**SECURE QR CODE**

**OBJECTIVE:**

To design a program to generate a secure QR code using Tkinter module.

**ABSTRACT:**

This Python program generates the random password of a given length and also generates the QR code of that password. You can simply scan the QR code to identify the password or see the password. QR code stands for Quick Response code is basically a barcode on steroids. While the barcode holds information horizontally, the QR code does so both horizontally and vertically. This enables the QR code to hold over a hundred times more information.

**INTRODUCTION :**

QR Code QR i.e. "Quick Response" code is a 2D matrix code that is designed by keeping two points under consideration, i.e. it must store large amount of data as compared to 1D barcodes and it must be decoded at high speed using any handheld device like phones.

QR Code is a two-dimensional bar code used for its fast readability and comparatively large storage capacity. It consists of black squares arranged in a square grid on a white background.

**SOFTWARE REQUIREMENTS:**

Software: JUPYTER NOTEBOOK – Python 3

**CONCEPTS/WORKING PRINCIPLE:**

Through this code, we are trying to develop a GUI application using “Tkinter” that will generate a random password based on the user’s preference. For this we are using the “pyqrcode” module. And to copy the random password to the clipboard “pyperclip” module is used.

Step-1:

Import the required libraries

Tkinter

Pyqrcode

re (inbuilt).

Step-2 :

Create a from line user interface using Tkinter library 6 to 9 in code is used to resect Create main Event with main window ”top”.

Step-3:

Creating Here a Canvas in the main event loop. Canvas1 is created in main window “top”. with dimension (400x300).

Step-4:

the User's Data and another for Entering the Password. and now create two buttons

Create two Entry boxes one for Entering one for generating the arcade for the given user's data in the 1st Entry box and another button for Submitting the Password

\* when button I is pressed then it will call getqr coder function for generating arcade.

\* Similarly button 2 will call password generating and verifying function i.e (“password()”).

after clicking Submit button will be verified.

Step 5:

In ”get qrcode() “ function qrcode will be generated using [Pyay code «create() function. and this will be visible to user only if he Enter's correct password. This is done by using ("BitmapImage()") function.

Step-6:

In Password() function the

Passed to re. findall () function.

It will return the 1st letter. of the each word in user data.Those letters are joined using join()function and assinged to pwd variable.

Now, The Stored Password in Pwd will be used to verify the password given by the user,

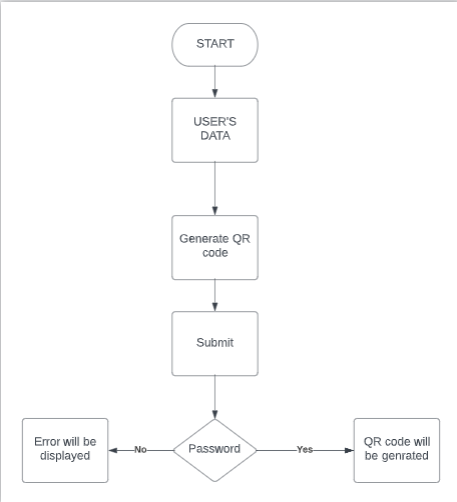
Password (2nd entry) if password in 2nd entry box is stored in variable ‘password’.

After clicking the submit button password will be verified.

And if the qr code is displayed if the Password is correct.

or else, An error message will be displayed.

**FLOWCHART:**

****

**PROGRAM/ APPROACH/METHODOLOGY:**

import tkinter as tk

from tkinter import \*

import re

import pyqrcode

from PIL import Image,ImageTk

top = tk.Tk()

top.geometry("500x500")

top.pack\_propagate(False) # tells the root to not let the widgets inside it determine its size.

top.resizable(0, 0)

canvas1 = tk.Canvas(top, width = 400, height = 300)

canvas1.pack()

s\_var = tk.Entry(top)

canvas1.create\_window(200, 100, window=s\_var)

password\_entry = tk.Entry(top, show="\*")

canvas1.create\_window(200, 170, window=password\_entry)

def password():

s = s\_var.get()

pwd = re.findall(r'\b([a-zA-Z]|\d+)', s)

pwd = "".join(pwd)

passwrd = password\_entry.get()

if passwrd == pwd:

img\_lbl.config(image=img)

output.config(text="QR code ")

else :

w = tk.Label(top, text='error', font="50")

w.pack()

msg = tk.Message(top, text="please enter correct password")

msg.pack()

def getqrcode():

global qr,img

p = s\_var.get()

qr = pyqrcode.create(p)

img = BitmapImage(data=qr.xbm(scale=8))

#qr.save('my\_qrcode.png')

button1 = tk.Button(text='Generate qr code', command=getqrcode)

canvas1.create\_window(200, 140, window=button1)

button2 = tk.Button(text='submit', command=password)

canvas1.create\_window(200, 200, window=button2)

img\_lbl = tk.Label(top)

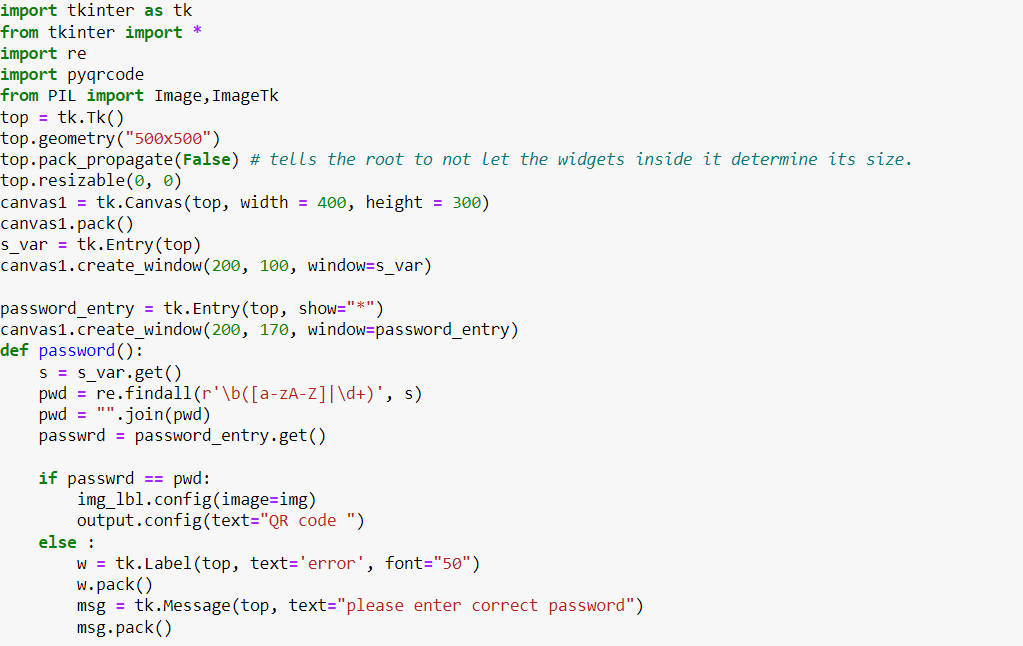
img\_lbl.pack()

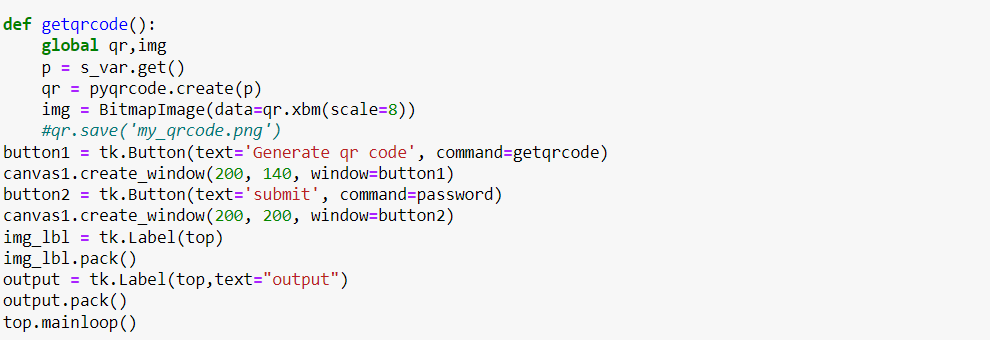
output = tk.Label(top,text="output")

output.pack()

top.mainloop()

**OUTPUT:**

****

****

****

**CONCLUSION:**

Hence, a program to secure qrcode was designed using Tkinter module in Python.

**REFERENCES:**

https://www.geeksforgeeks.org

https://www.geeksforgeeks.org/python-gui-tkinter/