Introduction:

This is a python based project in which various concepts of the programming language and some feature-rich libraries are used. This project helped us to learn and understand some really good and interesting features of the python language which we can further use in our future projects. Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming. There are various opportunities for Python programmers in networking and AI. One can explore career options in this industry and work as a network engineer or an AI analyst by studying the more complex principles from Python programming classes.

About the project:

We have made a QR code generator, reader and scanner application with the use of python programming language. With the help of this, a user can generate a QR code for themselves by just writing the text or pasting the link in the edit text space and clicking on ‘generate’. The QR code that will be generated will be unique and can be scanned by any scanner and from any device. A user can also read any QR code given to the application either by uploading an image of it from the user’s device or scanning it from the camera of the device. The program will read that QR and will display the output in the text space provided and will open that appropriate link in the user’s device browser.

Technologies and Libraries used:

Programming language: PYTHON

Libraries:

- QtPy5
- Open CV (cv2)
- Qrcode
- Pyzbar
- Pillow
- Webbrowser, time

QtPy5:

PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including iOS and Android.

Open CV (cv2):

OpenCV is an open-source software library for computer vision and machine learning. The OpenCV full form is Open Source Computer Vision Library. It was created to provide a shared infrastructure for applications for computer vision and to speed up the use of machine perception in consumer products. OpenCV, as a BSD-licensed software, makes it simple for companies to use and change the code. There are some predefined packages and libraries that make our life simple and OpenCV is one of them.

Qrcode:

A Quick Response Code or a QR Code is a two-dimensional bar code used for its fast readability and comparatively large storage capacity. It consists of black squares arranged in a square grid on a white background. Python has a library “qrcode” for generating QR code images. It can be installed using pip.

Pyzbar:

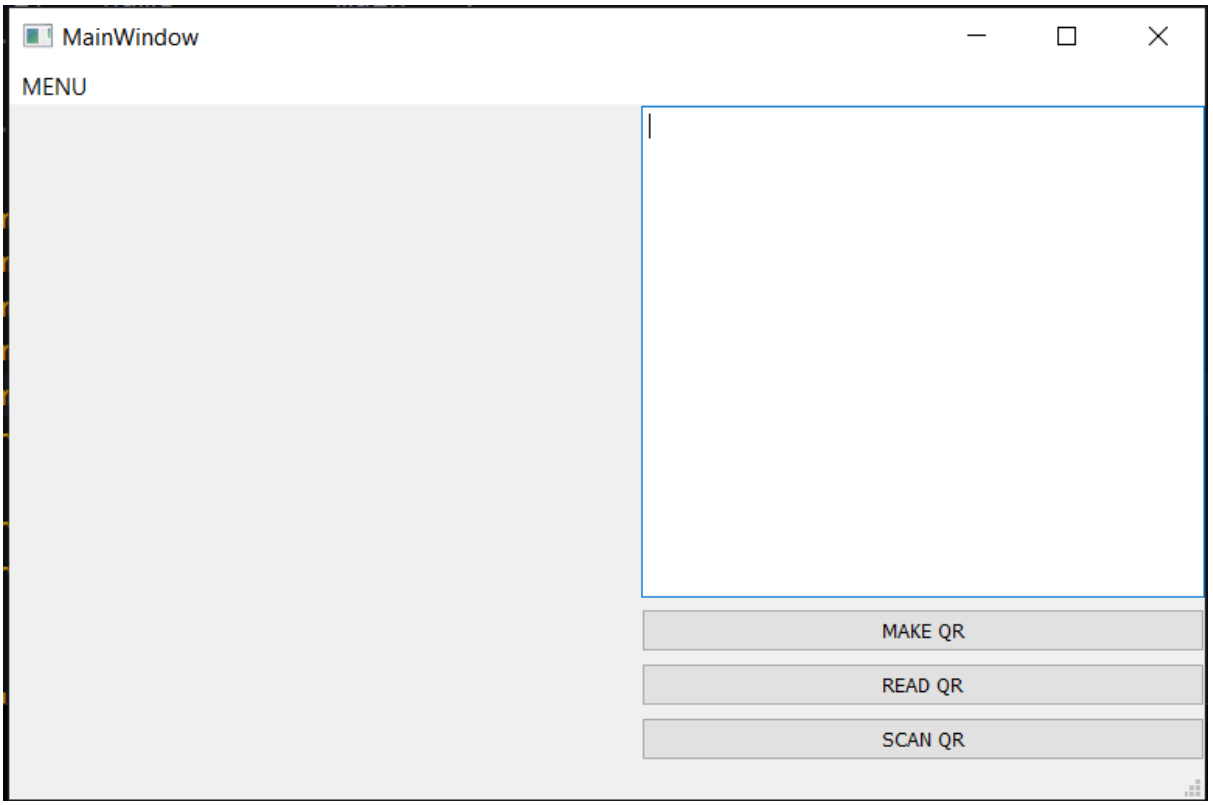
a pure Python library that reads one-dimensional barcodes and QR codes using the ‘zbar’ library, an open-source software suite for reading bar codes from various sources, such as video streams, image files and raw intensity sensors.

Pillow:

Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

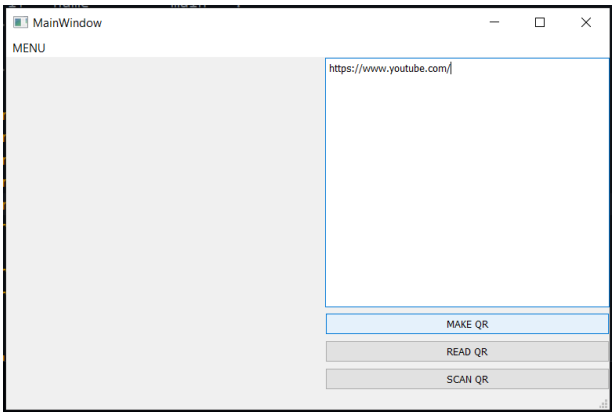
Working:

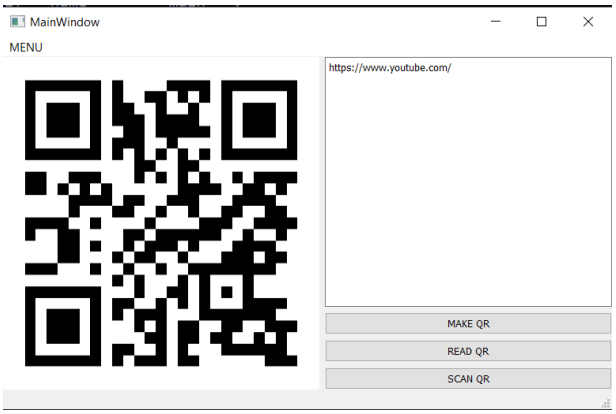
Step 1: Run the program.



A GUI appears. One can see all the options provided.

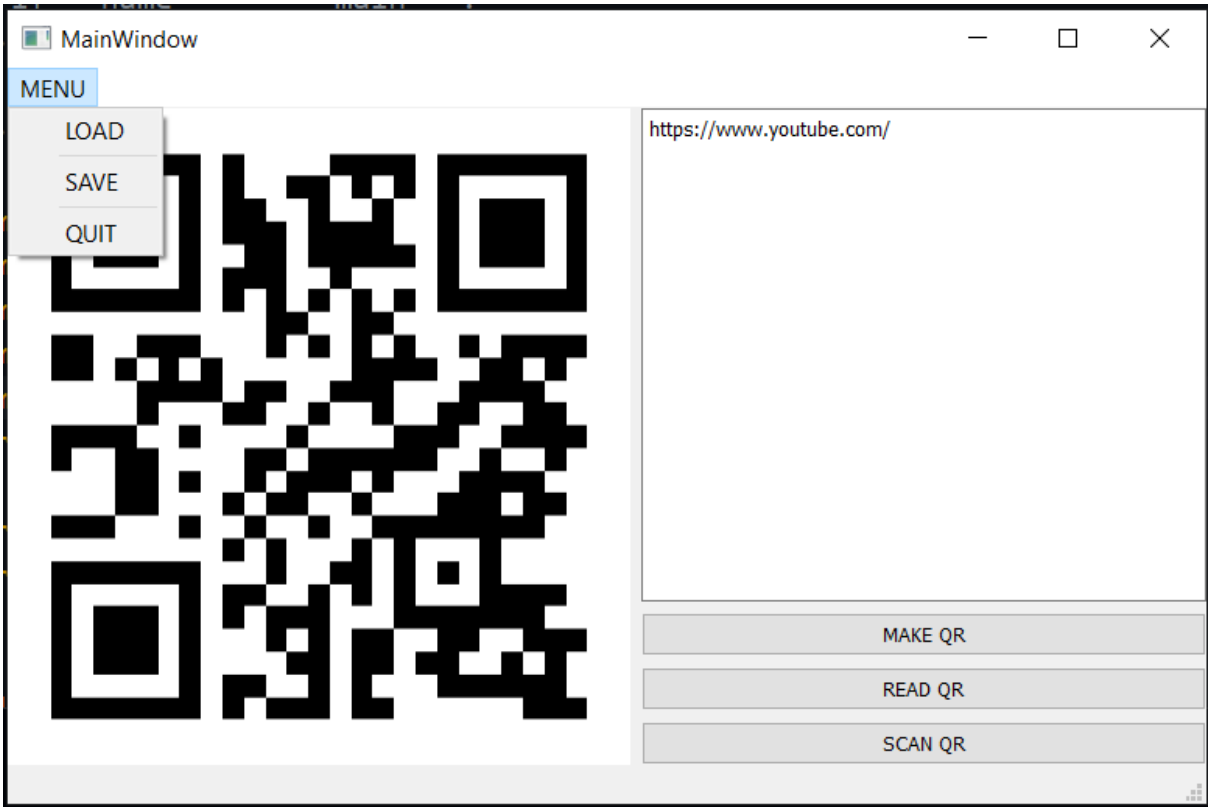
Step 2: Write the text or URL in the text field for which you want to generate the QR and click on ‘Make QR’.





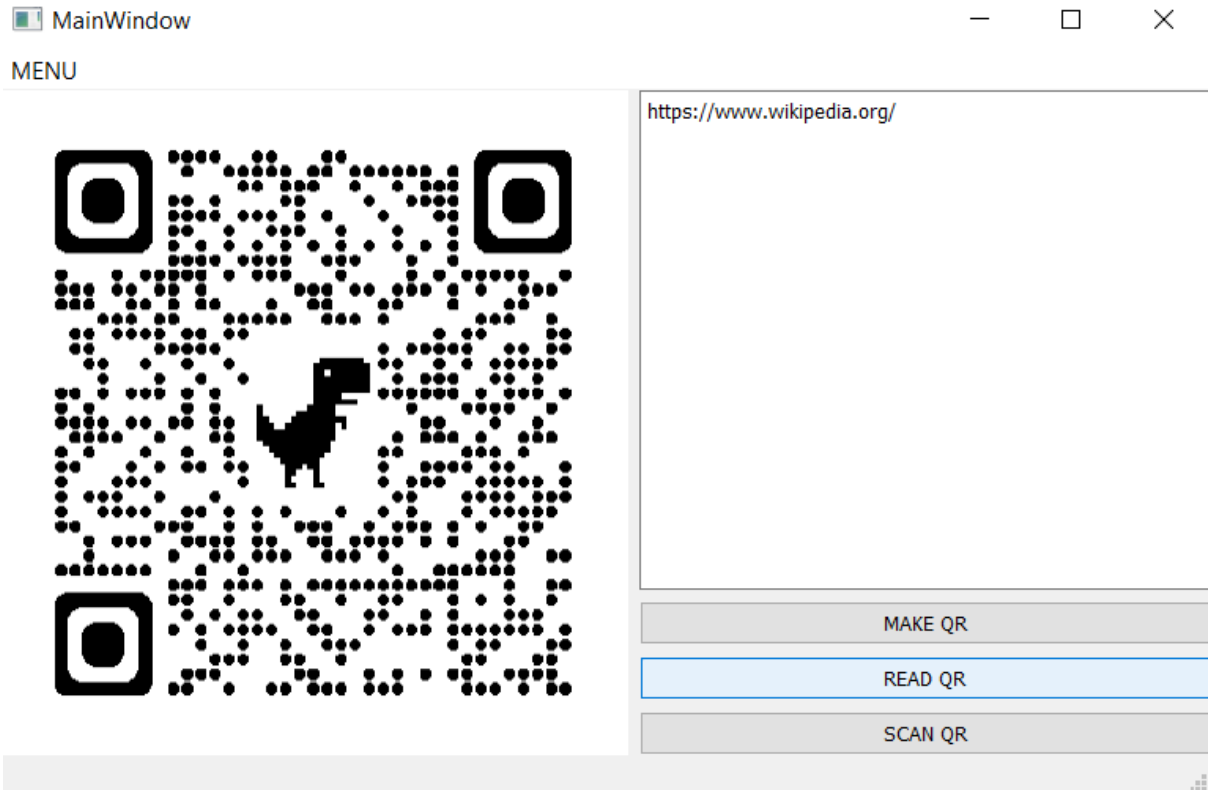
Your QR code will be generated.

Step 3: If you want to save the QR that is generated click on the 'MENU' option in the status bar and choose 'Save'.

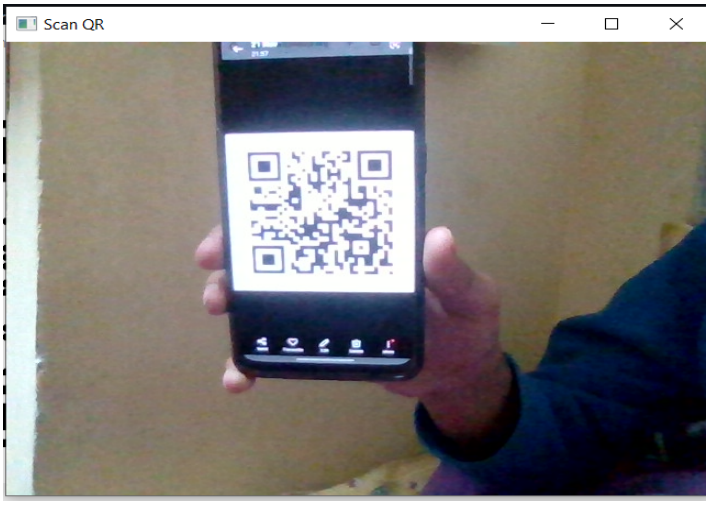


You will be able to save your QR code on your device as an image.

Step 4: If you want to read the QR code from your device then click on ‘MENU’ and click ‘Load’. You can browse your device files and choose a QR code to read.
Click on ‘Read QR’. The output will be displayed in the text field.



Step 5: If you want to scan a QR code from the device camera then click on ‘SCAN QR’. Your device camera will open and you can scan using your camera.



Step 6: If you want to exit or terminate your program click on ‘MENU’ and select option ‘Quit’. It will close all the windows and the program will exit.


```
exiting.....
```

Code:

```
import qrcode
import cv2
import webbrowser
import sys
import time
from pyzbar.pyzbar import decode

from PyQt5.QtWidgets import *
from PyQt5 import QtGui, uic


class MyGUI(QMainWindow):
    def __init__(self):
        super(MyGUI, self).__init__()
        uic.loadUi("QR_code.ui", self)
        self.show()

        self.current_file = ""
        self.actionLOAD.triggered.connect(self.load_image)
        self.actionSAVE.triggered.connect(self.save_image)
        self.pushButton.clicked.connect(self.generate_code)
        self.pushButton_2.clicked.connect(self.read_code)
        self.pushButton_3.clicked.connect(self.scan_qr)
        self.actionQUIT.triggered.connect(self.quit_window)

    def load_image(self):
        options = QFileDialog.Options()
        filename, _ = (QFileDialog.getOpenFileNames(self, "Open File",
"", "All Files (*)", options=options))
        print(filename)

        if filename != "":
            self.current_file = filename[0]
            pixmap = QtGui.QPixmap(self.current_file)
            pixmap = pixmap.scaled(400, 400)
            self.label.setScaledContents(True)
            print(pixmap)
            self.label.setPixmap(pixmap)
        else:
```

```
            print("some unexpected error occurred. TRY AGAIN!!")


    def save_image(self):
        print("saving generated qr......")
        options = QFileDialog.Options()
        filename, _ = QFileDialog.getSaveFileName(self, "Save File",
"", "PNG (*.png)", options=options)

        if filename != "":
            img = self.label.pixmap()
            img.save(filename, "PNG")

    def generate_code(self):
        print("generating qr......")
        time.sleep(1)
        qr = qrcode.QRCode(version=1,
error_correction=qrcode.constants.ERROR_CORRECT_L, box_size=20,
border=2)
        qr.add_data(self.textEdit.toPlainText())
        qr.make(fit=True)

        img = qr.make_image(fill_colr="black", back_color="white")
        img.save("currentqr.png")
        pixmap = QtGui.QPixmap("currentqr.png")
        pixmap = pixmap.scaled(400, 400)
        self.label.setScaledContents(True)
        self.label.setPixmap(pixmap)

    def read_code(self):
        print("reading qr.......")
        time.sleep(2)
        img = cv2.imread(self.current_file)
        detector = cv2.QRCodeDetector()
        data, _, _ = detector.detectAndDecode(img)

        if data:
            self.textEdit.setText(data)
        else:
            print("can't read the qr code!!")

    def quit_window(self):
        print("exiting.....")
```

```
        time.sleep(2)
        cv2.destroyAllWindows()
        sys.exit(0)


    def scan_qr(self):
        print("opening camera......")
        cam = cv2.VideoCapture(1)
        detector = cv2.QRCodeDetector()
        camera = True
        print("scanning...")
        while camera == True:
            _, frame = cam.read()
            data,one,_ = detector.detectAndDecode(frame)
            if data:
                self.textEdit.setText(data)
                a = data
                self.generate_code()
                break
            cv2.imshow("Scan QR", frame)
            if cv2.waitKey(1) == ord('q'):
                break
        b = webbrowser.open(str(a))
        # cam.release(a)
        cv2.destroyAllWindows()



def main():
    app = QApplication([])
    window = MyGUI()
    app.exec_()


if __name__ == "__main__":
    main()
```

Conclusion:

In a world where we expect everything at our fingertips, QR codes provide instant interaction between businesses and customers. QR codes were steadily growing in popularity. However, the pandemic escalated businesses' and marketers’ appetite for them, and it’s not hard to see why. QR codes help to drive traffic to landing pages, share important information and increase engagement with customers. This project helped in solving a real-life problem of generating and reading a QR code and helped us learn new and interesting about this technology.