Title:Cybersecurity project with python programming

A project Report submitted to

Vaultofcodes

By

Ms.Maneesha Gangapatnam

#develop a password manager with strong encryption

```
import random
import string
import hashlib

class PasswordManager:
    def _init_(self):
        self.password_table = {}

    def generate_password(self, length=12):
        characters = string.ascii_letters + string.digits + string.punctuation
        password = ''.join(random.choice(characters) for _ in range(length))
        return password

    def generate_strength(self, password):
        # This is a basic strength measure, you may want to implement a more sophisticated approach
        if len(password) >= 12:
            return "Strong"
        elif len(password) >= 8:
            return "Moderate"
        else:
            return "Weak"

    def create_password_table(self, website, username, password):
        if website not in self.password_table:
            self.password_table[website] = {}
        self.password_table[website][username] = password

    def hash_password(self, password):
        # Use a strong hashing algorithm, like SHA-256, in a real-world scenario
        hashed_password = hashlib.sha256(password.encode()).hexdigest()
```

```
        return hashed_password

    def store_password(self, website, username, password):
        hashed_password = self.hash_password(password)
        self.create_password_table(website, username, hashed_password)

    def verify_password(self, website, username, password):
        if website in self.password_table and username in self.password_table[website]:
            stored_password = self.password_table[website][username]
            hashed_input_password = self.hash_password(password)
            return hashed_input_password == stored_password
        return False

# Example usage:
password_manager = PasswordManager()

# Generate a password
generated_password = password_manager.generate_password()
print(f"Generated Password: {generated_password}")

# Determine password strength
password_strength = password_manager.generate_strength(generated_password)
print(f"Password Strength: {password_strength}")

# Store a password
website = "example.com"
username = "your_username"
password_manager.store_password(website, username, generated_password)

# Verify a password
input_password = "your_input_password"
```

```
is_verified = password_manager.verify_password(website, username, input_password)

print(f"Password Verification: {is_verified}")
```

output:

```
= RESTART: C:\Users\G.saikalyan\cybersecurity.py
Generated Password: E>iN}9\U@yJ7
Password Strength: Strong
```