GIVE THE OUTPUT**

```
53
54 ▼ def translate():
55     global translated
56
57     translator = Translator()
58     entry1 = entry_from.get()
59     result = translator.translate(entry1,src=from_var.get(),dest=to_var.get())
60     translated.set(result.text)
61
62
```



-
- **SELECT THE LANGUAGES WHICH YOU WANT TO TRANSLATE**



CREATE A TRANSLATE BUTTON

```

149
150 translate = Button(root,text='Translate',bg='white',fg='black',font=('bold',10),activebackground='orange',command=translate)
151 translate.place(x=260,y=350)
152

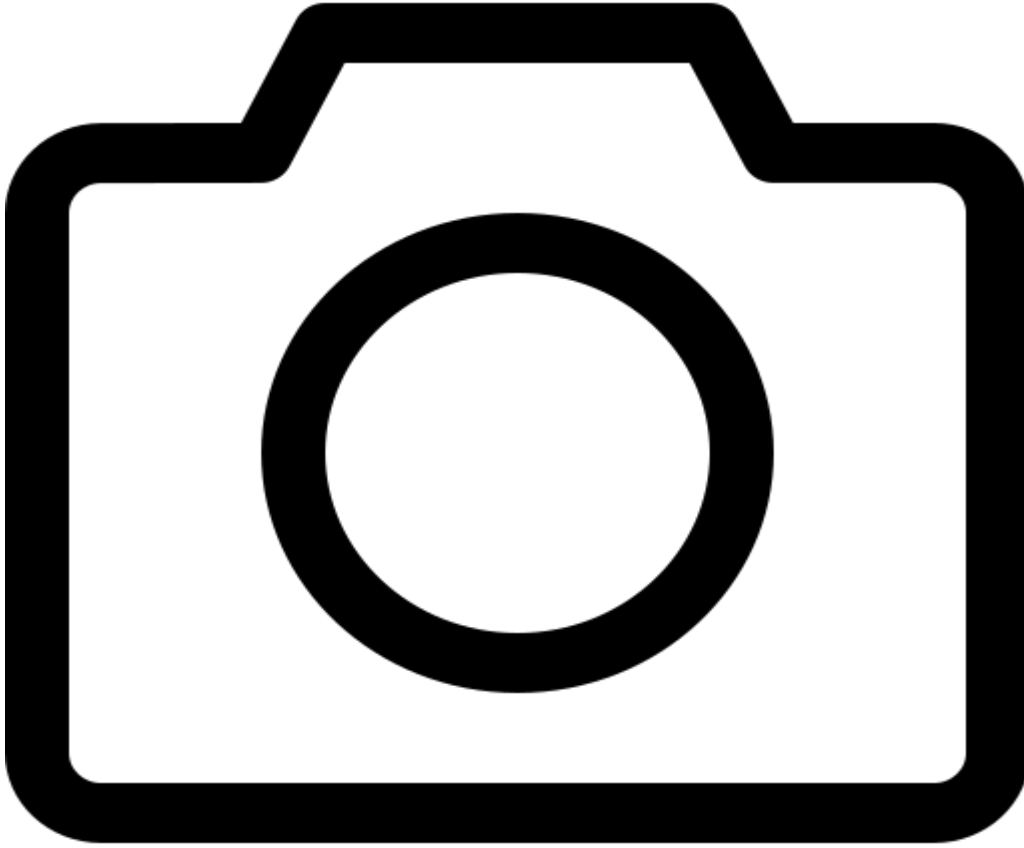
```

-
- **LOGO OF GOOGLE TRANSLATE**



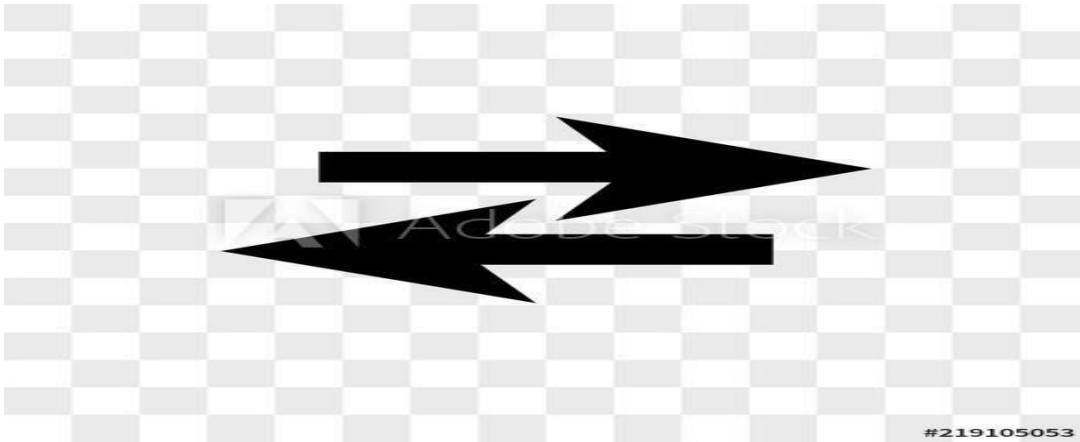
```
80 img = Image.open('logo.png')
81 img = img.resize((45,45),Image.ANTIALIAS)
82 img = img.save('image.ppm','ppm')
83 img = PhotoImage(file='image.ppm')
84
```

- **CAMERA PIC**



```
92 img3 = Image.open('camera.png')
93 img3 = img3.resize((30,30))
94 img3 = img3.save('image3.png','png')
95 img3 = PhotoImage(file='image3.png')
96
97 title = Label(root,text='MULTI LINGUAL',fg='black',bg='white',font=('bold',15))
98 title.place(x=220,y=30)
99
100 logo = Label(root,image=img3,bg='white')
101 logo.place(x=175,y=23)
102
```

- **SWAP PIC**



```
86 img2 = Image.open('swap.png')
87 img2 = img2.resize((30,30))
88 img2 = img2.save('image2.ppm', 'ppm')
89 ⚠ img2 = PhotoImage(file='image2.ppm')
90
```

TESTING

Testing is the process of exercising software with the intent of finding errors and ultimately correcting them. The following testing techniques have been used to make this project free of errors.

Content Review

The whole content of the project has been reviewed thoroughly to uncover typographical errors, grammatical error and ambiguous sentences.

Navigation Errors

Different users were allowed to navigate through the project to uncover the navigation errors. The views of the user regarding the navigation flexibility and user friendliness were taken into account and implemented in the project.

Unit Testing

Focuses on individual software units, groups of related units.

- Unit – smallest testable piece of software.
- A unit can be compiled /assembled / linked/loaded; and put under a test harness.
- Unit testing done to show that the unit does not satisfy the application and /or itsimplemented software does not match the intended designed structure.

Integration Testing

Focuses on combining units to evaluate the interaction among them

- Integration is the process of aggregating components to create larger components.
- Integration testing done to show that even though components were individually satisfactory, the combination is incorrect and inconsistent.

System testing

Focuses on a complete integrated system to evaluate compliance with specified requirements (test characteristics that are only present when entire system is run)

- A system is a big component.
- System testing is aimed at revealing bugs that cannot be attributed to a component as such, to inconsistencies between components or planned interactions between components.
- Concern: issues, behaviors that can only be exposed by testing the entire integrated system (e.g., performance, security, recovery) each form encapsulates (labels, texts, grid etc.). Hence in case of project in V.B. form are the basic units. Each form is tested thoroughly in term of calculation, display etc.

Regression Testing

Each time a new form is added to the project the whole project is tested thoroughly to rectify any side effects. That might have occurred due to the addition of the new form. Thus regression testing has been performed.

White-Box testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements. Techniques used in white-box testing include:

API testing (application programming interface)– testing of the application using public and private APIs

Code coverage– creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)

Fault injection methods– intentionally introducing faults to gauge the efficacy of testing strategies

Black-box testing

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation. The tester is only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing and specification-based testing. Specification-based testing aims to test the functionality of software according to the applicable requirements. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behaviour), either "is" or "is not" the same as the expected value specified in the test case. Test cases

are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional. Specification-based testing may be necessary to assure correct functionality, but it is insufficient to guard against complex or high-risk situations. One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight." Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, or leaves some parts of the program untested. This method of test can be applied to all levels of software testing: unit, integration, system and acceptance. It typically comprises most if not all testing at higher levels, but can also dominate unit testing as well.

Alpha Testing

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

Beta Testing

Beta testing comes after alpha testing and can be considered a form of external user acceptance testing. Versions of the software, known as beta versions, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

MODULAR DESCRIPTION

PART 1 -Completed

Our first step is to import ttk modules from tkinter library and Translator, LANGUAGES modules from googletrans library.

PART 2- Completed

We use tkinter library to create a window where we'll enter the text which we want to convert into target text.

- ✦ **Tk()** initialized tkinter which means window created
- ✦ **geometry()** set the width and height of the window
- ✦ **resizable(0,0)** set the fixed size of the window ✦ **bg = ''** use to set the background color
- ✦ **title()** used to set the title of the window
- ✦ **Label()** widget use to display one or more than one line of text that users aren't able to modify.
- ✦ **root** is the name which we refer to our window
- ✦ **text** which we display on the label
- ✦ **font** in which the text is written
- ✦ **pack** organized widget in block

PART 3- Completed

Our next step is to create an input-output text widget

PART 4- Completed

Define Combobox to select the language.

PART 5-Completed

Define Translate function-The Translate function will translate the message and give the output.

- ✦ **src** gets the language selected as input text language
- ✦ **dest** gets the language select to translate
- ✦ **text** gets the input text entered by the user.”1.0” means that the input should be read from zero characters to line one
- ✦ 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

PART 6-Completed

Create translate button-When we click on the Translate button it will call the translate function

Button() widget used to display button on our window

- + **command** is called when we click the button
- + **activebackground** sets the background color to use when the button is active

root.mainloop() is a method that executes when we want to run our program.

FUTURE SCOPE

In this advanced age, geographical boundaries are fast becoming history. The online medium is closing down the sections of society. Thanks to the emergence of India, Japan and China as superpowers, the English language is also losing its strong hold. In this scenario, the jobs of translators and interpreters are considered as a lucrative career option by large number of people.

The appearance of new language centers significantly depict that there is no shortage of economic opportunities for a prominent translation company in Delhi and those who desire to take up interpretation and translation as their profession. Russian, Arabic, Japanese and Chinese translators and interpreters are nowadays highly in demand.

Today, foreign languages become a real asset for anyone who knows how to master them. With the internationalization of companies, companies are looking for more and more people who speak one or more languages.

The priority remains English. All other foreign languages are only optional. You can be brought to work in Germany, without speaking a word of German, as long as you know how to deal with the language of Shakespeare. The other languages will be able to prove to be an asset, according to the proposed post. If a company sells products in China, and you master Mandarin, your profile will grab their attention. And you will be even more likely to be offered the job if you have a good knowledge of the local culture. Moreover, English is a very spoken language, companies are not really surprised to have candidates who know how to speak fluently English; you will make the difference if you speak another language.

For information, the three most commonly used languages in the world are: Mandarin, Spanish and English. French is only in twelfth place!

So do not hesitate to indicate your language level in your CV. Even if a language that you speak will not be essential for your future work, indicating it will show your intellectual aptitude and openness. Be careful, however, not to exaggerate: never pretend that you have a certain level in a language, if you do not have it! The company will quickly realize if you speak a language fluently, or not .

REFERENCES

<https://www.kaggle.com/>

<https://github.com/>

<https://towardsdatascience.com/>