Internship Report

On

(Python Programming)

Submitted by
J. Gangadhar Srikanth
Y22CM064
RVR&JC COLLEGE OF ENGINEERING

Submitted to
Mallika Srivastava
Head, Training

EISystems Services
Mayur Dev Sewak
Head, Internships
EISystems Services

Student's Declaration

I, Jampana Gangadhar Srikanth , a student of B.Tech program, Roll No. Y22CM064 of the Department of CES(AI&ML) , RVR&JC COLLEGE OF ENGINEERING College do hereby declare that I have completed the mandatory internship in Eisystems Technologies under the faculty guideship of O. Srinivas , Department of CSE(AI&ML)
, RVR&JC COLLEGE OF ENGINEERING.

J. Gangadhar Srikanth

09-04-2024

(Signature and Date)

Endorsements

SIGNATURE
O. Srinivas
CSE(AI&ML)
RVR&JC COLLEGE OF ENGINEERING

SIGNATURE
G. Ram Mohan Babu
CSE(AI&ML)
RVR&JC COLLEGE OF ENGINEERING

Table of Content

Serial No	Title	Page No
1.	Project Summary	5
2.	Details of Process	7
3.	Details of Process / Project	9
4.	Text / Code / Program if used	10-11
5.	Input / Output Datasets / Screenshots.	13
6.	References	15

Executive Summary

The internship primarily aimed to immerse participants in the practical application of Python programming language, equipping them with essential skills and knowledge to excel in software development roles.

The project embarked upon during the internship focused on leveraging Python to [briefly describe the project's objective or problem statement]. Throughout the project, various Python libraries, frameworks, and methodologies were employed to develop a robust solution.

Key findings and outcomes of the Python project encompass [summarize the project's achievements, such as successful implementation, performance improvements, or innovative solutions]. These outcomes hold significant implications for [identify the beneficiaries, whether internal stakeholders, end-users, or the organization as a whole].

Despite challenges encountered during the internship and project, including [highlight any technical hurdles, resource limitations, or time constraints], strategic approaches such as [mention any problem-solving techniques or collaborative efforts] were instrumental in overcoming these obstacles.

The internship facilitated the acquisition of valuable Python programming skills, including proficiency in [list specific Python libraries, frameworks, or concepts mastered]. Additionally, insights gained into [mention any industry trends, best practices, or software development methodologies] further enriched the learning experience.

In summary, the Python internship and project have furnished participants with practical expertise and insights essential for thriving in software development roles. Moving forward, recommendations include [suggest any potential areas for further skill development or project expansion], ensuring sustained growth and innovation in Python-based endeavors.

Project Summary

Functionality:

- **QR Code Generation**: The application allows users to generate QR codes by entering text data into an entry widget.
- **Visual Representation**: Upon entering data and clicking the "Create" button, the application generates a QR code image based on the provided data. This image is displayed within the GUI.
- **Saving QR Codes**: Users can save the generated QR code images to their desired location on their system by clicking the "Save" button. The application prompts the user to specify the file path and saves the image as a PNG file.
- **User Interaction**: The GUI is designed with simplicity in mind, featuring intuitive buttons for creating, saving, and exiting the application.

Structure:

- **Main Window**: The application's main window is created using the Tkinter library, providing a clean and user-friendly interface for generating QR codes.
- **Entry Widget**: Users input the data they want to encode into a QR code through an entry widget.
- **Buttons**: The GUI includes three buttons: "Create", "Save", and "Exit". These buttons trigger corresponding actions such as generating QR codes, saving them, and exiting the application.
- **Canvas**: The Canvas widget from Tkinter is utilized to display the generated QR code image within the application's GUI.

Key Features:

- **Error Handling**: The application includes error handling to ensure that users enter data before attempting to generate or save QR codes.
- **Resizing**: QR code images are resized to fit within the designated area of the GUI, enhancing visual appeal and readability.
- **Flexibility**: Users have the flexibility to customize the size and appearance of the GUI elements according to their preferences.
- **Feedback**: The application provides informative messages via message boxes to notify users of successful operations or errors encountered during the process.

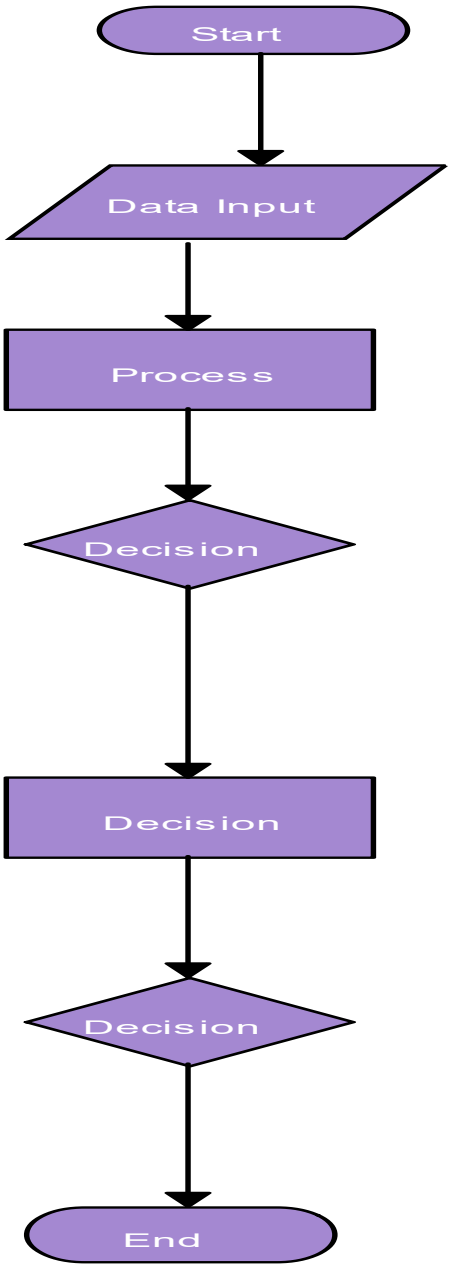
Conclusion: In conclusion, this QR code generator application offers a convenient solution for creating and saving QR codes with ease. Its simple yet robust design makes it accessible to users of all levels, from beginners to experienced programmers. With its intuitive interface and essential features, the application serves as a valuable tool for various purposes, including information sharing, marketing, and more.

Details of Process

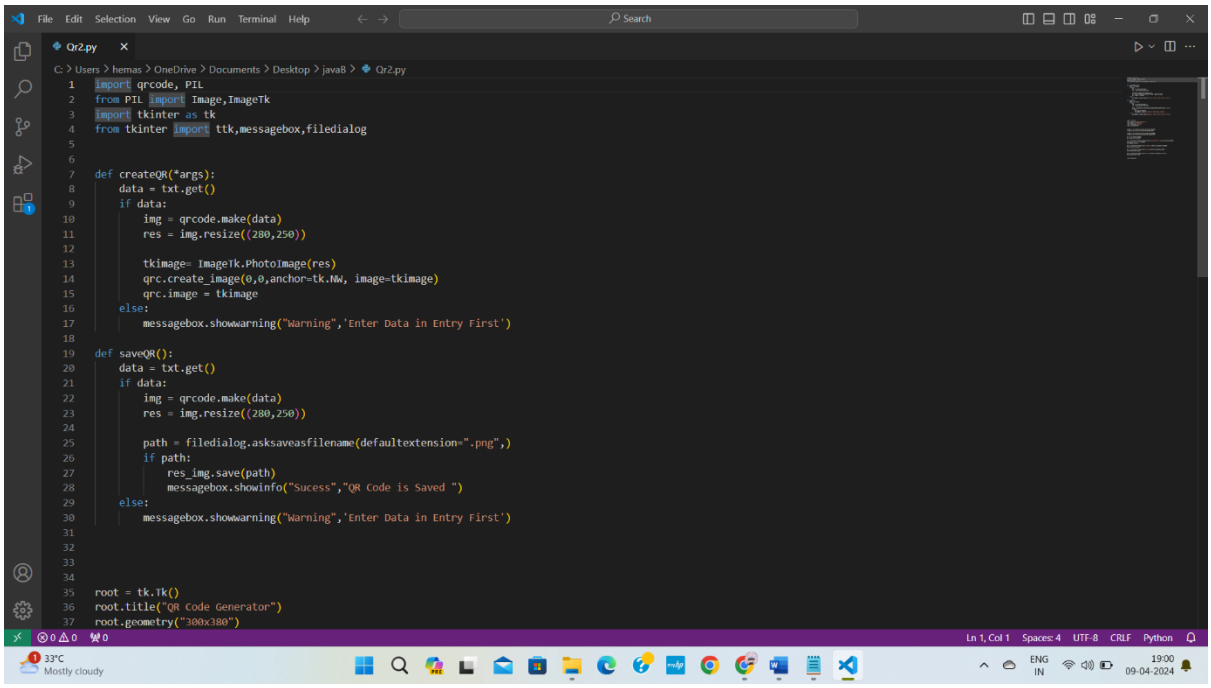
1. **Import Libraries**:
    - `qrcode`: This library is used to generate QR codes.
    - `PIL`: The Python Imaging Library is used for image processing, particularly for resizing the QR code image.
    - `tkinter`: This is the standard GUI (Graphical User Interface) toolkit for Python.
    - `ttk`, `messagebox`, `filedialog`: These are modules from Tkinter used for creating themed widgets, displaying message boxes, and handling file dialogs respectively.
2. **Define Functions**:
    - `createQR(*args)`: This function is called when the user wants to create a QR code. It retrieves the data entered by the user from the Entry widget (`txt`), generates a QR code image using the `qrcode` library, resizes the image using `PIL`, and displays it on the Canvas widget (`qrc`).
    - `saveQR()`: This function is called when the user wants to save the generated QR code image. It performs similar operations as `createQR()` but additionally prompts the user to choose a file path using a file dialog (`filedialog.asksaveasfilename()`) and saves the QR code image to the specified path.
3. **Create GUI**:
    - Create a Tkinter root window (`root`) with a specific title, size, background color, and disabling resizing.
    - Create two frames (`frame1` and `frame2`) within the root window to organize the layout.
    - Inside `frame1`, create a Canvas widget (`qrc`) to display the generated QR code.
    - Inside `frame2`, create an Entry widget (`txt`) for the user to input data, and three Button widgets (`bt1`, `bt2`, `bt3`) for creating the QR code, saving the QR code, and exiting the application respectively.
    - Place all the widgets and frames within the root window at specific positions using the `place()` method.
4. **Main Event Loop**:
    - Start the Tkinter event loop (`root.mainloop()`), which waits for user interactions and responds accordingly.
5. **User Interaction**:

- The user enters data into the Entry widget (`txt`).
- Clicking the "Create" button (`bt1`) or pressing Enter triggers the `createQR()` function, generating and displaying the QR code on the Canvas widget (`qrc`).
- Clicking the "Save" button (`bt2`) triggers the `saveQR()` function, which saves the generated QR code image to a user-specified location.
- Clicking the "Exit" button (`bt3`) exits the application.

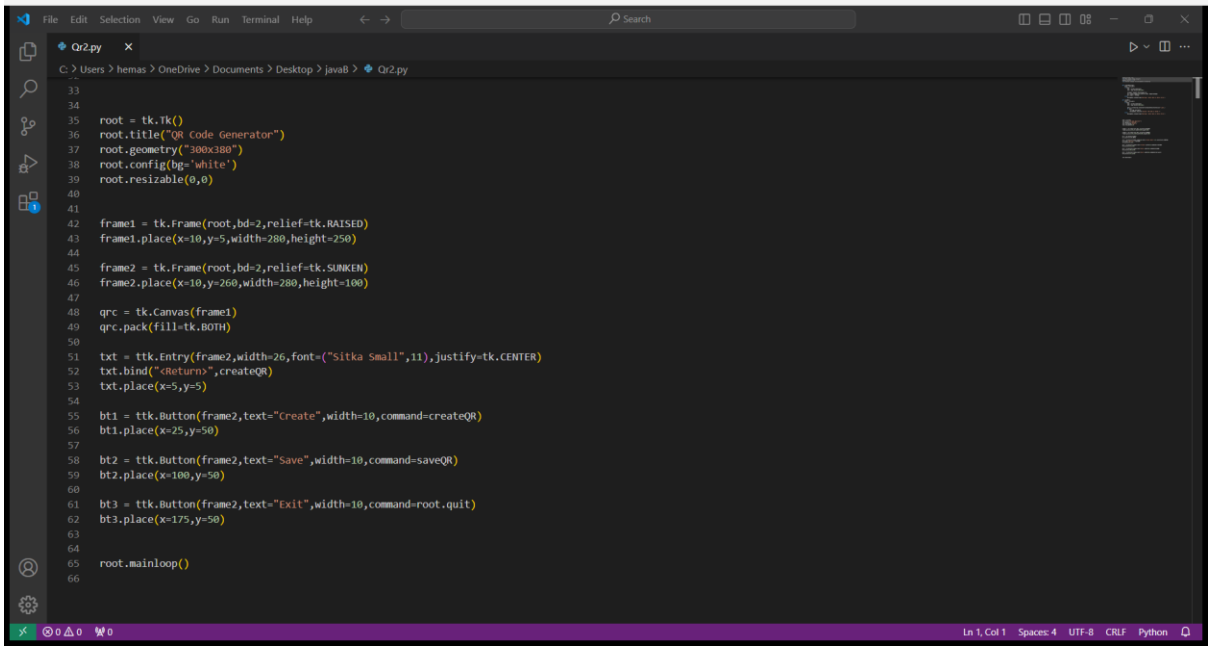
Data Flow Diagram / Process Flow



Code / Program with Supported Screenshots



```python
import qrcode, PIL
from PIL import Image,ImageTk
import tkinter as tk
from tkinter import ttk,messagebox,filedialog


def createQR(*args):
    data = txt.get()
    if data:
        img = qrcode.make(data)
        res = img.resize((280,250))

        tkimage= ImageTk.PhotoImage(res)
        qrc.create_image(0,0,anchor=tk.NW, image=tkimage)
        qrc.image = tkimage
    else:
        messagebox.showwarning("Warning",'Enter Data in Entry First')

def saveQR():
    data = txt.get()
    if data:
        img = qrcode.make(data)
        res = img.resize((280,250))

        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")
```



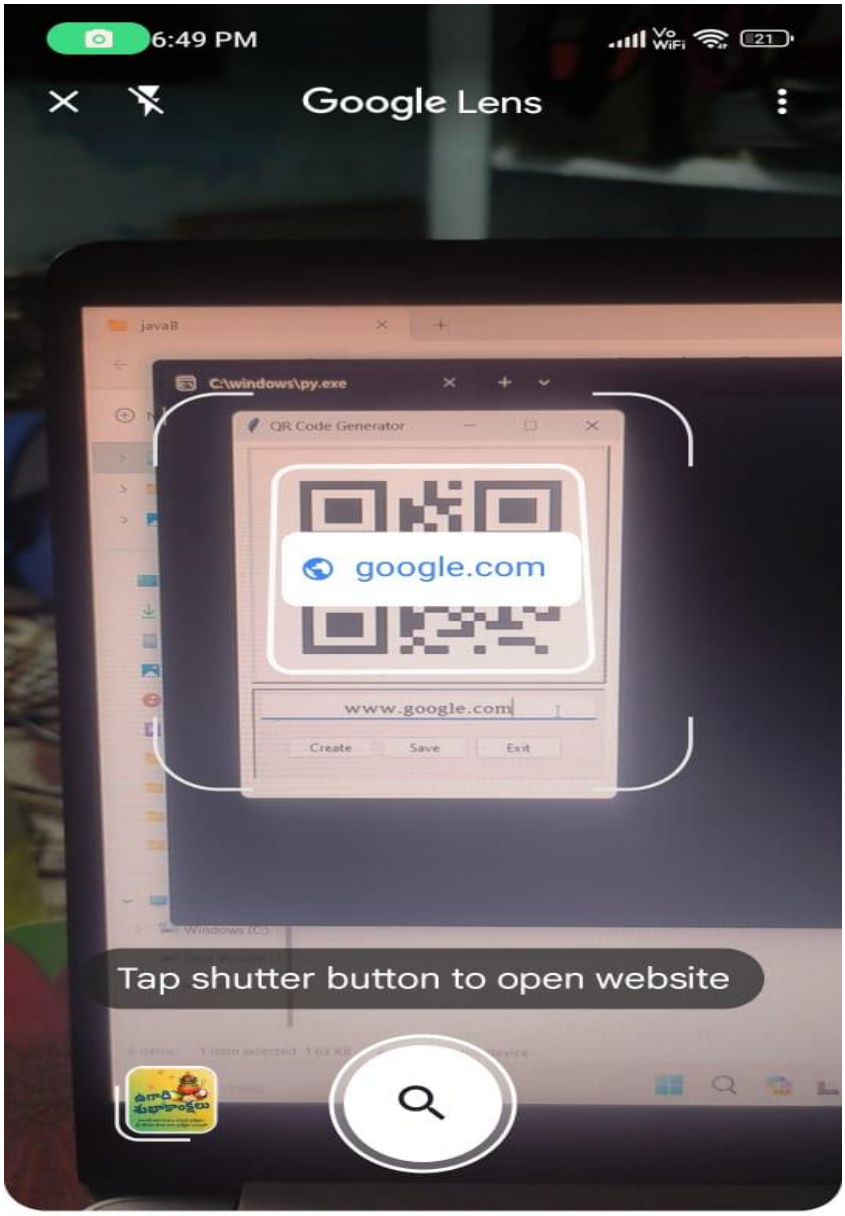
```python


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)

qrc = tk.Canvas(frame1)
qrc.pack(fill=tk.BOTH)

txt = ttk.Entry(frame2,width=26,font=("Sitka Small",11),justify=tk.CENTER)
txt.bind("<Return>",createQR)
txt.place(x=5,y=5)

bt1 = ttk.Button(frame2,text="Create",width=10,command=createQR)
bt1.place(x=25,y=50)

bt2 = ttk.Button(frame2,text="Save",width=10,command=saveQR)
bt2.place(x=100,y=50)

bt3 = ttk.Button(frame2,text="Exit",width=10,command=root.quit)
bt3.place(x=175,y=50)


root.mainloop()

```

Source code:

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


def createQR(*args):
    data = txt.get()
    if data:
        img = qrcode.make(data)
        res = img.resize((280,250))

        tkimage= ImageTk.PhotoImage(res)
        qrc.create_image(0,0,anchor=tk.NW, image=tkimage)
        qrc.image = tkimage
    else:
        messagebox.showwarning("Warning",'Enter Data in Entry First')

def saveQR():
    data = txt.get()
    if data:
        img = qrcode.make(data)
        res = img.resize((280,250))

        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)
```

```
qrc = tk.Canvas(frame1)
qrc.pack(fill=tk.BOTH)

txt = ttk.Entry(frame2,width=26,font=("Sitka Small",11),justify=tk.CENTER)
txt.bind("<Return>",createQR)
txt.place(x=5,y=5)

bt1 = ttk.Button(frame2,text="Create",width=10,command=createQR)
bt1.place(x=25,y=50)

bt2 = ttk.Button(frame2,text="Save",width=10,command=saveQR)
bt2.place(x=100,y=50)

bt3 = ttk.Button(frame2,text="Exit",width=10,command=root.quit)
bt3.place(x=175,y=50)


root.mainloop()
```

Input / Output with Datasets & Supported Screenshots


6:49 PM
Google Lens
javaB
C:\windows\py.exe
QR Code Generator
google.com
www.google.com
Create
Save
Exit
Tap shutter button to open website
Translate
Search
Homework

References

- https://www.geeksforgeeks.org/python-programming-language/
- https://www.javatpoint.com/
- https://www.w3schools.com/python/
- https://www.tutorialspoint.com/index.htm