

PROJECT REPORT
On
“OTP GENERATOR”

Submitted By:-

Divesh C Lambat

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 **Divesh Lambat** 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
CHAPTER 2 METHODOLOGY	3
CHAPTER 3 TOOLS/PLATFORMS	4-5
CHAPTER 4 DESIGN & IMPLEMENTATION	6-8
4.1 ALGORITHM	
4.2 FLOWCHART	
4.3 SOURCE CODE	
CHAPTER 5 RESULT & DISCUSSION	9-11
5.1 OUTPUT	
5.2 DISCUSSION	
5.3 APPLICATION	
CHAPTER 6 CONCLUSION	12
REFERENCES	13

LIST OF FIGURE

FIG. NO.	TITLE OF FIGURE	PAGE NO.
4.2.1	Flowchart	7
5.1.1	The otp send successfully in the registered number.	9
5.1.2	While check verefied otp is correct or not (correct otp)	9
5.1.3	While check verefied otp is correct or not (not correct otp)	10

CHAPTER 1

INTRODUCTION

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.

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

Here you have to write Methodology of your project.

- 1) First we have to import Twilio and Random library.
- 2) And then register a number in twilio account.
- 3) Genarate a otp using a randint function of random library.
- 4) Login into a twilio account using ssid no and token key.
- 5) Now we will get the snippet of code from the twilio official website which contain the ssid and the token key.
- 6) Just copy that snippet into the code we made for OTP.
- 7) Now execute the code
- 8) You will now , Just enter your mobile number and you will receive the otp.
- 9) Now after writing the otp in the output section , if the otp is right you entered then it will show the message ‘successful’ otherwise unsuccessful.

CHAPTER 3

TOOLS/PLATFORMS

3.1 SOFTWARE REQUIREMENT

- a. **CLIENT-SIDE TECHNOLOGY:** Python
 - b. **SERVER-SIDE TECHNOLOGY:** Python
 - c. **IDE / FRAMEWORK:** Visual Studio.
 - d. **LIBRARIES:** Random
 - e. **OPERATING SYSTEM:** Windows 11
1. **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
 2. **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.
 3. **PYTHON**

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.
- 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.

4. **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.

CHAPTER 4

DESIGN & IMPLEMENTATION

4.1 ALGORITHM

Step 1: Start

Step 2: Enter your number.

Step 3: Then otp will be send in registered mobile number.

Step 4: Now you can entered your received otp.

Step 5: The result will be displayed successfully.

Step 6: And then verified it the otp will be correct or not.

4.2 FLOWCHART

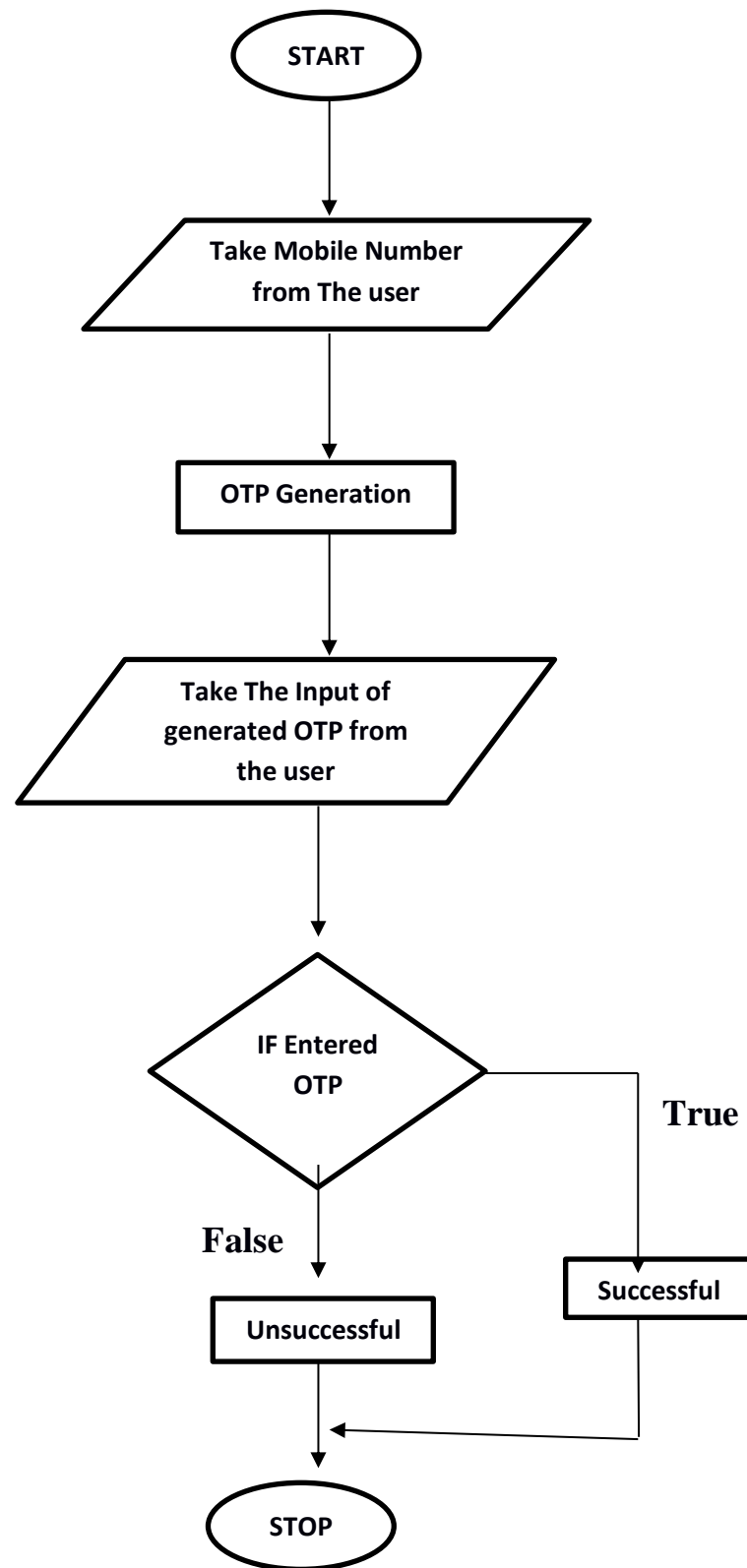


Fig.4.2.1 Flowchart

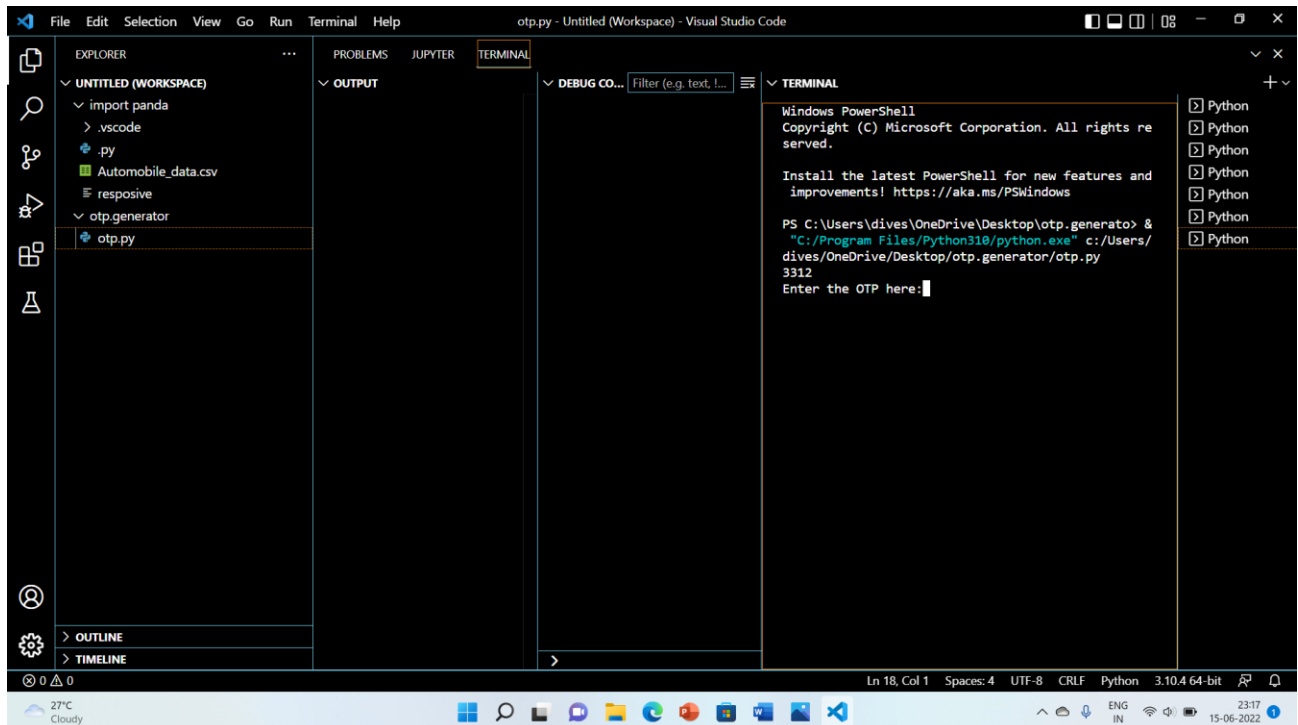
4.3 SOURCE CODE

```
import twilio
from twilio.rest import Client
import random
otp="".join([str(random.randint(0,9)) for i in range(4)])
print(otp)
Client=Client("AC892de81ef8f9dcf7f7ae04ac84dac799", "deb4540b03069adfe2385ea5418234ea" )
my_no="+917719818494"
my_twilio="+19706590551"
messages=Client.messages(f'Your otp is {otp} Please verify it....')
Client.messages.create(to=my_no, from_= my_twilio, body=messages)
a=input("Enter the OTP here:")
a1=int(a)
b=int(otp)
if a1==b:
    print("entered otp is correct")
else:
    print("Entered OTP is not correct")
```

CHAPTER 5

RESULT & DISCUSSION

5.1 OUTPUT

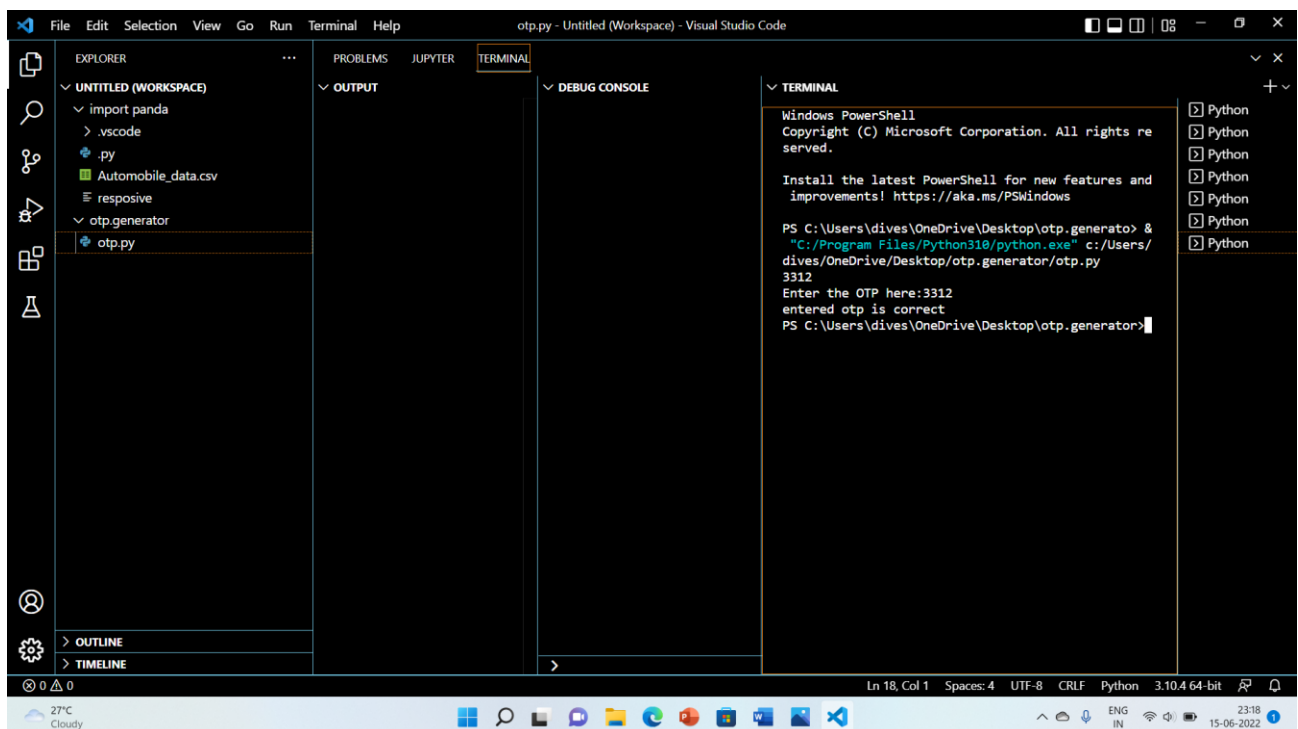


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\dives\OneDrive\Desktop\otp.generator> & "C:/Program Files/Python310/python.exe" c:/Users/dives/OneDrive/Desktop/otp.generator/otp.py
3312
Enter the OTP here:
```

Fig- 5.1.1 The otp send successfully in the registered number.



```
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\dives\OneDrive\Desktop\otp.generator> & "C:/Program Files/Python310/python.exe" c:/Users/dives/OneDrive/Desktop/otp.generator/otp.py
3312
Enter the OTP here:3312
entered otp is correct
PS C:\Users\dives\OneDrive\Desktop\otp.generator>
```

Fig.5.1.2 While check veriefed otp is correct or not (correct otp)

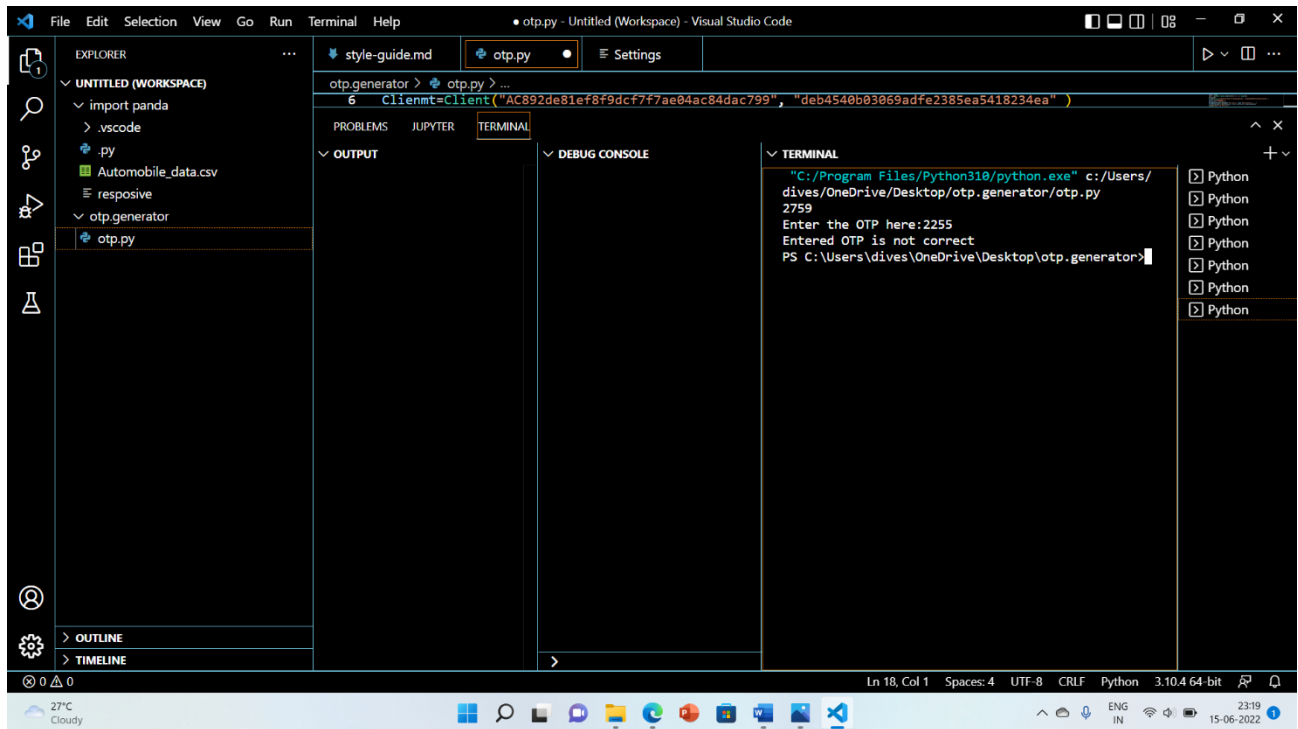


Fig- 5.1.3 While check veriefied otp is correct or not (not correct otp)

5.2 DISCUSSION

As we have seen output of our project:

One-time Passwords (OTP) is a password that is valid for only one login session or transaction in a computer or a digital device. Now a days OTP's are used in almost every service like Internet Banking, online transactions, etc. They are generally combination of 4 or 6 numeric digits or a 6-digit alphanumeric.

random() function can be used to generate random OTP which is predefined in random library. Let's see how to generate OTP using Python.

UsedFunction:

random.random(): This function returns any random number between 0 to 1.

math.floor(): It returns floor of any floating number to a integer value.

Using the above function pick random index of string array which contains all the possible candidates of a particular digit of the OTP.

5.3 APPLICATION

The most important advantage addressed by OTPs is that, in contrast to static passwords, they are not vulnerable to replay attack. 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. 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 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 6

CONCLUSION

The OTP generated is **valid for a limited duration** from the time of its generation. On the expiry of the time limit, the OTP can no longer be used to authenticate the transaction and must be regenerated. The freshly generated OTP is unique and unrelated to the previous OTP, also valid for a specified period from the time of its generation.

The OTP is a numeric code that is randomly and uniquely generated during each authentication event. This adds an additional layer of security, as the password generated is fresh set of digits each time an authentication is attempted and it offers the quality of being unpredictable for the next created session.

REFERENCE

1. <https://youtu.be/VMP1oQOxfM0>
2. <https://www.geeksforgeeks.org/python-gui-tkinter/>
3. <https://stackoverflow.com/questions/6920302/how-to-pass-arguments-to-a-button-command-in-tkinter>
4. https://www.tutorialspoint.com/python/tk_colors.htm#:~:text=The%20colors%20%22white%22%2C%20%22magenta%22%20will%20always%20be%20availabl

