



HEXAMDA – Hash Simulator

UNDER ESTEEMED GUIDANCE OF “DR. Shwetha Gaur” Prof Maths



BY – S Praveen Kumar

1CR21AD044

O-05

Hexamda – Hashing

Simulator using GUI

PROBLEM STATEMENT:

Write a Python Code to Emulate Hashes using Python using a GUI.

REQUIREMENT AND TOOLS:

Laptop and Basic Knowledge on Python and its Libraries.

THEORY:

Hashes are Cryptographic Converted Codes which consists of letters and numbers in varying order signifying a particular String or File. The most known form of Hashes are MD5 (128 bits), SHA-1 (160 bits), SHA-2 256 (256 bits). Hashes are used for storing classified information such as important files, passwords, etc. The main Key Function of Hashes is that once its Encrypted it can't be Decrypted in the given time. So an inbuilt function is called for Hashing letters.

For the GUI we use Tkinter a Basic GUI Emulator using Python. And This Entire File is Converted to a Working Executable File is using py-installer package

METHODOLOGY:

The Main Functions of this GUI is to be able to Change into Different Themes, Emulate Hashes and Exit using a GUI Function.

There are Two Themes : Light and Dark Mode .

There are Five Different Functions Possible in this GUI .

- Image Hashing Using a URL :
Basically we search for an Image on the Browser and we copy the Image link address from the options when we right click. We just paste using Ctrl+V and then we hash.
- Image Hashing : Path of the Image is used to Hash.
(Selects and Hashes the File rather than the path itself).
- File Hashing : Path of the File is Used to Hash (Selects and Hashes the File rather than the path itself).
- MD5 Hashing : Most Popular Form for Basic hashing it uses the Input String.
- SHA256 : Similar to MD5 it also Considers the String.

ACKNOWLEDGEMENT

I would also like to extend my gratitude to the principal Sir “Dr.Sanjay Jain” and vice principal sir “Dr. B. Narasimha Murthy” and also our HODs “Dr. Raveesha K H” and “Dr. Manjunath M” for providing us with all the facility that was required for this project

I would like to express my special thanks of gratitude to my teacher “Dr. Shwetha Gaur” and also my mentor “Dr. Sam Gilvine” for their able guidance and support in completing my project

DECLARATION BY THE **CANDIDATE**

We hereby declare that that the project entitled
“**Hexamda - Hashing Simulator with GUI**” is an
outcome of our own efforts under the guidance of
“**Dr. Paras Nath Singh**” and our mentor
“**Dr. Sam Gilvine**”.

This project is submitted to **CMR Institute of**
technology.

DATE:

S.Praveen Kumar (O-05)

CODE LISTING

```
from cProfile import label

from logging import root
from tkinter import *
import tkinter as tk
from tkinter import messagebox
from PIL import Image, ImageTk
import urllib.request
from PIL import Image
import imagehash
import hashlib

#setting main variables
root = tk.Tk()
root.title("Hexamda")
root.geometry('800x600')
#root.iconbitmap('icon.png')
#img=ImageTk.PhotoImage(file='ico.png')
#root.iconphoto(True, img)
root.minsize(800,600)
root.resizable(width=True,height=True)
root.config(bg='#121212') # making darkmode as default

#setting light mode
def Light() :
    root.config(bg='#00FFFF')
    lbl.config(bg='#00FFFF',fg='#000000')
    imgurl.config(bg='#00FFFF',fg='#000000')
    imgloc.config(bg='#00FFFF',fg='#000000')
    flloc.config(bg='#00FFFF',fg='#000000')
    md5.config(bg='#00FFFF',fg='#000000')
    sha.config(bg='#00FFFF',fg='#000000')
    credits.config(bg='#00FFFF',fg='#000000')

#setting dark mode
def Dark() :
    root.config(bg='#121212')
    lbl.config(bg='#121212',fg='#00FFFF')
    imgurl.config(bg='#121212',fg='#00FFFF')
    imgloc.config(bg='#121212',fg='#00FFFF')
    flloc.config(bg='#121212',fg='#00FFFF')
    md5.config(bg='#121212',fg='#00FFFF')
    sha.config(bg='#121212',fg='#00FFFF')
    credits.config(bg='#121212',fg='#00FFFF')
```

```
#image url hashing
def urlhash() :
    link=str(url.get())
    urllib.request.urlretrieve(link,'rng')
    var = imagehash.average_hash(Image.open('rng'))
    return messagebox.showinfo('Your Hashed Code is ',var)
```

```
#Image hashing
def imghash() :
    imgloc=ien.get()
    var1 = imagehash.average_hash(Image.open(imgloc))
    return messagebox.showinfo('Your Hashed Code is ',var1)
```

```
#File hashing
def filehash():
    floc=fent.get()
    h = hashlib.sha256()
    with open(floc,'rb') as file:
        chunk = 0
        while chunk != '':
            chunk = file.read(1024)
            h.update(chunk)
        return messagebox.showinfo('Your Hashed Code is ',h.hexdigest())
```

```
#MD5 Hashing
def md5hash() :
    mhash=ment.get()
    var2 = hashlib.md5(str(mhash).encode('utf-8'))
    return messagebox.showinfo('your Hashed Code is :',var2.hexdigest())
```

```
#Sha-256 Hashing
def shahash() :
    shash=sent.get()
    var3 = hashlib.sha256(str(shash).encode('utf-8'))
    return messagebox.showinfo('your Hashed Code is :',var3.hexdigest())
```

```
#setting main label
lbl = tk.Label(root,text="Hexamda The Hashing Tool",font='Helvetica 20 bold italic underline',bg='#121212',fg='#00FFFF')
lbl.place(relx=0.5, rely=0.05,anchor=CENTER)
```

```
#creating a menu options
menu = tk.Menu(root)
```

```
#making and adding all options while ending with menu name
item = tk.Menu(menu,tearoff=False)
#item.add_command(label='Hash Image',command='hashing')
```

```

#item.add_separator()
item.add_command(label='Exit',command=root.destroy)
menu.add_cascade(label='File', menu=item)

#menu two for color mode
item2 = tk.Menu(menu,tearoff=False)
item2.add_command(label='Light',command=Light)
item2.add_command(label='Dark',command=Dark)
menu.add_cascade(label='Theme', menu=item2)

#finishing it with a menu option
root.config(menu=menu)

#setting up image from website
imgurl = tk.Label(root,text="Enter Image URL :",font='Helvetica
16',bg='#121212',fg='#00FFFF')
imgurl.place(relx=0.25,relx=0.15,anchor=CENTER)
url=tk.Entry(root,width=50,)
url.place(relx=0.67,relx=0.15,anchor=CENTER)

#Setting Button 1
button1=tk.Button(root,text='Hash',command=urlhash)
button1.place(relx=0.67,relx=0.20,anchor=CENTER)

#setting up using device image
imgloc = tk.Label(root,text="Enter Image Path:",font='Helvetica
16',bg='#121212',fg='#00FFFF')
imgloc.place(relx=0.25,relx=0.25,anchor=CENTER)
ien=tk.Entry(root,width=50,)
ien.place(relx=0.67,relx=0.25,anchor=CENTER)
iloc=ien.get()

# setting button 2
button2=tk.Button(root,text='Hash',command=imghash)
button2.place(relx=0.67,relx=0.30,anchor=CENTER)

#file location
flloc = tk.Label(root,text="Enter File Path :",font='Helvetica
16',bg='#121212',fg='#00FFFF')
flloc.place(relx=0.25,relx=0.35,anchor=CENTER)
fent=tk.Entry(root,width=50,)
fent.place(relx=0.67,relx=0.35,anchor=CENTER)

#seetting button 3
button3=tk.Button(root,text='Hash',command=filehash)
button3.place(relx=0.67,relx=0.40,anchor=CENTER)

#md5 hash compare

```



```

md5 = tk.Label(root,text="MD5 Hashing :",font='Helvetica
16',bg='#121212',fg='#00FFFF')
md5.place(relx=0.25,rely=0.45,anchor=CENTER)
ment=tk.Entry(root,width=50,)
ment.place(relx=0.67,rely=0.45,anchor=CENTER)

#setting button 4
button4=tk.Button(root,text='Hash',command=md5hash)
button4.place(relx=0.67,rely=0.50,anchor=CENTER)

#sha Compare
sha = tk.Label(root,text="SHA256 Hashing :",font='Helvetica
16',bg='#121212',fg='#00FFFF')
sha.place(relx=0.25,rely=0.55,anchor=CENTER)
sent=tk.Entry(root,width=50,)
sent.place(relx=0.67,rely=0.55,anchor=CENTER)

#setting button 5
button5=tk.Button(root,text='Hash',command=shahash)
button5.place(relx=0.67,rely=0.60,anchor=CENTER)

# credits part
credits=tk.Label(root,text='This is a Program Created to Simulate Basic Hashing\
nBY : S . Praveen Kumar\n1CR21AD044\n@Kanoro.2003',font='Helvetica 15 bold italic
underline',bg='#121212',fg='#00FFFF')
credits.place(relx=0.5,rely=0.8,anchor=CENTER)

#finalizing the program and outputing it.
root.mainloop()

```

SCREEN SHOTS OF OUTPUTS

Hexamda
File Theme

Hexamda The Hashing Tool

Enter Image URL :

Enter Image Path:

Enter File Path :

MD5 Hashing :

SHA256 Hashing :

This is a Program Created to Simulate Basic Hashing
BY : S . Praveen Kumar
1CR21AD044
@Kanoro.2003

Hexamda
File Theme

Hexamda The Hashing Tool

Enter Image URL :

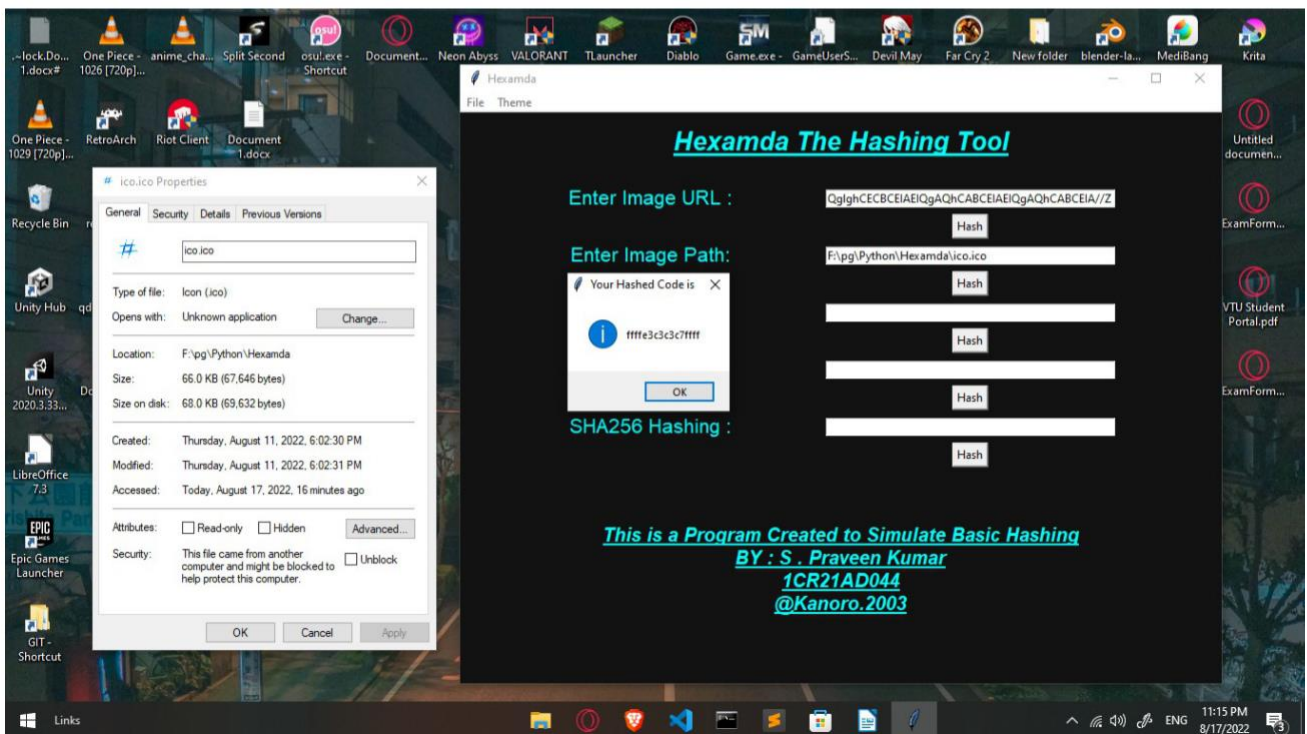
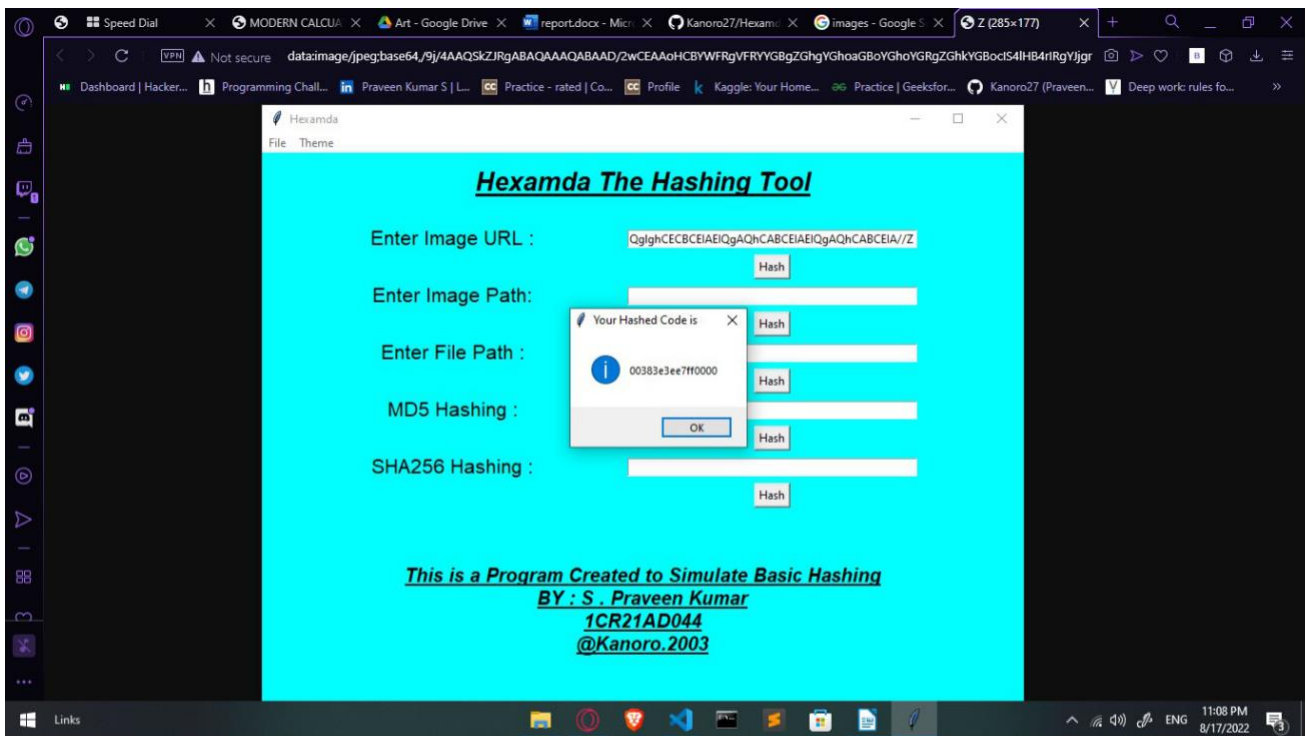
Enter Image Path:

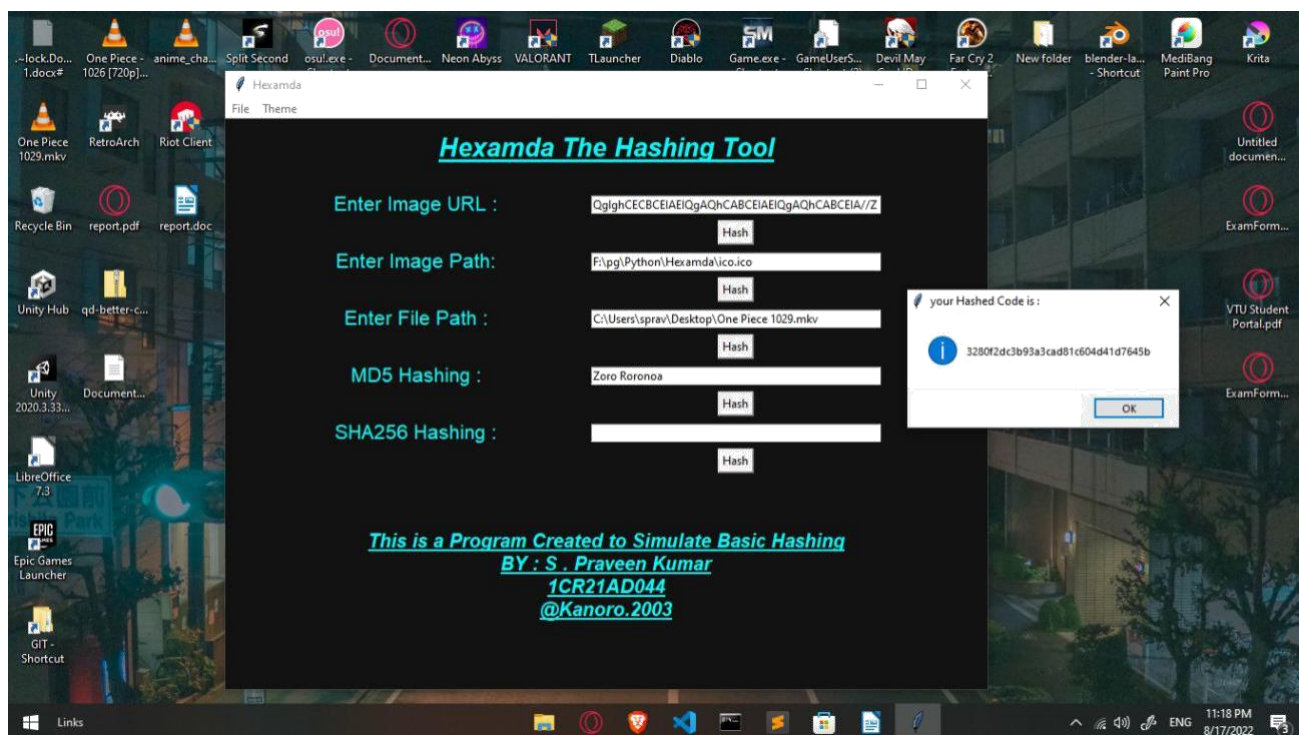
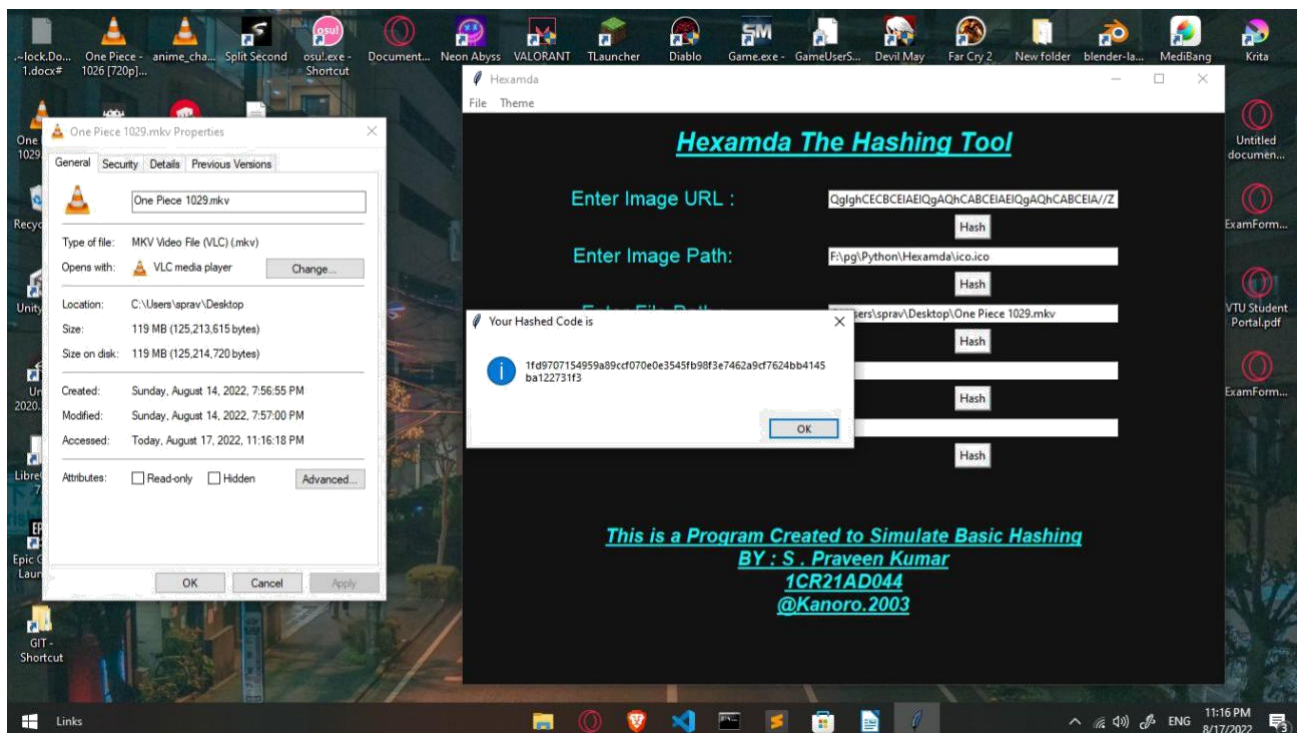
Enter File Path :

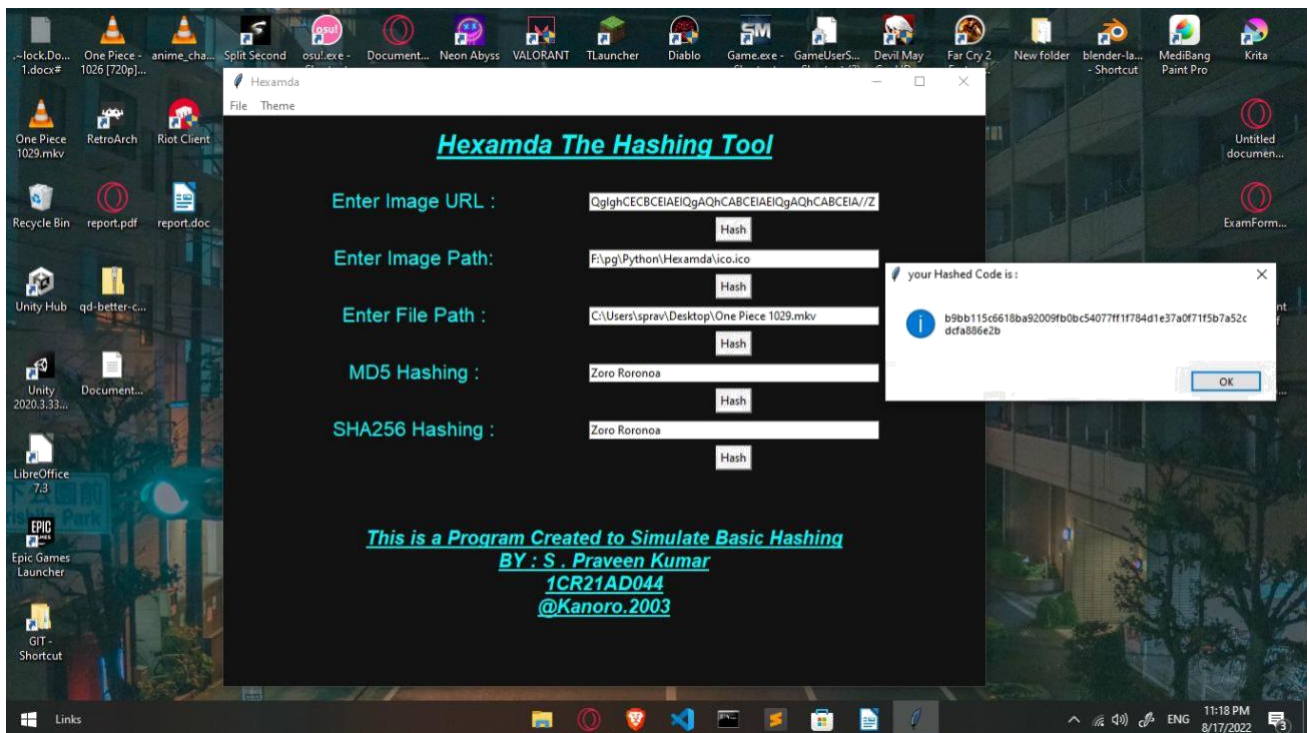
MD5 Hashing :

SHA256 Hashing :

This is a Program Created to Simulate Basic Hashing
BY : S . Praveen Kumar
1CR21AD044
@Kanoro.2003







Future Scope :

The Implementation of Various Functions in Python using Tkinter Proved to be useful in various ways.

The output given is in the form of Message Box which could be changed to a direct display widget which allows us to copy the hashes this can be achieved by using py-game a simple 2D game emulator in python.

We could even Improving it by adding more Hashing Systems and the Function of Right Click select to be added so that Pasting a path or Website or Address is Easier.

Conclusion :

The Objective To Learn the Use of Tkinter in Python was Observed. We even Acquired the Basic Knowledge on Hashes and its Simple applications in the Internet World. The Emulation of Hashes were Made Successfully.

BIBLIOGRAPHY :

[Tutorials Teacher](#) – For Basic Tkinter GUI

[Python Guides](#) – For Basic Tkinter Option Menu GUI

[Geeks for Geeks](#) – For Importing Images In Python

[Python Guides](#) – For Message Box in Tkinter

Wikipedia for MD5 - <https://en.wikipedia.org/wiki/MD5>

Wikipedia for SHA256- <https://en.wikipedia.org/wiki/SHA-2>

[autopy to exe](#) – for converting python into a Executable File

