Python programming internship

Project name

Password Generator

Name – sandhya pal

Introduction

This project is password genrator and password is not good for a system that demands high confidentiality and security of user credentials. It turns out that people find it difficult to make up a strong password that is strong enough to prevent unauthorized users from memorizing it.

Objective

This project task is to design and build a Python program that generates strong, secure passwords. These passwords should meet modern security standards and be suitable for various applications.

Requirements

- Create a Python script that generates random passwords.
- Ensure the passwords are a mix of uppercase and lowercase letters, numbers, and special characters.
- Allow users to specify the length and number of passwords to generate.

Coding

```
import random
import array

# maximum length of password needed
# this can be changed to suit your password length
MAX_LEN = 40

# declare arrays of the character that we need in out password
# Represented as chars to enable easy string concatenation
DIGITS = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
LOCASE_CHARACTERS = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',
                     'i', 'j', 'k', 'm', 'n', 'o', 'p', 'q',
                     'r', 's', 't', 'u', 'v', 'w', 'x', 'y',
                     'z']

UPCASE_CHARACTERS = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H',
                     'I', 'J', 'K', 'M', 'N', 'O', 'P', 'Q',
                     'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y',
                     'Z']

SYMBOLS = ['@', '#', '$', '%', '=', ':', '?', '.', '/', '|', '~', '>',
           '*', '(', ')', '<']

# combines all the character arrays above to form one array
COMBINED_LIST = DIGITS + UPCASE_CHARACTERS + LOCASE_CHARACTERS + SYMBOLS
```

```
# randomly select at least one character from each character set above
rand_digit = random.choice(DIGITS)
rand_upper = random.choice(UPCASE_CHARACTERS)
rand_lower = random.choice(LOCASE_CHARACTERS)
rand_symbol = random.choice(SYMBOLS)

# combine the character randomly selected above
# at this stage, the password contains only 4 characters but
# we want a 40-character password
temp_pass = rand_digit + rand_upper + rand_lower + rand_symbol


# now that we are sure we have at least one character from each
# set of characters, we fill the rest of
# the password length by selecting randomly from the combined
# list of character above.
for x in range(MAX_LEN - 5):
    temp_pass = temp_pass + random.choice(COMBINED_LIST)
```

```
# convert temporary password into array and shuffle to
    # prevent it from having a consistent pattern
    # where the beginning of the password is predictable
    temp_pass_list = array.array('u', temp_pass)
    random.shuffle(temp_pass_list)

# traverse the temporary password array and append the chars
# to form the password
password = ""
for x in temp_pass_list:
        password = password + x

# print out password
print(password)
```

Output:

```
PS C:\Users\admin1\san> & "C:/Program Files/Python310/python.exe" c:/Users/admin1/san/sam.py

ab1pKTjd=8?xtSs|xE2JAJq:tJBCFgW|eyG2RtUR

PS C:\Users\admin1\san> & "C:/Program Files/Python310/python.exe" c:/Users/admin1/san/sam.py

~NrP<hO%oqg1=?NUFrhr2Tvs#8xDb3Juo4TgSIfh

PS C:\Users\admin1\san> & "C:/Program Files/Python310/python.exe" c:/Users/admin1/san/sam.py

OnYZGvgZ@odc.)#*W6e*WSu0~Kgk.)dh3K?:gmIT

PS C:\Users\admin1\san> & "C:/Program Files/Python310/python.exe" c:/Users/admin1/san/sam.py

2NNORppg?TA*g2K@SNa$8s*Rag$p<ob)muB$BuIr

PS C:\Users\admin1\san> & "C:/Program Files/Python310/python.exe" c:/Users/admin1/san/sam.py

t(G/1gDF5kCI.qs#D>bEZ)Y?:1p/Cja:?U)v0xZ#

PS C:\Users\admin1\san>
```

Thank you