INDUSTRIAL TRAINING REPORT

QR GENERATOR AND QR DECODER

Submitted in partial fulfillment of the

Requirements for the award of

Degree of Bachelor of Technology in Computer Science Engineering

Submitted By

Name: DANISH KHAN

University Roll No. 1814310071

SUBMITTED TO:

Department of Computer Science & Engineering
IMS ENGINEERING COLLEGE
GHAZIABAD (UP).

DECLARATION

I hereby declare that the Industrial Training Report entitled ("QR Generator and Decoder program") is an authentic record of my own work as requirements of Industrial Training during the period from 06.06.2019 to 15.07.2019 for the award of degree of B.Tech. (Computer Science & Engineering), IMS ENGINEERING COLLEGE, GZB , under the guidance of Mukesh Kumar Sir.

(Signature of student)
()
(ROLL NO - 1814310071)

Date: _____

Head of Department
(Signature and Seal)

ACKNOWLEDGEMENT

An endeavor is not complete and successful till the people who make it possible are given duecredit for making it possible. I take this opportunity to thank all those who have made theendeavor of mine successful for me and for all.

I take the opportunity to express my gratitude to all of them who in some or the other way helpedme to accomplish this project. The study cannot be completed without their guidance, assistance,inspiration and co-operation.

I would like to express my deep sense of gratitude to my respected project guide Mr. Mayur Varshney for his support & cooperation.

Secondly I would like to thank my close friends & well wishers who were there for me whenever required.

Last but not the least WWW which provided with all the required data & information.

About Company/Industry/Institute

Red Hat, Inc. is an American multinational software company providing open-source software products to the enterprise community. Founded in 1993, Red Hat has its corporate headquarters in Raleigh, North Carolina with other offices worldwide. It became a subsidiary of IBM on July 9, 2019.

Red Hat has become associated to a large extent with its enterprise operating system Red Hat Enterprise Linux. With the acquisition of open-source enterprise middleware vendor JBoss, Red Hat also offers Red Hat Virtualization (RHV), an enterprise virtualization product. Red Hat provides storage, operating system platforms, middleware, applications, management products, and support, training, and consulting services.

Red Hat creates, maintains, and contributes to many free software projects. It has acquired several proprietary software product codebases through corporate mergers and acquisitions and has released such software under open-source licenses. As of March 2016, Red Hat is the second largest corporate contributor to the Linux kernel version 4.14 after Intel.

On October 28, 2018, IBM announced its intent to acquire Red Hat for $34 billion. The acquisition closed on July 9, 2019. Red Hat's lead financial adviser in the transaction was Guggenheim Securities

CERTIFICATE

IMS ENGINEERING COLLEGE
GHAZIABAD

Department of Information Technology

CERTIFICATE OF PARTICIPATION

This is to certify that

DANISH KHAN

has Completed the 6 week internship on "*RedHat Linux & Python with Machine Learning*" organized by Department of Information Technology, IMS Engineering College & COSS INDIA (Red Hat Academy Partner) on **15 July 2019.**

Redhat Academy Partner

COSS

RED HAT ACADEMY

Coordinator
Red Hat Academy,
COLLEGE AUTHORITY.

Authorised Signatory
Complete Open Source Solutions
www.cossindia.net

Serial #: COSS/RHA/IMSEC/18-19/005

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

QR code (abbreviated from **Quick Response code**) is the trademark for a type of matrix barcode (or two-dimensional barcode) first designed in 1994 for the automotive industry in Japan. A barcode is a machine-readable optical label that contains information about the item to which it is attached. In practice, QR codes often contain data for a locator, identifier, or tracker that points to a website or application. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte/binary, and kanji) to store data efficiently; extensions may also be used.

The Quick Response system became popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC barcodes. Applications include product tracking, item identification, time tracking, document management, and general marketing.

A QR code consists of black squares arranged in a square grid on a white background, which can be read by an imaging device such as a camera, and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data is then extracted from patterns that are present in both horizontal and vertical components of the image.

HISTORY

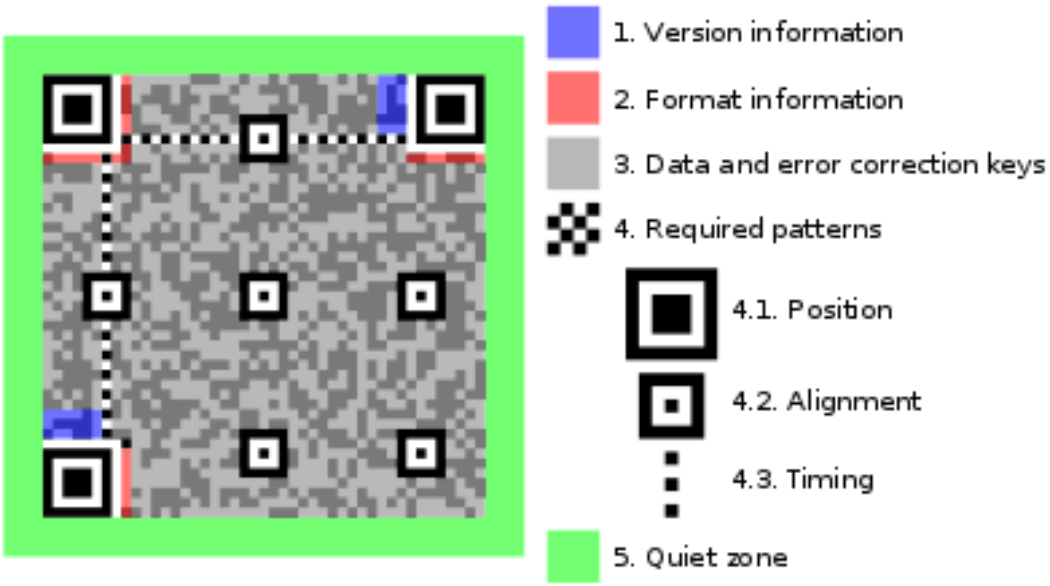
The QR code system was invented in 1994 by the Japanese company Denso Wave. Its purpose was to track vehicles during manufacturing; it was designed to allow high-speed component scanning. QR codes are now used in a much broader context, including both commercial tracking applications and convenience-oriented applications aimed at mobile-phone users (termed mobile tagging). QR codes may be used to display text to the user, to add a vCard contact to the user's device, to open a Uniform Resource Identifier (URI), to connect to a wireless network, or to compose an email or text message. There are a great many QR code generators available as software or as online tools. The QR code has become one of the most-used types of two-dimensional code.

STANDARDS

There are several standards that cover the encoding of data as QR codes:

- October 1997 – AIM (Association for Automatic Identification and Mobility) International
- January 1999 – JIS X 0510
- June 2000 – ISO/IEC 18004:2000 *Information technology – Automatic identification and data capture techniques – Bar code symbology – QR code* (now withdrawn)
  Defines QR code models 1 and 2 symbols.
- 1 September 2006 – ISO/IEC 18004:2006 *Information technology – Automatic identification and data capture techniques – QR code 2005 bar code symbology specification* (now withdrawn)
  Defines QR code 2005 symbols, an extension of QR code model 2. Does not specify how to read QR code model 1 symbols, or require this for compliance.
- 1 February 2015 – ISO/IEC 18004:2015 *Information – Automatic identification and data capture techniques – QR Code barcode symbology specification*
  Renames the QR Code 2005 symbol to QR Code and adds clarification to some procedures and minor corrections.

At the application layer, there is some variation between most of the implementations.
Japan's NTT DoCoMo has established de facto standards for the encoding of URLs, contact information, and several other data types.[10] The open-source "ZXing" project maintains a list of QR code data types.



USES

QR codes have become common in consumer advertising. Typically, a smartphone is used as a QR code scanner, displaying the code and converting it to some useful form (such as a standard URL for a website, thereby obviating the need for a user to type it into a web browser). QR code has become a focus of advertising strategy, since it provides a way to access a brand's website more quickly than by manually entering a URL. Beyond mere convenience to the consumer, the importance of this capability is that it increases the conversion rate: the chance that contact with the advertisement will convert to a sale. It coaxes interested prospects further down the conversion funnel with little delay or effort, bringing the viewer to the advertiser's website immediately, where a longer and more targeted sales pitch may lose the viewer's interest.

Although initially used to track parts in vehicle manufacturing, QR codes are used over a much wider range of applications. These include commercial tracking, entertainment and transport ticketing, product and loyalty marketing and in-store product labeling. Examples of marketing include where a company's discounted and percent discount can be captured using a QR code decoder which is a mobile app, or storing a company's information such as address and related information alongside its alpha-numeric text data as can be seen in Yellow Pages directory.

They can also be used in storing personal information for use by organizations. An example of this is Philippines National Bureau of Investigation (NBI) where NBI clearances now come with a QR code. Many of these applications target mobile-phone users (via mobile tagging). Users may receive text, add a vCard contact to their device, open a URL, or compose an e-mail or text message after scanning QR codes. They can generate and print their own QR codes for others to scan and use by visiting one of several pay or free QR code-generating sites or apps. Google had an API, now deprecated, to generate QR codes, and apps for scanning QR codes can be found on nearly all smartphone devices

MOBILE OPERATING SYSTEM

QR codes can be used on various mobile device operating systems. iPhones running on iOS 11 and higher and some Android devices can natively scan QR codes without downloading an external app. The camera app is able to scan and display the kind of QR code (only on iPhone) along with the link (both on Android and iPhone). These devices support URL redirection, which allows QR codes to send metadata to existing applications on the device. Many paid or free apps are available with the ability to scan the codes and hard-link to an external URL.

CONSTRUCTION SITES

QR Codes application in the field of Construction has already been started in India which helps in saving time, paper and energy. QR Code usage in Safety Vertical of Construction industry aids in usage of Resources, allocation of time and usage of critical information. Dynamic nature of Construction Industry requires the need for dynamic technologies and QR Code aids in developing that technology for future usages.

URLs

URLs aided marketing conversion rates even in the pre-smartphone era, but during those years faced several limitations: ad viewers usually had to type the URL and often did not have a web browser in front of them when they first viewed the ad. The chances were high that they would forget to visit the site later, not bother to type a URL, or forget what URL to type. Semantic URLs decreased these risks but did not eliminate them. With the advent of smartphones the issue of viewers not being able to access a website immediately has become less of an issue, however the trouble of typing in URLs still remained and thus QR codes were utilised in order to allow redirecting to URLs for instant access. Several QR Code Generators offer an additional feature - Dynamic QR Codes. Dynamic QR Codes can be edited over and over again since they use a placeholder URL which also makes it easier and faster to scan than their counterpart - Static QR Codes

QR CODE PAYMENT

QR codes can be used to store bank account information or credit card information, or they can be specifically designed to work with particular payment provider applications. There are several trial applications of QR code payments across the world. In developing countries like India and China, QR code payment is a very popular and convenient method of making payments.

In November 2012, QR code payments were deployed on a larger scale in the Czech Republic when an open format for payment information exchange — a Short Payment Descriptor — was introduced and endorsed by the Czech Banking Association as the official local solution for QR payments. In 2013, the European Payment Council provided guidelines for the EPC QR code enabling SCT initiation within the Eurozone.

WIFI NETWORK LOGIN

By specifying the SSID, encryption type, password/passphrase, and if the SSID is hidden or not, mobile device users can quickly scan and join networks without having to manually enter the data. Note that this technique is valid for specifying only static SSID passwords (i.e. PSK); dynamic user credentials (i.e. Enterprise/802.1x) cannot be encoded in this manner.

The format of the encoded string is:

```
WIFI:S:<SSID>;T:<WPA|WEP|>;P:<password>;H:<true|false|>;
```

Order of fields does not matter. Special characters """ (quotation mark), ";" (semicolon), "," (comma) and ":" (colon) should be escaped with a backslash ("\") as in MECARD encoding. For example, if an SSID were `"foo;bar\baz"`, with quotation marks part of the literal SSID name itself, this would be encoded as: `WIFI:S:\"foo\;bar\\baz\";;`

As of January 2018, iPhones have this feature built into the camera app under iOS 11.x. Android users may have the feature built into one of the device's stock apps (e.g. Samsung Galaxy S8/S8+/Note8 users can launch the stock browser, tap the browser's 3-dot menu, then choose "Scan QR code") or can install one of several available free apps such as "Barcode Scanner" or "QR Droid" to perform the QR Wi-Fi join

DESIGN

Unlike the older, one-dimensional barcodes that were designed to be mechanically scanned by a narrow beam of light, a QR code is detected by a 2-dimensional digital image sensor and then digitally analyzed by a programmed processor. The processor locates the three distinctive squares at the corners of the QR code image, using a smaller square (or multiple squares) near the fourth corner to normalize the image for size, orientation, and angle of viewing. The small dots throughout the QR code are then converted to binary numbers and validated with an error-correcting algorithm

STORAGE

The amount of data that can be stored in the QR code symbol depends on the datatype (*mode*, or input character set), version (1, ..., 40, indicating the overall dimensions of the symbol, i.e. 4 × version number + 17 dots on each side), and error correction level. The maximum storage capacities occur for version 40 and error correction level L (low), denoted by 40-L:

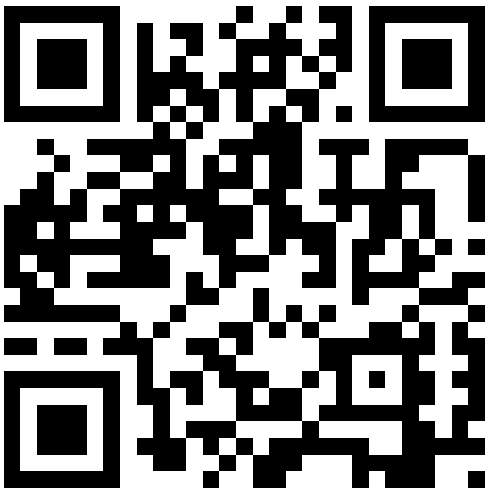
Maximum character storage capacity (40-L)
***character* refers to individual values of the input mode/datatype**

Input mode	Max. characters	Bits/char.	Possible characters, default encoding
Numeric only	7,089	3⅓	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Alphanumeric	4,296	5½	0–9, A–Z (upper-case only), space, $, %, *, +, -, ., /, :
Binary/byte	2,953	8	ISO 8859-1
Kanji/kana	1,817	13	Shift JIS X 0208

Here are some sample QR code symbols:

Version 1 (21×21). Content: "Ver1"

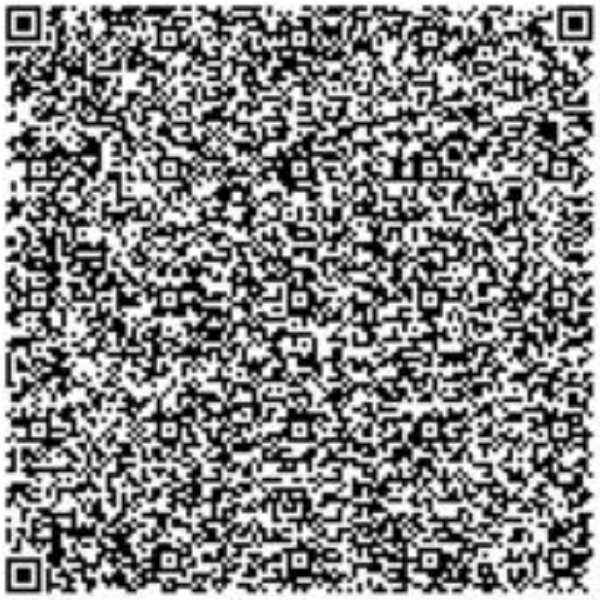
Version 2 (25×25). Content: "Version 2"



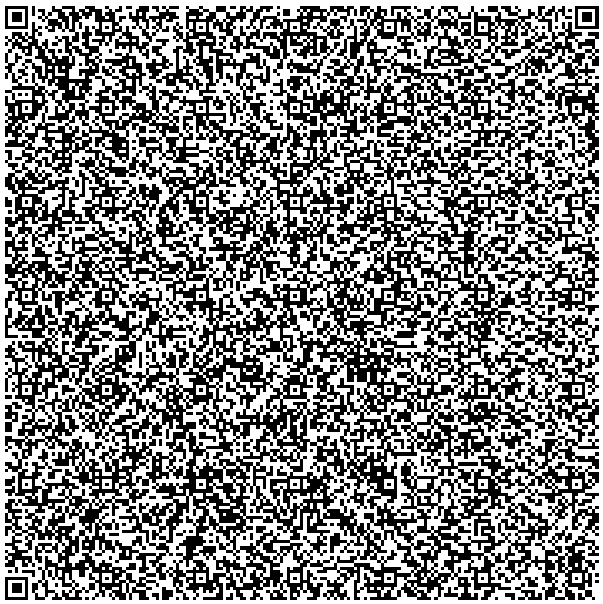
Version 3 (29×29). Content: "Version 3 QR Code"

Version 4 (33×33). Content: "Version 4 QR Code, up to 50 char"

Version 10 (57×57). Content: "VERSION 10 QR CODE, UP TO 174 CHAR AT H LEVEL, WITH 57X57 MODULES AND PLENTY OF ERROR-CORRECTION TO GO AROUND. NOTE THAT THERE ARE ADDITIONAL TRACKING BOXES"



Version 25 (117×117 enlarged to 640x640)



Version 40 (177×177). Content: 1,264 characters of ASCII text describing QR codes

ERROR CORRECTION

Codewords are 8 bits long and use the Reed–Solomon error correction algorithm with four error correction levels. The higher the error correction level, the less storage capacity. The following table lists the approximate error correction capability at each of the four levels:

Level L (Low)	7% of codewords can be restored.
Level M (Medium)	15% of codewords can be restored.
Level Q (Quartile)	25% of codewords can be restored.
Level H (High)	30% of codewords can be restored.

In larger QR symbols, the message is broken up into several Reed–Solomon code blocks. The block size is chosen so that at most 15 errors can be corrected in each block; this limits the complexity of the decoding algorithm. The code blocks are then interleaved together, making it less likely that localized damage to a QR symbol will overwhelm the capacity of any single block.

Due to error correction, it is possible to create artistic QR codes that still scan correctly, but contain intentional errors to make them more readable or attractive to the human eye, as well as to incorporate colors, logos, and other features into the QR code block.

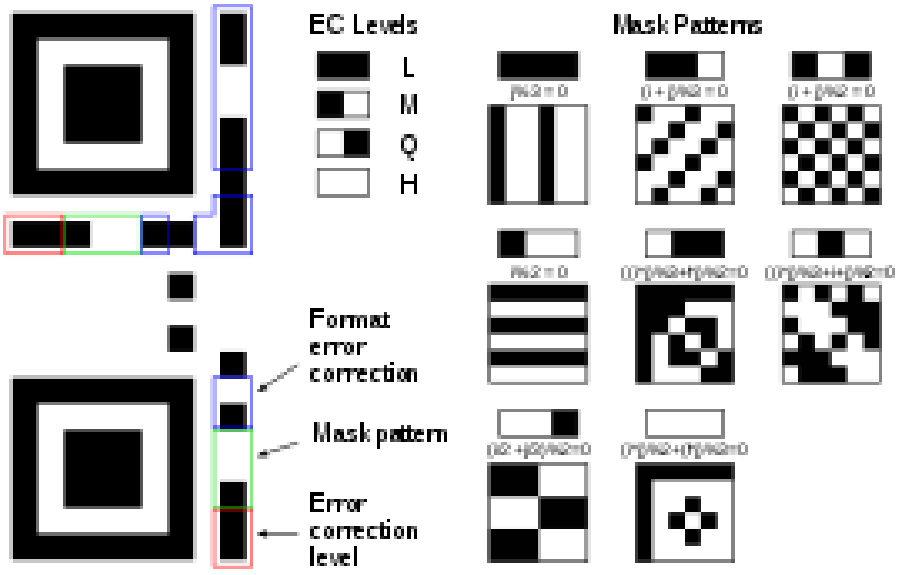
It is also possible to design artistic QR codes without reducing the error correction capacity by manipulating the underlying mathematical constructs.

Damaged but still decodable QR code

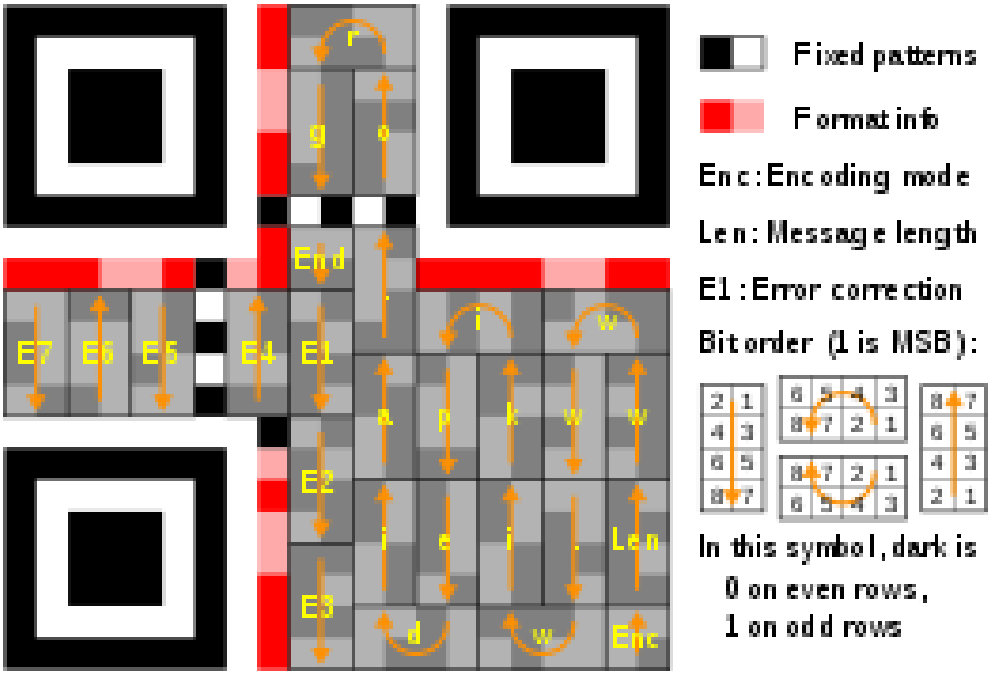
ENCODING

The format information records two things: the error correction level and the mask pattern used for the symbol. Masking is used to break up patterns in the data area that might confuse a scanner, such as large blank areas or misleading features that look like the locator marks. The mask patterns are defined on a grid that is repeated as necessary to cover the whole symbol. Modules corresponding to the dark areas of the mask are inverted. The format information is protected from errors with a BCH code, and two complete copies are included in each QR symbol.[2]

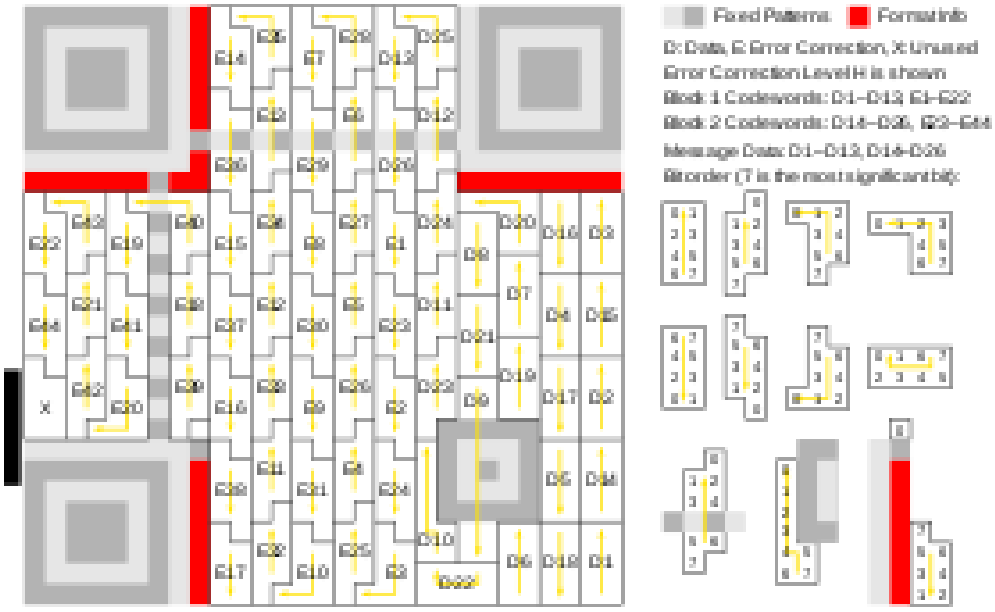
The message dataset is placed from right to left in a zigzag pattern, as shown below. In larger symbols, this is complicated by the presence of the alignment patterns and the use of multiple interleaved error-correction blocks



Meaning of format information. Encoding: In the above figure, the format information is protected by a (15,5) BCH code, which can correct up to 3 bit errors. The total length of the code is 15 bits, and 5 indicates the number of data bits (EC level plus mask pattern). 10 extra bits are used for error correction. The mask for these 10 bits is: [1001010101] (excludes the fixed pattern squares). The first 5 bits are unmasked.



Message placement within a QR symbol. The message is encoded using a (255,249) Reed Solomon code (shortened to (24,18) code by using "padding") which can correct up to 3 byte errors.



Larger symbol illustrating interleaved blocks. The message has 26 data bytes and is encoded using a (255,211) Reed Solomon code (shortened to (70,26) code) which can correct up to 22 byte errors in a single burst. For error correction, 44 extra "parity" bytes are appended to the data bytes (resulting in a total of 70 code bytes). The symbol achieves level H error correction

The general structure of a QR encoding is as a sequence of 4 bit indicators with payload length dependent on the indicator mode (e.g. byte encoding payload length is dependent on the first byte).

Mode indicator	Description	Typical structure '[type : sizes in bits]'
0001	Numeric	[0001 : 4] [Character Count Indicator : variable] [Data Bit Stream : 10 × charcount]
0010	Alphanumeric	[0010 : 4] [Character Count Indicator : variable] [Data Bit Stream : 11 × charcount]
0100	Byte encoding	[0100 : 4] [Character Count Indicator : variable] [Data Bit Stream : 8 × charcount]
1000	Kanji encoding	[1000 : 4] [Character Count Indicator : variable] [Data Bit Stream : 13 × charcount]
0011	Structured append	[0011 : 4] [Symbol Position : 4] [Total Symbols: 4] [Parity : 8]
0111	ECI	[0111 : 4] [ECI Assignment number : variable]
0101	FNC1 in first position	[0101 : 4] [Numeric/Alphanumeric/Byte/Kanji payload : variable]
1001	FNC1 in second position	[1001 : 4] [Application Indicator : 8] [Numeric/Alphanumeric/Byte/Kanji payload : variable]
0000	End of message	[0000 : 4]

Note:

- Character Count Indicator depends on how many modules are in a QR code (Symbol Version).
- ECI Assignment number Size:
  - 8 × 1 bits if ECI Assignement Bitstream starts with '0'
  - 8 × 2 bits if ECI Assignement Bitstream starts with '10'
  - 8 × 3 bits if ECI Assignement Bitstream starts with '110'

Four-bit indicators are used to select the encoding mode and convey other information.

Encoding modes

Indicator	Meaning
0001	Numeric encoding (10 bits per 3 digits)
0010	Alphanumeric encoding (11 bits per 2 characters)
0100	Byte encoding (8 bits per character)
1000	Kanji encoding (13 bits per character)
0011	Structured append (used to split a message across multiple QR symbols)
0111	Extended Channel Interpretation (select alternate character set or encoding)
0101	FNC1 in first position (see Code 128 for more information)
1001	FNC1 in second position

0000	End of message (Terminator)

Encoding modes can be mixed as needed within a QR symbol. (e.g., a url with a long string of alphanumeric characters)

[Mode Indicator][Mode bitstream] --> [Mode Indicator][Mode bitstream] --> etc... --> [0000 End of message (Terminator)]

After every indicator that selects an encoding mode is a length field that tells how many characters are encoded in that mode. The number of bits in the length field depends on the encoding and the symbol version.

Number of bits in a length field (Character Count Indicator)

Encoding	Ver. 1–9	10–26	27–40
Numeric	10	12	14
Alphanumeric	9	11	13
Byte	8	16	16
Kanji	8	10	12

Alphanumeric encoding mode stores a message more compactly than the byte mode can, but cannot store lower-case letters and has only a limited selection of punctuation marks, which are sufficient for rudimentary web addresses. Two characters are coded in an 11-bit value by this formula:

$$V = 45 \times C_1 + C_2$$

This has the exception that the last character in an alphanumeric string with an odd length is read as a 6-bit value instead.

Alphanumeric character codes									
Code	**Character**	**Code**	**Character**	**Code**	**Character**	**Code**	**Character**	**Code**	**Character**
00	0	09	9	18	I	27	R	36	Space
01	1	10	A	19	J	28	S	37	$
02	2	11	B	20	K	29	T	38	%
03	3	12	C	21	L	30	U	39	*
04	4	13	D	22	M	31	V	40	+
05	5	14	E	23	N	32	W	41	–
06	6	15	F	24	O	33	X	42	.
07	7	16	G	25	P	34	Y	43	/
08	8	17	H	26	Q	35	Z	44	:

VARIENTS

Model 1

Model 1 QR code is an older version of the specification. It is visually similar to the widely seen model 2 codes, but lacks alignment patterns.

Micro QR code

Micro QR code is a smaller version of the QR code standard for applications where symbol size is limited. There are four different versions (sizes) of Micro QR codes: the smallest is 11×11 modules; the largest can hold 35 numeric characters.

- 

Micro QR code example

- 

Micro QR code functional regions

- 

Model 1 QR code example

- 

Model 1 QR code functional regions

MINI PROJECT

CONVERT TEXT INTO QR CODE AND DECODE IT

DISCRIPTION

QR IDENTITY SYSTEM is made up of two parts QR GENERATOR and IDENTIFICATION SYSTEM. QR GENERATOR is a program that creates many QR in a single time while on the other hand DECODING SYSTEM is a program that will help by decoding QR moreover it creates a NOTEPAD FILE and store information in that NOTEPAD FILE with DATE and TIME

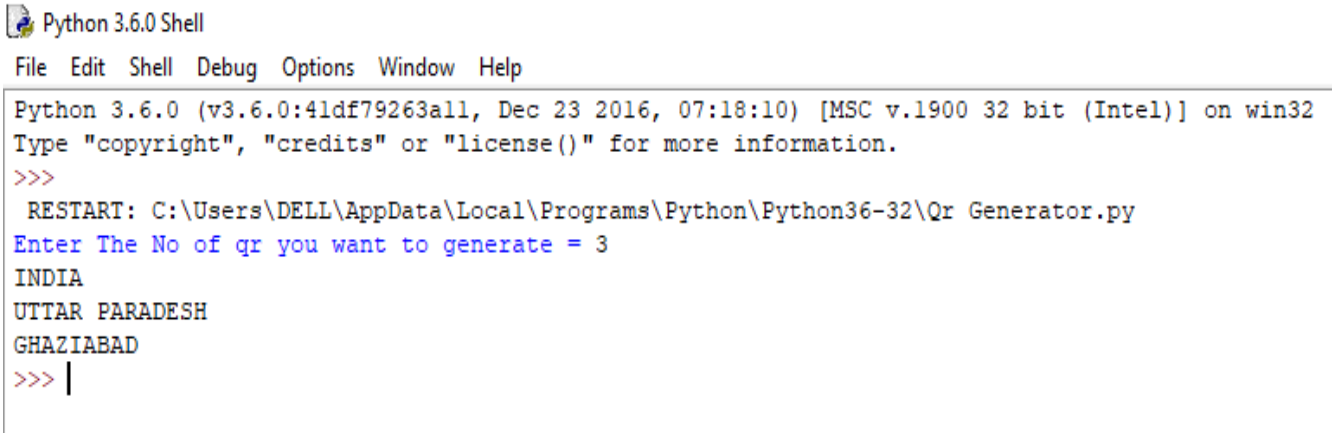
BARRIERS

There are some problems that I face while making this project.

- Which MODULE should be imported to perform a particular task
- How to install a MODULE
- How to Clean Data of previously made QR
- How to Decrease Time complexity and space complexity

INPUT

QR GENERATOR



PROGRAM OF QR GENERATOR

```
import qrcode

# Create qr code instance
qr = qrcode.QRCode(
    version = 1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size = 10,
    border = 4,
)
n=int(input("Enter The No of Student"))
l=[]

# The data that you want to store
for i in range(n):
    x=input()
    l.append(x)


for i in range(len(l)):
    data=l[i]
    # Add data
    qr.add_data(l[i])
    qr.make(fit=True)

    # Create an image from the QR Code instance
```

```
img = qr.make_image()

# Save it somewhere, change the extension as needed:
# img.save("image.png")
# img.save("image.bmp")
# img.save("image.jpeg")
img.save(f"{l[i]}.jpg")
qr.clear()
```

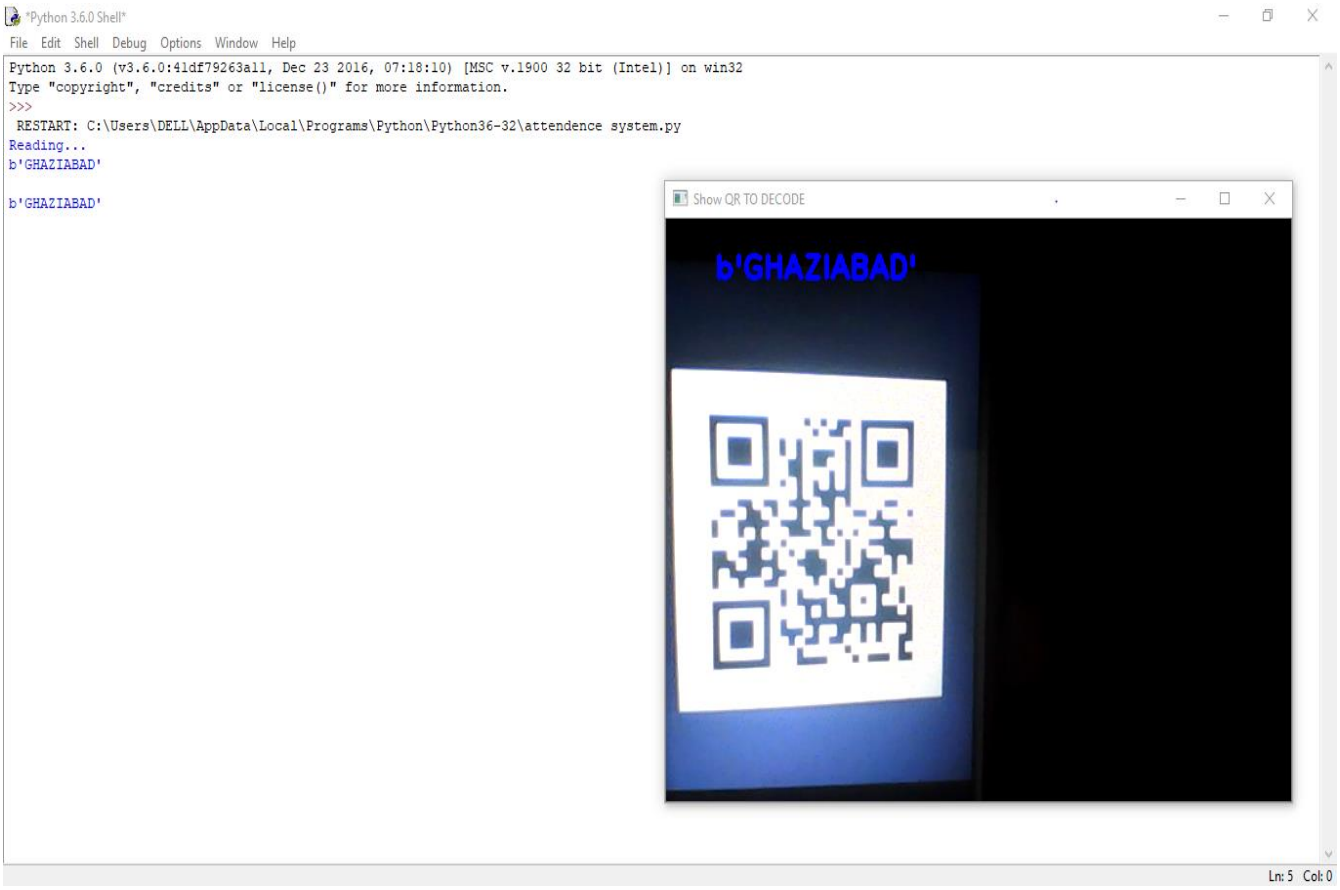
OUTPUT

QR CODE WILL GENERATE

INPUT

SHOW QR TO DECODE

SAMPLE 1:



SAMPLE 2

PROGRAM OF QR DECODER

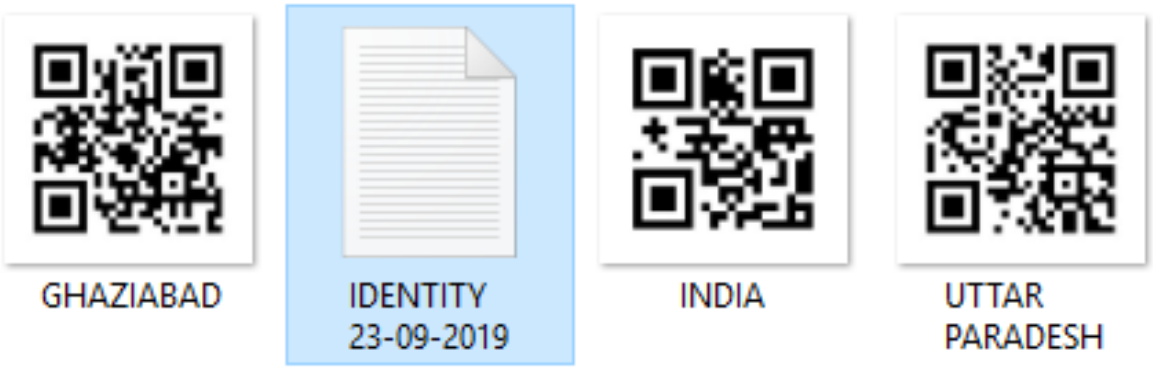
```
import cv2
import numpy as np
import pyzbar.pyzbar as pyzbar
import sys
import time
import datetime

cap = cv2.VideoCapture(0)

d_date = datetime.datetime.now()
d = d_date.strftime("%d-%m-%Y")


fob = open(f'IDENTITY {d}.txt', 'w+')

names = []



def enterData(z,c):
    if z in names:
```

```
        pass
    else:
        names.append(z)
        z = ''.join(str(z))
        fob.write(z + '\n')
        fob.write(c + '\n')
    return names


print('Reading...')


def checkData(data):
    try:
        data = str(data.decode())
    except:
        print('Invalid ID !!! try new one')
        return
    if data in names:
        pass
    else:
        #print('\n' + str(len(names) + 1) + '\n' + data)
        d_date = datetime.datetime.now()
        c = d_date.strftime("%I:%M:%S %p")
        enterData(data,c)
        cv2.putText(frame, str(obj.data), (50, 50), cv2.FONT_HERSHEY_PLAIN, 2,
                    (255, 0, 0), 3)


while True:
    _, frame = cap.read()

    decodedObjects = pyzbar.decode(frame)
    for obj in decodedObjects:
        checkData(obj.data)
        print("Data", obj.data)
        #enterData(obj.data,c)
        #print(len(names))
        sys.stdout.write(str(obj.data))
        print("\n")
        sys.stdout.flush()
        time.sleep(1.5)

    cv2.imshow("Show QR TO DECODE ", frame)

    if cv2.waitKey(1) & 0xFF == ord('s'):
```

```
        cv2.destroyAllWindows()
        break

fob.close()
```

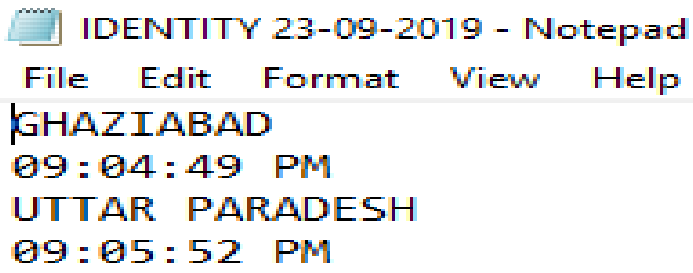
OUTPUT

NOTEPAD FILE GENERATE BY THE NAME OF IDENTITY



OUTPUT

INSIDE IDENTITY NAMED NOTEPAD FILE



CONCLUSION AND FUTURE SCOPE

For something that is commonplace in modern (urban) existence, it is remarkable how infrequently QR codes are used. Most technologically-aware people would be able to recognize one if they saw it, but the number of people that have actually used them is small, with even fewer using them with any regularity. For all those out there that use QR codes, there are countless others that don't even really know what they are, and tech and graphic people tend to dislike them on account of them being ugly.

The potential uses for QR codes are numerous – Wi-Fi network login, code payments, virtual stores, URLs, to name but a few – but no single use has exploded to the point where they are legitimately part of daily life. In terms of digital marketing, QR codes have the potential to help audiences find out more about a campaign or access special offers through the codes, though again no single use has stood out as a must for marketing teams at large.

QR codes enjoyed a brief renaissance thanks to Snapchat, which introduced its very similar 'Snapcodes' in 2015. Using exactly the same principle at QR codes, the Snapcodes were a quick and novel way for the messaging app's younger users to add each other, using unique formations of black dots to act as an ID card for users' profiles. The codes threw the basic idea behind QR straight back into the mainstream, with everyone from Bernie Sanders to Snapchat's army of teenagers getting on board.

Last year, Snapchat partnered with Starbucks to allow customers to scan QR codes in store and receive free coffees. Crucially, the users didn't have to have the Starbucks app to qualify for the giveaway, instead using Snapchat as the portal. According to the coffee chain, the numbers were 'incredible,' and the success of the promotion suggests that QR codes are held back more by their tethering to apps, rather than the medium more fundamentally. This is why it makes sense for an app like Snapchat to incorporate QR code recognition into its software – if users can do more in an app, they'll have fewer reasons to leave it.