

QR CODE GENERATOR USING PYTHON

```
# Import the modules
import qrcode
from PIL import ImageTk, Image
import tkinter as tk
from tkinter import ttk, filedialog, messagebox

# Define Functions
def createQR(*args):
    data = text_entry.get()

    if data and data != "Enter Here...": # Check if data is not the
placeholder text
        img = qrcode.make(data) # generate QR code
        res_img = img.resize((280, 250)) # resize 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):
    data = text_entry.get()
    if data and data != "Enter Here...": # Check if data is not the
placeholder text
        img = qrcode.make(data) # generate QR code
        res_img = img.resize((280, 250)) # resize QR Code Size

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

# Function to handle focus event to clear the placeholder text
def on_focus_in(event):
    if text_entry.get() == "Enter Here...":
        text_entry.delete(0, tk.END) # Delete the placeholder text
when the user clicks in the box
```

```
text_entry.config(foreground="#000000") # Change text color
to black when user starts typing
```

```
# Function to handle focus out event to reinsert placeholder text if
the entry is empty
```

```
def on_focus_out(event):
    if text_entry.get() == "":
        text_entry.insert(0, "Enter Here...") # Insert the
placeholder text back
        text_entry.config(foreground="#888888") # Change text color
to gray for placeholder
```

```
# GUI Code
```

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

```
# Frames
```

```
frame1 = tk.Frame(root, bd=1, relief=tk.RAISED)
frame1.place(x=10, y=0, width=280, height=250)
```

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

```
# Cover Image Placeholder
```

```
cover_img = tk.PhotoImage(file="QR_Code.png")
```

```
qr_canvas = tk.Canvas(frame1)
qr_canvas.create_image(0, 0, anchor=tk.NW, image=cover_img)
qr_canvas.image = cover_img
qr_canvas.bind("<Double-1>", saveQR)
qr_canvas.pack(fill=tk.BOTH)
```

```
# Entry Box with Placeholder
```

```
text_entry = ttk.Entry(frame2, width=26, font=("Sitka Small", 11),
justify=tk.CENTER)
text_entry.insert(0, "Enter Here...") # Insert placeholder text
text_entry.config(foreground="#888888") # Set text color for
placeholder
text_entry.bind("<Return>", createQR)
text_entry.bind("<FocusIn>", on_focus_in) # Bind focus-in event to
clear placeholder
text_entry.bind("<FocusOut>", on_focus_out) # Bind focus-out event
to reinsert placeholder if needed
text_entry.place(x=5, y=5)
```

```
# Buttons
```

```
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)

# Run the GUI main loop
root.mainloop()
```