Project Report

# Secure Password Generator using Python

## 1. Introduction

In today’s digital age, security is a major concern. One of the most basic yet critical practices for securing online accounts and systems is the use of strong, unpredictable passwords. However, users often struggle to create passwords that are both secure and easy to remember.  
 This project titled 'Secure Password Generator using Python' is developed to assist users in generating random and strong passwords based on customizable options such as length, character sets, and special symbols. This version of the application is designed as a Command Line Interface (CLI) tool written purely in Python.

## 2. Objective

* To build a secure password generator tool using Python.
* To allow user control over password characteristics such as length, use of numbers, uppercase letters, and symbols.
* To ensure generated passwords are random, secure, and meet general security guidelines.
* To practice good coding practices and improve logical thinking using Python.

## 3. Tools and Technologies Used

* Programming Language: Python 3.x
* Libraries: string, random, secrets
* Interface: Command Line Interface (CLI)
* OS Compatibility: Cross-platform (Windows/Linux/Mac)

## 4. System Requirements

* Python 3.x installed
* Terminal or command prompt
* Text editor (VS Code, Notepad++, etc.)

## 5. Project Description

The Password Generator is a Python script that allows users to customize their password by selecting:

* Desired length of the password
* Whether to include uppercase letters
* Whether to include digits
* Whether to include special symbols

The program ensures that at least one character from each selected category is included in the final password. The rest of the password is filled using secure random choices from the allowed character set. The password is shuffled to avoid predictable patterns.

## 6. Code Explanation

**a. Imports:**import string  
import random  
import secrets

- string: Provides character sets (uppercase, lowercase, digits, punctuation).  
- random: Used to shuffle the password list.  
- secrets: Used to securely pick characters to prevent predictability.

**b. Password Generation Function:**  
def generate\_password(length=12, use\_symbols=True, use\_numbers=True, use\_uppercase=True):

This function generates a password of a given length with options for symbols, numbers, and uppercase letters.  
- Ensures at least one character from each selected category is present.  
- Uses secrets.choice() for security and randomness.  
- Uses random.shuffle() to mix characters randomly before returning the password.

**c. Main Function:**  
def main():  
- Accepts user input for password length and preferences.  
- Handles errors like non-integer input.  
- Calls the password generation function and displays the output.

**d. Sample Output:**

Secure Password Generator   
Enter password length (e.g. 12): 14  
Include uppercase letters? (y/n): y  
Include numbers? (y/n): y  
Include symbols? (y/n): y  
Generated Password: @Xk9v&bgRm23H!

## 7. Security Considerations

* Passwords are generated using the secrets module, which is cryptographically secure.
* The application ensures strong entropy by combining multiple character sets.
* The password contains at least one symbol, one digit, and one uppercase character (based on user selection).

## 8. Key Learnings

* Understanding the importance of randomness and entropy in security.
* Usage of built-in libraries like string, random, and secrets.
* Handling user inputs and input validation.
* Writing clean, reusable Python code.
* Following coding best practices.

## 9. Future Enhancements

* Add a GUI interface using Tkinter.
* Add a password strength meter.
* Option to generate multiple passwords at once.
* Save password history locally in encrypted format.

## 10. Conclusion

The project was successful in building a reliable and flexible tool to generate secure passwords through the command line. It improved my skills in Python programming, secure coding, input validation, and modular design. This tool can be a helpful utility for anyone who needs secure, customizable passwords instantly.