

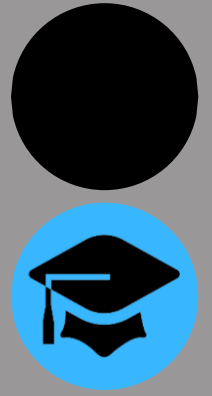


FORGE YOUR AMBITION

HUNAR INTERN

WWW.HUNARINTERN.LIVE

LET'S GET STARTED



Task- 1(Easy)

Name: Leburi Sriram

Date: 04/03/2024

Task: Create a Password Strength Checker

Description:

A Python program to analyze passwords and determine if they are weak, okay, or strong. Assess password strength based on length and other factors.

Steps to Follow:

•Write Code:

```
import re
```

```
class PasswordStrengthChecker:
```

```
    def __init__(self, password):
```

```
        self.password = password
```

```
    def check_length(self):
```

```
        return len(self.password) >= 8
```

```
def check_digits(self):  
    return bool(re.search(r"\d", self.password))
```

```
def check_uppercase(self):  
    return bool(re.search(r"[A-Z]", self.password))
```

```
def check_lowercase(self):  
    return bool(re.search(r"[a-z]", self.password))
```

```
def check_special_characters(self):  
    return bool(re.search(r"[ !#$%&'()*+,-./[\\\]^_`{|}~" + r""]',  
self.password))
```

```
def check_strength(self):  
    checks = [  
        ('Length', self.check_length()),  
        ('Digits', self.check_digits()),  
        ('Uppercase', self.check_uppercase()),  
        ('Lowercase', self.check_lowercase()),  
        ('Special Characters', self.check_special_characters())  
    ]  
    return [(criteria, passed) for criteria, passed in checks if not passed]
```

```
def get_strength(self):  
    failed_checks = self.check_strength()
```

```
if failed_checks:
```

```
    return "Weak Password. Missing criteria: {}".format(',  
' + join([check[0] for check in failed_checks]))
```

```
else:
```

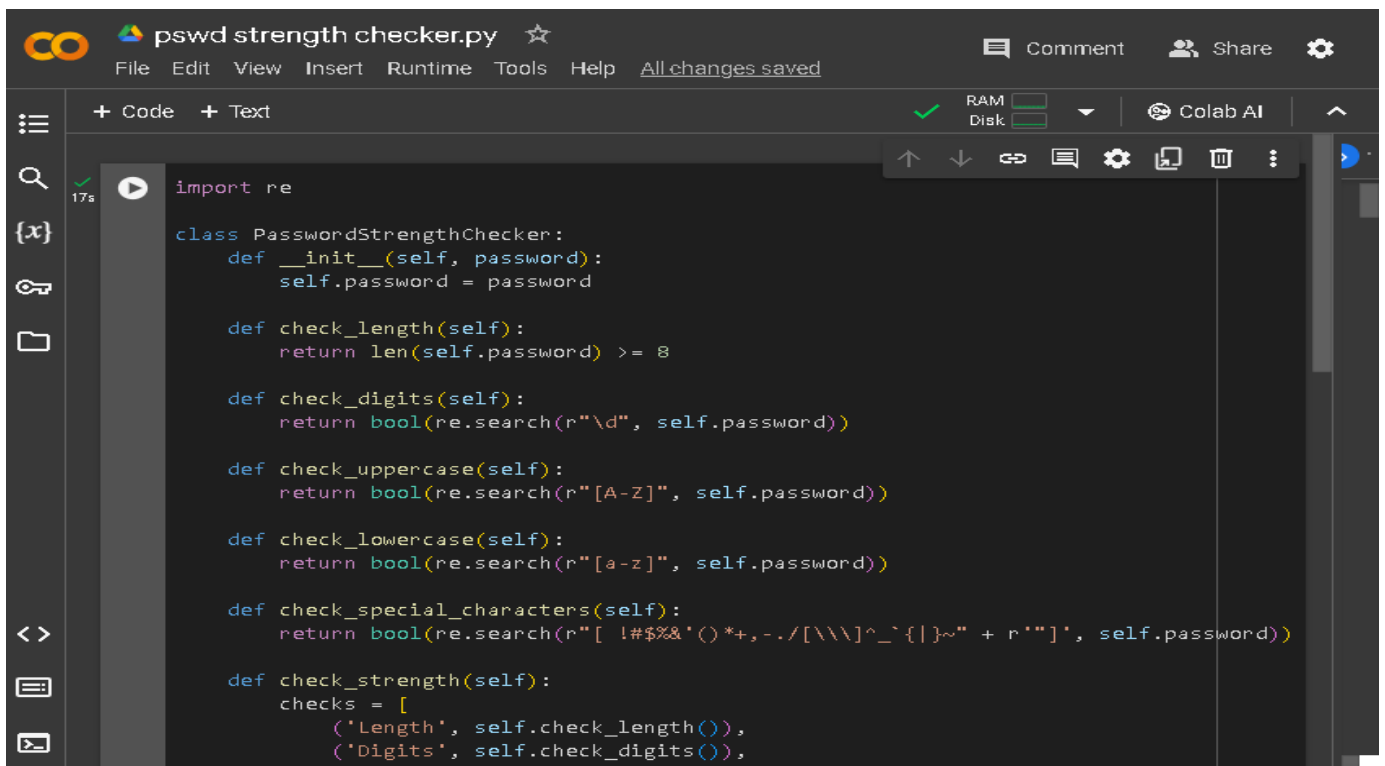
```
    return "Strong Password"
```

```
# Example usage
```

```
password = input("Enter your password: ")
```

```
checker = PasswordStrengthChecker(password)
```

```
print(checker.get_strength())
```



The screenshot shows a Google Colab notebook titled "pswd strength checker.py". The code defines a class `PasswordStrengthChecker` with the following methods:

```
import re

class PasswordStrengthChecker:
    def __init__(self, password):
        self.password = password

    def check_length(self):
        return len(self.password) >= 8

    def check_digits(self):
        return bool(re.search(r"\d", self.password))

    def check_uppercase(self):
        return bool(re.search(r"[A-Z]", self.password))

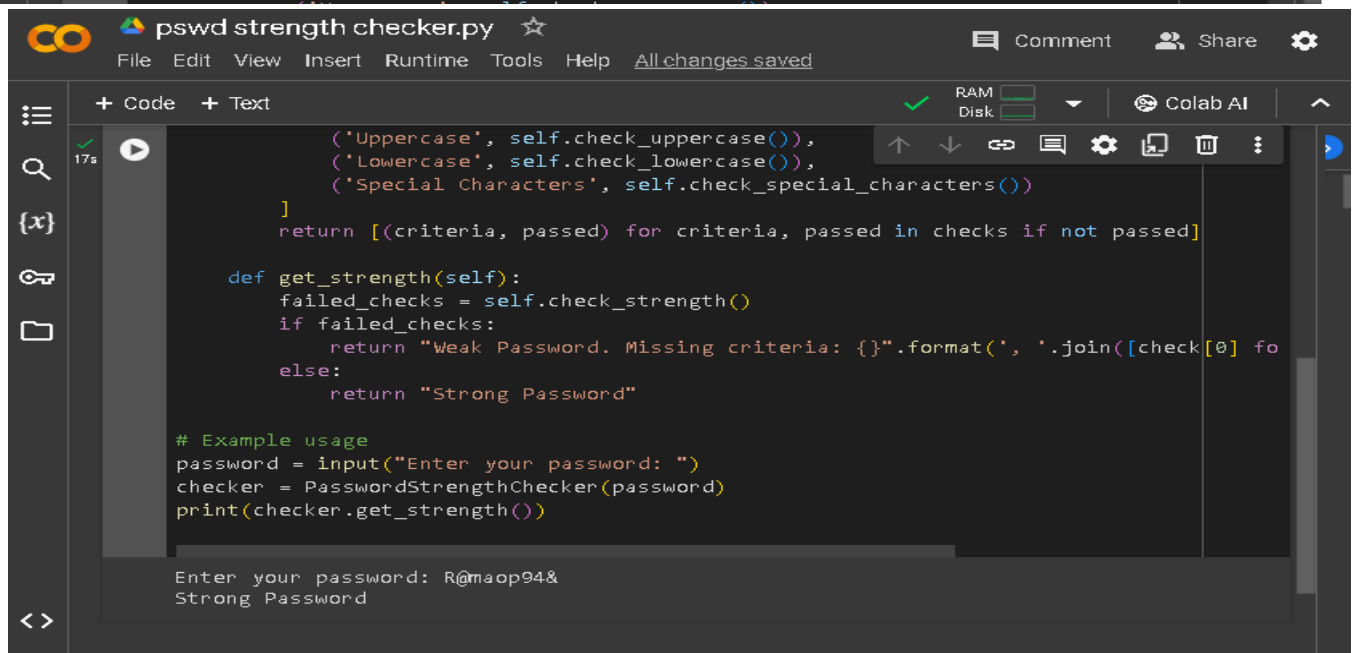
    def check_lowercase(self):
        return bool(re.search(r"[a-z]", self.password))

    def check_special_characters(self):
        return bool(re.search(r"[!#$%&'()*+,-./[\]\^_`{|}~" + r"'"]', self.password))

    def check_strength(self):
        checks = [
            ('Length', self.check_length()),
            ('Digits', self.check_digits()),
            ('Uppercase', self.check_uppercase()),
            ('Lowercase', self.check_lowercase()),
            ('Special Characters', self.check_special_characters())
        ]
        return [(criteria, passed) for criteria, passed in checks if not passed]

    def get_strength(self):
        failed_checks = self.check_strength()
        if failed_checks:
            return "Weak Password. Missing criteria: {}".format(', '.join([check[0] for check in failed_checks]))
        else:
            return "Strong Password"

# Example usage
password = input("Enter your password: ")
checker = PasswordStrengthChecker(password)
print(checker.get_strength())
```



The screenshot shows the continuation of the Python script from the previous image. It includes the `get_strength` method and an example usage section. The output of the script is displayed in a terminal window at the bottom:

```
Enter your password: R@maop94&
Strong Password
```

Additional Suggestions:

- ✓ Length of the password should be with minimum of 8 characters
- ✓ Must contain combination of alphabets (uppercase, lowercase), numbers, special characters.
- ✓ Password should be unique and difficult to crack

Conclusion:

This project is implementing a python program to understand the password strength and to improve security system. Due to strong passwords avoid of easy data loss,avoid easy crack of devices.