PYTHON MINOR PROJECT

A Minor Project Report

Submitted to the School of Management- IIT Mandi

in partial fulfillment of requirements of the 1st sem Assignment work for

MBA course

in

Data Science and AI

by

Priya Rani (MB23041)

Vijay Vasisht V (MB23042)

Submitted to: Dr. Manoj Thakur

SCHOOL OF MANAGEMENT

INDIAN INSTITUTE OF TECHNOLOGY, MANDI

20th December'23

0.1 OBJECTIVE

QR CODE GENERATOR

This Project involves developing a Python program that generates QR (Quick Response) codes. The program will take input data (such as a URL, text, or other information) and create a corresponding QR code image that can be scanned by QR code readers.

0.2 TOOLS AND LIBRARIES USED

- **Juypter**: This application was primarily used to write and run our python code.
- **qrcode Library**: This library allows the program to create QR code instances, add data to them, and generate the corresponding image representation. It provides functionalities to specify the QR code version, error correction level, box size, border, and more.

0.3 BACKGROUND

1. QR codes are two-dimensional barcodes that encode information in a matrix of black and white squares.
2. They are widely used for various applications, including encoding URLs, contact information, and more

0.4 PYTHON PROGRAM

To generate QR codes in Python, the following steps are followed; also the application we worked on is Anaconda Jupyter.

- **STEP 1:** Install 'qr code' library as it simplifies the process of creating QR codes from various types of data, such as URLs, text, or other information.
- **STEP 2:** By using **def** function of python we created a code in which **generate_qr_code(data)** takes a parameter data, which represents the information (URL, text, etc.) to be encoded into the QR code. The specific settings under this functon are:

```
In [1]: pip install qrcode

Collecting qrcode
  Downloading qrcode-7.4.2-py3-none-any.whl (46 kB)
     ---------------------------------------- 0.0/46.2 kB ? eta -:--:--
     ---------------------------------------- 46.2/46.2 kB 2.4 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions in c:\users\hp\documents\anaconda\lib\site-packages (from qrcode) (4.7.1)
Collecting pypng (from qrcode)
  Downloading pypng-0.20220715.0-py3-none-any.whl (58 kB)
     ---------------------------------------- 0.0/58.1 kB ? eta -:--:--
     ---------------------------------------- 58.1/58.1 kB ? eta 0:00:00
Requirement already satisfied: colorama in c:\users\hp\documents\anaconda\lib\site-packages (from qrcode) (0.4.6)
Installing collected packages: pypng, qrcode
Successfully installed pypng-0.20220715.0 qrcode-7.4.2
Note: you may need to restart the kernel to use updated packages.
```

Figure 1: Installing **qr code** library

- **version=1**: Specifies the size of the QR code
- **error_correction=qrcode.constants.ERROR_CORRECT_H:** Sets error correction level to high
- **box_size=10:** Defines the size of each box (pixel) in the QR code.
- **border=4:** Determines the size of the border around the QR code.

In the second part, The **add_data()** adds the provided data to the QR code instance and the **make()** command with fit=True is used here to generate the QR code based on the provided data.

```
In [2]: import qrcode

def generate_qr_code(data):
    # Create QR code instance
    qr = qrcode.QRCode(
        version=1,
        error_correction=qrcode.constants.ERROR_CORRECT_H,
        box_size=10,
        border=4,
    )

    # Add data to the QR code
    qr.add_data(data)
    qr.make(fit=True)

    # Create an image from the QR code instance
    img = qr.make_image(fill_color="black", back_color="white")
    return img

def main():
    user_data = input("Enter data for QR code: ")

    qr_image = generate_qr_code(user_data)

    # Display the generated QR code image
    qr_image.show()

if __name__ == "__main__":
    main()

Enter data for QR code: https://chat.openai.com/c/6ad8e51f-f5c1-4220-b6ed-c01d207cac5c
```

Figure 2: **main()** function prompting the user to input the data

- **STEP 3:** The **make_image()** code creates an image representation of the QR code using the specified fill color (**'black'**) for the foreground (data pixels) and **'white'** for the background. This function returns the generated QR code image. For example: Fig.3 is the QR code of the chat gpt link provided by us in the **main()** function.

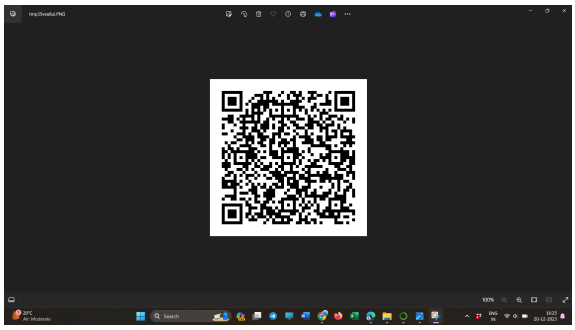

Figure 3: QR code for chat gpt link

0.4.1 FLOW CHART of the code:

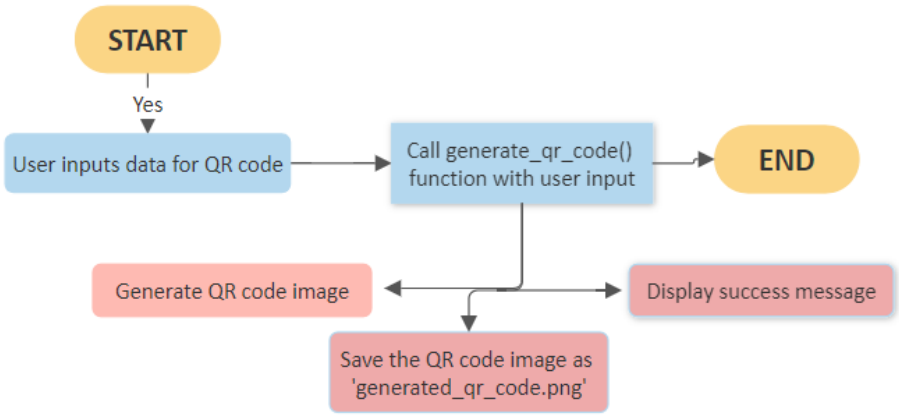

Figure 4: Flow Chart for Figure 2 code

0.5 CONCLUSION

This project allows you to delve into working with libraries, handling user input, and generating QR codes, offering a practical application of encoding data into QR format using Python. Here, the code used is a simple Python program to create a QR code from user input and save it as an image file using the **qrcode** library.