**Project Report: Password Generator in Python**

**Project Title**

**Project 11: Password Generator**

**Abstract**

This project focuses on building a simple password generator using Python. The tool helps users create strong, secure passwords based on custom criteria like length and character types (uppercase, lowercase, numbers, symbols). Strong passwords are critical for digital security and this project demonstrates how to generate them programmatically. It is suitable for beginners and showcases core Python concepts like loops, string manipulation, and GUI (optional).

**Objective**

* To create a random password generator.
* To allow user-defined options (length and character types).
* To enhance understanding of string and random modules in Python.
* To promote good security practices by generating strong passwords.

**Tools & Technologies Used**

| Technology | Purpose |
| --- | --- |
| Python 3.x | Programming Language |
| random module | For random password generation |
| string module | To access letters, digits, symbols |
| Tkinter (Optional) | For graphical interface (not used here) |

**System Requirements**

* Python 3.x installed
* No external libraries required

**Methodology**

1. User Input: Get desired password length and character types.
2. Character Pool: Combine characters based on selected options.
3. Password Generation: Randomly pick characters to form password.
4. Output: Display generated password to the user.

**Code Implementation (Simple Console Version)**

**import random**

**import string**

**def generate\_password(length, use\_upper, use\_lower, use\_digits, use\_symbols):**

**characters = ""**

**if use\_upper:**

**characters += string.ascii\_uppercase**

**if use\_lower:**

**characters += string.ascii\_lowercase**

**if use\_digits:**

**characters += string.digits**

**if use\_symbols:**

**characters += string.punctuation**

**if not characters:**

**return "Error: No character types selected!"**

**password = ''.join(random.choice(characters) for \_ in range(length))**

**return password**

**def main():**

**print("=== Password Generator ===")**

**try:**

**length = int(input("Enter desired password length: "))**

**except ValueError:**

**print("Invalid input. Length must be a number.")**

**return**

**use\_upper = input("Include uppercase letters? (y/n): ").lower() == 'y'**

**use\_lower = input("Include lowercase letters? (y/n): ").lower() == 'y'**

**use\_digits = input("Include numbers? (y/n): ").lower() == 'y'**

**use\_symbols = input("Include special characters? (y/n): ").lower() == 'y'**

**password = generate\_password(length, use\_upper, use\_lower, use\_digits, use\_symbols)**

**print(f"\nGenerated Password: {password}")**

**if \_\_name\_\_ == "\_\_main\_\_":**

**main()**

**Sample Output**

**=== Password Generator ===**

**Enter desired password length: 12**

**Include uppercase letters? (y/n): y**

**Include lowercase letters? (y/n): y**

**Include numbers? (y/n): y**

**Include special characters? (y/n): n**

**Generated Password: aQ9bXr83WnqT**

**Advantages**

* **Fully customizable password settings**
* **Simple and easy to understand**
* **No third-party libraries required**
* **Improves security by generating strong passwords**

**Limitations**

* **Command-line based (no GUI)**
* **No option to save passwords**
* **Doesn’t check for password strength balance (e.g., at least one of each type)**

**Future Enhancements**

* **Add GUI using Tkinter**
* **Add password strength meter**
* **Option to copy password to clipboard**
* **Save password history securely**

**References**

1. [**Python random Module Documentation**](https://docs.python.org/3/library/random.html)
2. [**Python string Module Documentation**](https://docs.python.org/3/library/string.html)
3. [**GeeksforGeeks – Python Password Generator**](https://www.geeksforgeeks.org/python-generate-random-passwords/)
4. [**Python Official Docs**](https://docs.python.org/3/)
5. [**OWASP Password Recommendations**](https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html)

**Code execution with pdf link:**

https://www.programiz.com/onlinecompiler/1XKFCrf1aQUJV