Assignment Title: Building a Secure Password Generator in Python

Overview:

Your task is to develop a Python application that generates secure passwords for users. The password generator should create strong, random passwords that meet various security criteria. This project will help you understand how to work with strings, randomization, user input, and basic security principles in Python.

Objectives:

- Create an application that generates passwords of specified lengths and complexities.
- Ensure the passwords meet common security standards (e.g., inclusion of uppercase letters, lowercase letters, numbers, and special characters).
- Allow users to customize the password generation process by choosing the desired length and character types.
- Implement basic security measures to ensure that the generated passwords are unpredictable and difficult to crack.

Note : Make sure to check the samples Given Below for more clarity in term of expected inputs and outputs

Task Breakdown:

1. User Input for Password Criteria

Develop functionality that allows users to specify the length of the password and the types of characters it should include (e.g., uppercase letters, lowercase letters, numbers, special characters). The application should validate the user's input to ensure it meets minimum security standards.

2. Generating a Secure Password

Write a function that generates a random password based on the user's specified criteria. The password should be randomly generated using a secure method and should include a mix of characters to enhance security.

3. Ensuring Password Strength

Implement rules to ensure that the generated passwords are strong. For example, passwords should have a minimum length of 8 characters and include at least one character from each selected category (uppercase, lowercase, numbers, special characters).

4. Displaying the Generated Password

Once the password is generated, display it to the user in a secure manner. Ensure that the password is not stored or displayed insecurely.

5. Optional: Saving Passwords Securely

As an optional task, allow the user to save the generated password to a file securely. Implement encryption to protect the saved passwords from unauthorized access.

Submission Requirements:

- Submit the complete Python application with clear, well-organized code.
- Include user input validation to ensure the application behaves as expected.
- Provide comments or a user guide explaining how the application works and how to use it.
- (Optional) If you implement the password-saving feature, include documentation on how the passwords are encrypted and how users can securely access their saved passwords.

Evaluation Criteria:

- **Functionality:** Does the application generate secure passwords according to the user's specifications?
- **Security:** Are the passwords generated in a secure manner, and do they meet common security standards?
- **Code Quality:** Is the code well-organized, easy to understand, and properly commented?
- **User Experience:** Is the application user-friendly, and does it validate inputs appropriately?
- **(Optional) Encryption:** If included, is the password-saving feature implemented securely with proper encryption?

Sample Inputs and Outputs for Secure Password Generator Assignment

Sample 1: Basic Password Generation

User Input:

```
Enter the desired password length: 12
Include uppercase letters? (yes/no): yes
Include lowercase letters? (yes/no): yes
Include numbers? (yes/no): yes
Include special characters? (yes/no): yes
```

Output:

```
Generated Password: @fG7hZ!q2xT&
```

Sample 2: Custom Password Generation with Specific Criteria

User Input:

```
Enter the desired password length: 16
Include uppercase letters? (yes/no): yes
Include lowercase letters? (yes/no): yes
Include numbers? (yes/no): no
Include special characters? (yes/no): yes
```

Output:

```
Generated Password: qZ@wH!gF$rUvLt@s
```

Sample 3: Minimum Length Validation

User Input:

```
Enter the desired password length: 6
Include uppercase letters? (yes/no): yes
Include lowercase letters? (yes/no): yes
Include numbers? (yes/no): yes
Include special characters? (yes/no): yes
```

Output:

```
Error: The password length must be at least 8 characters for secur
ity reasons. Please enter a longer password length.
```

Sample 4: Password Generation with Limited Character Set

User Input:

```
Enter the desired password length: 10
Include uppercase letters? (yes/no): no
Include lowercase letters? (yes/no): yes
Include numbers? (yes/no): yes
Include special characters? (yes/no): no
```

Output:

```
Generated Password: g5f8b2k9m1
```

Sample 5: Optional Feature - Saving the Password Securely

User Input:

```
Enter the desired password length: 14
Include uppercase letters? (yes/no): yes
Include lowercase letters? (yes/no): yes
Include numbers? (yes/no): yes
Include special characters? (yes/no): yes

Would you like to save this password to a file? (yes/no): yes
Enter the file name to save the password: passwords.txt
```

Output:

```
Generated Password: T#u2LgQ!z9FkxD
Password saved securely to passwords.txt
```

Sample 6: Handling Invalid Inputs

User Input:

```
Enter the desired password length: twelve
Include uppercase letters? (yes/no): sure
Include lowercase letters? (yes/no): yes
Include numbers? (yes/no): yes
Include special characters? (yes/no): yes
```

Output:

```
Error: Invalid input for password length. Please enter a numerical
value
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```