Minor project

Title: password strength classification

```
import re

def classify_password(password: str) -> str:
    score = 0

    # Length check
    if len(password) >= 8:
        score += 1
    if len(password) >= 12:
        score += 1

    # Character type checks
    if re.search(r"[a-z]", password):
        score += 1
    if re.search(r"[A-Z]", password):
        score += 1
    if re.search(r"[0-9]", password):
        score += 1
    if re.search(r"[!@#$%^&*(),.?\":{}|<>]", password):
        score += 1

    # Classification
    if score <= 2:
```

```
        return "Weak"
    elif score <= 4:
        return "Medium"
    else:
        return "Strong"


# Test passwords
passwords = [
    "password",
    "Password1",
    "Pass1!",
    "StrongPass123!",
    "Very$StrongPassword2024!"
]

for pwd in passwords:
    print(f"{pwd:30} -> {classify_password(pwd)}")
```

output:

```
password                       -> Weak
Password1                      -> Medium
Pass1!                         -> Medium
StrongPass123!                 -> Strong
Very$StrongPassword2024!       -> Strong
```

```
def password_strength(pwd):
    if len(pwd) < 6:
        return "Weak"
    elif len(pwd) < 10:
        return "Medium"
    else:
        return "Strong"

tests = ["abc", "welcome12", "strongpassword123"]
for t in tests:
    print(t, ":", password_strength(t))
```

output:

abc : Weak

welcome12 : Medium

strongpassword123 : Strong

```
import re

def password_strength(pwd):
    score = 0
    if re.search("[a-z]", pwd): score += 1
    if re.search("[A-Z]", pwd): score += 1
    if re.search("[0-9]", pwd): score += 1
    if re.search("[!@#$%^&*]", pwd): score += 1

    if score <= 1:
        return "Weak"
    elif score <= 3:
        return "Medium"
    else:
        return "Strong"

tests = ["hello", "Hello12", "Hello@123"]
for t in tests:
    print(t, ":", password_strength(t))
```

output:

hello : Weak

Hello12 : Medium

Hello@123 : Strong