SAVEETHA SCHOOL OF ENGINEERING

Affiliated to AICTE | IET-UK Accreditation

PYTHON PROGRAMMING FOR QR CODE GENERATOR USING GUI

GUIDED BY-

Raveena Selvanarayanan

Team members

192210363-P.Khaleel Ahamed

192110114-T.Vithesh

192210137-N.Sai Chandu

INTRODUCTION

- A QR code generator with a graphical user interface (GUI) is a software tool designed to simplify the process of creating QR (Quick Response) codes for users of all technical backgrounds.
- Unlike command-line interfaces, GUI-based generators offer a visually intuitive environment where users can interact with buttons, input fields, and dropdown menus to customize their QR codes.
- One of the primary benefits of using such a tool is its ease of use, as it eliminates the need for users to have in-depth technical knowledge.

ABSTRACT

- Additionally, GUI-based generators provide immediate visual feedback, allowing users to see real-time changes to their QR codes as they make adjustments
- Some generators even allow users to add logos, images, or branding elements to their QR codes.
- Furthermore, GUI-based QR code generators are often compatible with multiple operating systems, ensuring accessibility for users regardless of their preferred platform.
- Overall, a GUI-based QR code generator offers a user-friendly, customizable solution for a variety of applications, including marketing, advertising, inventory management, and more.

LITERATURE

AUTHOR NAME	CONCEPT	ADVANTAGES	FUTURE SCOPE	ALGORITHM USED
Nielsen, J. (1993)	Usability and User Experience	Firstly, GUI-based applications are inherently more user-friendly,	Implementation of inclusive designs	Block chain algorithm
Lazar, J.Goldstein.	Ensuring Digital Accessibility Through Process and Policy.	This ease of use extends to individuals with varying levels of technical expertise,	Create more intuitive and immersive user interfaces	Augumented reality and virtual reality algorithms
Oinas-Kukkonen	Key Issues, Process Model, and System Features.	making the QR code generator accessible to a broader audience.	Integration of big data analytics	(mvc)model-view-controller algorithm

AUTHOR NAME	CONCEPT	ADVANTAGES	FUTURE SCOPE	ALGORITHM USED
Fogg, B. J.	Using Computers to Change What We Think and Do.	This visual confirmation enhances user confidence and ensures that the desired QR code has been created accurately	Improve user privacy and control over personal data	(mvvm)model view-view model algorithm
Cavoukian, A.	Privacy by Design in the Age of Big Data.	The cross-platform compatibility of GUI frameworks like Tkinter	Solutions for secure and verifiable QR code transactions	Model view-present(mvp) algorithm
Schneier, B	Enabling the Trust that Society Needs to Thrive.	creating a QR code generator with a GUI provides a user-friendly	Enable real-time updates and customization	Recommedation algorithms

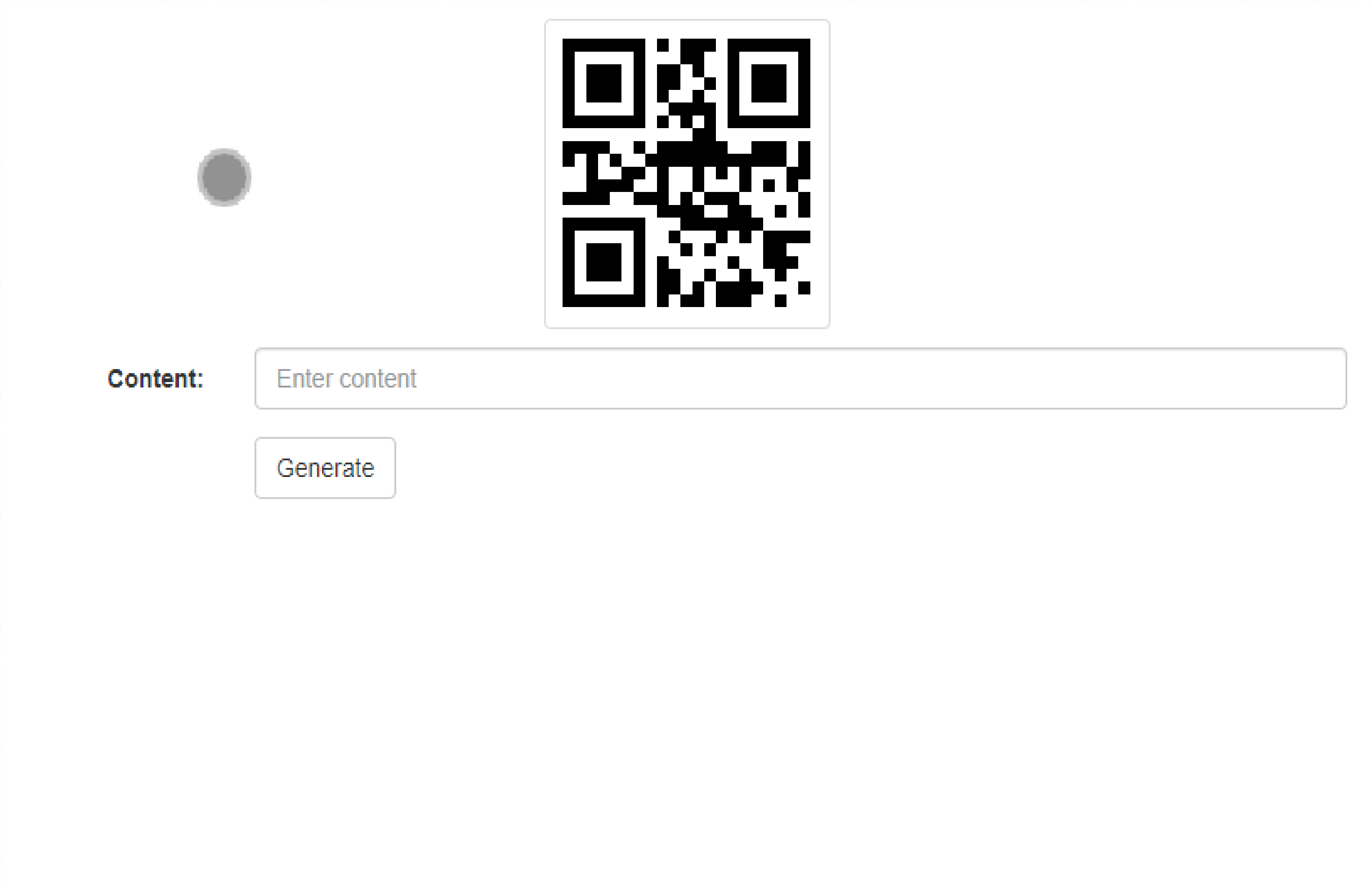
CODE/PROGRAM

```
# pip install qrcode pillow
import qrcode, PIL
from PIL import ImageTk
import tkinter as tk
from tkinter import ttk,messagebox,filedialog
from PIL import ImageTk

def createQR(*args):
    data = text_entry.get()
    if data:
        img = qrcode.make(data) #generate QRcode
        res_img = img.resize((280,250)) # reszie QR Code Size
        #Convert To photoimage
        tkimage= ImageTk.PhotoImage(res_img)
        qr_canvas.delete('all')
        qr_canvas.create_image(0,0,anchor=tk.NW, image=tkimage)
        qr_canvas.image = tkimage
```

```
 else:
        messagebox.showwarning("Warning",'Enter Data in Entry First')

def saveQR(*args):
    ata = text_entry.get()
    if data:
        img = qrcode.make(data) #generate QRcode
        res_img = img.resize((280,250)) # reszie QR Code Size

        path = filedialog.asksaveasfilename(defaultextension=".png",)
        if path:
            res_img.save(path)
            messagebox.showinfo("Sucess","QR Code is Saved ")
    else:
        messagebox.showwarning("Warning",'Enter Data in Entry First')

root = tk.Tk()
root.title("QR Code Generator")
root.geometry("300x380")
root.config(bg='white')
root.resizable(0,0)
```

```
frame1 = tk.Frame(root,bd=2,relief=tk.RAISED)
frame1.place(x=10,y=5,width=280,height=250)
frame2 = tk.Frame(root,bd=2,relief=tk.SUNKEN)
frame2.place(x=10,y=260,width=280,height=100)

coverImg = tk.PhotoImage(file="qrCodeCover.png")
qr_canvas = tk.Canvas(frame1)
qr_canvas.bind("<Double-1>",saveQR)
qr_canvas.create_image(0,0,anchor=tk.NW, image=coverImg)
qr_canvas.image = coverImg
qr_canvas.pack(fill=tk.BOTH)
text_entry = ttk.Entry(frame2,width=26,font=("Sitka Small",11),justify=tk.CENTER)
text_entry.place(x=5,y=5)

btn_1 = ttk.Button(frame2,text="Create",width=10,command=createQR)
btn_1.place(x=25,y=50)
btn_2 = ttk.Button(frame2,text="Save",width=10,command=saveQR)
btn_2.place(x=100,y=50)
btn_3 = ttk.Button(frame2,text="Exit",width=10,command=root.quit)
btn_3.place(x=175,y=50)
root.mainloop()
```

SAVEETHA UNIVERSITY
2005
Knowledge Conquers All



EXISTING WORK

* The process typically involves capturing user input, encoding it into a QR code using the qrcode library, and displaying the generated QR code image within the GUI interface.

* The example provided demonstrates a basic implementation where users can input text or links, click a button to generate the corresponding QR code, and view the generated QR code within the application window.

HARDWARE/SOFTWARE USED:

- Software used for the research of QR(QUICK RESPONSE) code generator using GUI (GRAPHICAL USER INTERFACE) is **VIRTUAL STUDIO CODE** or else we may use the **GOOGLE COLAB** software.
- GOOGLE COLAB --- One of its most notable features is its accessibility, as it requires no installation or configuration, leveraging Google's infrastructure for computation.
- Visual Studio Code (VS Code) is a lightweight yet powerful source code editor d. It provides support like syntax highlighting, intelligent code completion, debugging capabilities, and version control integration.

PROPOSED MODEL

- It involves several key elements includes:
- Literature Review
- Problem Definition
- Methodology
- Design and Development
- Testing and Evaluation

RESEARCH GAP

- In the context of developing a QR code generator with a graphical user interface (GUI), several potential research gaps may exist
- Security and Privacy: Limited research may exist on the security and privacy implications associated with GUI-based QR code generators.
- Investigating potential vulnerabilities, privacy risks, and security best practices for handling sensitive information within QR codes could contribute to the development of more secure and privacy-aware applications.

CONCLUSION

- The development of a QR code generator with a graphical user interface (GUI) has been successfully achieved. This project aimed to provide users with a simple yet powerful tool to create QR codes efficiently. Throughout the development process, several key components were implemented, including a user-friendly interface, QR code generation functionality, and the ability to customize QR code content and design.

REFFRENCE

•Arase K. (2020) *QR Code Generator for JavaScript.* *http://d-project.googlecode.com/svn/trunk/misc/qrcode/js/qrcode.js [accessed 9 July 2019]*

•Denso Wave Incorporated. (2018) *About QR code. http://www.qrcode.com/en/aboutqr.html [accessed 9 July 2019]*

•Denso Wave Inc. (n.d.c). QR code standardization. Retrieved Nov. 14, 2021, from http://www.denso-wave.com/qrcode/qrstandard-e.htm#notes01

•Hadro, J. (2019, Aug. 19). Can mobile tech, collaboration, and a focus on user experience rejuvenate reference services? Library Journal. Retrievec Nov. 14, 2018 fromhttp://www.libraryjoumal.com/lj/community/academiclibraries/886418-

THANK YOU