

Experiment 7: Creating and Reading QR codes

Name: Deep Nayak UID: 2019130045 TE COMPS

AIM : To create and read QR Codes for the given exercise

PROCEDURE:

Installing qrencode

```
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~$ sudo apt install qrencode
[sudo] password for deep:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gyp javascript-common libc-ares2 libjs-inherits libjs-is-typedarray libjs-psl
  libjs-typedarray-to-buffer libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib libssl-dev
  libuv1-dev nodejs-doc python-pkg-resources python2 python2-minimal python2.7 python2.7-minimal
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libqrencode4
The following NEW packages will be installed:
  libqrencode4 qrencode
0 upgraded, 2 newly installed, 0 to remove and 122 not upgraded.
Need to get 47.6 kB of archives.
After this operation, 133 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 libqrencode4 amd64 4.0.2-2 [23.6 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 qrencode amd64 4.0.2-2 [24.0 kB]
Fetched 47.6 kB in 0s (109 kB/s)
Selecting previously unselected package libqrencode4:amd64.
(Reading database ... 201520 files and directories currently installed.)
```

```
The following NEW packages will be installed:
  libqrencode4 qrencode
0 upgraded, 2 newly installed, 0 to remove and 122 not upgraded.
Need to get 47.6 kB of archives.
After this operation, 133 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 libqrencode4 amd64 4.0.2-2 [23.6 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 qrencode amd64 4.0.2-2 [24.0 kB]
Fetched 47.6 kB in 0s (109 kB/s)
Selecting previously unselected package libqrencode4:amd64.
(Reading database ... 201520 files and directories currently installed.)
Preparing to unpack .../libqrencode4_4.0.2-2_amd64.deb ...
Unpacking libqrencode4:amd64 (4.0.2-2) ...
Selecting previously unselected package qrencode.
Preparing to unpack .../qrencode_4.0.2-2_amd64.deb ...
Unpacking qrencode (4.0.2-2) ...
Setting up libqrencode4:amd64 (4.0.2-2) ...
Setting up qrencode (4.0.2-2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~$
```

1. The quickest way to create a QR code is with the Qrcode command-line utility. Any major distribution can install Qrcode via the package manager. The following command then creates a QR code containing the text “Hello World!”:
`$ qrcode -o qrcode.png 'Hello World!'`



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



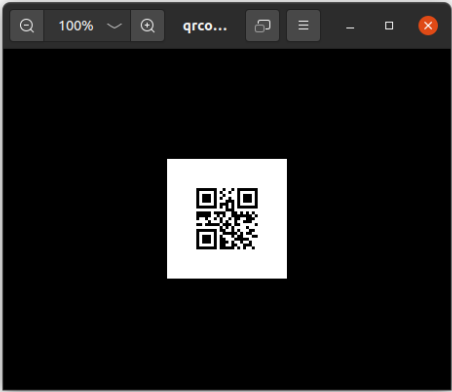
3. `$ qrcode -t ASCII -o qrcode.txt 'Hello World!'`

```
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx: ~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7
-t ASCII -o qrcode.txt 'Hello World!'
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ cat qrcode.txt

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

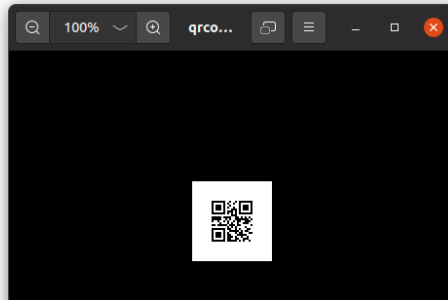
4. \$ qrencode -m 10 -o qrcode.png 'Hello World!'

```
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx: ~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ qrencode
-m 10 -o qrcode.png 'Hello World!'
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$
```



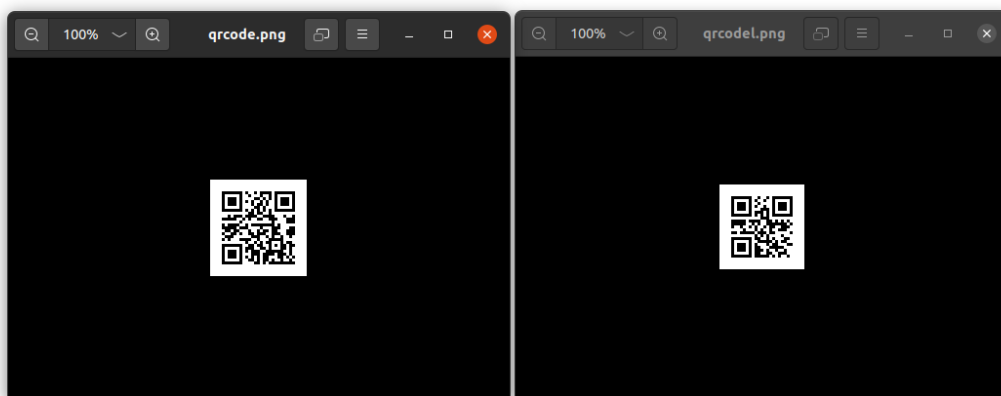
5. \$ qrencode -s 2 -m 10 -o qrcode.png 'Hello World!'

```
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx: ~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ qrencode
-m 10 -o qrcode.png 'Hello World!'
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ qrencode
-s 2 -m 10 -o qrcode.png 'Hello World!'
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$
```



2. Checking tolerance

```
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx: ~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ qrencode
-m 10 -o qrcode.png 'Hello World!'
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ qrencode
-s 2 -m 10 -o qrcode.png 'Hello World!'
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ qrencode
-o qrcode.png -l H "Hello World!"
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ qrencode
-o qrco1.png -l L "Hello World!"
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$
```



Creating new keys for encryption

```
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx: ~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ gpg --gen-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Deep Nayak
Email address: deep.nayak@spit.ac.in
You selected this USER-ID:
"Deep Nayak <deep.nayak@spit.ac.in>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key E36F194519BE9F44 marked as ultimately trusted
gpg: revocation certificate stored as '/home/deep/.gnupg/openpgp-revocs.d/5B6175B9859CF0516DD737CEE36F194519BE9F44.rev'
public and secret key created and signed.

pub   rsa3072 2021-12-10 [SC] [expires: 2023-12-10]
       5B6175B9859CF0516DD737CEE36F194519BE9F44
uid           Deep Nayak <deep.nayak@spit.ac.in>
sub   rsa3072 2021-12-10 [E] [expires: 2023-12-10]

deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$
```

Exporting newly generated keys to public and private keys

```
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx: ~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ gpg --export -a "Deep Nayak" > deep_public.key
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ gpg --export-secret-key -a "Deep Nayak" > deep_private.key
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 3 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 3u
gpg: next trustdb check due at 2023-12-06
/home/deep/.gnupg/pubring.kbx
-----
pub   rsa3072 2021-12-06 [SC] [expires: 2023-12-06]
       7EDF6D6B9DA2EAF3F15D574AB175AA1FCDD8345D
uid           [ultimate] Deep Nayak <deepmumbail6@gmail.com>
sub   rsa3072 2021-12-06 [E] [expires: 2023-12-06]

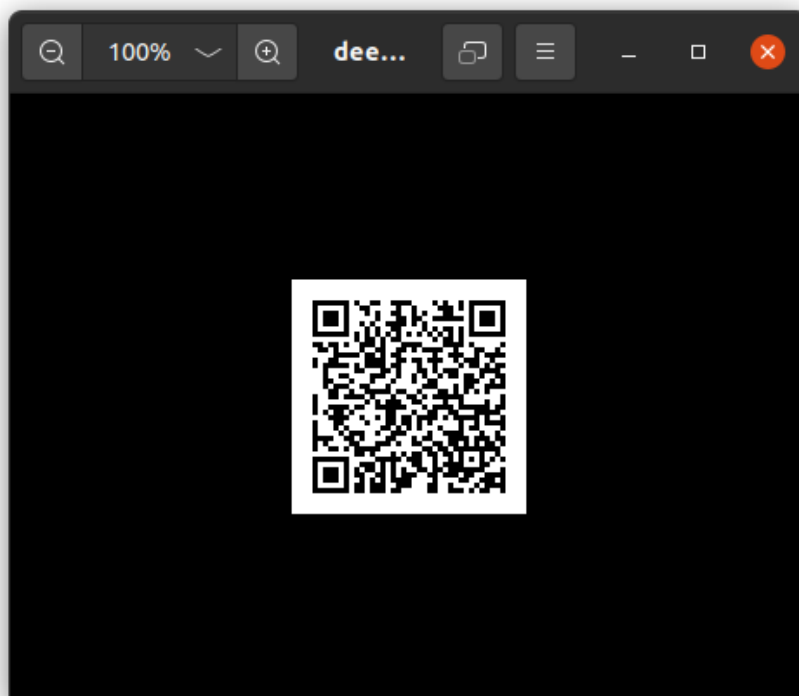
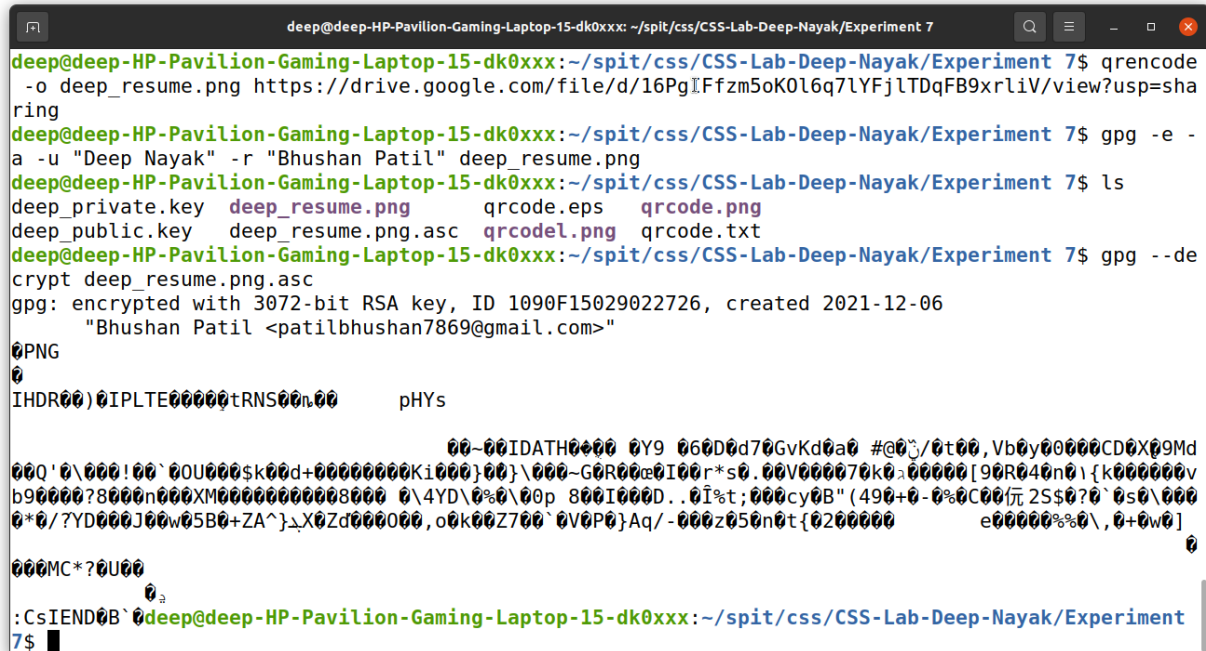
pub   rsa3072 2021-12-06 [SC] [expires: 2023-12-06]
       9ED23F26496294B553440E916792AE0CEDE8F7A2
uid           [ultimate] Bhushan Patil <patilbhushan7869@gmail.com>
sub   rsa3072 2021-12-06 [E] [expires: 2023-12-06]

pub   rsa3072 2021-12-10 [SC] [expires: 2023-12-10]
       5B6175B9859CF0516DD737CEE36F194519BE9F44
uid           [ultimate] Deep Nayak <deep.nayak@spit.ac.in>
```

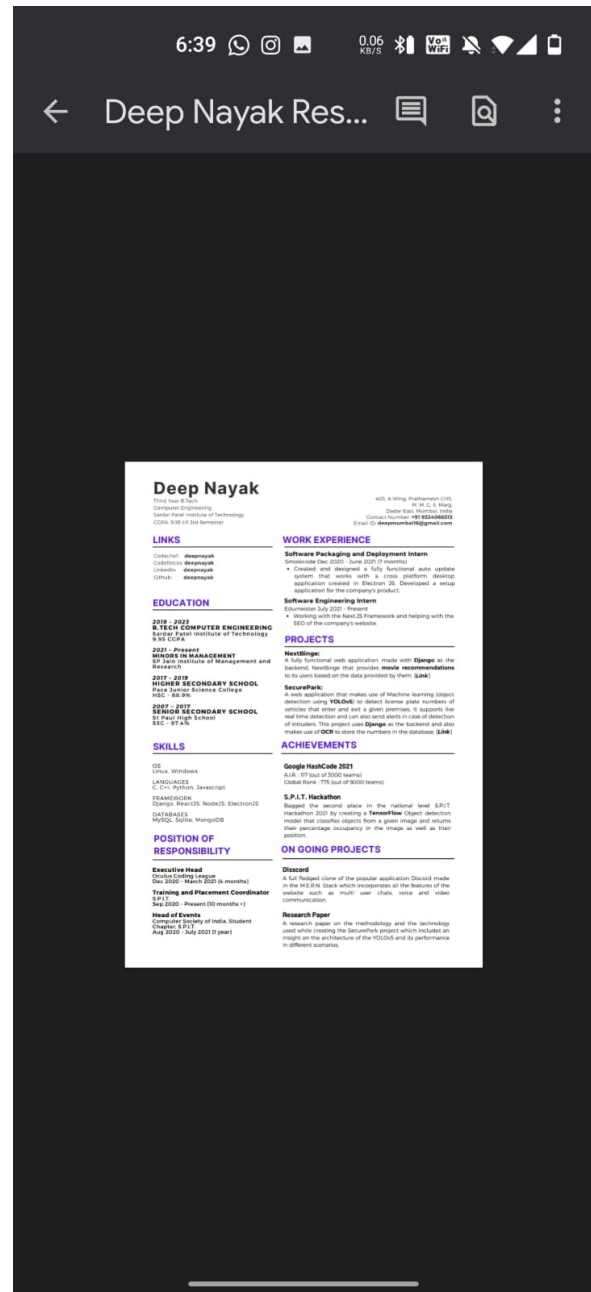
Encoding the URL of the Resume using qrencode

```
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx: ~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ qrencode -o deep_resume.png https://drive.google.com/file/d/16PgIFfzm5oK0l6q7lYFj1TDqFB9xrliV/view?usp=sharing
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ gpg -e -a -u "Deep Nayak" -r "Bhushan Patil" deep_resume.png
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ ls
deep_private.key  deep_resume.png  qrcode.eps  qrcode.png
deep_public.key  deep_resume.png.asc  qrcodel.png  qrcode.txt
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$
```

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



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



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

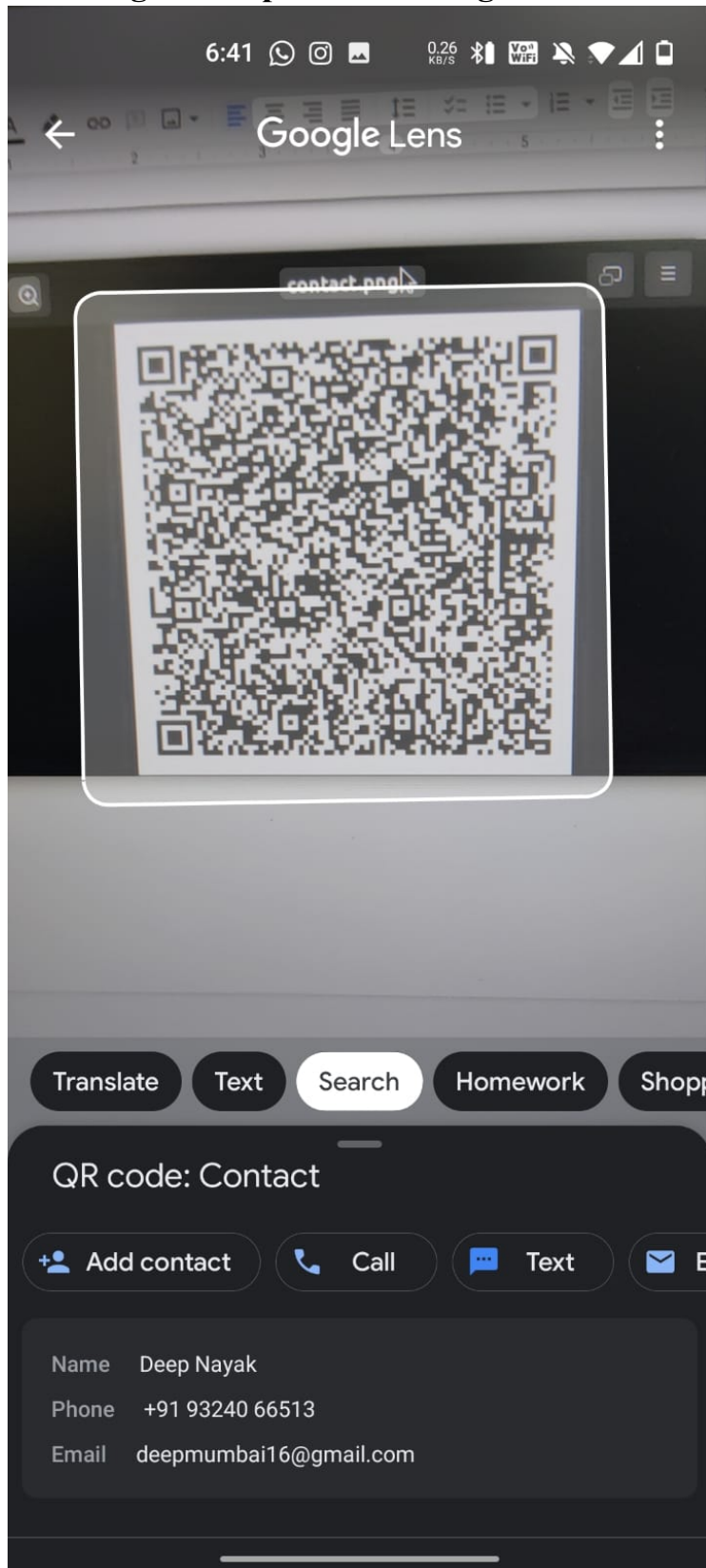
```
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx: ~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ touch contact.txt
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ cat contact.txt
BEGIN:VCARD
VERSION:2.1
N:Deep
FN:Deep Nayak
TEL;TYPE=voice,cell,pref:9324066513
TITLE:Software Engineer
ORG:SPIT
EMAIL:deepmumbai16@gmail.com
URL:www.https://www.linkedin.com/in/deep-nayak-4a81751b2/
END:VCARD

deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$ qrencode -s 6 -l H -o "contact.png" < contact.txt
deep@deep-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/spit/css/CSS-Lab-Deep-Nayak/Experiment 7$
```

QR Code for Contact Card



Checking the output of scanning the contact QR code on a mobile device.



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 and depending on the use case, which tolerance value is preferable.
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, email addresses, SMS, etc in a QR Code and output the same in the form of an image, document and ASCII text.

Github Link:

<https://github.com/deepnayak/CSS-Lab-Deep-Nayak/tree/master/Experiment%207>