**QR CODE GENERATOR (PATENT)**

**A Project Work Synopsis**

*Submitted in the partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE WITH SPECIALIZATION IN**

**BLOCKCHAIN TECHNOLOGY**

**Submitted by:**

Sanskar Sharma 21BCS6601



**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,**

**PUNJAB**

**September, 2022**

# Abstract

QR i.e." Quick Response" law is a 2D matrix law that's designed by keeping two points under consideration, i.e.it must store large quantities of data as compared to 1D barcodes and it must be decrypted at high speed using any handheld device like phones. QR law provides high data storehouse capacity, fast scanning, omnidirectional readability, and numerous other advantages including, errorcorrection( so that damaged law can also be read successfully) and different types of performances. Different kinds of QR law symbols like totem QR law, translated QR law, iQR Code are also available so that stoner can choose among them according to their need. currently, a QR law is applied in different operation aqueducts related to marketing, security, academics etc. and gain fashionability at a really high pace. Day by day more people are getting apprehensive of this technology and use it consequently. The fashionability of QR law grows fleetly with the growth of smartphone druggies and therefore the QR law is fleetly arriving at high situations of acceptance worldwide. The fashion ability of QR code grows fleetly with the growth of smartphone druggies and therefore the QR law is hastily arriving at high situations of acceptance worldwide. With the wide perpetration of QR law, the protection point of QR law is serious, like data leakage and data revision. This paper emphasizes on the analysis of QR law and its applications. This platform could be used by different security heart associations. Text lines or word system could be translated into QR Code and be read by a mobile device, etc. The work is achieved by the use of python beaker frame which is the main interface for generating the QR Canons.

# Table of Contents

|  |  |
| --- | --- |
| Title Page | i |
| Abstract | ii |
| 1.    Introduction |  |
| 1.1           Problem Definition |  |
| 1.2 Project Overview |  |
| 1.3 Hardware Specification |  |
| 1.4 Software Specification |  |
| 2.    Literature Survey |  |
| 2.1 Existing System |  |
| 2.2 Proposed System |  |
| 2.3 Literature Review Summary |  |
| 3.    Problem Formulation |  |
| 4.    Research Objective |  |
| 5.    Methodologies |  |
| 6.    Experimental Setup |  |
| 7.    Conclusion |  |
| 8.    Tentative Chapter Plan for the proposed work |  |
| 9.    Reference |  |

# 1. INTRODUCTION

## 1.1 Problem Definition

To build the QR Code Generator project using python by following the below steps:-

1. Importing the Modules
2. Creating the main window
3. Taking the input of the text/URL, location to store the QR Code, name of the QR Code and size of the QR Code.

# 1.2 Problem Overview

Since the internet has become more widely used over the past ten years, the majority of Smart- phone manufacturers have included QR Code scanning functionality to their built-in camera programs. This eliminates the need for individuals to download unofficial programs in order to scan a QR Code. The best aspect is that QR Codes may be utilized for a variety of purposes across all industries. They are capable of sharing text, multimedia, and website links. Compared to a bar-code of the same size, a QR code can store far more data. Your end consumers may be redirected to see multimedia content including photographs, videos, and audio as well. Colors and logos can be added to QR Codes to make them more unique. Hence, for their campaigns, marketers frequently employ branded QR Codes. With the audience's attention captured, they may increase brand recognition and increase the number of scans. It provides ease of accessibility. For their scanning activity, QR Codes can be monitored. Even some QR code producers let you keep track of the precise GPS position of scans. All of this information is useful for evaluating the success of your current campaign and planning for future ones. You don't need to set up special hand-held scanners in order to read QR Codes. Your smart- phone is all that is required to scan a QR Code. No need to spend money on any expensive setup. Many manufacturers include QR Codes on the packaging of their goods. The consumers can check the validity of the product on a landing page after scanning this QR Code. For a variety of purposes, several companies in the food and beverage sector are employing QR Codes. Marketing, inventory control, and even anti-counterfeiting efforts are among them. Counterfeiters have historically targeted the automotive industry. Many companies are eager to use QR Code labels as a solution to the problem. The usage of QR Codes in the health-care sector is growing across a wide range of applications, including preventing the sale of counterfeit drugs, checking the validity of vaccination certificates, accessing a patient's medical records quickly. One of the most important aspects of cashless transactions is the use of QR codes. They were already widely used and extremely well-liked in America and Europe, but in recent years, South and East Asia have begun to adopt them as well. In China, the use of QR codes has even overtaken the use of cash and credit cards. One of the biggest accomplishments for QR codes to date must be this. The use of QR codes is rapidly increasing in India, and a new era of cashless transactions is on the horizon for the nation. In recent years, many users have begun to favour these codes above others. The biggest drawback of QR codes is that they are just intended to send users to another web-page or website; they do not independently gather any data.

**1.3 Hardware Specification**:

* Smart-Phone: The QR code generator can be designed to work on any smartphone that supports camera and internet connectivity. The smartphone should have a good quality camera to capture QR code images and process them accurately.
* Tablet: Similar to smartphones, the QR code generator can be designed to work on tablets as well. The tablet should have a good quality camera and internet connectivity to scan and process QR codes.
* Laptop: The QR code generator can also be designed to work on laptops or desktops. In this case, the user would need to connect a webcam or a camera to the computer to scan the QR codes.

**1.4 Software Specification:**

* Python: Python is a high-level programming language that is easy to learn and widely used for various applications, including web development, machine learning, and data analysis. You can use Python to develop a QR code generator using libraries like qrcode, Pillow, and PyQRCode.
* Application Software: There are various application software available in the market for generating QR codes. You can use any of these software, depending on your requirements and budget. Some popular QR code generators include QR-Code Studio, QR Code Generator, and QR Code Monkey.
* C++: C++ is a powerful programming language used for developing system software, games, and other high-performance applications. You can use C++ to develop a QR code generator by using libraries like Zint, QREncode, and qrcodegen.
* IDE like VS Code: An Integrated Development Environment (IDE) is a software application that provides a comprehensive environment for writing, debugging, and testing code. VS Code is a popular IDE used by many developers. You can use VS Code to develop your QR code generator, along with appropriate extensions for Python and C++ development.
* Web Development: You can also develop a QR code generator as a web application using HTML, CSS, and JavaScript. You can use libraries like qrcode.js, QRious, and zxing-js to generate QR codes in the browser.

# 2. LITERATURE SURVEY

* Quick Response code is usually authenticated with the help of the camera of one’s mobile phone. QR codes can easily scanned through mediums like Tablets, laptops and personal computer desktops. The system automatically generates the ID of the user and its password. The characteristic which makes QR codes stand out is they can still be scanned even if they are partially damaged. QR codes are a 2 dimensional printing code on a paper or a screen which makes it pretty vulnerable from various type of cyber-attacks. It can harm your device by unknowingly directing you to a virus contaminated page or website. To avoid this, one must verify the origin of a particular QR code and must have a full understanding of the data type of that particular QR code. There are many attacks involving QR codes as well as their solutions. QR codes are becoming quite popular nowadays because of the rapid increase in smart devices by the normal people around the world. Obviously, 2D QR code is way better and store huge amount of encoded information compared to the old traditional 1D codes. People are using smartphones to do authentications and for this the QR codes are the most ideal way to do it. Many types of QR codes are getting popular nowadays including logo QR code, encrypted QR code, iQR Code etc. 3. METHODOLOGY QR codes are becoming popular day by day in the upcoming generation as it offers way easier authentication that the traditional old fashioned user id and password. QR codes offers many advantages such as greater storage capacity, fast readability, 360 degree reading, small print size, error correction, support for more languages and durability against soil and damage. Many firms who are relatively new in the online business are tend to use these codes instead of normal login process. To fix the QR information / security issue, Xiaohe Cao proposed a safe QR code scheme based on visual cryptography. The security problem of QR code is severe, such as data loss and data tampering as the implementation of QR Code is wide enough. The QR code is spilt into two shared pictures which will be transmitted singly. The development of the two shared pictures is based on the pseudo-random matrix, i.e. the pixels are determined by the pseudo-random matrix values in the two shared pictures. The two images shared can only be stacked to revive the information. Simulation output demonstrates that the picture of the QR code can be masked well and can be efficiently reconditioned. Peter Kieseberg has examined how both automated systems and human interaction can be attacked using QR Codes. As the encoded data is meant to be machine readable only, one cannot differentiate between a legitimate and a harmful corrupted QR code. While automated readers are very much endangered to SQL injections and command injections, individuals might be prone to phishing attacks. Peter Kieseberg contribution is a survey of the QR code as an attack vector, demonstrating different attack plans for the attackers to read and explore their implications.

# PROBLEM FORMULATION

The current problem with the traditional bar-code system is that it can only store limited information and can be easily replicated. This leads to security concerns and limitations in use cases. Therefore, there is a need for a more advanced and secure way of storing information that can be easily scanned by devices.

The solution to this problem is the use of QR (Quick Response) codes, which can store significantly more information than traditional bar-codes and are more secure due to their complex structure. However, there is currently a lack of user-friendly and accessible QR code generator tools that can be used by individuals and businesses.

Therefore, the problem that needs to be addressed is the development of a user-friendly and accessible QR code generator that can be used by anyone, regardless of their technical expertise. The generator should be able to create QR codes for various use cases, including product labeling, event ticketing, and contact information sharing. It should also be able to provide customization options for color, size, and format. Additionally, the generator should be easily accessible through various platforms, including web and mobile applications.

# OBJECTIVES

The objective of the research on the QR code generator project is to identify the current market demand for QR code generation tools and the features that are considered most essential by users. The research should also focus on identifying the target audience for the QR code generator, including individuals, small businesses, and large corporations.

The research will begin by investigating the existing QR code generator tools in the market and their strengths and weaknesses. This will help in identifying the gaps and opportunities for improvement in the current market offerings. For instance, some of the existing QR code generators might not be user-friendly or accessible to people with limited technical expertise. Other generators might not be customizable, which could be a major limitation for businesses that want to use QR codes for branding purposes.

In addition, the technical aspects of QR code generation, including the encoding and decoding process, error correction capabilities, and compatibility with various devices and platforms, will be explored. This will help in ensuring that the QR code generator tool developed meets the required technical standards. For example, the generator should be able to generate QR codes that can be easily scanned by different types of devices such as smartphones, tablets, and barcode scanners.

Furthermore, the research will investigate the potential use cases for QR codes in various industries such as healthcare, retail, logistics, and event management. This will help in identifying the specific needs of each industry and tailoring the QR code generator tool to meet those needs. For example, in the healthcare industry, QR codes can be used for patient identification, medication tracking, and medical record sharing. In the retail industry, QR codes can be used for product labeling, inventory management, and customer engagement.

The research will be conducted through market surveys, focus groups, and interviews with industry experts and potential users. The findings from the research will be used to develop a comprehensive QR code generator tool that is customizable, technically sound, and meets the needs of different industries and users. The ultimate goal is to create a user-friendly and accessible QR code generator tool that can cater to the diverse needs of individuals, small businesses, and large corporations. By identifying the current market demand, understanding the target audience, and exploring potential use cases, the project team can ensure that the QR code generator tool developed is tailored to meet the unique requirements of different industries and users.

# METHODOLOGY

Steganography is a technique used to conceal information in such a way that only the communicating party would know about the existence of the information. In this paper, Steganography is used to embed an encrypted QR Code into an image. The QR Code is encrypted using Triple DES algorithm in order to increase the level of security. The overall approach presented in this paper could potentially be used for secure communication and even for embedding signature/copyright information into an image. 1 Introduction to Steganography Steganography is a technique used to conceal information in such a way that only the communicating party would know about the existence of the information. The literally meaning of the word Steganography is ”covered writing”[6]. This paper provides a brief overview of Steganography and how it can be used to embed information into an image in an efficient manner. The main focus of the paper is to embed an encrypted QR Code into an image. A Cover Image is the one that would have information embedded in it. Then there is the actual information itself, this information could be anything like plain ASCII text or another image. It is also possible to encrypt the information before embedding it, thereby adding another level of security. Least Significant Bit Insertion is one of the most effective and widely used technique in order to achieve Steganography.

In a byte the right most bit is called Least Significant Bit as changing it has the least effect on the value of the Byte. In this technique modifies the last bit of the color components of pixels of the chosen cover image to embed the bit stream corresponding to the information to produce a stego image. Embedding the data:If you consider a 24-bit color image each pixel has three components(Red, Green and Blue) each having 8-bits, thus its possible to store 3 bits of information in every pixel, on contrary if you use a Grey scale image you can only store 1bit of information per pixel. Now in order to embed a letter A whose ASCII value is 65(i.e. 01000001) in an image would require 3 Pixels. Let the 3 pixels be as below Red Green Blue 11000100 10010100 00100100 10000100 11010101 01101100 11111101 11110100 01100100 Before Embedding the Data. After Embedding the data the resulting pixels would be. Red Green Blue 11000100 10010101 00100100 10000100 11010100 01101100 11111100 11110101 01100100 After Embedding the Data. In the above example it can be seen that only 3 bits were flipped. Normally when using this LSB insertion technique on an average 50% of the bits in an image are flipped[8]. Recovering the data: In order to retrieve the information from an image the 8-bit binary equivalent of each RGB color component of pixels is obtained. The LSB of this binary number represents a one bit of the hidden information that was embedded. Each of such bits are then stored in an output file.

One downside of LSB insertion is that it is susceptible to image processing operation such as cropping and compression. Another drawback of this technique is that if the original image was in GIF or BMP file format (i.e. which uses lossless compression technique) and was converted to JPEG file format (i.e. which uses lossy compression technique) and then converted back to the original format then the data in the LSBs would be lost. 2 Introduction to QR Codes Quick Response Code or better know as QR Code is a two dimensional barcode that allow high speed data encoding and decoding capabilities. It was invented by Denso-Wave[3] a Toyota subsidiary in 1994 in order to track the various parts during the vehicle manufacturing. Generally QR Codes are used for distributing small information like URL, a phone number or even small text. The Government of Canada uses QR Codes for efficient and faster processing of the Passport application forms.

A QR Code is embedded on the first page of their application form and the code gets updated as the form is being filled[11]. Also, a Dutch poet Chielie published a collection of 12 poems, QRCode that fits in one sheet of A4 paper. The most effective and efficient way of decoding a QR Code is using a smart phone equipped with a camera and a compatible decoding application. If users don’t have access to smart phones they can access websites such as Xzing[10] in order to decode the QR Codes, all they need to do is upload the QR Code image to the website and they website decode the information and display it to the user. Similarly there are multiple sites such as Kaywa[9] that can be used for generating the QR Code. Figure 1 shown below has a QR Code in it which was generated from Kaywa[9]. Figure 1: Generating a QR Code using Kaywa[9]. There has been increase in the use of QR Codes and the reason for this increase is due to the various features offered by the QR Codes.

One of the most desirable feature is its readability from any direction, also other features provided by QR Codes are high capacity encoding of data, small printout size, Dirt and Damage Resistant and so on[4]. 3 Introduction to Symmetric Key Encryption ”Triple DES” Symmetric Key Encryption techniques are the one in which the keys used for encryption and decryption purpose are the same. In this paper a Triple DES Algorithm[5] is used for the encrypting and decrypting QR Codes. Triple DES is a block cipher, which applies the Data Encryption Standard(DES) cipher algorithm three times to each data block(64-bit long). The advantage of using this algorithm is its ability to have a larger length keys, without designing a new block cipher algorithm. Triple DES utilizes three 56-bits key Key1, Key2andKey3. In order to encrypt the data it performs following sequence of operations, ciphertext = EncryptKey3(DecryptKey2(EncryptKey1(plaintext))) And to decrypt the data it performs following sequence of operations, plaintext = DecryptKey1(EncryptKey2(DecryptKey3(ciphertext))) plaintext = The original information. ciphertext = The encrypted information. EncryptKi = Encrypting using the key Ki . DecryptKi = Decrypting using the key Ki . 4 Understanding the Methodology for concealing the QR Code The Figure 2 shown below provides the big picture of the methodology used for embedding and extracting encrypted QR Codes into a cover image. This process can be divided into two subsections one which elaborates the embedding phase and the other which elaborates the extraction phase. Figure 2: Embedding & Extracting the QR Code using Steganography.

Once the QR Code has been embedded into a cover image, only the communicating parties will be aware about the existence of the QR Code, and even if some how the existence of the QR Code is discovered using Steganalysis it will be very difficult to decrypt it without the key. Thus this methodology can be used in security applications such as applications for exchanging confidential information or for embedding signature/copyright information in an image. Consider a situation where the owner of an image embeds an encrypted QR Code(which represents his ownership information) into the image. This will allow the owner to claim his ownership in case some one copies the image, also it will not be possible for the offender to modify the QR Code since it has been encrypted using a key that only the owner knows. 4.1 The Embedding Phase The first step in the embedding phase is to generate the desired QR Code. There are number of QR Code generator available on the Internet, the one that was used in this paper was Kaywa[9]. All you need to do visit http: // qrcode. kaywa. com and use their online application to generate the QR-Code. The Next step is choosing a cover image.

A complete paper can be written on this topic. Here are few pointers for choosing an effective cover image. Choosing a JPEG image as a cover image is probably not a good idea as JPEG use a lossy compression technique. A better candidate would be an image which has Bitmap(BMP) or Graphics Interchange Format(GIF) format. Apart from the file format it is also required that the chosen image has a palette that contains lots of variances in color as an image with less variance would have uniform patches with same color. If a QR Code is embedded in an image with less variance the distortions caused due to the embedding process would be quite visible. QR Code as described in the Section 2 is a black and white image. Thus any QR Code can be represented as a byte array in Java with ’0’ representing ’Black’ and ’1’ representing ’White’. This byte array is encrypted using a DES Algorithm. Java has a built in library javax.crypto that allows efficient encryption and decryption to be performed. In this paper the DESede ”Triple DES” Algorithm is used for encrypting the byte array representing the QR Code. In the last step of this phase the encrypted byte array is embedded[7] into a cover image. In order to achieve this the individual bytes from the byte array are converted into their equivalent 8-bit binary representation using the shift operators provided in Java and these individual bits are embedded using the Least Significant Bit(LSB) insertion technique as mentioned in Section 1.

At the end of Embedding Phase a Stego Image[7] is obtained which contains an encrypted QR Code embedded in it. 4.2 The Extraction Phase In order to extract a QR Code from a Stego Image following operations are performed. The first step involves going over Pixels and fetching the LSBs from the Red, Green and Blue component of the pixel and arranging them in a group of 8-bits to form a byte which would be part of the byte array. Here it is assumed that the amount of data present in the image is known before hand, however this can easily be implemented dynamically by first embedding the length of the data in the image. Once this encrypted byte array is obtained the next step involved is to decrypt the byte array using the key used during the embedding phase. In this step the DESede ”Triple DES” Algorithm is used for decryption purpose to obtain the original byte array which contained only 1’s and 0’s. This byte array represents the QR Code, which can be painted on a canvas( i.e. 0 corresponds to a Black pixel and 1 corresponds to a White pixel) in order to obtain the QR Code. The last step of this phase involves decoding the QR Code, this can be done using either a smart phone enable with a camera and a compatible decoding application or using the Internet .

# EXPERIMENTAL SETUP

1.Use a frame with a call-to-action (CTA) text. By adding a frame for your QR Code to "stay on," it can add more prominence and draw attention to the QR Code. Select from a variety of frames with a Scan Me text that urges people to take the next step when they encounter your QR Code.

2.Change the shape of your QR Code. Changing the shape means that you're changing the pattern of the small, pixelate squares that represents the QR Code's data. You can differentiate your QR Code from your competitors just by making small changes to its appearance.

3.Pick and choose your own color. Using our color picker, you can select from 16,777,216 HEX colors using the #RRGGBB notation. It is impossible to design a QR Code with a color you dislike, so feel free to experiment. Just one piece of advice: Stick with darker shades to ensure that your QR Code remains scan-able.

4. Add your logo. For some finishing touches, you can choose a SCAN ME logo for the center of your QR Code or upload your own when you sign up. By using your logo, you can foster brand awareness and establish trust among your audience.

# FUTURE SCOPE

Integration with other software: Integrating the QR code generator with other software can enhance its usability and expand its functionality. For instance, if it is integrated with a document management system, the QR code generator could automatically generate and attach QR codes to documents, making it easy for users to scan and retrieve information about the document. This integration would save users time by eliminating the need for manual QR code generation and attachment, thus streamlining the document management process.

Improved error handling: Error handling is a crucial aspect of any software application, and the QR code generator is no exception. Improving the error handling in the current version of the QR code generator would enable it to provide more detailed error messages, making it easier for users to understand what went wrong and how to fix the issue. This enhancement would enhance the user experience and help users generate accurate QR codes.

QR code scanning: Integrating QR code scanning functionality into the application would allow users to read and extract information from existing QR codes. This feature would make the application even more versatile and enable users to use the QR code generator to scan codes generated by other applications or sources. QR code scanning functionality could be integrated into the mobile application or the desktop application, depending on the target audience and use cases.

Customizable QR codes: Adding customizable QR codes to the application would allow users to modify the appearance of the QR codes they generate. For instance, users could add a logo or change the color scheme of the QR code to match their branding or personal preferences. This feature would add a level of personalization and customization to the application, making it even more attractive to users.

Mobile application: Developing a mobile application for the QR code generator would make it even more convenient for users to generate and scan QR codes on the go. A mobile application would allow users to generate and scan QR codes using their mobile devices, making it easier for them to use the application wherever they are. Additionally, a mobile application could integrate QR code scanning functionality, allowing users to scan existing QR codes from other sources. The mobile application could be developed for both iOS and Android platforms to reach a wider audience

# CONCLUSION

In conclusion, the paper highlights the increasing use of QR codes in various industries and the importance of their ability to encode more information than traditional barcodes. The paper also discusses the ease and convenience of creating QR codes through a web browser, which allows users to generate codes for various purposes such as websites, emails, business cards, and print ads.

The proposed method for generating QR codes was developed using open-source software such as Libqrencode, Drupal, and Ubuntu, which makes it more accessible for users to utilize without any licensing costs. The experimental results demonstrate the successful and accurate generation of QR codes, indicating that the proposed method is an effective and collaborative tool for generating QR codes.

Overall, the paper highlights the usefulness and versatility of QR codes and the potential for further developments such as integration with other software, improved error handling, and customization options. The proposed method provides a free and accessible solution for generating QR codes, which can benefit various industries and individuals in creating effective and efficient marketing strategies.

## 8. TENTATIVE CHAPTER PLAN FOR THE PROPOSED WORK

1.Importing the modules  
2. Creating the main window  
3. Taking the input of the text/URL, location to store the QR code, name of the QR code and the size of the QR code  
4. Writing the function to generate and save the QR Code

1. Importing the modules

The first step is to import the qrcode and the tkinter module. We use the messagebox in the tkinter module to show the pop up messages.

Code:

import qrcode

from tkinter import \*

from tkinter import messagebox

2. Creating the main window

Next, we create the main window with title, size and color.

#Creating the window

Code:

wn = Tk()

wn.title('DataFlair QR Code Generator')

wn.geometry('700x700')

wn.config(bg='SteelBlue3')

3. Taking the inputs

Now, we take the inputs from the user to create the QR Code. We take the following inputs:

1. Text/URL as Entry() named as ‘text’

2. Location to save the QR Code as Entry() named as ‘loc’

3. Name of the QR Code image when saved in the device as Entry() named as ‘name’

4. Size of the QR Code to be generated as Entry() named ‘size’. In this the user has to give the size in the range 1-40. 1 being the smallest size of 21×21.

Then we create a button when clicked generates the QR Code and saves it by executing the generateCode() function.

#Label for the window

headingFrame = Frame(wn,bg="azure",bd=5)

headingFrame.place(relx=0.15,rely=0.05,relwidth=0.7,relheight=0.1)

headingLabel = Label(headingFrame, text="Generate QR Code with DataFlair", bg='azure', font=('Times',20,'bold'))

headingLabel.place(relx=0,rely=0, relwidth=1, relheight=1)

#Taking the input of the text or URL to get QR code

Frame1 = Frame(wn,bg="SteelBlue3")

Frame1.place(relx=0.1,rely=0.15,relwidth=0.7,relheight=0.3)

label1 = Label(Frame1,text="Enter the text/URL: ",bg="SteelBlue3",fg='azure',font=('Courier',13,'bold'))

label1.place(relx=0.05,rely=0.2, relheight=0.08)

text = Entry(Frame1,font=('Century 12'))

text.place(relx=0.05,rely=0.4, relwidth=1, relheight=0.2)

#Getting input of the location to save QR Code

Frame2 = Frame(wn,bg="SteelBlue3")

Frame2.place(relx=0.1,rely=0.35,relwidth=0.7,relheight=0.3)

label2 = Label(Frame2,text="Enter the location to save the QR Code: ",bg="SteelBlue3",fg='azure',font=('Courier',13,'bold'))

label2.place(relx=0.05,rely=0.2, relheight=0.08)

loc = Entry(Frame2,font=('Century 12'))

loc.place(relx=0.05,rely=0.4, relwidth=1, relheight=0.2)

#Getting input of the QR Code image name

Frame3 = Frame(wn,bg="SteelBlue3")

Frame3.place(relx=0.1,rely=0.55,relwidth=0.7,relheight=0.3)

label3 = Label(Frame3,text="Enter the name of the QR Code: ",bg="SteelBlue3",fg='azure',font=('Courier',13,'bold'))

label3.place(relx=0.05,rely=0.2, relheight=0.08)

name = Entry(Frame3,font=('Century 12'))

name.place(relx=0.05,rely=0.4, relwidth=1, relheight=0.2)

#Getting the input of the size of the QR Code

Frame4 = Frame(wn,bg="SteelBlue3")

Frame4.place(relx=0.1,rely=0.75,relwidth=0.7,relheight=0.2)

label4 = Label(Frame4,text="Enter the size from 1 to 40 with 1 being 21x21: ",bg="SteelBlue3",fg='azure',font=('Courier',13,'bold'))

label4.place(relx=0.05,rely=0.2, relheight=0.08)

size = Entry(Frame4,font=('Century 12'))

size.place(relx=0.05,rely=0.4, relwidth=0.5, relheight=0.2)

#Button to generate and save the QR Code

button = Button(wn, text='Generate Code',font=('Courier',15,'normal'),command=generateCode)

button.place(relx=0.35,rely=0.9, relwidth=0.25, relheight=0.05)

#Runs the window till it is closed manually

wn.mainloop()

4. Creating the function to generate QR code and save it

Finally, we create the function to generate the code that runs on clicking the button. In this,

1. First we create the QRCode object with the version/size that user gave as input in the size() entry

2. Then we add the text that we need to encode by getting from the entry ‘text’

3. Then we get the QR code and save it in the directory that user gave as input

4. After this, we show the pop up message to confirm the user that the image is saved

#Function to generate the QR code and save it

def generateCode():

#Creating a QRCode object of the size specified by the user

qr = qrcode.QRCode(version = size.get(),

box\_size = 10,

border = 5)

qr.add\_data(text.get()) #Adding the data to be encoded to the QRCode object

qr.make(fit = True) #Making the entire QR Code space utilized

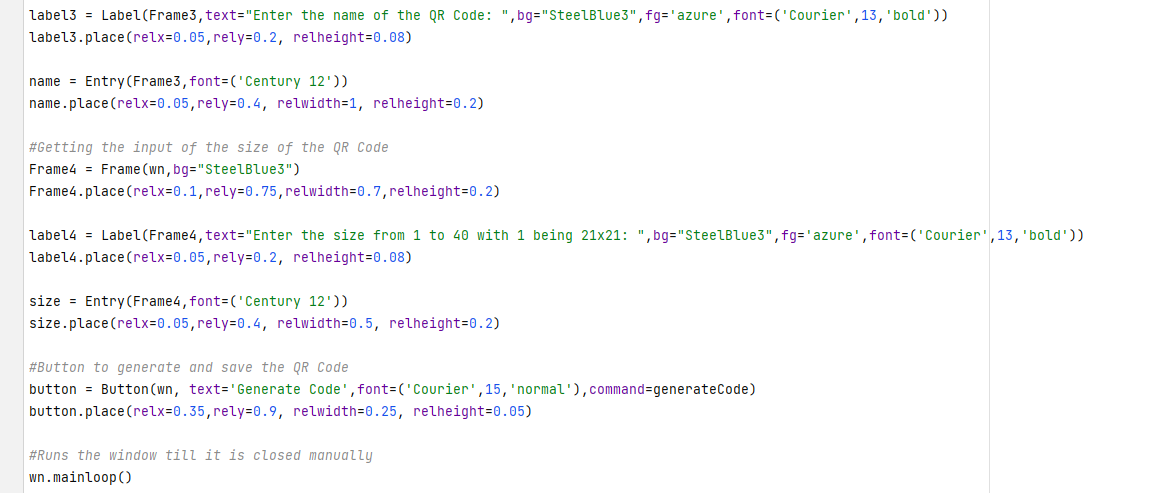
img = qr.make\_image() #Generating the QR Code

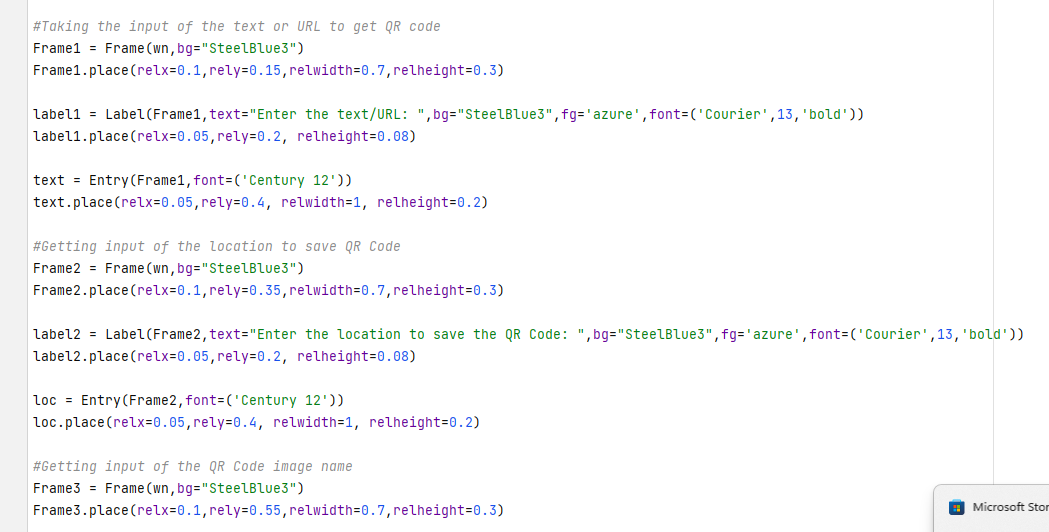
fileDirec=loc.get()+'\\'+name.get() #Getting the directory where the file has to be save

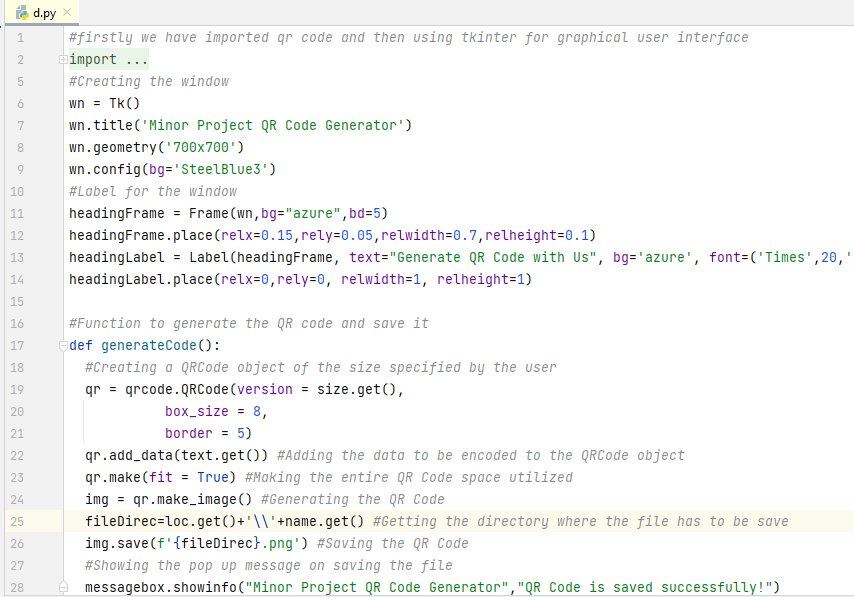
img.save(f'{fileDirec}.png') #Saving the QR Code

#Showing the pop up message on saving the file

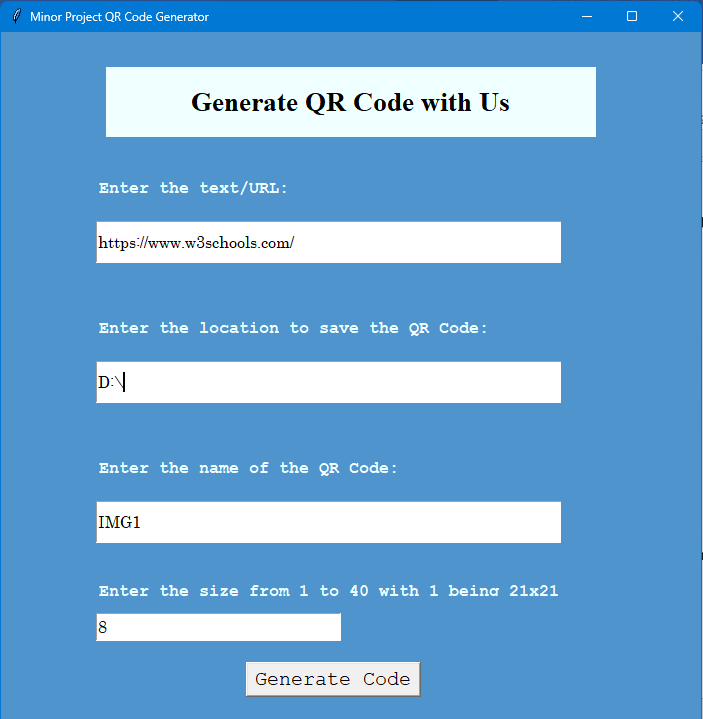
messagebox.showinfo("DataFlair QR Code Generator","QR Code is saved successfully!")



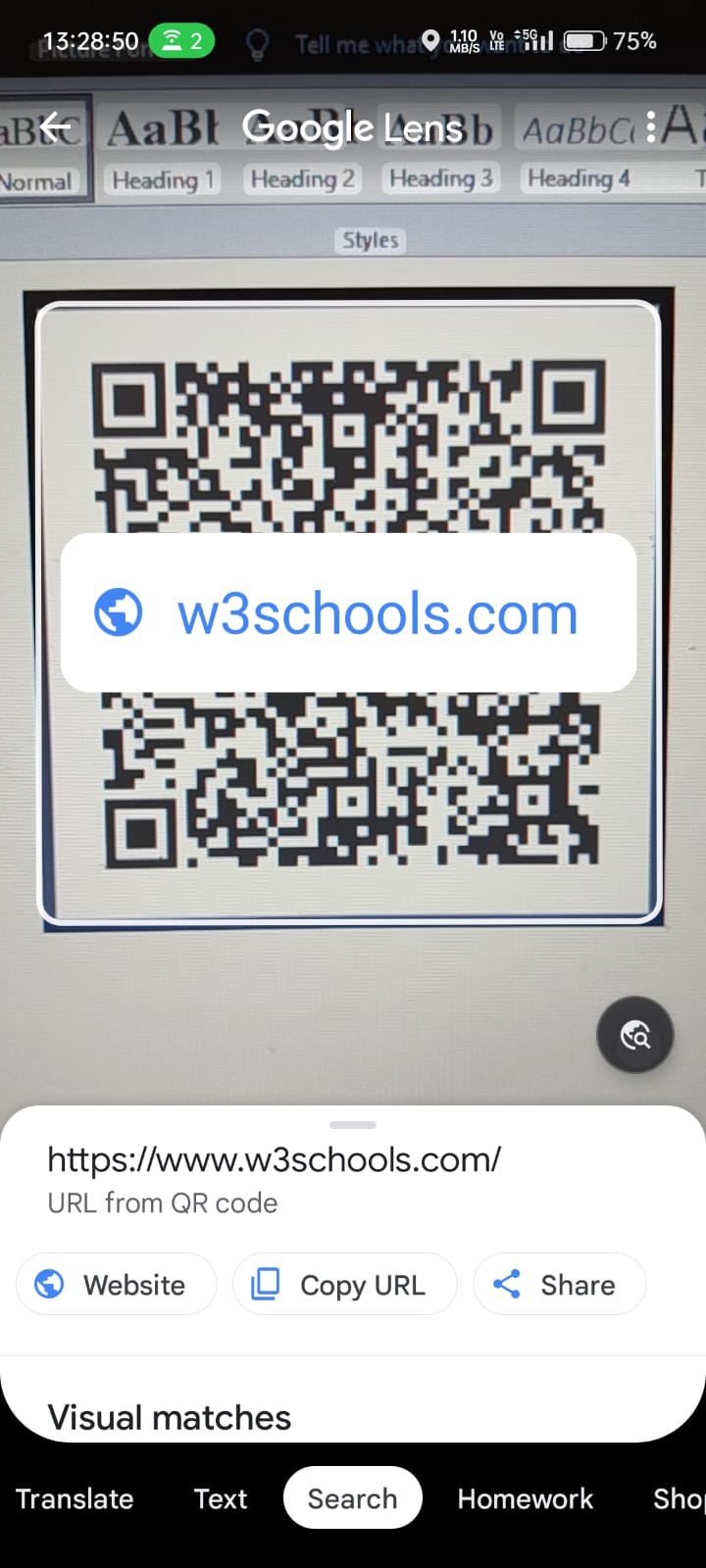




**OUTPUT:**

****

****

****

**REFERENCES**

1.Aidong Sun, Yan Sun and Caixing Liu, “The QR-code reorganization in illegible snapshots taken by mobile phones,” International Conference on Computational Scienceand its Applications, 2007. ICCSA 2007, pp. 532-538, 26-29 Aug. 2007.

2."AVG Cautions: Beware of Malicious QR Codes". PC World. 28 June 2011. Archivedfrom the original on 7 September 2012. Retrieved 31 August 2011.

3."Barcode Scanner". 1 June 2011. Archived from the original on 15 September 2012.Retrieved 31 August 2011.

4.Chen, Rongjun; Yu, Yongxing; Xu, Xiansheng; Wang, Leijun; Zhao, Huimin; Tan,Hong-Zhou (11 December 2019). "Adaptive Binarization of QR Code Images for FastAutomatic Sorting in Warehouse Systems". Sensors. 1924): 5466.Bibcode:2019Senso..19.5466C. doi: 10.3390/s19245466. PMC 6960674. PMID31835866.

5."From Japanese auto parts to ubiquity: A look at the history of QR codes". Mainichi DailyNews. 9 November 2021. Retrieved 11 November 2021.

6."Getting Started with Infographics (deprecated)". google-developers.appspot.com. 26May 2015. Archived from the original on 29 June 2016. Retrieved 27 May 2016.

7.Gupta, Kishor Datta; Ahsan, Md Manjurul; Andrei, Stefan (January 2018). "Extending the Storage Capacity And Noise Reduction of a Faster QR-Code". Brain Broad Research in Artificial Intelligence and Neuroscience. 9 (1): 59-71.

8.Hajra Shannon (18 October 2010). "Form Meets Function: Functionality and Form of QR Codes". Retrieved 29 July 2010.

9. ISO/IEC 18()04:2000. Information technology-Automatic identification and data capture

techniques-Bar code Symbology-QR Code, 2000.

10. Jenny Lee (4 January 2012). "Tesco's cool QR code advertising campaign". Archived

from the original on 3 June 2016. Retrieved 26 May 2016.

11.Joe Waters. "How to Use the Top QR Code Generators". Dummies.com. Archived from the original on I I September 2017. Retrieved 5 June 2017.

12. Kastrenakes, Jacob (29 September 2020). "Square launches QR codes that let you order from your table at a restaurant". Verge.com. VOX Media LLC. Retrieved 24 November 2020.

13. Kim, Yeong Gug; Woo, Eunju (l July 2016). "Consumer acceptance of a quick response (QR) code for the food traceability system: Application of an extended technologyacceptance model (TAM)". Food Research International. 85: 266—272.doi: 10.1016/j.foodres.2016.05.002. ISSN 0963-9969. PMID 29544844.

13. Küch, Oliver (26 June 2020). "Color barcode becomes ISO standard" (Press release).Darmstadt: Fraunhofer Institute for Secure Information Technology. Retrieved 17 February 2021.

14.Novak, Asami (23 March 2008). "Japanese Gravestones Memorialize the Dead With QR Codes". Wired. Archived from the original on 15 February 2013. Retrieved 8 May 2013.