Report

LOVELY PROFESSIONAL UNIVERSITY

Phagwara, Punjab

Course Details:-

Course Code:- INT 213

CA-2

Project Details:-

PROJECT NO:-15

Title:- Simple Math Based Captcha using Python

Type: -MINI PROJECT

SUBMITED TO:-

Prof.NIDHI ARORA

Student Details :-

Student 1: -

NAME: - Ritesh Alakh Singh

REG NO:- 12109810

ROLL NO: - K21RGA15

Student 2: -

NAME: - Gedela Chaitanya

REG NO: - 12107737

ROLL NO: - RK21RGB40

Student 3: -

NAME: - Vivek Raj

REG NO: - 12109874

ROLL NO: - K21RGB64

ACKNOWLEDGEMENT

We have effort into this project. However, completing this project would not have been possible without the support and guidance of our colleagues. We would like to extend our sincere thanks to all of them. We are highly indebted to Prof. Nidhi Arora for their guidance and supervision. We would like to thank him for providing the necessary information and resources for this project. We are incredibly grateful to our teachers and professors who gave us a chance to work on this project. We would like to thank him for giving us valuable suggestions and ideas. We would like to express our gratitude towards our parents & our friends for their kind co-operation and encouragement which helped us in completing this project. Our thanks and appreciations also go to our colleague in developing the project. Thank you to all the people who have willingly helped us out with their abilities.

Aim: -

To differentiate the users weither the user is real user or automated user, such as bots. Who was loging into any online portal by creating image-based CAPTCHA software by using Python Programing language. CAPTCHAs provide challenges that are difficult for computers to perform but relatively easy for humans.

Objective: -

This program is basically a tool to differentiate the users. This CAPTCHA software is used by any website that wishes to restrict usage by bots. CAPTCHAs provided distorted or overlapping letters and numbers that a user then has to submit via a form field. The distortion of the letters made it difficult for bots to interpret the text and prevented access until the characters were verified. Since CAPTCHA was introduced, bots that use machine learning have been developed. These bots are better able to identify traditional CAPTCHAs with algorithms trained in pattern recognition. Due to this development, newer CAPTCHA methods are based on more complex tests. For example, reCAPTCHA requires clicking in a specific area and waiting until a timer runs out.

System Overview

A CAPTCHA system presents a visitor with an obscured word, words, or phrase. The obscuring is usually achieved by warping the words, distorting the background, or segmenting the word by adding lines.Users are asked to decode the image and enter the alphanumeric characters in the correct order (they may or may not be case sensitive) before submitting the form. Upon form submission, the response is verified, and users are either taken to the next step or presented with an error.

DETAILS OF RESOURCES USED :-

This program was built from scratch using Python Programming Language, we tried to keep the code simple and clean. We tried to make use of classes and objects for the code to be more efficient.

This program was kept to a minimal, using basic python functions and commands. Keeping the project basic gives it a very simple appearance and minimalistic look, which kind of works in its advantage.

The only main resource used in this program is the 'webbrowser' module, which enables the opening of various URLs. The opening of URLs is necessary for the user-friendliness of this program, as otherwise there would be no proper application of the program in any case.

No other external resources are needed for the execution of the program. Yes, the program would need a web browser and an internet connection for its best case use.

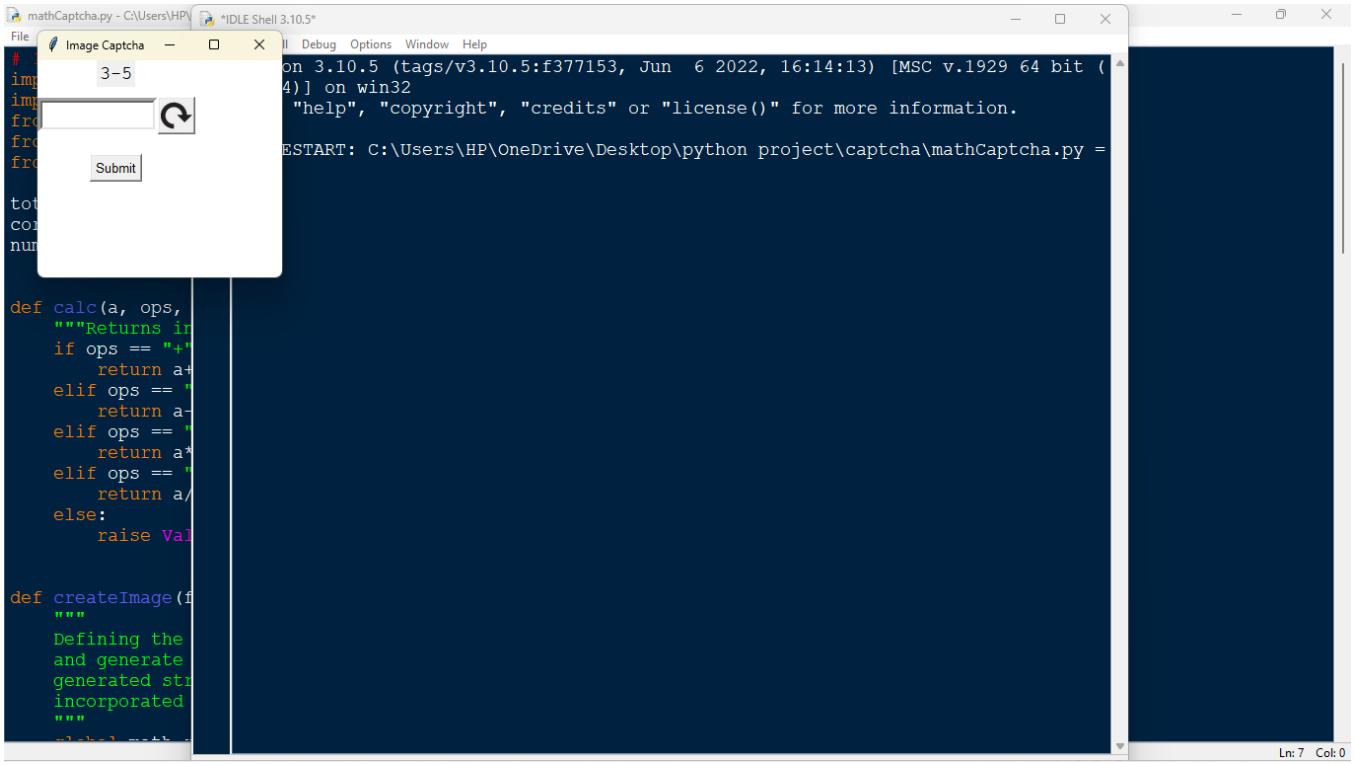
Problem Statement

Nowdays on every platform Bot is main reason for very high traffic on internet. “Bot” generally refers to any program that is set to automatically complete some process, whether it’s posting news on Twitter or leaving spam in website comment sections. Used correctly, these programs are fairly useful, but they can also be used to generate useless/ad-ridden/malicious content, overwhelm a site with signups, rig online poll results, scrape email addresses, or do any number of other unpleasant things. It’s just best not to let them in. To overcome this problem we need Captcha system. CAPTCHA motto goes, to create a task that is “Easy for people, hard for bots.”

SCREENSHOTS:-

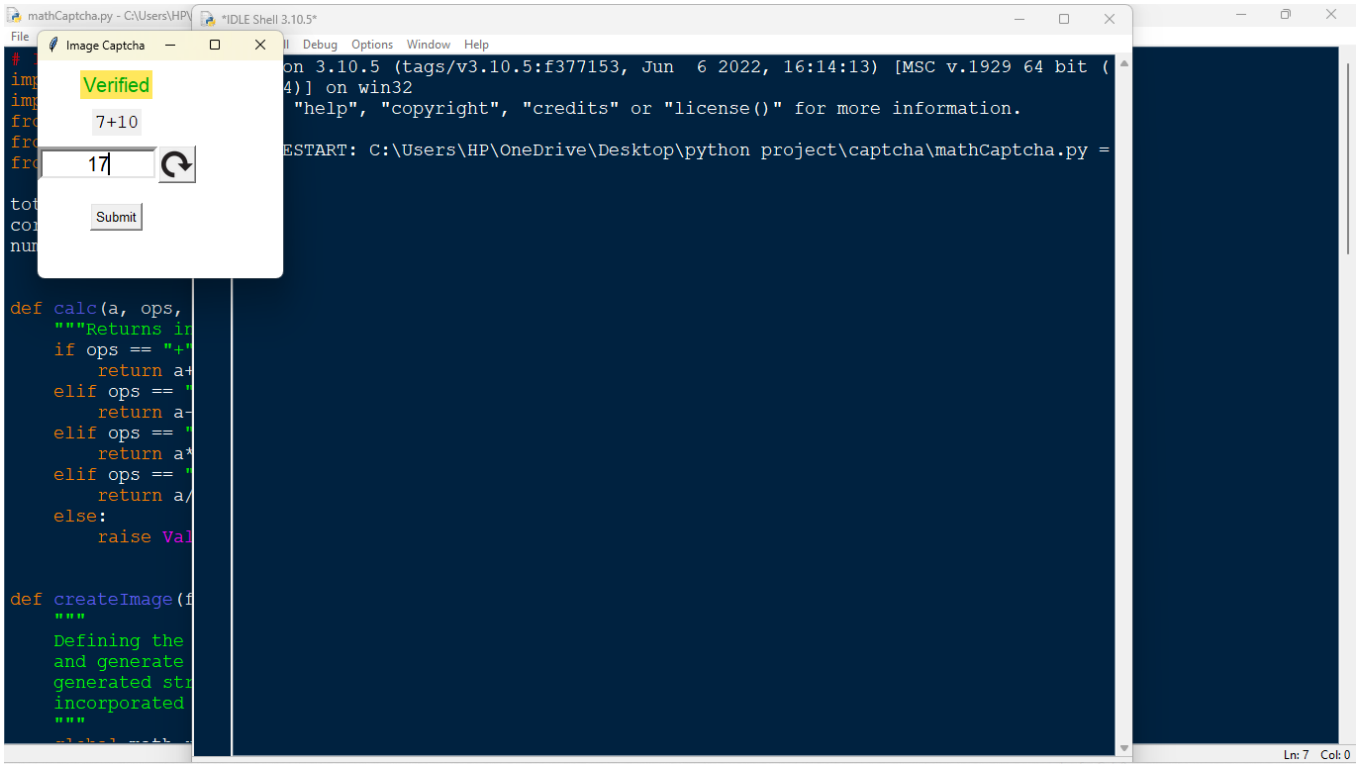
Case 1 :

Home Screen of the Captcha



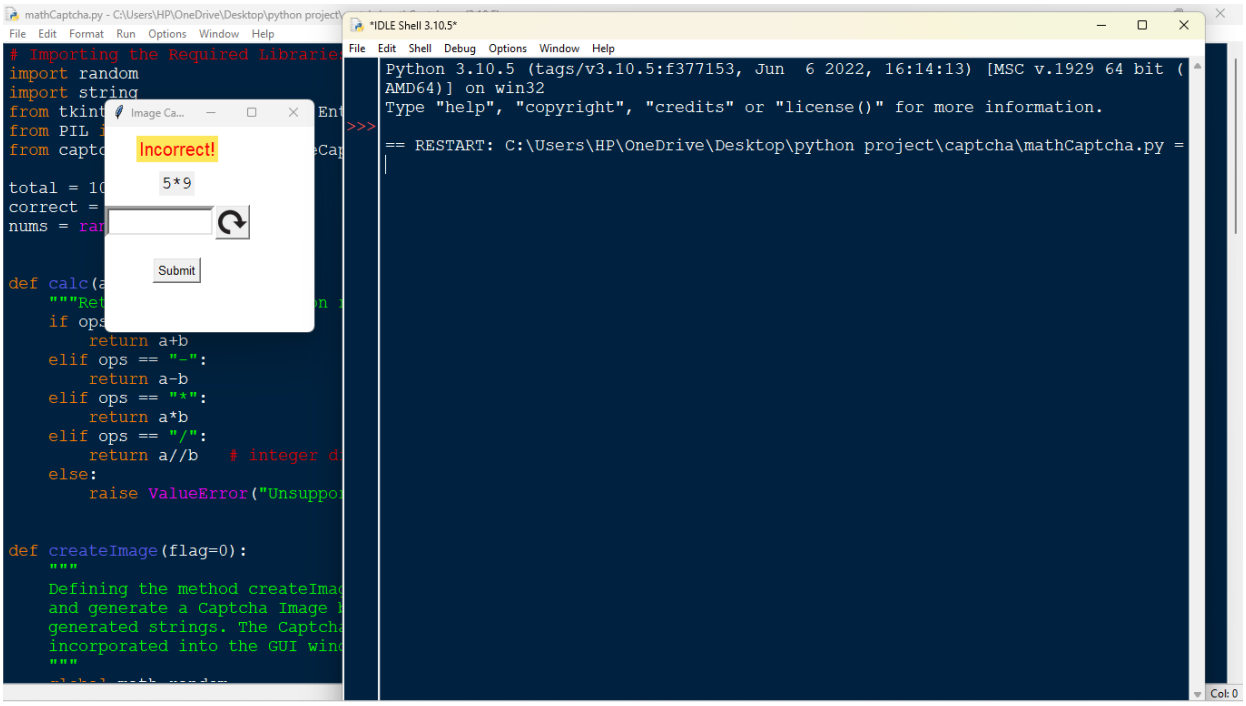
Case 2 :

When all the fields inputs are correct .



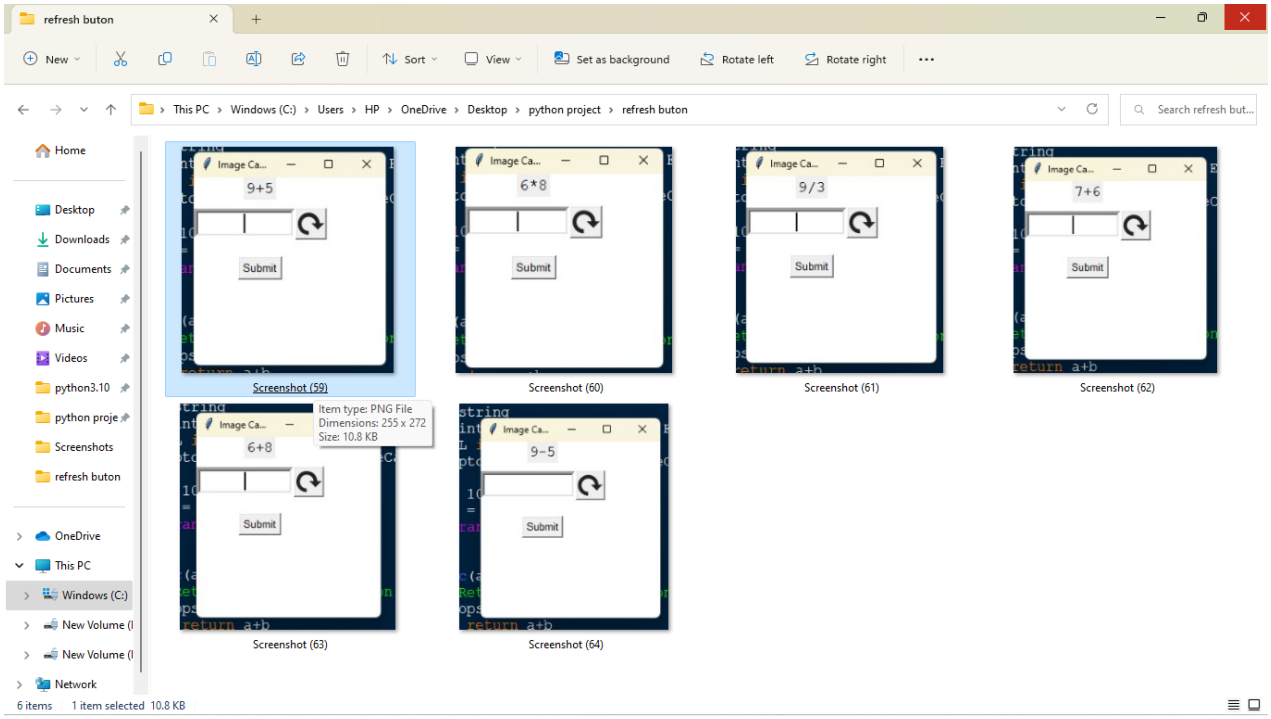
Case 3 :

When all the field inputs are wrong.



Additional Info Screenshots :-

The Refresh Button : We can refresh captcha using refresh this.



Source Code :-

```
# Importing the Required Libraries
import random
import string
from tkinter import Tk, Label, Entry, Button, Text, Label, END
from PIL import ImageTk, Image
from captcha.image import ImageCaptcha

total = 10
correct = 0
nums = range(1, 11)


def calc(a, ops, b):
    """Returns integer operation result from using : 'a','ops','b'"""
    if ops == "+":
        return a+b
    elif ops == "-":
        return a-b
    elif ops == "*":
        return a*b
    elif ops == "/":
        return a//b   # integer division
    else:
        raise ValueError("Unsupported math operation")


def createImage(flag=0):
    """
    Defining the method createImage() which will create
    and generate a Captcha Image based on a randomly
    generated strings. The Captcha Image generated is then
    incorporated into the GUI window we have designed.
    """
    global math_random
    global image_label
    global image_display
    global entry
    global verify_label
    global answer
    ops = random.choice("+-*/")
    a, b = random.choices(nums, k=2)
    answer = calc(a, ops, b)
    math_random = str(a)+str(ops)+str(b)
    # The if block below works only when we press the
    # Reload Button in the GUI. It basically removes
    # the label (if visible) which shows whether the
    # entered string is correct or incorrect.
    if flag == 1:
        verify_label.grid_forget()
    # Removing the contents of the input box.
    entry.delete(0, END)
    # Generating a random string for the Captcha

    # Create label
    l = Label(root, text=math_random)
    l.config(font=("Courier", 14))
    l.grid(row=1, column=0, columnspan=2, pady=0)

def check(x, y):
```

```
"""
    Defining the method check() which will check
    whether the string entered by the user matches
    with the randomly generated string. If there is
    a match then "Verified" pops up in the window.
    Otherwise, "Incorrect!" pops up and a new Captcha
    Image is generated for the user to try again.
    """
    # Making the scope of the below mentioned
    # variables because their values are accessed
    # globally in this script.
    global verify_label
    verify_label.grid_forget()
    if x == y:
        verify_label = Label(master=root,
                             text="Verified",
                             font="Arial 15",
                             bg='#ffe75c',
                             fg="#00a806"
                             )
        verify_label.grid(row=0, column=0, columnspan=2, pady=10)
    else:
        verify_label = Label(master=root,
                             text="Incorrect!",
                             font="Arial 15",
                             bg='#ffe75c',
                             fg="#fa0800"
                             )
        verify_label.grid(row=0, column=0, columnspan=2, pady=10)
        createImage()

if __name__ == "__main__":
    # Initializing Tkinter by creating a root widget,
    # setting Title and Background Color
    root = Tk()
    root.title('Image Captcha')
    root.configure(background='#fff')
    # Initializing the Variables to be defined later
    verify_label = Label(root)
    image_label = Label(root)
    # Defining the Input Box and placing it in the window
    entry = Entry(root, width=10, borderwidth=5,
                  font="Arial 15", justify="center")
    entry.grid(row=2, column=0)
    # Creating an Image for the first time.
    createImage()
    # Defining the path for the reload button image
    # and using it to add the reload button in the
    # GUI window
    path = './refresh.png'
    reload_img = ImageTk.PhotoImage(Image.open(
        path).resize((32, 32), Image.Resampling.LANCZOS))
    reload_button = Button(image=reload_img, command=lambda: createImage(1))
    reload_button.grid(row=2, column=1, pady=10)
    # Defining the submit button
    submit_button = Button(root, text="Submit", font="Arial 10",
                           command=lambda: check(int(entry.get()), answer))
    submit_button.grid(row=3, column=0, columnspan=2, pady=10)
    root.bind('<Return>', func=lambda Event: check(int(entry.get()), answer))
    # This makes the program loops till the user
    # doesn't close the GUI window
    root.mainloop()
```

CONCLUSION: -

Hence, the program was implemented successfully and the wedsite admin will prevent poll skewing by ensuring that each vote is entered by a human, prevent bots from spamming registration systems to create fake accounts, limit scalpers from purchasing large numbers of tickets for resale in ticket system and prevent bots from spamming message boards, contact forms, or review sites.

The real user-friendliness of the program ensures that the user feels comfortable while using the program and is satisfied at the completion of the program.