

PROJECT REPORT
On
“OTP GENERATOR”

Submitted By:-
Pranay S Tiwaskar

Guided By:-
Mr. Ratnesh K. Choudhary



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**S. B. JAIN INSTITUTE OF TECHNOLOGY
MANAGEMENT AND RESEARCH, NAGPUR.**

(An Autonomous Institute, Affiliated to RTMNU, Nagpur)

2021-2022

© S.B.J.I.T.M.R Nagpur 2022

**S.B. JAIN INSTITUTE OF TECHNOLOGY MANAGEMENT AND
RESEARCH, NAGPUR**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SESSION 2021-2022

CERTIFICATE

This is to certify that the Project titled **OTP generator** is a bonafide work of **Pranay Tiwaskar** carried out for the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering in **Computer Science & Engineering**.

Mr. Ratnesh K. Choudhary

Assistant Professor

Mr. Animesh Tayal

Head of Department

INDEX

CERTIFICATE	i
INDEX	ii
LIST OF FIGURES	iii
CHAPTER 1 INTRODUCTION	1-2
1.1 PROJECT BACKGROUND	
1.2 PURPOSE OF STUDY	
CHAPTER 2 METHODOLOGY	3
2.1 PROPOSED WORK	
2.2 SYSTEM ARCHITECTURE	
CHAPTER 3 TOOLS/PLATFORMS	4-5
3.1 SOFTWARE REQUIREMENT	
CHAPTER 4 DESIGN & IMPLEMENTATION	6-9
4.1 ALGORITHM	
4.2 FLOWCHART	
4.3 SOURCE CODE	
CHAPTER 5 RESULT & DISCUSSION	10-12
5.1 OUTPUT	
5.2 DISCUSSION	
5.3 APPLICATION	
CHAPTER 6 CONCLUSION	13
REFERENCES	14

LIST OF FIGURE

FIG. NO.	TITLE OF FIGURE	PAGE NO.
2.2.1	System Architecture	3
4.1.1	Flow chart for system architecture	7
5.1.1	Program run successfully on the Terminal	10
5.1.2	OTP generator GUI opened on screen	10
5.1.3	User entered the mobile number for receiving OTP	11
5.1.4	User received an OTP on Mobile phone	11
5.1.5	User entered OTP on GUI and verified it	12

CHAPTER 1

INTRODUCTION

1.1 PROJECT BACKGROUND

A one-time password (OTP), also known as a one-time PIN, one-time authorization code (OTAC) or dynamic password, is a password that is valid for only one login session or transaction, on a computer system or other digital device. OTPs avoid several shortcomings that are associated with traditional (static) password-based authentication; a number of implementations also incorporate two-factor authentication by ensuring that the one-time password requires access to something a person has (such as a small keyring fob device with the OTP calculator built into it, or a smartcard or specific cellphone) as well as something a person knows (such as a PIN).

OTP generation algorithms typically make use of pseudorandomness or randomness to generate a shared key or seed, and cryptographic hash functions, which can be used to derive a value but are hard to reverse and therefore difficult for an attacker to obtain the data that was used for the hash. This is necessary because otherwise, it would be easy to predict future OTPs by observing previous ones.

1.2 PURPOSE OF STUDY

OTPs have been discussed as a possible replacement for, as well as an enhancer to, traditional passwords. On the downside, OTPs can be intercepted or rerouted, and hard tokens can get lost, damaged, or stolen. Many systems that use OTPs do not securely implement them, and attackers can still learn the password through phishing attacks to impersonate the authorized user.

When correctly implemented, OTPs are no longer useful to an attacker within a short time of their initial use. This differs from passwords, which may remain useful to attackers years after the fact. As with passwords, OTPs are vulnerable to social engineering attacks in which phishers steal OTPs by tricking customers into providing them with their OTPs. Also like passwords, OTPs can be vulnerable to man-in-the-middle attacks, making it important to communicate them via a secure channel, for example Transport Layer Security.

The fact that both passwords and OTP are vulnerable to similar kinds of attacks was a key motivation for Universal 2nd Factor, which is designed to be more resistant to phishing attacks.

OTPs which don't involve a time-synchronization or challenge–response component will necessarily have a longer window of vulnerability if compromised before their use. In late 2005 customers of a Swedish bank were tricked into giving up their pre-supplied one-time passwords.[16] In 2006 this type of attack was used on customers of a US bank.

The most important advantage addressed by OTPs is that, in contrast to static passwords, they are not vulnerable to replay attacks. This means that a potential intruder who manages to record an OTP that was already used to log into a service or to conduct a transaction will not be able to use it, since it will no longer be valid.[1] A second major advantage is that a user who uses the same (or similar) password for multiple systems, is not made vulnerable on all of them, if the password for one of these is gained by an attacker. A number of OTP systems also aim to ensure that a session cannot easily be intercepted or impersonated without knowledge of unpredictable data created during the previous session, thus reducing the attack surface further.

There are also different ways to make the user aware of the next OTP to use. Some systems use special electronic security tokens that the user carries and that generate OTPs and show them using a small display. Other systems consist of software that runs on the user's mobile phone. Yet other systems generate OTPs on the server-side and send them to the user using an out-of-band channel such as SMS messaging. Finally, in some systems, OTPs are printed on paper that the user is required to carry.

CHAPTER 2

METHODOLOGY

2.1 PROPOSED WORK

Here we have the Methodology of this project:

- 1) First we have to import Twilio and Random library. Also Tkinter Library, we need to import in order to make GUI for our project.
- 2) Generate a otp using a randint function of random library.
- 3) Login into a twilio account using ssid no. and token key.
- 4) Now we will get the snippet of code from the twilio official website which contain the ssid and the token key.
- 5) Just copy that snippet into the code we made for Gui of OTP.
- 6) Now execute the code
- 7) You will now see the GUI Window. Just enter your mobile number and you will receive the otp.
- 8) Now after writing the otp in the label of GUI if the otp is right you entered then it will show the message “successful” otherwise unsuccessful.

2.2 SYSTEM ARCHITECTURE

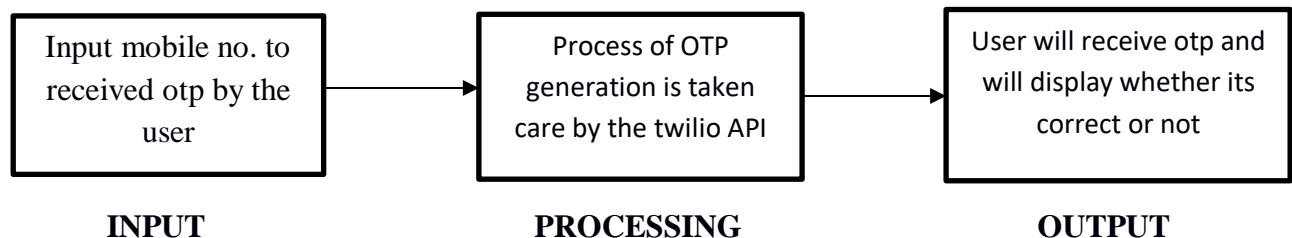


Fig- 2.2.1 System Architecture

CHAPTER 3

TOOLS/PLATFORMS

3.1 SOFTWARE REQUIREMENT

- a. **CLIENT-SIDE TECHNOLOGY:** Python3
- b. **IDE / FRAMEWORK:** Visual Studio
- c. **LIBRARIES:** tkinter, random, twilio
- d. **OPERATING SYSTEM:** Windows 11

1. Tkinter

TKinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit. Tkinter is included with standard GNU/Linux, Microsoft Windows and macOS installs of Python. The name Tkinter comes from Tk interface.

2. Random

Random module is an inbuilt module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, print random a value for a list of string, etc

3. Twilio

The Twilio Python Helper Library makes it easy to interact with the Twilio API from your Python application. The most recent version of the library can be found on PyPi. The Twilio Python Helper Library supports Python applications written in Python 3.6 and above.

4. PYTHON3

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting
- Python is easy to learn. Its syntax is easy and code is very readable.
- Python has a lot of applications. It's used for developing web applications, data science, rapid application development, and so on.
- o Python allows you to write programs in fewer lines of code than most of the programming languages.

- The popularity of Python is growing rapidly. Now it's one of the most popular programming languages.

5. VS CODE

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and available on your favorite platform - Linux, macOS, and Windows. Visual Studio Code - Code Editing.

6. SUBLIME TEXT

Sublime Text (opens new window) is a versatile, fun, and fast text editor for code and prose that automates repetitive tasks so you can focus on the important stuff. It is supported on macOS, Windows and Linux. Its versatility comes from a wide range of community-developed third-party packages that provide syntax highlighting, snippets, or other automation backed by Python (opens new window) plugins. The default distribution of Sublime Text aims to provide a basic but very functional set of features, but it can easily be turned into a full-fledged IDE, if so desired.

CHAPTER 4

DESIGN & IMPLEMENTATION

4.1 ALGORITHM

Step 1: Start

Step 2: Enter Your Mobile Number

Step 3: You will receive a otp

Step 4: Just write that otp to check whether it is correct or not

Step 5: if the otp entered by user is correct then successful otherwise will display unsuccessful

Step 6: Stop

4.2 FLOWCHART

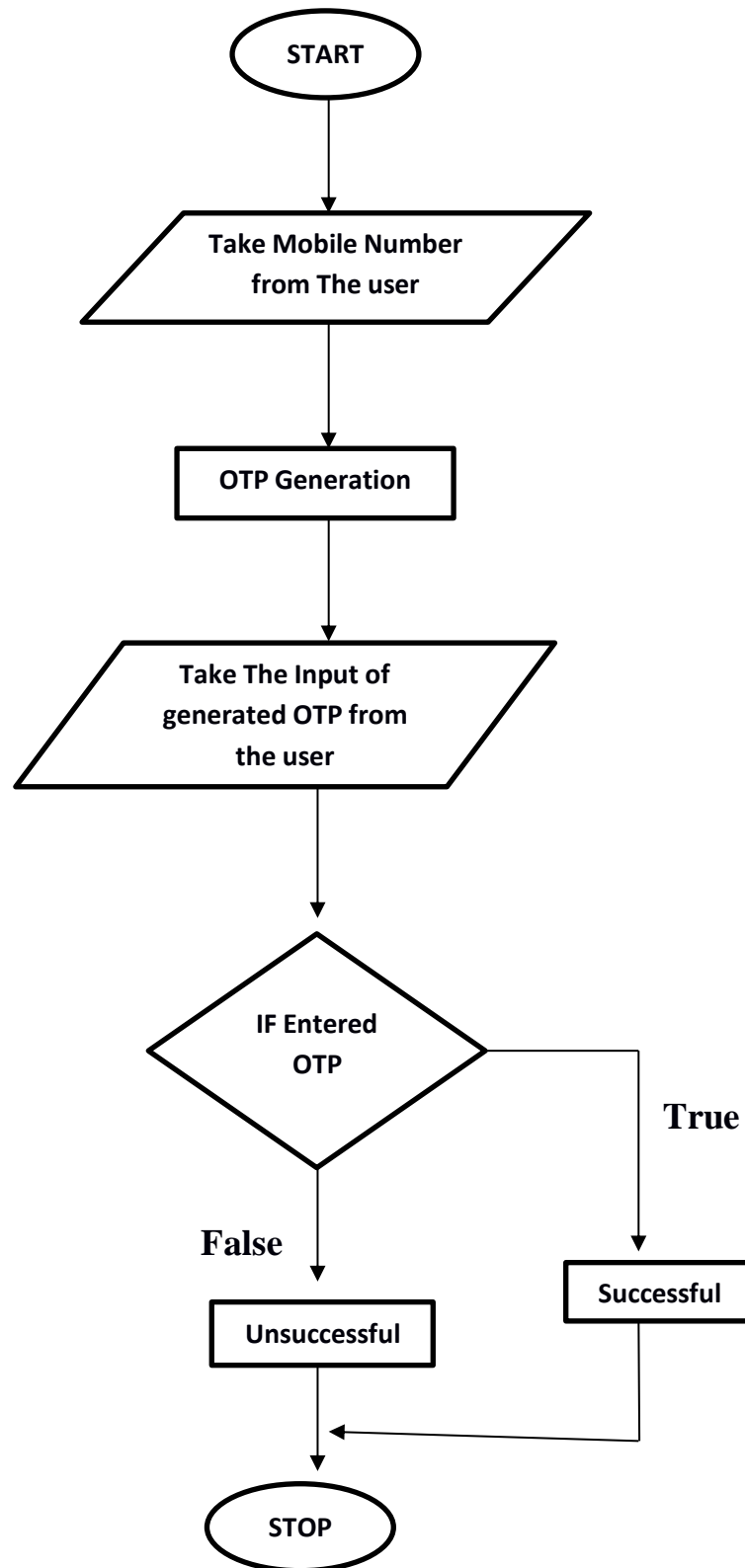


Fig 4.1.1 Flow chart of system architecture

4.3 SOURCE CODE

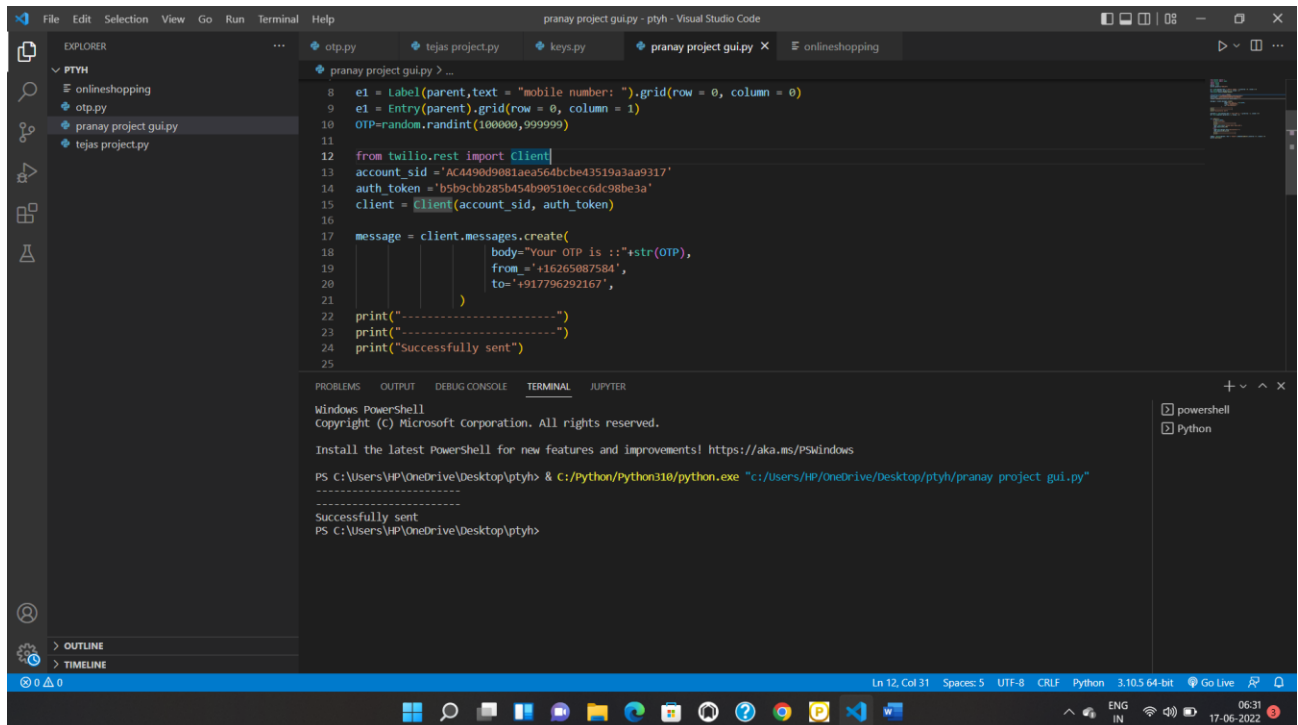
```
from tkinter import *
from tkinter import Label
import twilio
import random
parent = Tk()
parent.geometry("300x250")
e1 = Label(parent,text = "mobile number: ").grid(row = 0, column = 0)
e1 = Entry(parent).grid(row = 0, column = 1)
OTP=random.randint(100000,999999)
from twilio.rest import Client
account_sid ='AC4490d9081aea564bcbe43519a3aa9317'
auth_token ='b5b9cbb285b454b90510ecc6dc98be3a'
client = Client(account_sid, auth_token)
message = client.messages.create(
    body="Your OTP is ::"+str(OTP),
    from_='+16265087584',
    to='+917796292167',
)
print("-----")
print("-----")
print("Successfully sent")
password = Label(parent,text = "your OTP is: ").grid(row = 1, column = 0)
e2 = Entry(parent).grid(row = 1, column = 1)
def submit1():
    if e2==str(OTP):
        print("verified")
        print("-----")
        print("-----")
        lab1= Label(text='Please Enter Valid OTP')
        lab1.place(x=60,y=90)
    else:
        lab2=Label(parent,text="Successfull!!")
        lab2.place(x=100,y=110)
        print("-----")
```

```
print("-----")  
submit = Button(parent, text = "Submit",command=submit1).grid(row = 4, column = 0)  
parent.mainloop()
```

CHAPTER 5

RESULT & DISCUSSION

5.1 OUTPUT



The screenshot shows the Visual Studio Code interface with a Python script named `pranay project gui.py` open in the editor. The script uses the `twilio.rest` library to send an OTP. The terminal at the bottom shows the command `python.exe "c:/Users/HP/OneDrive/Desktop/ptyh/pranay project gui.py"` being executed, followed by the output `Successfully sent`. The Explorer sidebar on the left shows the project structure with files like `onlineshopping`, `otp.py`, `pranay project gui.py`, and `tejas project.py`.

```
8 e1 = Label(parent, text = "mobile number: ").grid(row = 0, column = 0)
9 e1 = Entry(parent).grid(row = 0, column = 1)
10 OTP = random.randint(100000, 999999)
11
12 from twilio.rest import Client
13 account_sid = 'AC4490d9081aea564bcbcd3519a3aa9317'
14 auth_token = 'b5b9cbb285b454b90510eccdc98be3a'
15 client = Client(account_sid, auth_token)
16
17 message = client.messages.create(
18     body="Your OTP is :"+str(OTP),
19     from_='+16265087584',
20     to='+917796292167',
21 )
22 print("-----")
23 print("-----")
24 print("Successfully sent")
25
```

Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\HP\OneDrive\Desktop\ptyh> & C:/Python/Python310/python.exe "c:/Users/HP/OneDrive/Desktop/ptyh/pranay project gui.py"

Successfully sent
PS C:\Users\HP\OneDrive\Desktop\ptyh>

Fig- 5.1.1 Program run successfully on the Terminal

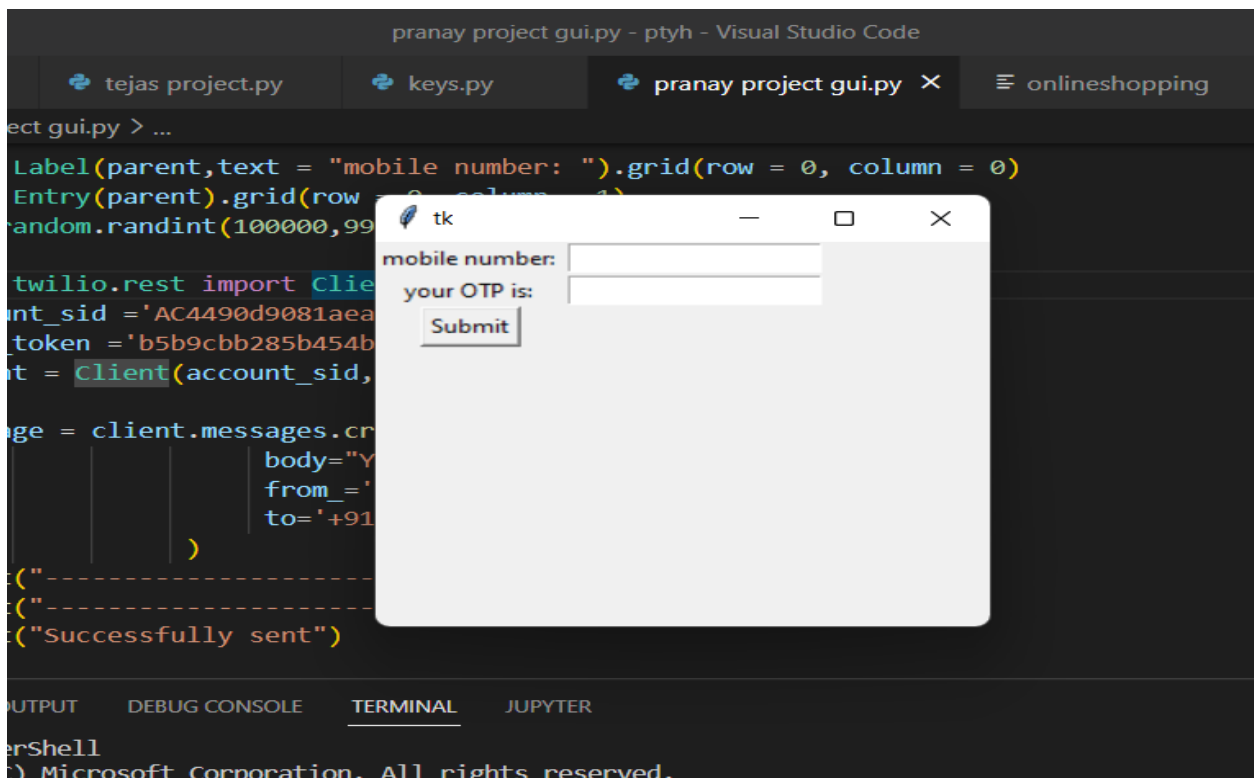


Fig- 5.1.2 OTP generator GUI opened on screen

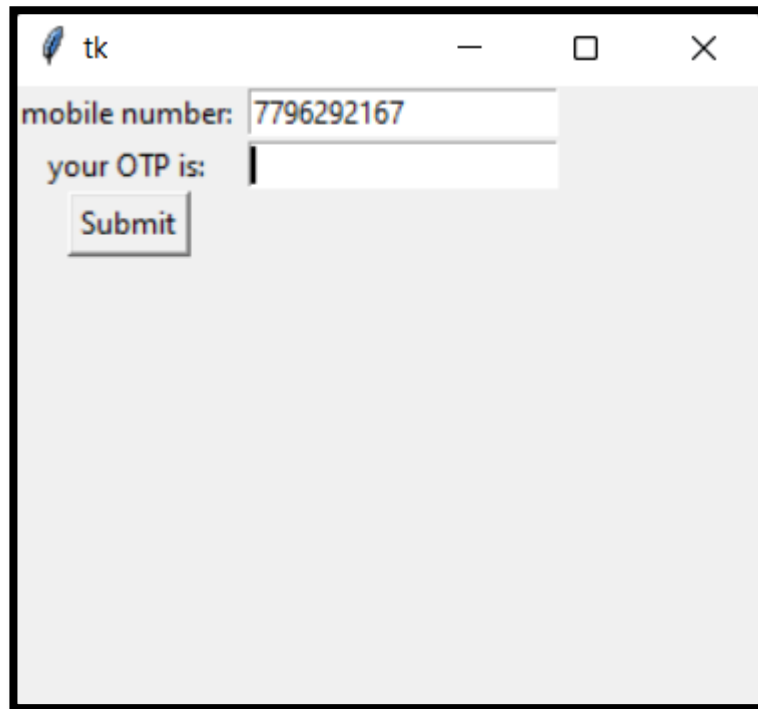


Fig- 5.1.3 User entered the mobile number for receiving OTP

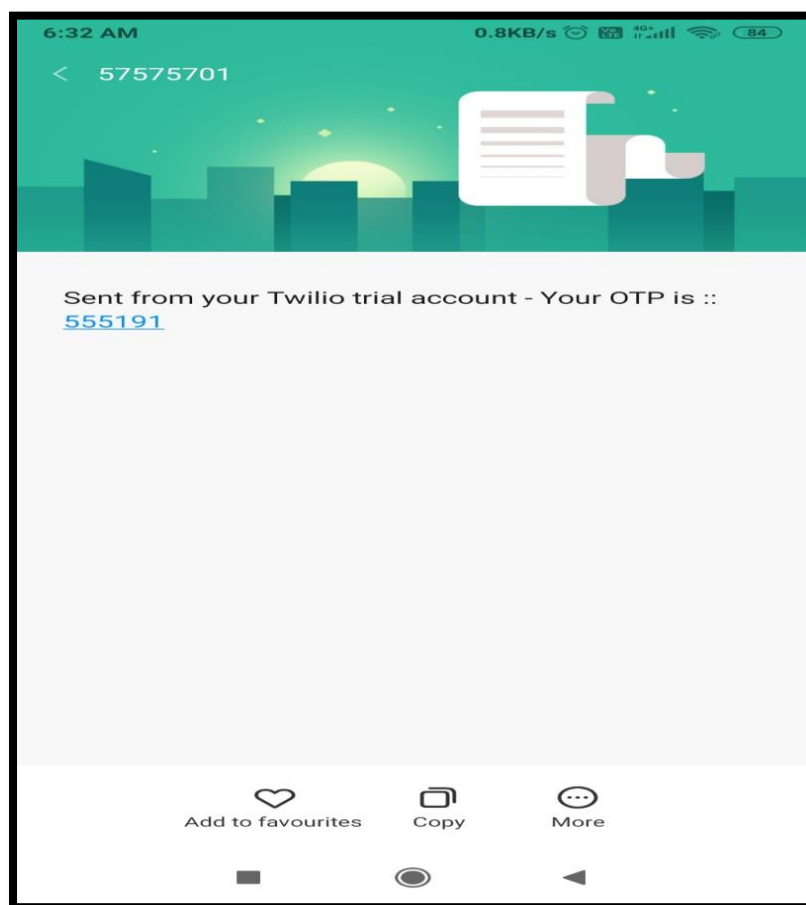


Fig- 5.1.4 User received an OTP on Mobile phone

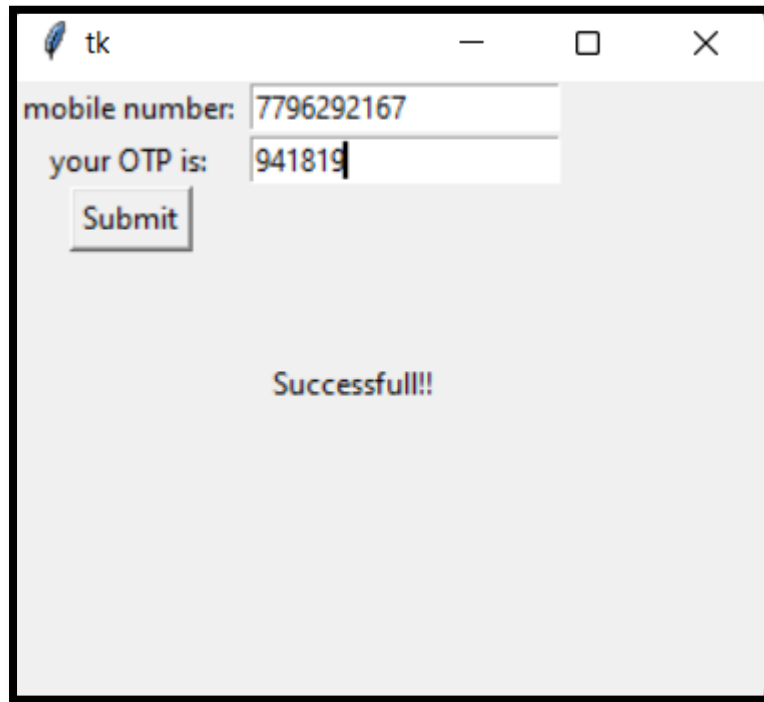


Fig- 5.1.5 User entered OTP on GUI and verified it

5.2 DISCUSSION

As we have seen output of our project:

This project is totally based on generating the otp. In this project I have also created a GUI using the Tkinter library. Then to verify it as for the securely login purpose, I have used the twilio API. Just registered on this API and used the ssid number and token key for receiving the otp. Also I have used the if-else statements to check the user entered otp is correct or not, if its correct then screen will display successful otherwise unsuccessful. We can also use another API's for this project but I have used this twilio as it is very easy to use this API.

5.3 APPLICATION

The OTP feature prevents some forms of identity theft by making sure that a captured user name/password pair cannot be used a second time. Typically the user's login name stays the same, and the one-time password changes with each login. One-time passwords are a form of strong authentication, providing much better protection to eBanking, corporate networks, for any application or website login and other systems containing sensitive data. They are popularly used in the Banking sectors for secure payments. It is used in a multifactor authentication (MFA) process to secure access to data.

CHAPTER 6

CONCLUSION

We have designed and developed OTP Generator and verifier on GUI by applying **engineering knowledge**. This Project allows user to proceed further into the websites or any other logins after entering the correct OTP(One Time Password) received by him on his mobile. After executing the program The Gui will be displayed on the screen .User have to enter their mobile number to receive an OTP. Then after receiving the otp it has to be entered in the label in same GUI so as to make a successful login. After the user enter correct otp, the message will be displayed on the GUI telling the user that he have entered the right otp.

REFERENCE

- 1) https://en.wikipedia.org/wiki/One-time_password#Use
- 2) <https://www.youtube.com/watch?v=qt1OKBR-pBs&t=230s>
- 3) <https://www.techtarget.com/searchsecurity/definition/one-time-password-OTP>
- 4) <https://www.geeksforgeeks.org/python-program-to-generate-one-time-password-otp/>
- 5) <https://code.visualstudio.com/docs/python/python-tutorial>

