Roshan Jain

Project Report

**OTP generator**

OTP is used in a lot of works, like resetting password, email verification, mobile verification and a lot.

If you are developing something in which you need to register users, you are going to need to generate OTP.

OTP is expected to be hard to guess and it should be random. It can be all digit, all character, or mixture of it. In this article, I'll talk about generating numeric OTP and alphanumeric OTP.

**Approach**

We will use python's **random function** to generate random numbers and this number will be used to generate our OTP.

**Implementation**

Generating 4 digit, all numeric OTP.

Here we have used Python tkinter module where we generate an OTP and check the credentials if it matches we the OTP generated a dialog box of Login successful appears else a dialog box of Login denied will appear

OTP is generated using a user defined function otpgenerator() and random module where to get a 4 digit OTP we apply a for loop of range 4 and randint function using random numbers from 1-9.This will generate a unique 4 digit OTP everytime we call the function

In the Tkinter part of the code we have labels and entry boxes for name ,city and password whereas a radiobutton for Gender.

**Password Generator**

A password, sometimes called a passcode, is a memorized secret, typically a string of characters, usually used to confirm the identity of a user, In other words is a string of characters used to verify the identity of a user during the authentication process.

**Approach**

Random Module:

Random module is used to perform the random generations. We are making use of random.sample module here. If you will observe in the output all characters will be unique. Random.sample() never repeats characters. If you don’t want to repeat characters or digits in the random string, then use random.sample() but it is less secure because it will reduce the probability of combinations because we are not allowing repetitive letters and digits.

String Module:

The string module contains a number of useful constants, classes and a number of functions to process the standard python string.

1. string.ascii\_letters: Concatenation of the ascii (upper and lowercase) letters
2. string.ascii\_lowercase: All lower case letters
3. string.ascii\_uppercase: All Upper case letters
4. string.digits: The string ‘0123456789’.
5. string.punctuation: String of ASCII characters which are considered punctuation characters in the C locale.

**Implementation**

Here we have made use of user defined python functions to generate 4 types of passwords

1. Only letters.

2. Capital and small letters

3. Capital and small letters and numbers

4. Capital and small letters and numbers and special characters

First we ask the user the length of the password that needs to be generated.

A condition is specifed where length has to be > 8 else it will show a message weak password and code will be stopped

Once we give the length more than 8 four options as mentioned above appear

For each type of password to be generated we have to define the data. We will make use of string module for the same.

lower = string.ascii\_lowercase  
upper = string.ascii\_uppercase  
num = string.digits  
symbols = string.punctuation

We have stored lowercase and uppercase letters along with numbers and symbols.We then combine and store the data in the all variable. Now that we have the data, we make use of random module to finally generate the password.

We are passing in the combined data along with the length of the password, and joining them at the end.Finally we print the password