QR-CODE

```
import tkinter as tk
import customtkinter as ctk
from PIL import Image, ImageTk
import qrcode
from tkinter import filedialog, messagebox



class App(ctk.CTk):
    def __init__(self):
        # set up window
        self.tk_image = None
        self.row_img = None
        ctk.set_appearance_mode('light')
        super().__init__(fg_color='white')

        # customization
        self.geometry('400x400')
        self.title('')

        row_img = tk.PhotoImage(file='/home/saabir/Music/Tkinter/GUI-development-tkinter/QR-Code Project/empty.png')
        self.tk.call('wm', 'iconphoto', self._w, row_img)

        # Input
        self.text_input = tk.StringVar()
        CTkEntery(self,  self.text_input, self.save_image)

        self.text_input.trace('w', self.generate_code)


        # qr code image
        self.qr_image = QRImage(self)
        # image = Image.open("Placeholder.png").resize((200, 200), 2)
        # placeholder_image = ImageTk.PhotoImage(image)
        # self.qr_image.update_image(placeholder_image)

        self.bind("<Return>", lambda event: self.save_image())

        # running the app
        self.mainloop()

    def save_image(self):
        if self.row_img:
            file = filedialog.asksaveasfile(mode='wb', defaultextension=".png")
            if file:
                self.row_img.save(file)  # saves the image to the input file name.

    def generate_code(self, *args):
        text = self.text_input.get()
        if text:
            self.row_img = qrcode.make(text).resize((200, 200), 2)
            self.tk_image = ImageTk.PhotoImage(self.row_img)
            self.qr_image.update_image(self.tk_image)
        else:
            self.qr_image.clear()
            self.row_img = None
            self.tk_image = None
```

```
class CTkEntery(ctk.CTkFrame):
    def __init__(self, parent, text_input,save_image):
        super().__init__(parent, fg_color='#021FB3', corner_radius=20)

        self.place(relx=0, rely=0.8, relwidth=1, relheight=0.4)

        self.rowconfigure((0, 1), weight=1)
        self.columnconfigure(0, weight=1)

        # frame
        frame = ctk.CTkFrame(self, fg_color='transparent')
        frame.grid(row=0, column=0, sticky='ew')

        # frame grid set up
        frame.rowconfigure((0, 1), weight=1, uniform='b')
        frame.columnconfigure(0, weight=1, uniform='b')
        frame.columnconfigure(1, weight=4, uniform='b')
        frame.columnconfigure(2, weight=2, uniform='b')
        frame.columnconfigure(3, weight=1, uniform='b')

        # the widgets
        entery = ctk.CTkEntry(frame, textvariable=text_input, fg_color='#2E54E8', border_width=0, text_color='white')
        button = ctk.CTkButton(frame, text='save', fg_color='#2E54E8', hover_color='#4266f1', command=save_image)
        entery.focus()

        entery.grid(row=0, column=1, sticky='news', padx=5)
        button.grid(row=0, column=2, sticky='news', padx=5)


class QRImage(tk.Canvas):
    def __init__(self, parent):
        super().__init__(parent, background='white', bd=0, highlightthickness=0, relief='ridge')

        self.place(relx=0.5, rely=0.4, width=200, height=200, anchor='center', )

    def update_image(self, image_tk):
        self.clear()
        self.create_image(0, 0, image=image_tk, anchor='nw')

    def clear(self):
        self.delete('all')


app = App()
```