URLOGY

PROJECT REPORT

PRAAJNA
G BARAGUR

Class XII

2021

TABLE OF CONTENTS

Project Details

Introduction

URLogy is an application that consists of various components that are designed to provide users with an all-in-one solution for various URL-related utilities.
The utilities and their functions are listed below.

- **Domain Availability Checker :**
  This Tool accepts a domain name from the user and lets the user know whether the domain name is available for use or not.

- **URL Shortener :**
  The URL Shortener tool accepts a URL as an input and returns a shorter link which redirects to the same link fed in by the user. It can be useful while sending messages or emails with links.

- **Web-Browser Utility :**
  This utility gives quick and easy access to multiple commonly used browsers and pages. It also allows the user to open their own custom created URL, or any URL of their choice.

User Manual

Main:

- Run the main.py file in your preferred IDLE or IDE.
- The Main screen displays 3 buttons: 'Domain Availability Tool', 'URL Shortener', 'Web Browser'.

Domain Availability:

- Click the 'Domain Availability Check' button. A new window will pop up. Enter the domain name in the entry box and click the 'check' button.
- If the domain is available, click the 'Yes' option if you want to register the domain. Otherwise, click 'No'.
- **If you clicked yes, the next time you input the same domain, the program will display that the domain is taken.**

URL Shortener:

- Click the 'URL Shortener' button. A new window will pop up. Enter the URL to be shortened and click 'Shorten'.
- The shortened link is then displayed in the same box in which the link was first entered. It is also automatically copied to the clipboard.
- Paste this copied link as required to utilize the same.

Web Browser:

- Click the 'Web Browser' button. A new window will pop up.
- Click the button of the respective website, if any of the websites you wish to open, is in the form of a button.
- If you do not find the website you wish to open, click the 'custom' button. It opens another window.
- Enter the URL you wish to open in the entry box and click the 'Enter' button. The website will be opened in your default web browser.

Working

Front End

- The Graphic User Interface and Menus were designed using Python's tkinter module.
- Stock images sourced from Google Images.
- Message boxes were from tkinter.message module.
- whois and pyshorteners modules were used in the Domain availability utility and URL Shortener utility respectively. (eg: whois.parser.PywhoisError)
- webbrowser module was used for various functions in the Web Browser Utility. (eg:webbrowser.open("www.yahoo.com")

Back End

- Domain Check (module-whois):
  - The domain is entered into the entry box and the check button is clicked.
  - WHOIS is a query and response protocol that is often used for querying databases that store registered domain names. It stores and delivers the content in a human-readable format. whois library simply queries a WHOIS server directly instead of going through an intermediate web service.
  - When the function whois.whois(URL) is called, it checks its databases and raises an exception for domains that don't exist.
  - When an exception is raised, our program shows that the domain is available. If it doesn't show any exception, the program returns the domain name and if it does, our program displays that the domain is already taken.
  - When a domain is available, it allows the user to choose whether to register the domain. If the user clicks 'Yes', the domain is then saved into a file called 'url'.
  - Every time the program runs, it also checks in the 'url' file whether a domain has already been registered through this tool. If it has been registered through the tool, it displays that the domain name is now taken.

- **URL Shortener Tool**
  - The module Pyshorteners is used. This module utilizes most of the famous URL shortening tools online like tinyurl.com, bit.ly, etc. to shorten or expand URLs.
  - We have utilized tinyurl.com to shorten our URLs.
  - The URL to be shortened is input into the entry box and the 'shorten' button is clicked. Then, the shortened URL from tinyurl.com is then displayed in the same entry box. This link now redirects to the link that the user had input.
  - The link is also automatically copied to the clipboard using Tkinter, and the user can now paste this anywhere he/she wishes to do so.

- **Web-Browser Utility**
  - The webbrowser module provides a high-level interface to allow displaying Web-based documents to users.
  - webbrowser.open(url) accepts a given URL and opens the passed URL through the computer's default browser.
  - By default, several commonly used pages/websites are linked via various buttons.
  - There is also a 'CUSTOM' button that accepts a custom URL given by the user.
  - The custom URL is retrieved from the text box and is passed through the open() function.

- **File Handling:**
  - Our program has incorporated the concept of file handling in the domain availability checker.
  - When a particular domain name entered by the user is available, we provide the user with an option to register this domain.
  - If they do register it by clicking 'Yes', we save the domain name into a file named 'url.txt'.
  - Every time the domain availability checking tool is run, it also goes through the file created by our program to check if a domain has already been registered. If yes, it displays the domain as taken.
  - The functions we have created for the same are, check_file(url) and write_to_file(url).

- Modules used:
  - Python-whois: Used for the domain checker tool. The library was pip installed into the device. This module uses an API to send requests and get data from the famous database which holds information about all the registered websites. This can be done manually by visiting www.whois.com. In our program, we have utilized python to automate this entire process with our application.
  - webbrowser - Used for the web browser to open websites. It is a built-in library.
  - Tkinter - Used for the graphic user interface of the entire application. It is built-in with python.
  - pyshorteners - Utilized for the URL shortener tool. The library was pip installed into the device.

CODE

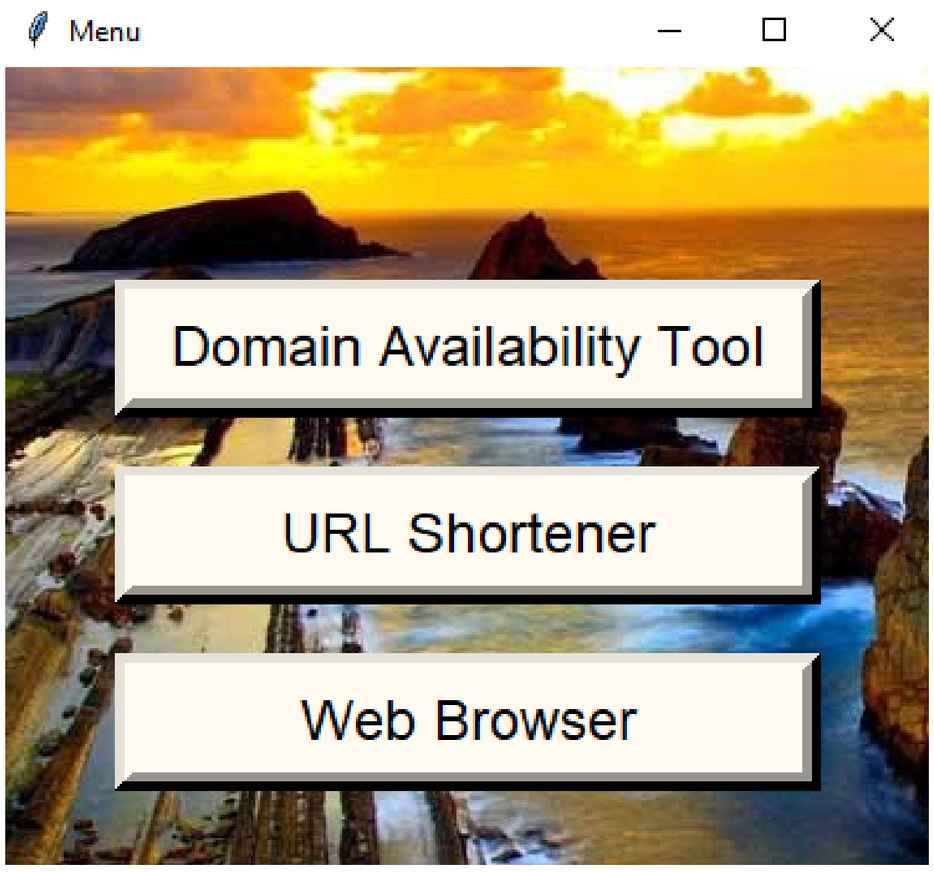
```python
#importing modules
import webbrowser
import tkinter as tk
from tkinter.messagebox import *
import whois,pyshorteners

#defining functions

#to check if domain is registered in the file url.txt
def check_file(url):
    try:
        with open('url.txt','r') as file:
            taken_domains = file.readlines()
            if f'{url}\n' in taken_domains:
                return False
            else:
                return True
    except Exception as e:
        return True

#function for domain availability checker utility
def domaincheck():
    global url
    url = url_entry.get()
    if url == '':
        showerror('Invalid',' Please Enter A URL')
        url_entry.delete(0,tk.END)
        return
    try:
        #whois.whois() function raises an exception for domains that doesn't exist.
        response=whois.whois(url)
        for x in response.domain_name:
            showerror('Taken',f"{url} is taken")
            break
    #whois.parser.PywhoisError is the exception raised when domain doesn't exist already
    except whois.parser.PywhoisError:
        if check_file(url):
            ans=askyesno('Available',f"{url} is available.\nWould you like to register this domain?")
            if ans:
                write_to_file(url)
        else:
            showerror('Taken',f"{url} is taken")
    url_entry.delete(0,tk.END)

#to copy registered domain to file url.txt
def write_to_file(url):
    with open('url.txt','a') as file:
        file.write(url+'\n')

#GUI for domain checker
def domaincheckmain():
    global url_entry
    #creating root window
    root = tk.Tk()
    root.title('Url Tool')
    root.geometry('400x300')

    title_label = tk.Label(root,text='ENTER DOMAIN BELOW',font=('Cabin Sketch',18))
    title_label.place(relx=0.5,rely=0.25,anchor='center')

    url_entry = tk.Entry(root,font=('Cabin Sketch',15))
    url_entry.place(relx=0.5,rely=0.5,anchor='center')
    btnRead=tk.Button(root, height=1, width=10, bd=4,bg='white',text="Check", command=domaincheck)
    btnRead.place(relx=0.5,rely=0.7,anchor='center')
    root.mainloop()

#logic for URL Shortener Utility
def shorten():
    url = web_entryl.get()
    if url == '':
        showerror('Invalid',' Please Enter A URL')
        web_entryl.delete(0,tk.END)
        return
    try:
        url = web_entryl.get()
        maker=pyshorteners.Shortener()
        #maker.tinyurl.short(url) utilises tinyurl.com to shorten URL
        short_url=maker.tinyurl.short(url)
        web_entryl.delete(0,tk.END)
        web_entryl.insert(tk.END,short_url)
        root=tk.Tk()
        root.withdraw()
```

```
        root.clipboard_clear()
        root.clipboard_append(short_url)
        showinfo(title='Information', message='Link Copied!')
        root.destroy()

    #won't stop the program in case of an error. User might have to click the button again.
    except Exception as e:
        print()

#GUI for URL Shortener
def linkshortenermain():
    global web_entryl
    #creating root window
    root = tk.Tk()
    root.title('Shortener Tool')
    root.geometry('400x300')

    title_label = tk.Label(root,text='ENTER URL TO SHORTEN',font=('Cabin Sketch',18))
    title_label.place(relx=0.5,rely=0.25,anchor='center')

    web_entryl = tk.Entry(root,font=('Cabin Sketch',15))
    web_entryl.place(relx=0.5,rely=0.5,anchor='center')
    btnRead=tk.Button(root, height=1, width=10,bd=4,bg='white', text="Shorten", command=shorten)
    btnRead.place(relx=0.5,rely=0.7,anchor='center')
    root.mainloop()

#functions to call in Web Browser Utility
def google():
    webbrowser.open('www.google.com')

def bing():
    webbrowser.open('www.bing.com')

def yahoo():
    webbrowser.open("www.yahoo.com")

def facebook():
    webbrowser.open("www.facebook.com")

def twitter():
    webbrowser.open("www.twitter.com")

def youtube():
    webbrowser.open("www.youtube.com")

def instagram():
    webbrowser.open("www.instagram.com")

def nps():
    webbrowser.open("https://npsrnr.com/")

#function for custom URL
def customopen():
    url = url_entry2.get()
    if url == '':
        showerror('Invalid',' Please Enter A URL')
        web_entryl.delete(0,tk.END)
        return
    try:
        webbrowser.open(url)
    except Exception as e:
        return
#UI for custom URL opener
def custom():
    global url_entry2
    root = tk.Tk()
    root.title('Custom Link')
    root.geometry('400x300')

    title_label = tk.Label(root,text='ENTER LINK BELOW',font=('Cabin Sketch',18))
    title_label.place(relx=0.5,rely=0.25,anchor='center')

    url_entry2 = tk.Entry(root,font=('Cabin Sketch',15))
    url_entry2.place(relx=0.5,rely=0.5,anchor='center')
    btnRead=tk.Button(root, height=1, width=10, bd=4,bg='white',text="Enter", command=customopen)
    btnRead.place(relx=0.5,rely=0.7,anchor='center')
    root.mainloop()
```

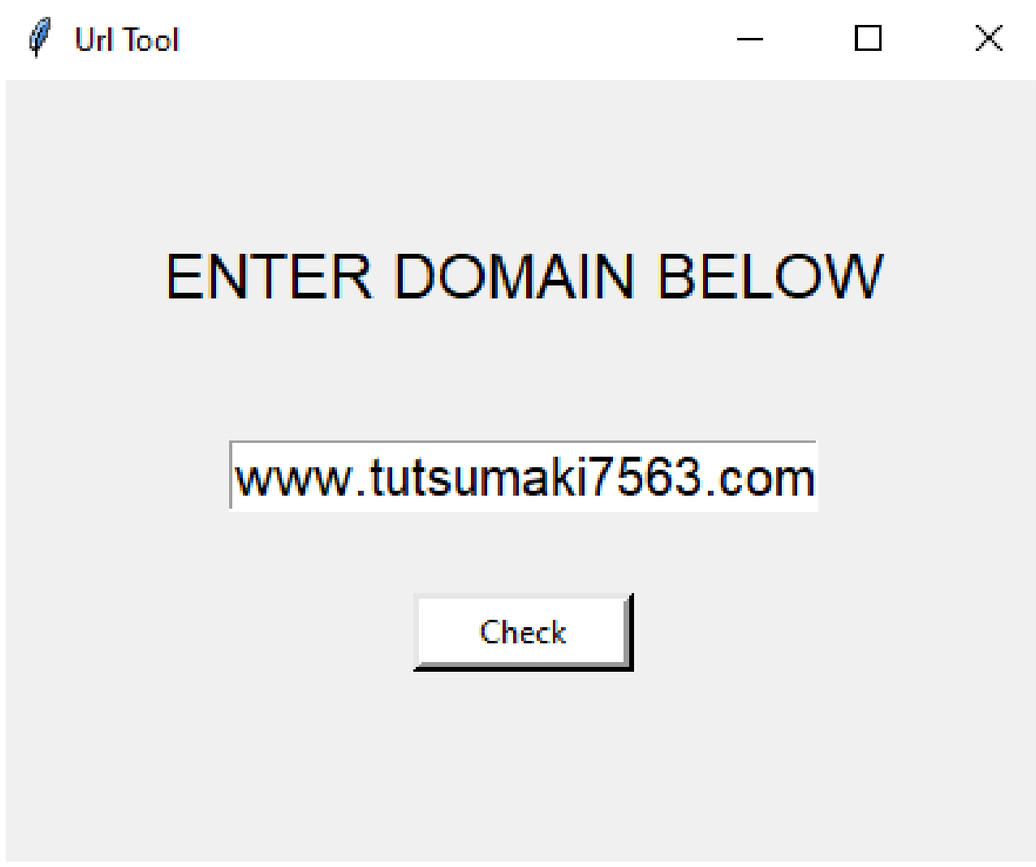
```
#GUI for Browser Utility
def webbrowsermain():
    global google,bing,yahoo,npsrnr,instagram,twitter,youtube,facebook
    #creating root window
    root = tk.Tk()
    root.title("WebBrowsers")
    root.geometry("330x300")

    tk.Label(root, text="WEBROWSERS",font=('Cabin Sketch',15)).place(relx=0.5,rely=0.07,anchor='center')
    tk.Label(root,text="Select website",font=('Cabin Sketch',15)).place(relx=0.5,rely=0.15,anchor='center')

    #buttons
    google =tk.Button(root,height=1, width=9,text="GOOGLE",command=google,bd=4, font=('Cabin Sketch',15),fg='white',bg='black')
    google.place(relx=0.3,rely=0.31,anchor='center')

    customl = tk.Button(root,height=1, width=9,text="CUSTOM", command=custom,bd=4,font=('Cabin Sketch',15),fg='white',bg='black')
    customl.place(relx=0.7,rely=0.85,anchor='center')

    yahoo = tk.Button(root,height=1, width=9,text="YAHOO", command=yahoo,bd=4,font=('Cabin Sketch',15),fg='white',bg='black')
    yahoo.place(relx=0.3,rely=0.49,anchor='center')

    facebook = tk.Button(root,height=1, width=9, text="FACEBOOK", command=facebook,bd=4,font=('Cabin Sketch',15),fg='white',bg='black')
    facebook.place(relx=0.7,rely=0.49,anchor='center')

    twitter = tk.Button(root, height=1, width=9,text="TWITTER", command=twitter,bd=4,font=('Cabin Sketch',15),fg='white',bg='black')
    twitter.place(relx=0.3,rely=0.67,anchor='center')

    youtube = tk.Button(root, height=1, width=9,text="YOUTUBE", command=youtube,bd=4,font=('Cabin Sketch',15),fg='white',bg='black')
    youtube.place(relx=0.7,rely=0.67,anchor='center')

    instagram = tk.Button(root,height=1, width=9, text="INSTAGRAM", command=instagram,bd=4,font=('Cabin Sketch',15),fg='white',bg='black')
    instagram.place(relx=0.3,rely=0.85,anchor='center')

    npsrnr=tk.Button(root,height=1, width=9, text="NPS-R", command=nps,font=('Cabin Sketch',15),bd=4,fg='white',bg='black')
    npsrnr.place(relx=0.7,rely=0.31,anchor='center')


    #running the mainloop()
    root.mainloop()


# main

menu = tk.Tk()
menu.title('Menu')
menu.geometry('400x400')
bg = tk.PhotoImage(file='Background.png')
bg_lab = tk.Label(menu,image=bg)
bg_lab.place(x=-200,y=-200)

#buttons for main menu
Domain_button=tk.Button(menu, height=1, width=20, text="Domain Availability Tool", command=domaincheckmain,bd=8,bg='floral white',fg='black',font=('Cabin Sketch',18))
Domain_button.place(relx=0.5,rely=0.3,anchor='center')

short_button=tk.Button(menu, height=1, width=20, text="URL Shortener", command=linkshortenermain,bd=8,bg='floral white',fg='black',font=('Cabin Sketch',18))
short_button.place(relx=0.5,rely=0.5,anchor='center')

browser_button=tk.Button(menu, height=1, width=20, text="Web Browser", command=webbrowsermain,bd=8,bg='floral white',fg='black',font=('Cabin Sketch',18))
browser_button.place(relx=0.5,rely=0.7,anchor='center')
menu.mainloop()
```

OUTPUT

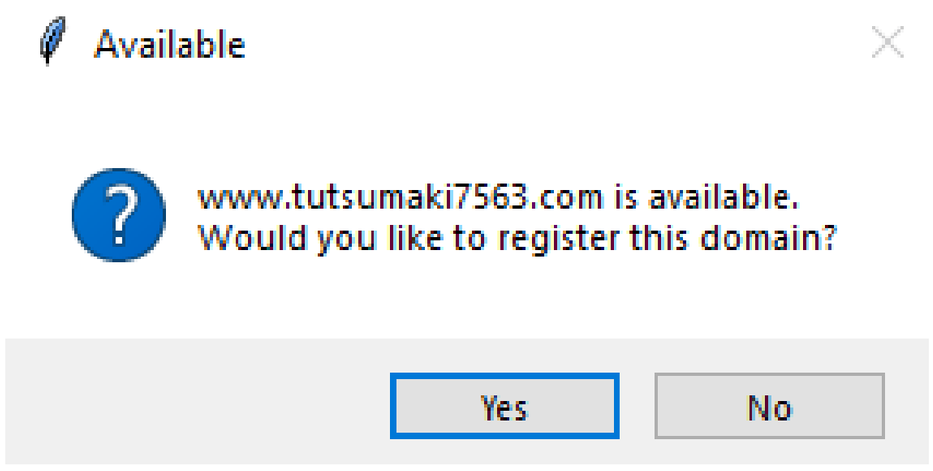
Output



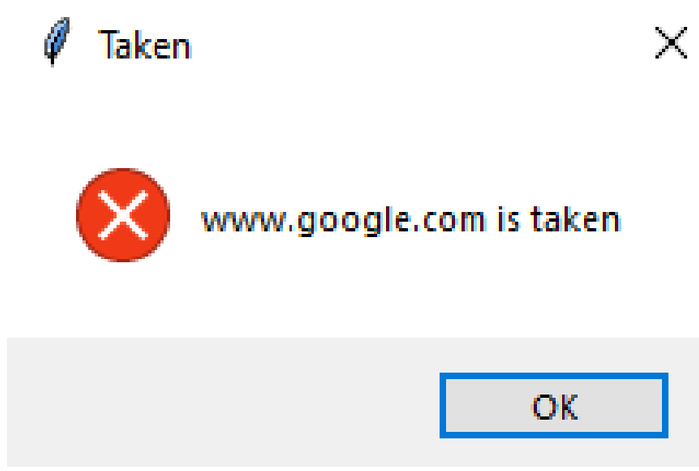
Main menu



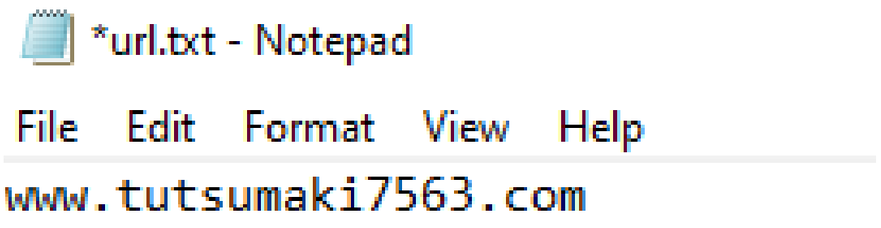
Domain Availability tool



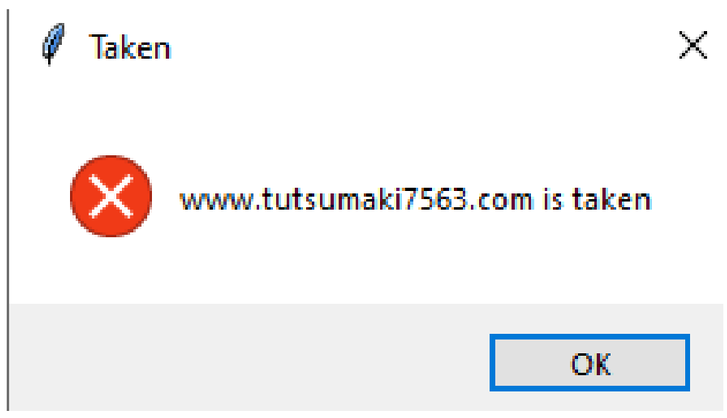
Registering new domain



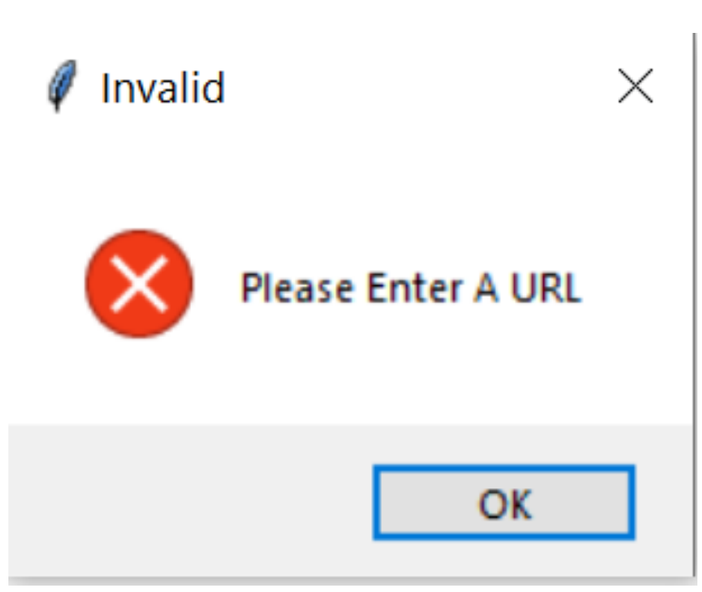
If domain is taken



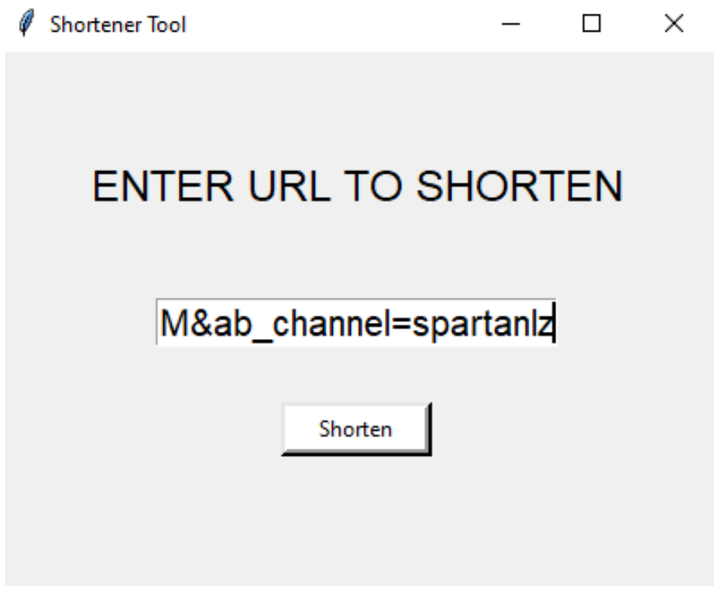
Registered URLs are stored in url.txt



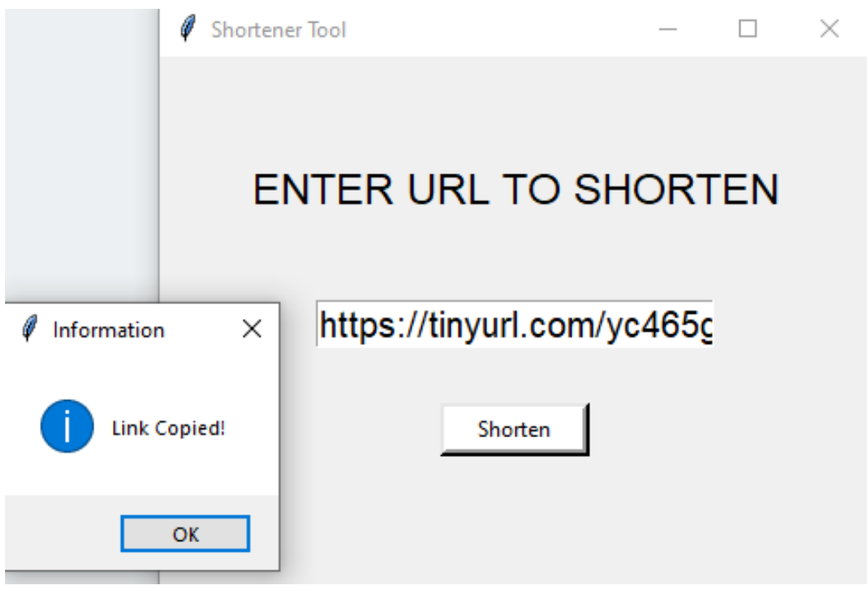
When domain is registered



If no URL is entered before checking for domain or shortening.

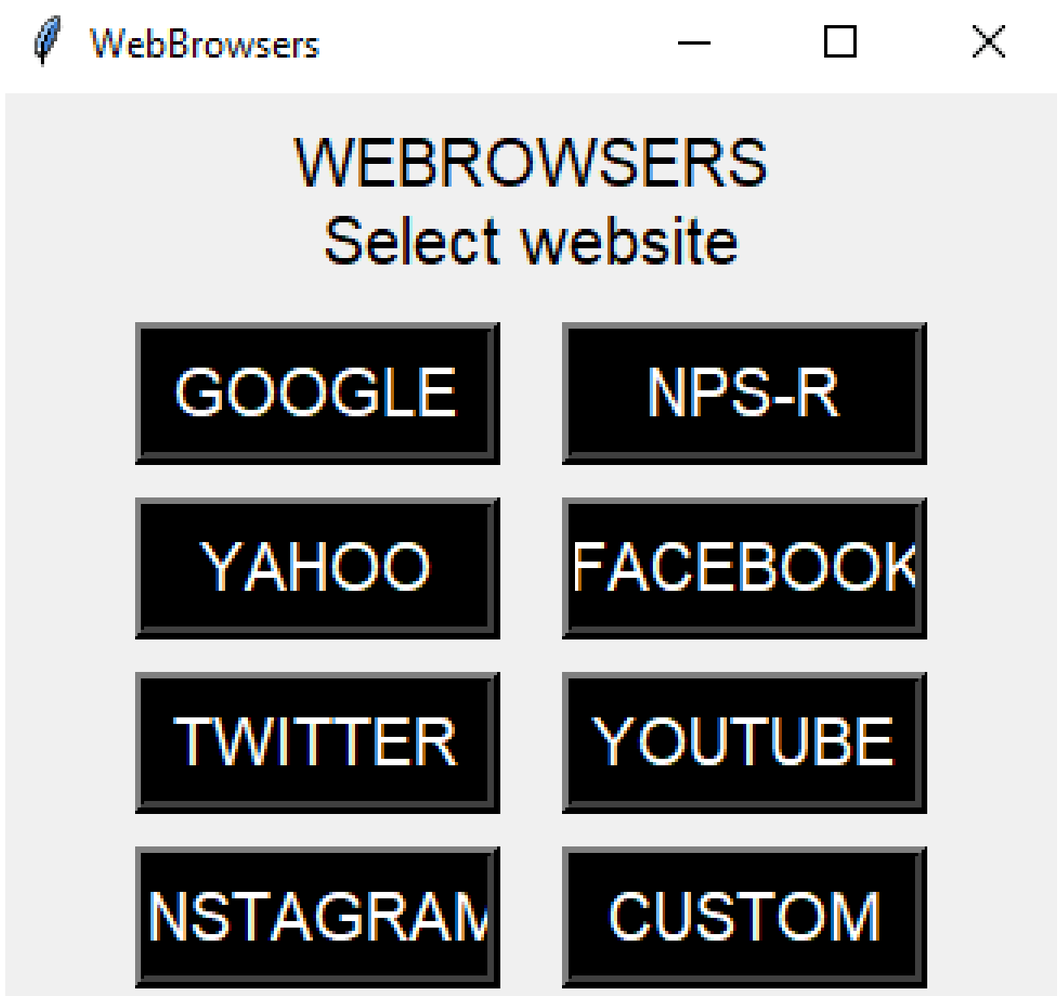


URL Shortener Utility

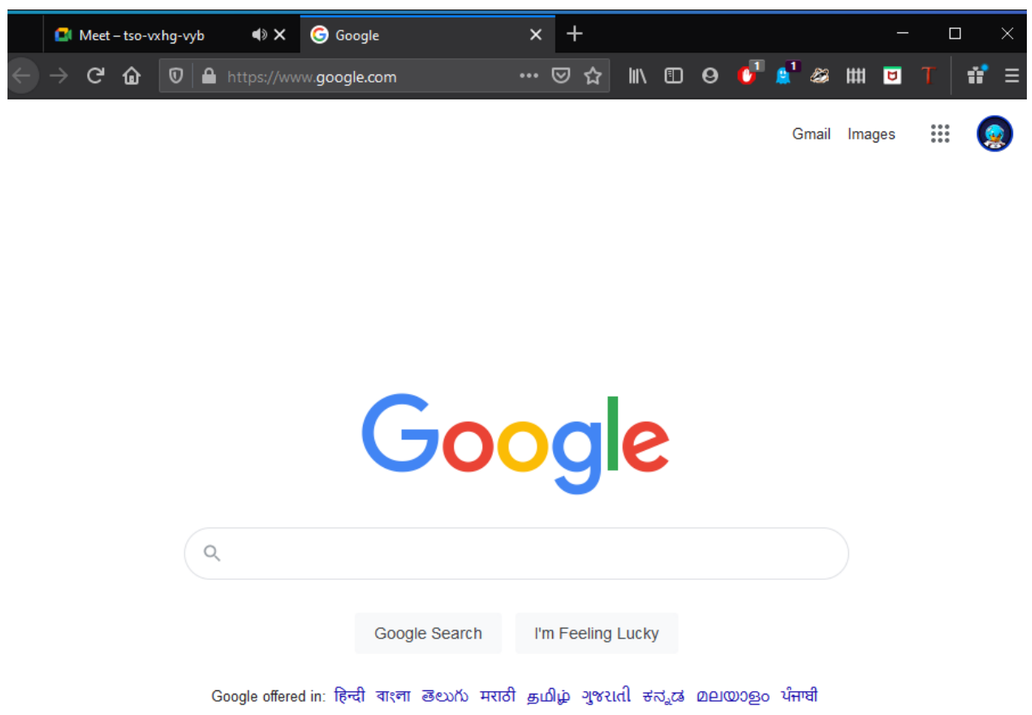


Input URL:
https://www.youtube.com/watch?v=Scnb853CNHM&ab_channel=spartanlz

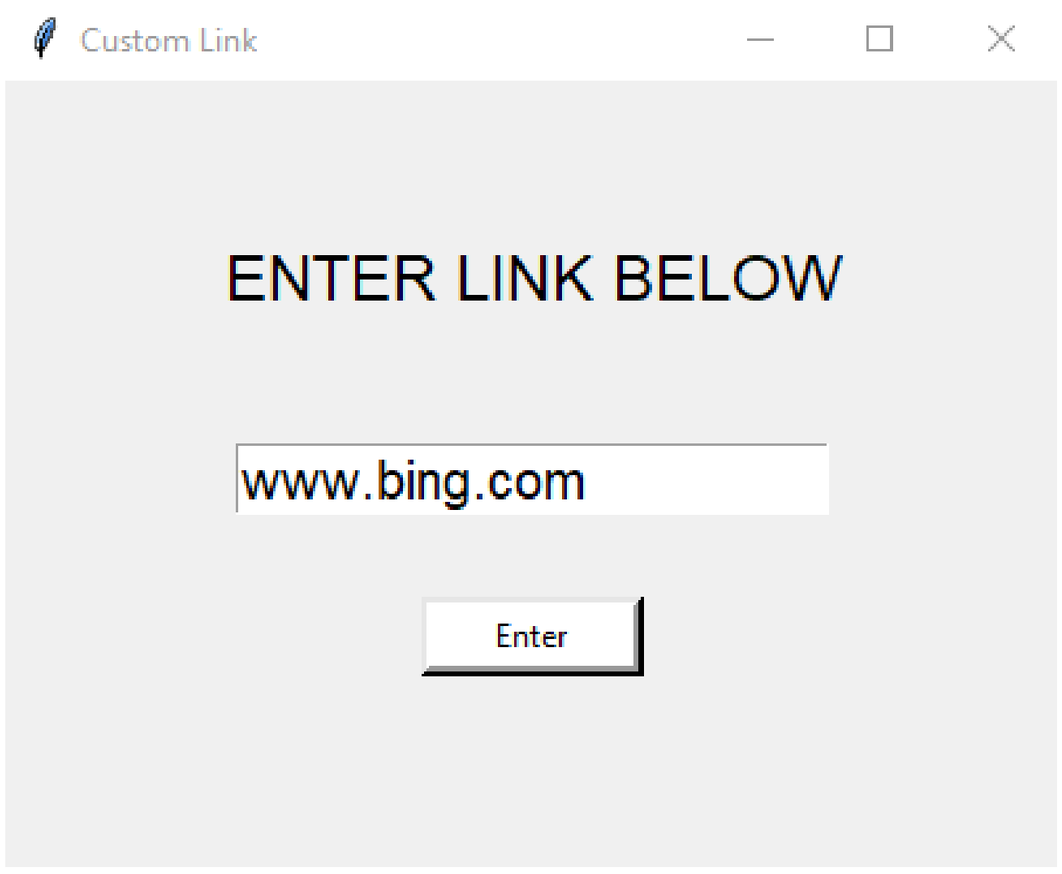
Output URL:
https://tinyurl.com/yc465gub



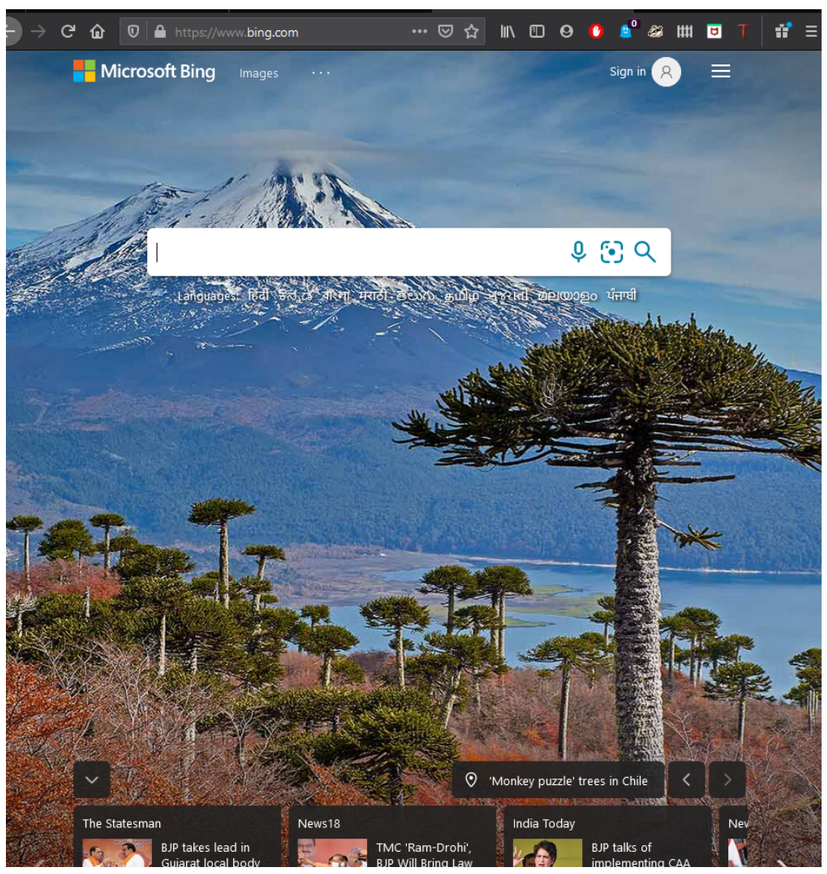
Web Browser Utility



On clicking 'GOOGLE'

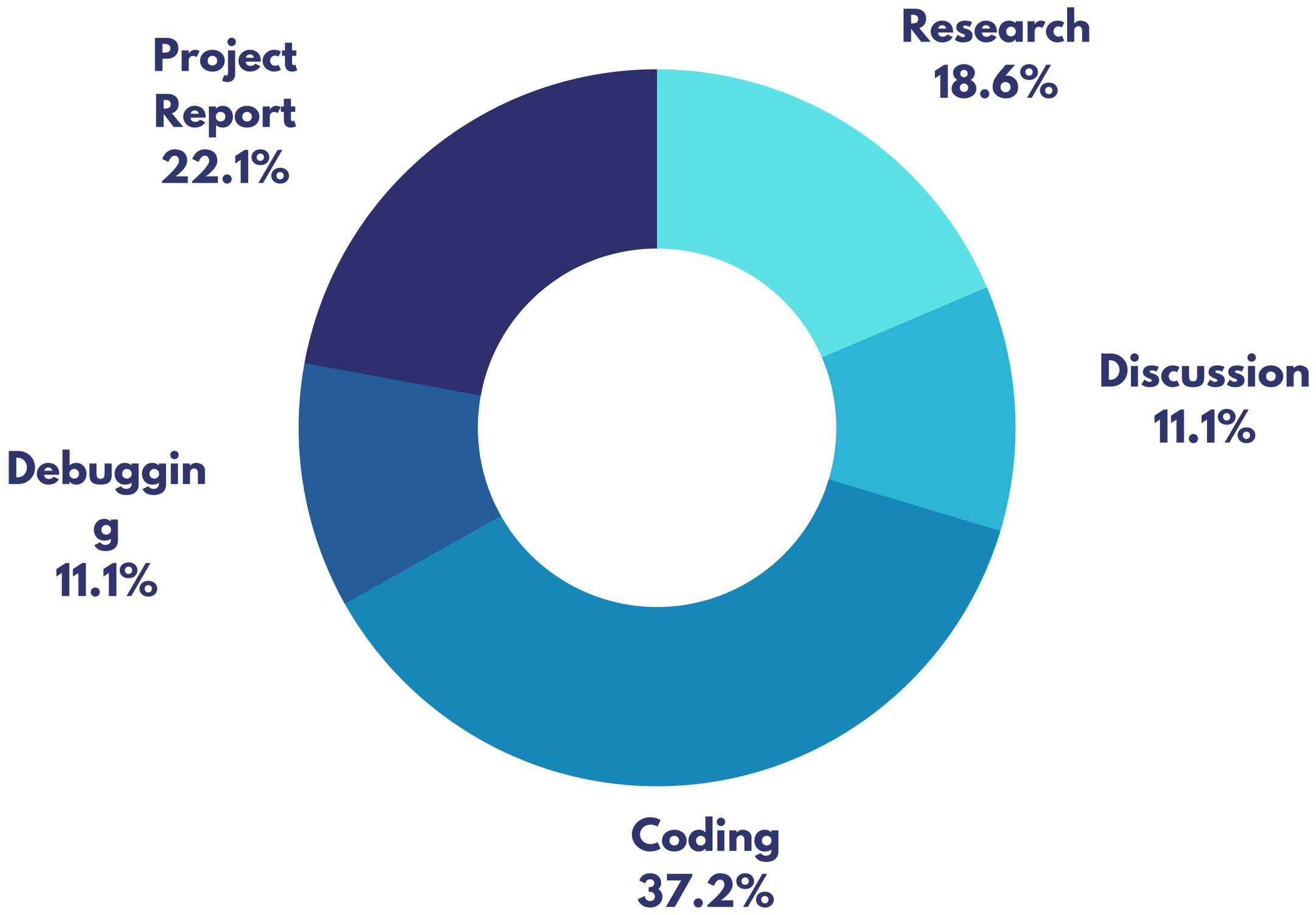


Custom URL Entry



Custom URL output

Time Chart



Research	5 hours
Discussion	3 hours
Coding	10 hours
Debugging	3 hours
Project Report	6 hours
Total	27 hours

Bibliography

- https://docs.python.org/3/library/webbrowser.htm
- https://docs.python.org/3/library/tk.html
- https://pypi.org/project/python-whois/
- https://www.thepythoncode.com/article/extracting-domain-name-information-in-python
- https://docs.python.org/3/library/tkinter.messagebox.html