

QR CODE GENERATOR

Presentation by Group 7
Dec 2022

TABLE OF CONTENT

- 1.** OUR TEAM
- 2.** ABSTRACT
- 3.** INTRODUCTION
- 4.** PROJECT DESIGN
- 5.** CODE WITH EXPLANATION
- 6.** RESULTS



1. OUR TEAM



JAWAD KHAN



MAHENDRA KUMAR



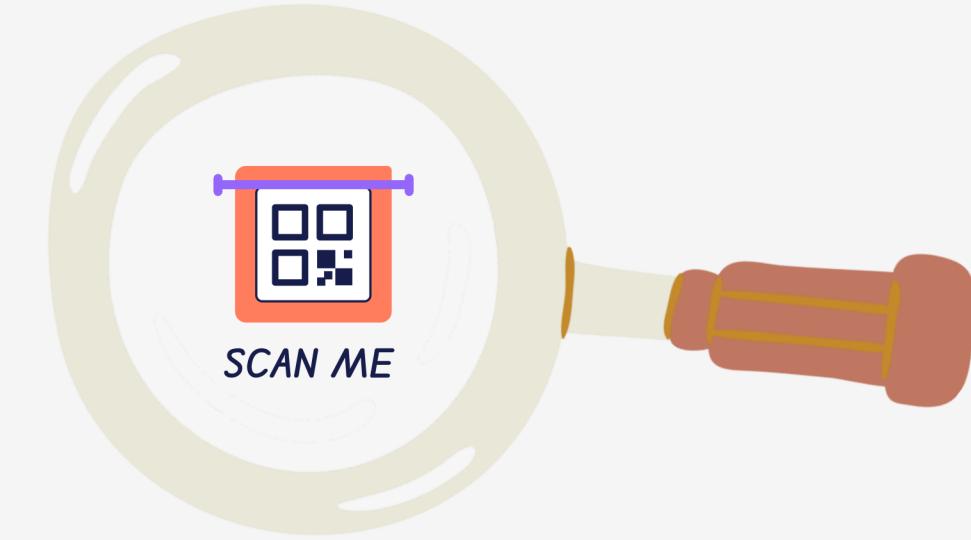
SRINIVAS



SESHA SAI

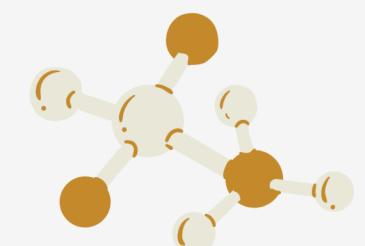


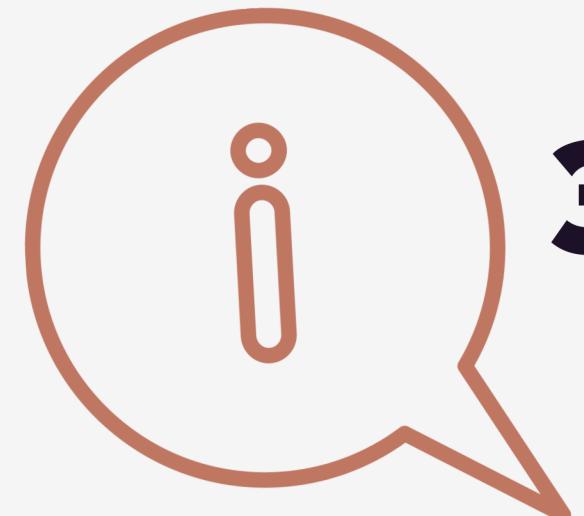
2. ABSTRACT



WHAT WE MADE ?

In this project we develop QR code generator. The use of QR code-based technologies and applications has become prevalent in recent years where QR codes are accepted to be a practical and intriguing data representation / processing mechanism amongst worldwide users.

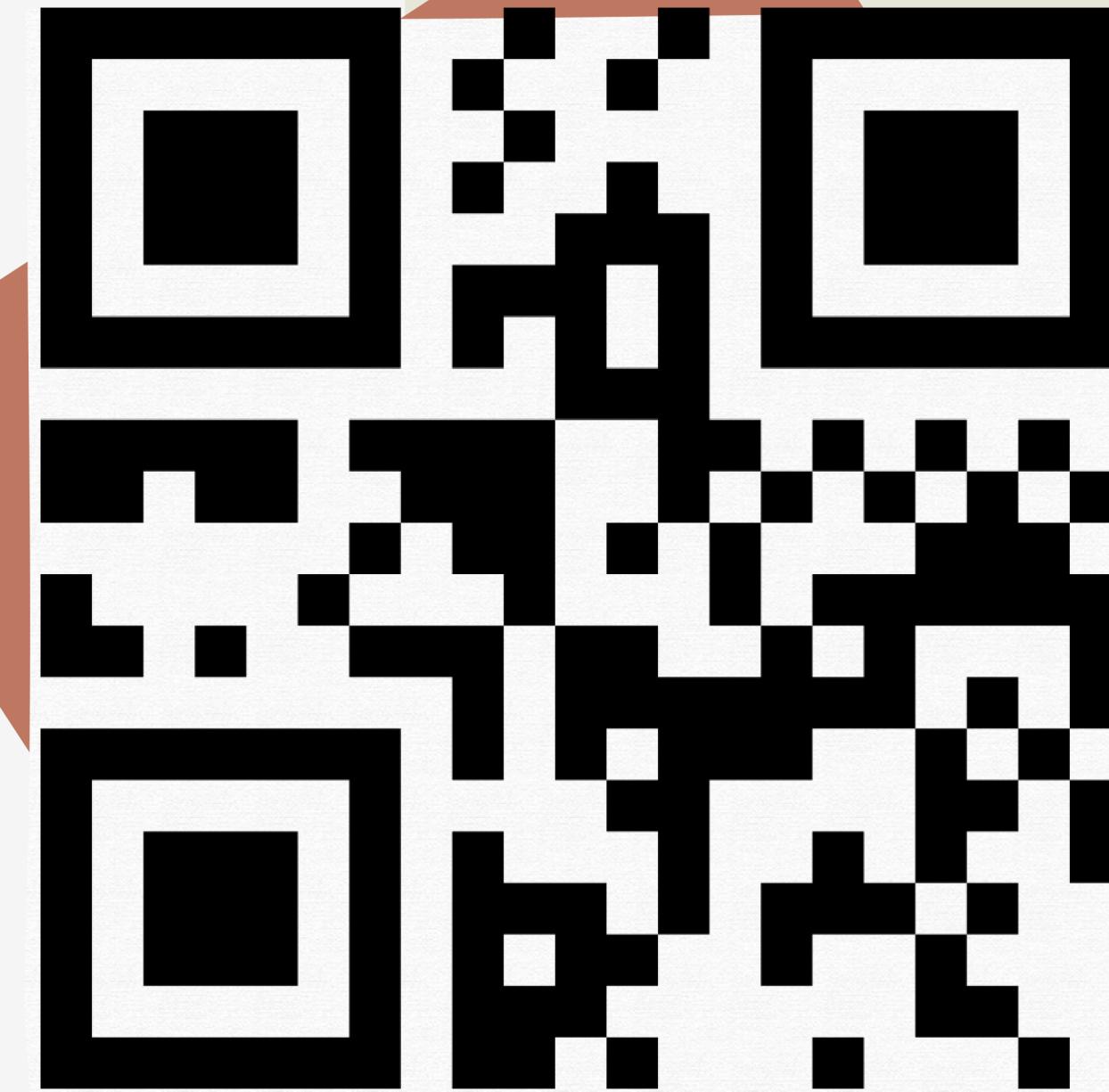




3. INTRODUCTION

QR CODE

A QR (quick response) code is a box-shaped matrix form of a 2-dimensional barcode (also called checkerboard-type barcode) that contains some meaningful data or linkage.



A barcode is an optical label with information about the item it is attached to that can be read by machines. QR code is the trademark for a type of matrix barcode

QR Code was first introduced in the automotive industry of Japan. The QR Code system has become popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC barcodes.

An active QR Code is one that a QR Code scanner can scan easily, prompting users to click on an external link upon scanning. The external link delivers relevant information that the user is looking for.



PARTS OF THE QR CODE

THE MOST IMPORTANT PARTS OF A QR CODE ARE:



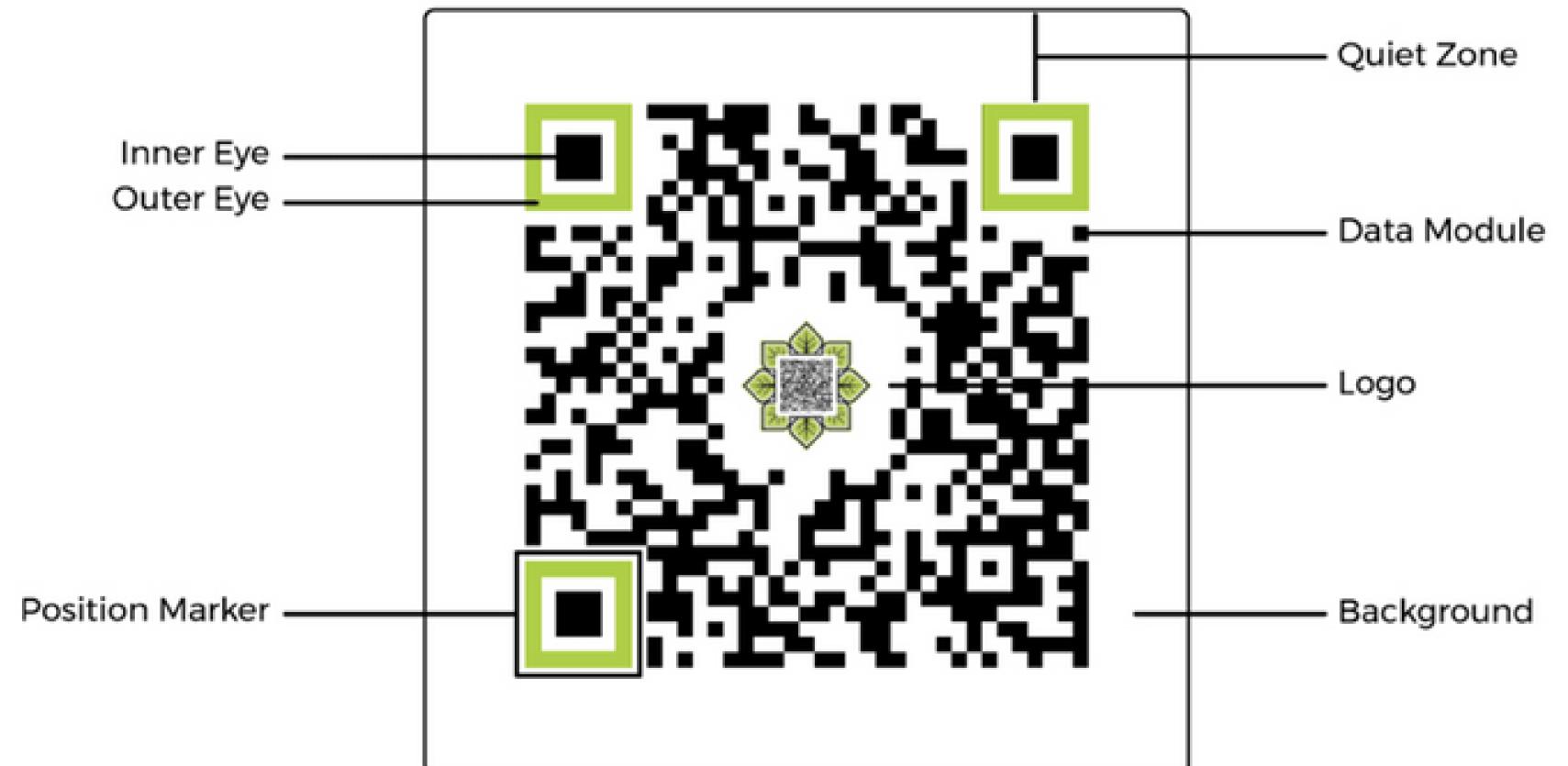
DATA MODULE.



POSITION MARKER.



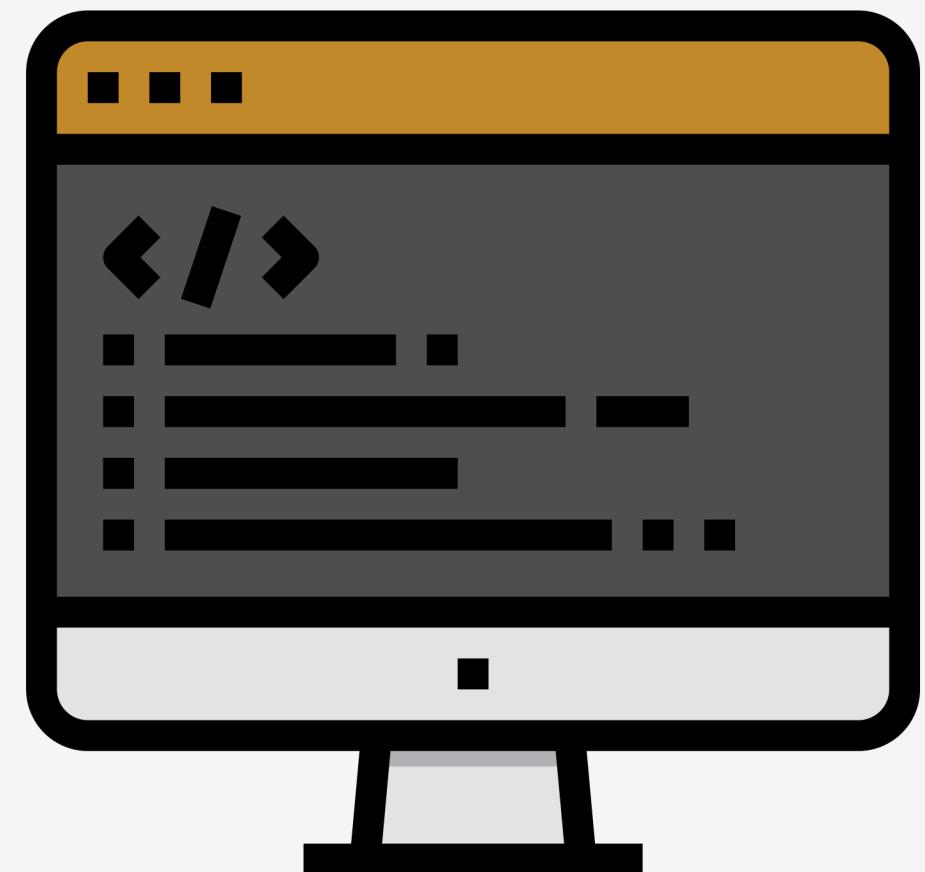
QUIET ZONE.



4. PROJECT DESIGN

This project has been designed majorly with the help of pyqr code module and the Tkinter GUI library which is available in python.

- The pyqr code module is used to create the QR code when the user gives input as a string.
- The Tkinter library is used to create a Graphical User Interface (GUI) so the output is more interactive and easily readable by the end user.



5. CODE WITH EXPLAINATION

```
✓ from tkinter import *
from tkinter import messagebox
import pyqrcode

ws = Tk()
```

- First, we will import `tkinter` and all its sub-modules using the `*`.
- Then we will also import `messagebox` and `pyqrcode`.
`Tkinter` module helps in creating GUI windows and apps where users can interact with events.
- We will generate object `ws` by assigning it with the `Tk()` constructor.



```
def generate_QR():
    if len(user_input.get()) != 0:
        global qr, img
        qr = pyqrcode.create(user_input.get())
        img = BitmapImage(data = qr.xbm(scale = 5))
    else:
        messagebox.showwarning('warning', 'All Fields are Required!')
try:
    display_code()
except:
    pass
```

This is a user defined function where all our QR code logic will reside. The `pyqrcode.create()` generates the QR code as it fetches the string through `user_input.get()` method from the text box. Also, this function will get executed when the `user_input.get()` is not equal to 0. Once the qr code gets generated using the `pyqrcode.create()`, we have to use the `BitmapImage()` and pass the `data = qr.xbm()` along with a scale size (here 5) that will generate Bitmapimage of 5x5. If the `user_input.get()` is equals 0, then the control goes to the else block where `messagebox. showwarning()` shows a warning to fill the data in that text box. Then within the try block we call the `display_code()`.



```
def display_code():
    displayimage.config(image = img,bg='white')
    output.config(text = "SUCCESSFULLY GENERATED THE QR CODE OF YOUR INPUT ", font=("Terminal",15),bg='white', fg='black')
```

This user-defined function displays the QR code and shows the message “SUCCESSFULLY GENERATED QR CODE OF YOUR INPUT”. It is a function that shows the image details as well as the status info that we can see at the bottom of that application.



```
def team_info():
    f3 = LabelFrame(ws,bg = '#00B9F1',highlightthickness=0,borderwidth=0,padx=5,pady=25)
    f3.pack()
    f3.place(x=30,y=600)

    lheading = Label(f3,text = "DONE BY: ",font=("bauhaus 93",20),bg = "#002E6E", fg='white',padx=130)
    l2 = Label(f3,text="Jawad Khan",font=("Arial rounded MT bold",20),highlightthickness=0,borderwidth=0,padx=83,bg='white')
    l3 = Label(f3,text="Srinivas M",font=("Arial rounded MT bold",20),highlightthickness=0,borderwidth=0,padx=95,bg='white')
    l4 = Label(f3,text="Mahendra Kumar",font=("Arial rounded MT bold",20),highlightthickness=0,borderwidth=0,padx=50,bg='white')
    l5 = Label(f3,text="Sesha Sai",font=("Arial rounded MT bold",20),highlightthickness=0,borderwidth=0,padx=98,bg='white')
    lheading.grid(row=0,column=0,pady=10)
    l2.grid(row=2,column=0,padx=30,pady=10)
    l3.grid(row=3,column=0,padx=30,pady=10)
    l4.grid(row=4,column=0,padx=10,pady=10)
    l5.grid(row=5,column=0,padx=10,pady=10)
```

This user-defined function is to display the names of the team members by whom this project is done. It has an object f3 of Labelframewidget. This object f3 is created inside the main frame ws. We set the required attributes of this widget. Inside this frame, we have different Label objects namely lheading,l2,l3,l4, and l5 which are used to show the names of the team members.



```

lmain=Label(ws,text= "QR CODE GENERATOR",font=("bauhaus 93",30),bg="#00B9F1",fg='white',padx=675)
lmain.pack()
lmain.place(x=0,y=20)

#******(end of main heading)

f1 = LabelFrame(ws,bg = '#00B9F1',highlightthickness=0,borderwidth=0,padx=5,pady=50)
f1.pack()
f1.place(x=30,y=100)

l1 = Label(f1,text = "ENTER ANY STRING TO GENERATE THE QR CODE ",font=("bauhaus 93",20),bg = "#002E6E", fg='white',padx=50)
l1.grid(row=0,column=0,pady=10)

user_input = StringVar()

e = Entry(f1,textvariable = user_input,font=("Terminal",20),width=35)
e.grid(row=1,column=0,pady=50)

b1 = Button(f1,text="GENERATE QR",font=("bauhaus 93",20),highlightthickness=0,borderwidth=0,command=generate_QR,padx=50,bg='white')
b2 = Button(f1,text="EXIT",font=("bauhaus 93",20),highlightthickness=0,borderwidth=0,command=ws.destroy,padx=105,bg='white')
b3 = Button(f1,text="TEAM INFO",font=("bauhaus 93",20),highlightthickness=0,borderwidth=0,command=team_info,padx=65,bg='white')
b1.grid(row=2,column=0,padx=10,pady=10)
b2.grid(row=3,column=0,padx=10,pady=10)
b3.grid(row=4,column=0,padx=10,pady=10)

#******(end of frame 1)

f2=LabelFrame(ws,highlightthickness=0,borderwidth=0,bg='white')
f2.pack()
f2.place(x=900,y=100)

displayimage = Label(f2, bg= "#002E6E")
displayimage.grid(row=0,column=0)

output = Label(f2,text = "",bg = "#002E6E")
output.grid(row=1,column=0)

#******(end of frame 2)

```

Throughout, we have to use the **Label()**, **Entry()**, and **Button()** constructors of the Tkinter module to generate the labels, text box, and buttons. We will then store these Constructor initialization in different objects namely, **lmain**, **l1**, **f1**, **f2** ,**e** ,**b1** ,**b2** ,**b3**

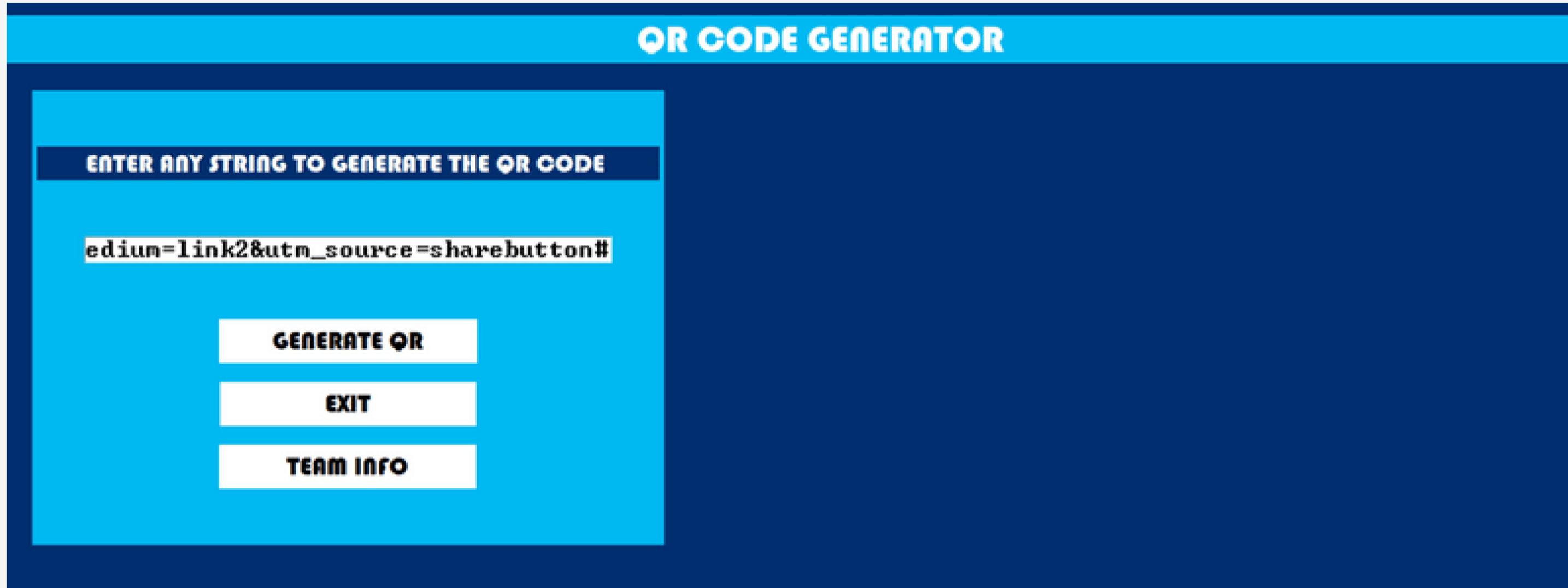


```
ws.attributes('-fullscreen',True)  
ws.config(bg="#002E6E")  
ws.mainloop()
```

Lastly, the `mainloop()` is used, which is an infinite loop implemented for running the application, waiting for an event to occur, and processing the event as long as the window does not get closed by the user manually.



6. RESULTS



The user giving an input string (link of a website) of which they wants to generate a QR Code.



QR CODE GENERATOR

ENTER ANY STRING TO GENERATE THE QR CODE

edium=link2&utm_source=sharebutton#

GENERATE QR

EXIT

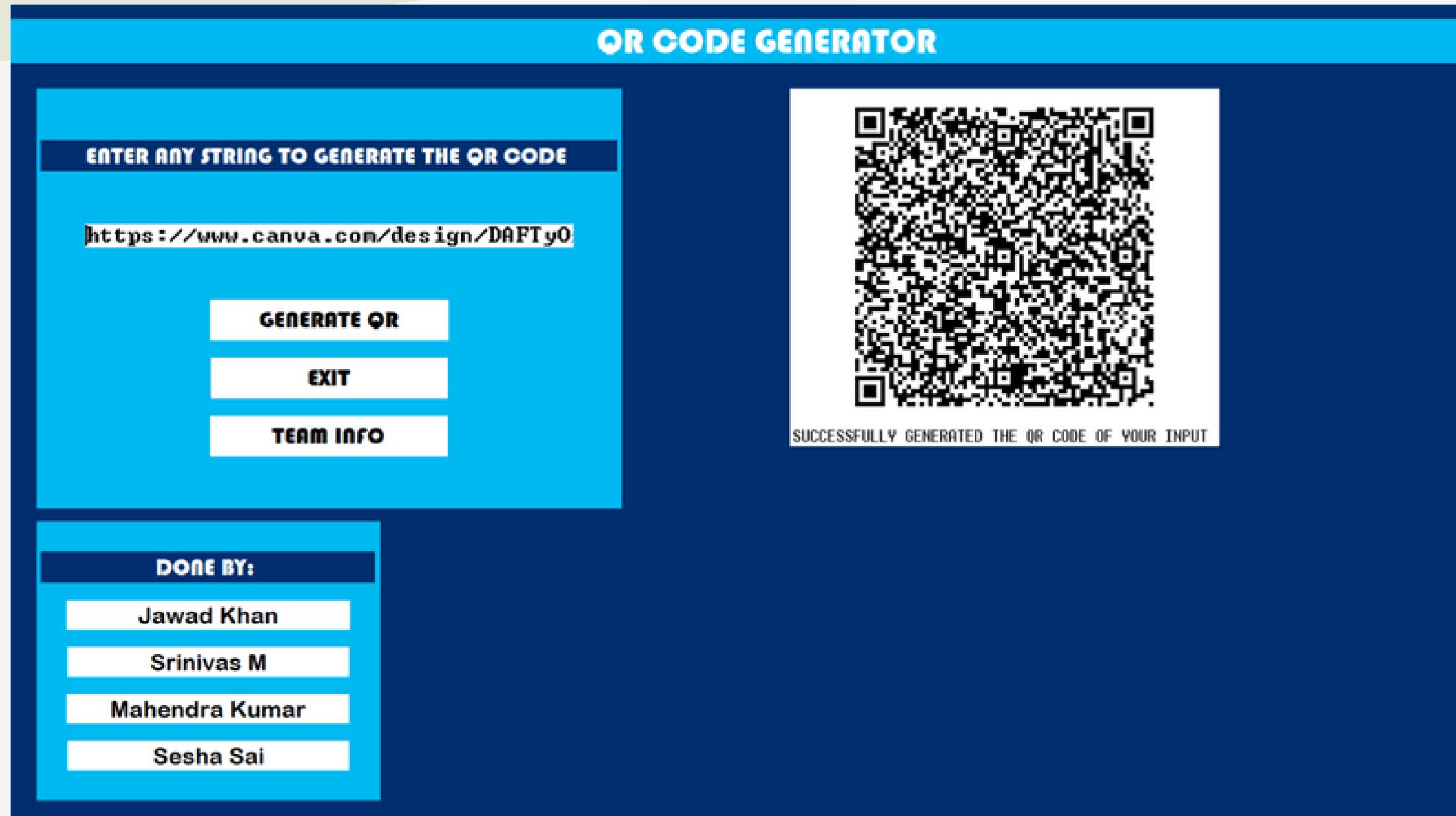
TEAM INFO



SUCCESSFULLY GENERATED THE QR CODE OF YOUR INPUT

After pasting the site link, when the user clicks on "GENERATE QR" button, a unique QR Code of the site is generated





After this, if the user clicks "EXIT" button, the QR Code Generator interface gets closed.

After generating the QR Code if the user clicks on "TEAM INFO" button, a frame displaying all the team members appears. (This step isn't absolutely necessary).



**THANK
YOU**



