Shashank Sarma
2019130053
TE COMPS CSS EXP 7

LABORATORY

CEL62: Cryptography and System Security
Winter 2021

Experiment 7:	**Creating and Reading QR Codes** **Ref:** **https://www.linux-magazine.com/Online/Features/Generating-QR-Codes-in-Linux**

Note: Students are advised to read through this lab sheet before doing experiment. On-the-spot evaluation may be carried out during or at the end of the experiment. Your performance, teamwork/Personal effort, and learning attitude will count towards the marks.

Experiment 6: Creating and Reading QR codes

1 OBJECTIVE

Creating and Reading QR Codes for given exercise

2 INTRODUCTION

With the right tools, you can create your own QR code squares with information you want to share on a business card, in a letter, or on your website.

Read errors of unreliable, one-dimensional bar codes often caused interruptions in industrial production, prompting companies like Toyota and its subsidiary Denso Wave to develop as early as 1994 a new code for acquiring stock data. The new matrix bar code was designed to store more information than the traditional bar code and to stay legible, even if the label was dirty, wrinkled, or partially destroyed.

The quick response code, or QR code, comprises a matrix of square dots instead of the usual lines. Measuring up to 177 by 177 dots, the QR code encodes up to 4,296 characters, compared with a bar code that encodes just 13.

Thanks to numerous free reader apps for smartphones, QR codes have gained in popularity in recent years. Posters, catalogs, magazines, business cards, and even television screens display the small squares, offering additional information or URLs for microsites.

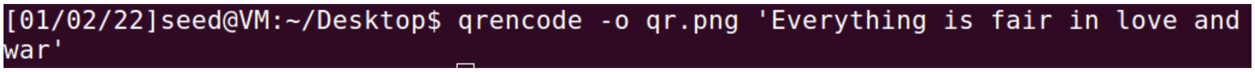
Data Grabber

If you want to encode your address into a QR code for a business card, you could try one of the many services on the Internet. However, restrictions typically apply. For example, some services allow only QR codes in a certain size or in limited quantities. Also, you never know for sure, what the services do with your data. For example, some sites explicitly allow sharing and selling the data to third parties in their terms of use, more or less pre-programming an increase in junk mail in the future. Fortunately, several QR code tools can help you avoid these problems in Linux.

3 PROCEDURE/TASK

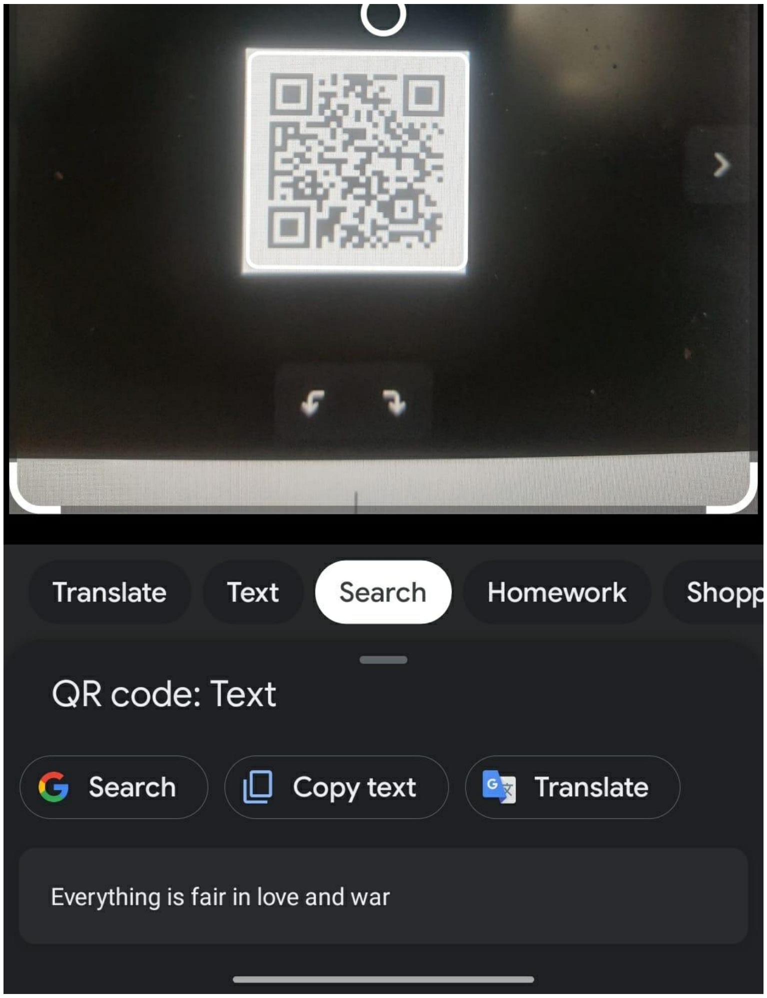
The quickest way to create a QR code is with the Qrencode command-line utility. Any major distribution can install Qrencode via the package manager. The following command then creates a QR code containing the text “Hello World!”:

$ qrencode -o qrcode.png 'Hello World!'



```
[01/02/22]seed@VM:~/Desktop$ qrencode -o qr.png 'Everything is fair in love and
war'
```

qr.png



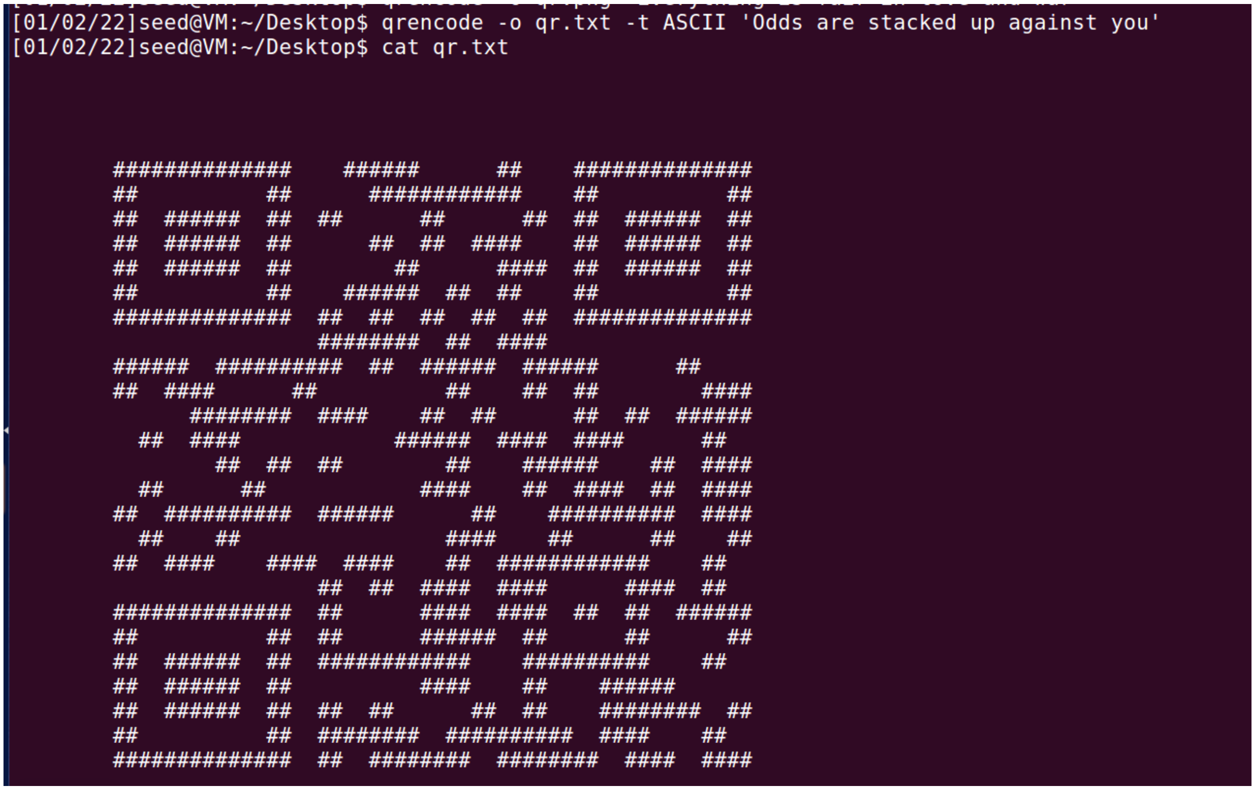
Everything is fair in love and war

QR Code relevant Everything is fair in love and war will pop up: Qrencode encoding the string "Everything is fair in love and war" as a QR code and storing it in the qr.png file.

The generated QR code ends up in the *qr.png* file. If the file already exists, it's overwritten without prompting. If you have at least version 3.3.0 (*qrencode -V*), Qrencode can generate an EPS graphic,

```
$ qrencode -t EPS -o qrcode.eps 'Hello World!'
```

or ASCII output.

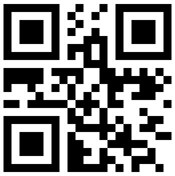


```
[01/02/22]seed@VM:~/Desktop$ qrencode -o qr.txt -t ASCII 'Odds are stacked up against you'
[01/02/22]seed@VM:~/Desktop$ cat qr.txt



        ##############    ######      ##    ##############
        ##          ##      ############    ##          ##
        ##  ######  ##  ##      ##      ##  ##  ######  ##
        ##  ######  ##      ##  ##  ####    ##  ######  ##
        ##  ######  ##        ##      ####  ##  ######  ##
        ##          ##    ######  ##  ##    ##          ##
        ##############  ##  ##  ##  ##  ##  ##############
                        ########  ##  ####
        ######  ##########  ##  ######  ######      ##
        ##  ####      ##          ##    ##  ##        ####
              ########  ####    ##  ##      ##  ##  ######
          ##  ####            ######  ####  ####      ##
                ##  ##  ##        ##    ######    ##  ####
          ##      ##            ####    ##  ####  ##  ####
        ##  ##########  ######      ##    ##########  ####
          ##    ##                ####    ##      ##    ##
        ##  ####    ####  ####    ##  ############    ##
                        ##  ##  ####  ####      ####  ##
        ##############  ##      ####  ####  ##  ##  ######
        ##          ##  ##      ######  ##      ##      ##
        ##  ######  ##  ############    ##########    ##
        ##  ######  ##          ####    ##    ######
        ##  ######  ##  ##  ##      ##  ##    ########  ##
        ##          ##  ########  ##########  ####    ##
        ##############  ##  ########  ########  ####  ####
```

Outputting QR codes in ASCII characters just for fun. Each # corresponds to a dot.

In the QR code image, the software creates a white border the width of one dot. This facilitates the process of deciphering the code for programs or the smartphone later on. If desirable, you can increase or decrease the edge with the *-m* parameter; in the following example the border width would be 10 code pixels:

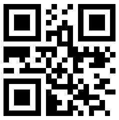
```
$ qrencode -m 10 -o qrcode.png 'Hello World!'
```



If you are saving the QR code in PNG format, the *-s* parameter specifies the height of a black QR code pixel. By default, Qrencode draws every black dot three by three pixels. The program creates a slightly smaller QR code with quite a wide margin with the following command:

```
$ qrencode -s 2 -m 10 -o qrcode.png 'Hello World!'
```

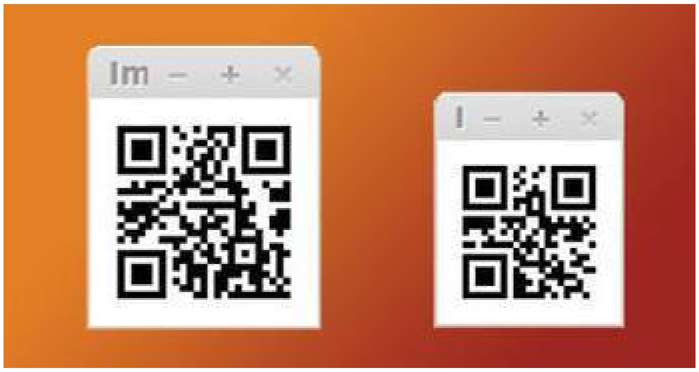
A white border appears around the QR code that is exactly 20 screen pixels wide (10 QR code dots in width, with each dot two pixels). If you are creating EPS images, the only parameter available is *-m*; *-s* moves the entire QR code out of the image in this case. To control the resolution of PNGs further, use the *-d* parameter to define the dpi.



Tolerance

Besides the specified data, the QR code contains additional error correction information. If a portion of the image is damaged, it allows you to reconstruct the missing or illegible data. The more additional information the QR code contains, the more heavily damaged it can be without becoming useless.

Increasing the error tolerance increases the size of the image because you need more black dots



Both QR codes contain the text Hello World!; the one on the left uses the highest error correction level, H, and therefore is more robust – but also larger.

The QR code standard thus uses four levels of error correction: *H* level allows you to read all data if 30 percent of the QR code is destroyed, *Q* level if 25 percent is unreadable, and *M* level if just

15 percent is unintelligible. At the lowest level, *L*, only 7 percent of the data can be faulty. In Qrencode, the *-l* parameter selects the error correction level. The possible values are pretty much what you would expect: *L*, *M*, *Q*, and *H*.

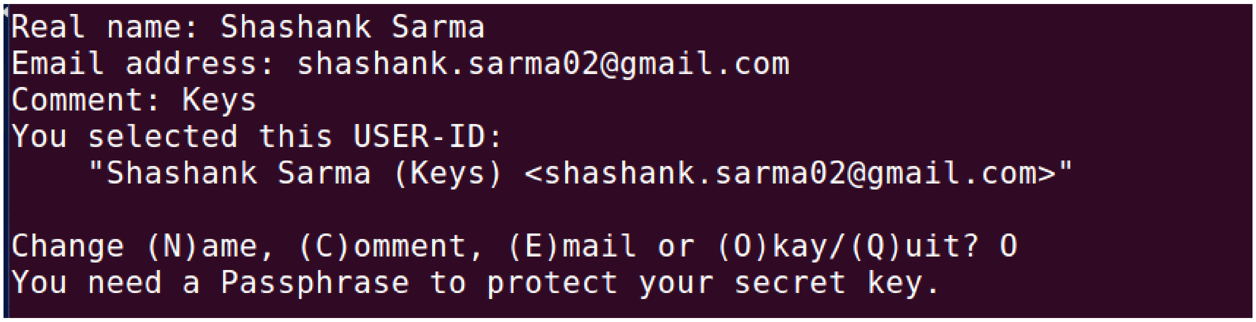
Of the tested programs, Portable QR-Code Generator offers the greatest functionality. Data entry is also convenient in a variety of tabs. Qrencode creates QR codes more quickly, and it is easy to scripted so you generate QR codes in batches. KBarcode4-light is deprecated, but it is the only program that can generate PDF files; the EPS images created by Qrencode offer similar output.

EXERCISE

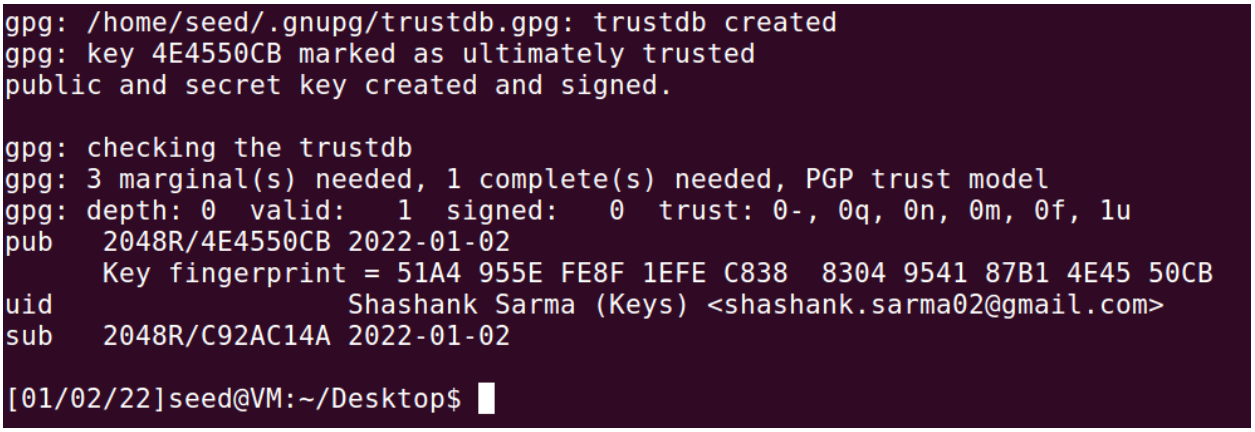
TPO of SPIT has asked you to submit simple QR code of your CV so that company can make off line process to create short list of potential candidates. Hence as a part of exercise create CV showing your Profile, Ability and Capability and generate QR code of CV. This QR code can only be shared between you and company. You will provide your digital signature as an authentication to company. Once authentication is successful company is able to scan this QR code of your CV.

Show Implementation steps that you will generate QR code and sign it with your digital signature. Share it with your colleague from same lab with who is capable to verify your digital signature. Successful verification makes scanning of QR code feasible. Show you creativity in terms of robustness of generating digital signature, generating of QR code and Scanning of QR code if relevant secret key is entered.

Creating Keys for encryption:

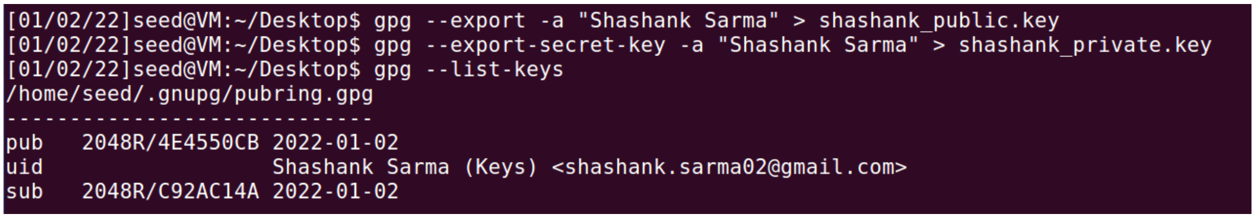


```
Real name: Shashank Sarma
Email address: shashank.sarma02@gmail.com
Comment: Keys
You selected this USER-ID:
    "Shashank Sarma (Keys) <shashank.sarma02@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.
```



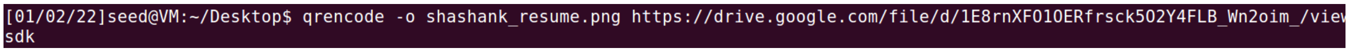
```
gpg: /home/seed/.gnupg/trustdb.gpg: trustdb created
gpg: key 4E4550CB marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   2048R/4E4550CB 2022-01-02
      Key fingerprint = 51A4 955E FE8F 1EFE C838  8304 9541 87B1 4E45 50CB
uid                  Shashank Sarma (Keys) <shashank.sarma02@gmail.com>
sub   2048R/C92AC14A 2022-01-02

[01/02/22]seed@VM:~/Desktop$ 
```

Exporting newly generated keys to public and private keys:



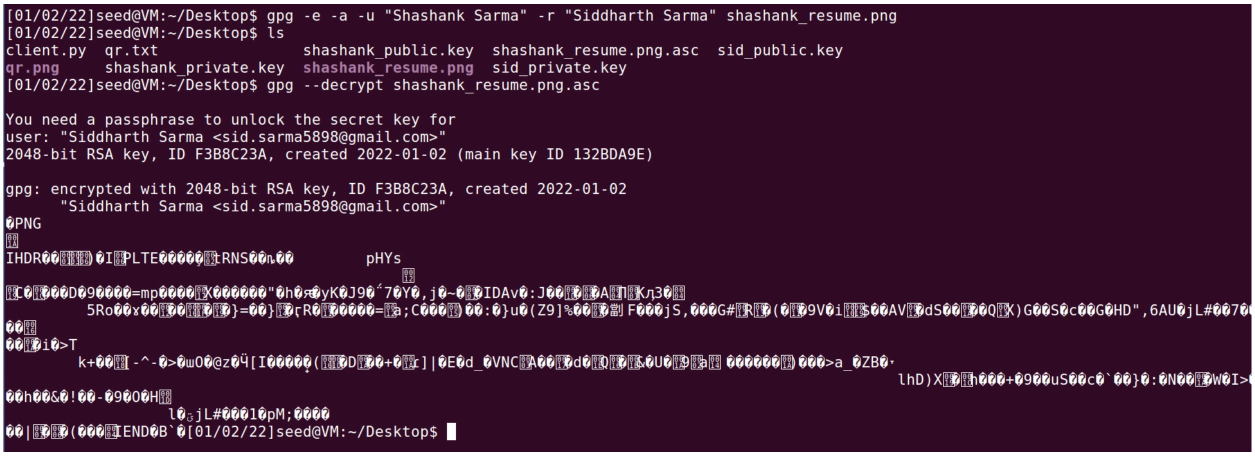
```
[01/02/22]seed@VM:~/Desktop$ gpg --export -a "Shashank Sarma" > shashank_public.key
[01/02/22]seed@VM:~/Desktop$ gpg --export-secret-key -a "Shashank Sarma" > shashank_private.key
[01/02/22]seed@VM:~/Desktop$ gpg --list-keys
/home/seed/.gnupg/pubring.gpg
-----------------------------
pub   2048R/4E4550CB 2022-01-02
uid                  Shashank Sarma (Keys) <shashank.sarma02@gmail.com>
sub   2048R/C92AC14A 2022-01-02
```

Encoding the URL of the Resume using qrencode

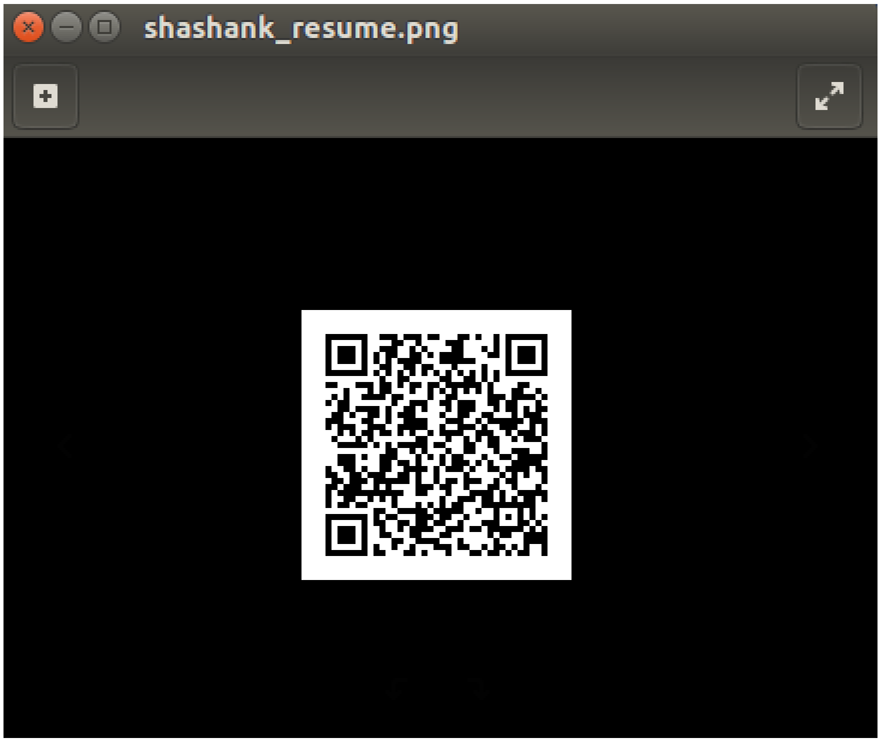


```
[01/02/22]seed@VM:~/Desktop$ qrencode -o shashank_resume.png https://drive.google.com/file/d/1E8rnXFO1OERfrsck5O2Y4FLB_Wn2oim_/view
sdk
```

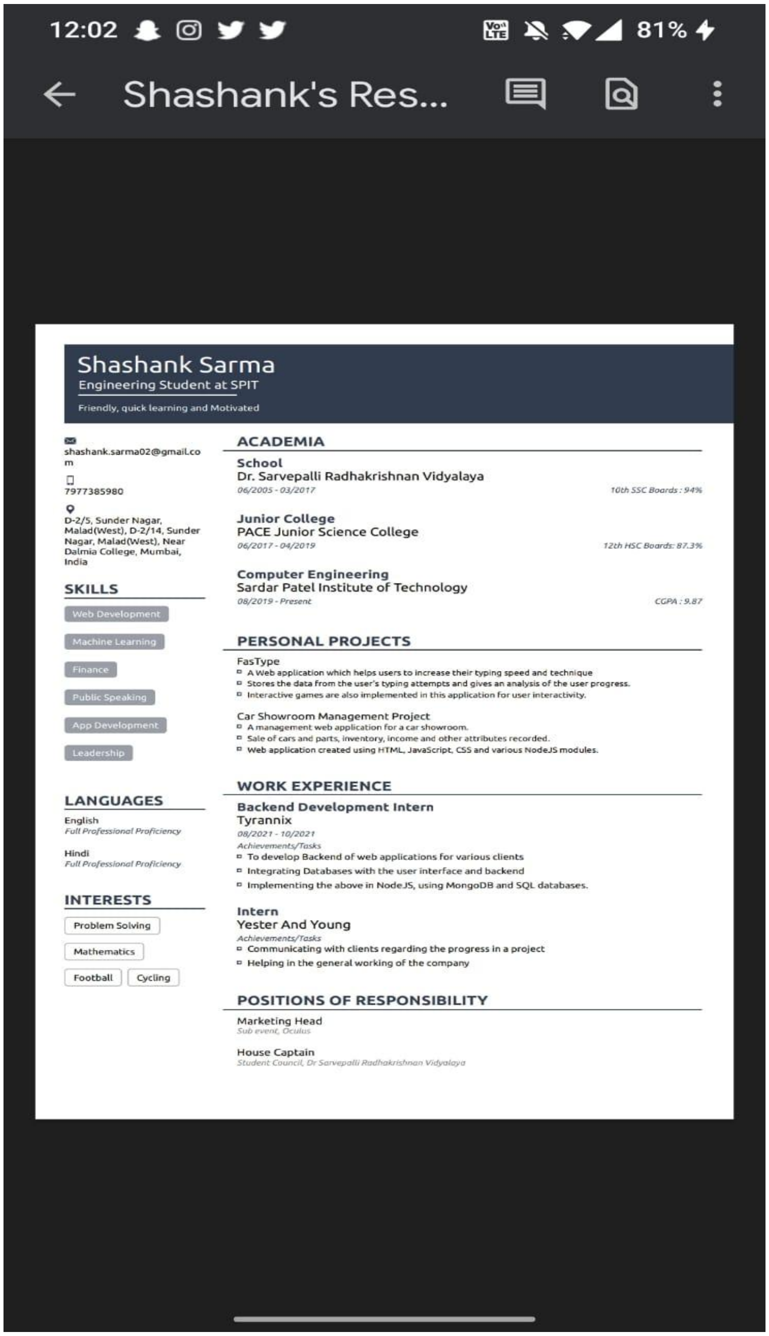
This makes sure that only the intended users are able to access the resume from the QR Code.



```
[01/02/22]seed@VM:~/Desktop$ gpg -e -a -u "Shashank Sarma" -r "Siddharth Sarma" shashank_resume.png
[01/02/22]seed@VM:~/Desktop$ ls
client.py  qr.txt             shashank_public.key  shashank_resume.png.asc  sid_public.key
qr.png     shashank_private.key  shashank_resume.png  sid_private.key
[01/02/22]seed@VM:~/Desktop$ gpg --decrypt shashank_resume.png.asc

You need a passphrase to unlock the secret key for
user: "Siddharth Sarma <sid.sarma5898@gmail.com>"
2048-bit RSA key, ID F3B8C23A, created 2022-01-02 (main key ID 132BDA9E)

gpg: encrypted with 2048-bit RSA key, ID F3B8C23A, created 2022-01-02
      "Siddharth Sarma <sid.sarma5898@gmail.com>"
�PNG
�
IHDR������)�I�PLTE������tRNS��ɲ��        pHYs
                                                  �
�C�����D�9����=mp�����X������"�h�я�yK�J9�´7�Y�,j�~��IDAv�:J������A�Ӆ�Kл3��
         5Ro��ɤ����������}=��}��ɼR�������=�a;C�����)��:�}u�(Z9]%����剷F���jS,���G#�R��(���9V�i���$��AV��dS������Q�X)G��S�c��G�HD",6AU�jL#��7�
���
����i�>T
        k+���[-^-�>�ɯO�@z�Ӵ[I�����(����D����+���ɾ]|�E�d_�VNC�A�����d���Q����&�U���9��a� ��������)���>a_�ZB�ˑ
                                                                                                                        lhD)X���h���+�9��uS��c�`��}�:�N����W�I>�
��h��&�!��-�9�O�H��
                    l�ˀjL#���1�pM;����
��|����(�����IEND�B`�[01/02/22]seed@VM:~/Desktop$ █
```



Checking the output of scanning the QR code on an external device (Mobile Phone)



Shashank Sarma
Engineering Student at SPIT

Friendly, quick learning and Motivated

shashank.sarma02@gmail.com

7977385980

D-2/5, Sunder Nagar, Malad(West), D-2/14, Sunder Nagar, Malad(West), Near Dalmia College, Mumbai, India

SKILLS

Web Development

Machine Learning

Finance

Public Speaking

App Development

Leadership

LANGUAGES

English
Full Professional Proficiency

Hindi
Full Professional Proficiency

INTERESTS

Problem Solving

Mathematics

Football

Cycling

ACADEMIA

School
Dr. Sarvepalli Radhakrishnan Vidyalaya
06/2005 - 03/2017 *10th SSC Boards : 94%*

Junior College
PACE Junior Science College
06/2017 - 04/2019 *12th HSC Boards: 87.3%*

Computer Engineering
Sardar Patel Institute of Technology
08/2019 - Present *CGPA : 9.87*

PERSONAL PROJECTS

FasType
- A Web application which helps users to increase their typing speed and technique
- Stores the data from the user's typing attempts and gives an analysis of the user progress.
- Interactive games are also implemented in this application for user interactivity.

Car Showroom Management Project
- A management web application for a car showroom.
- Sale of cars and parts, inventory, income and other attributes recorded.
- Web application created using HTML, JavaScript, CSS and various NodeJS modules.

WORK EXPERIENCE

Backend Development Intern
Tyrannix
08/2021 - 10/2021
Achievements/Tasks
- To develop Backend of web applications for various clients
- Integrating Databases with the user interface and backend
- Implementing the above in NodeJS, using MongoDB and SQL databases.

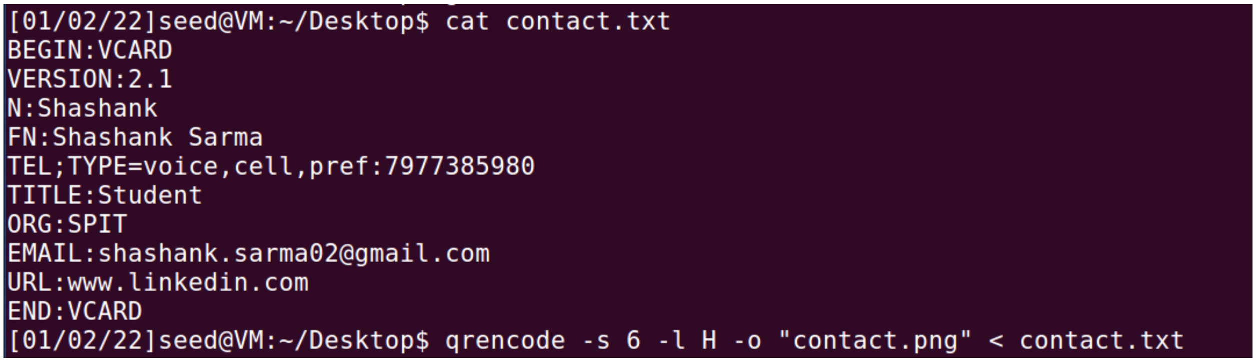
Intern
Yester And Young
Achievements/Tasks
- Communicating with clients regarding the progress in a project
- Helping in the general working of the company

POSITIONS OF RESPONSIBILITY

Marketing Head
Sub event, Oculus

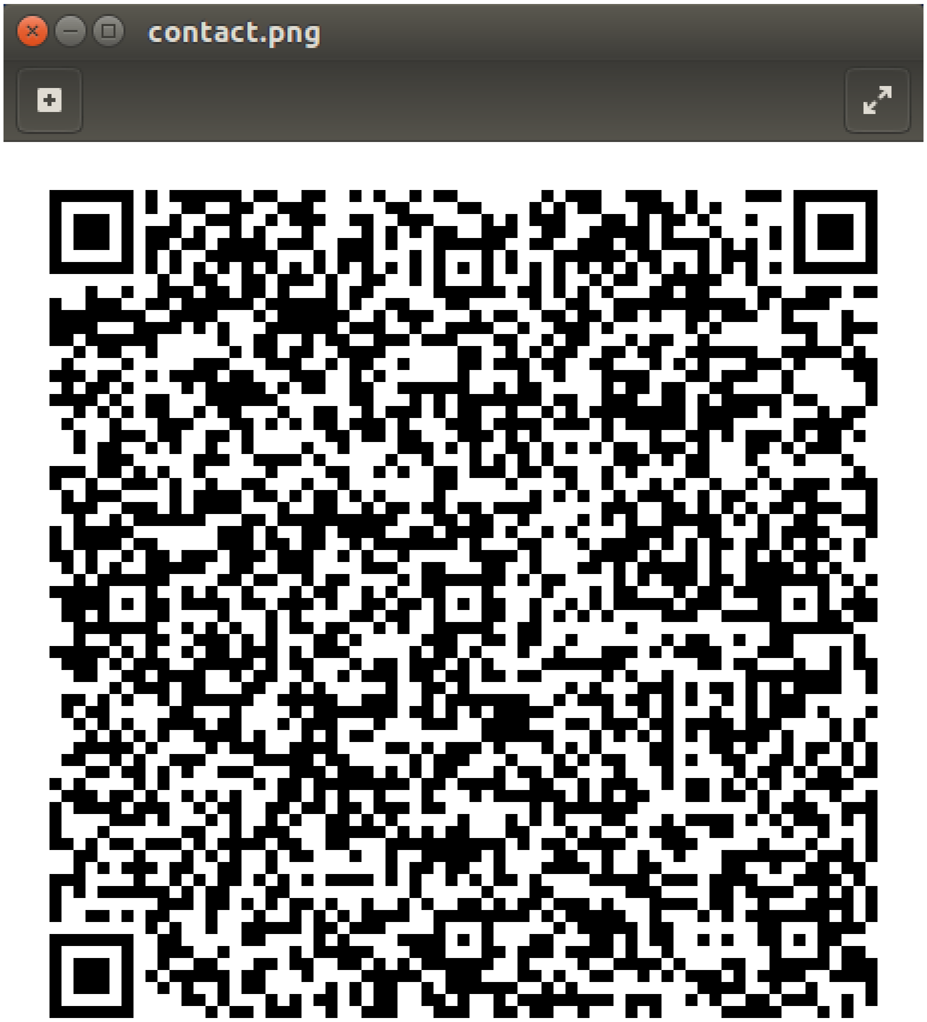
House Captain
Student Council, Dr Sarvepalli Radhakrishnan Vidyalaya

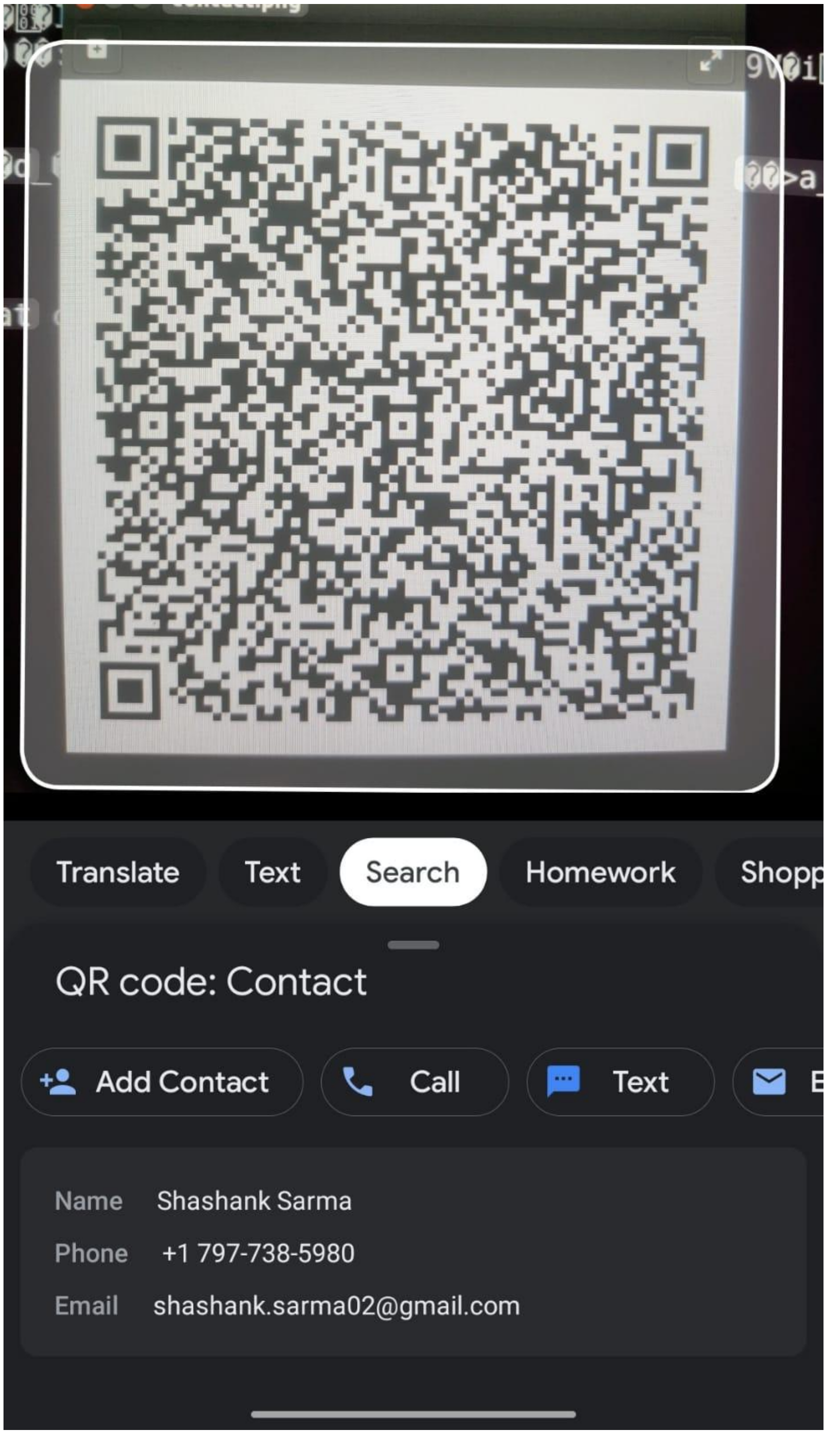
Creating a custom contact card and encapsulating the information in a QR Code



```
[01/02/22]seed@VM:~/Desktop$ cat contact.txt
BEGIN:VCARD
VERSION:2.1
N:Shashank
FN:Shashank Sarma
TEL;TYPE=voice,cell,pref:7977385980
TITLE:Student
ORG:SPIT
EMAIL:shashank.sarma02@gmail.com
URL:www.linkedin.com
END:VCARD
[01/02/22]seed@VM:~/Desktop$ qrencode -s 6 -l H -o "contact.png" < contact.txt
```

QR Code for Contact Card





QR code: Contact

Name	Shashank Sarma
Phone	+1 797-738-5980
Email	shashank.sarma02@gmail.com

CONCLUSION:

1. Through this experiment I learnt about the usage and working of qrencode. QR codes are frequently used to track information about products in a supply chain and – because many smartphones have built-in QR readers – they are often used in marketing and advertising campaigns.
2. I learnt how to customize the QR Code by tweaking the tolerance value.
3. I also learnt how to share the link to my resume using the QR Code and encrypt the information so that only authorized users can access the resume.
4. Lastly, I learnt how to encapsulate different types of information such as text, URLs, contact cards in a QR Code and output the same in the form of an image, document and ASCII text.