**Command-Line Password Generator**

import random

import string

import pyperclip

def generate\_password(length, use\_uppercase, use\_lowercase, use\_digits, use\_symbols):

character\_set = ''

if use\_uppercase:

character\_set += string.ascii\_uppercase

if use\_lowercase:

character\_set += string.ascii\_lowercase

if use\_digits:

character\_set += string.digits

if use\_symbols:

character\_set += string.punctuation

if not character\_set:

raise ValueError("At least one character set must be selected.")

password = ''.join(random.choice(character\_set) for \_ in range(length))

return password

def main():

print("Welcome to the Password Generator!")

# Get user inputs

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

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

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

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

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

# Generate password

try:

password = generate\_password(length, use\_uppercase, use\_lowercase, use\_digits, use\_symbols)

print(f"Generated Password: {password}")

# Copy to clipboard

copy\_to\_clipboard = input("Copy password to clipboard? (y/n): ").lower() == 'y'

if copy\_to\_clipboard:

pyperclip.copy(password)

print("Password copied to clipboard.")

except ValueError as e:

print(e)

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

main()

**Install Required Libraries**

pip install pyperclip

**Running the Command-Line Version**

python password\_generator.py

**GUI Password Generator**

import tkinter as tk

import random

import string

import pyperclip

def generate\_password():

length = int(length\_entry.get())

use\_uppercase = uppercase\_var.get()

use\_lowercase = lowercase\_var.get()

use\_digits = digits\_var.get()

use\_symbols = symbols\_var.get()

character\_set = ''

if use\_uppercase:

character\_set += string.ascii\_uppercase

if use\_lowercase:

character\_set += string.ascii\_lowercase

if use\_digits:

character\_set += string.digits

if use\_symbols:

character\_set += string.punctuation

if not character\_set:

result\_label.config(text="At least one character set must be selected.")

return

password = ''.join(random.choice(character\_set) for \_ in range(length))

result\_label.config(text=f"Generated Password: {password}")

# Copy to clipboard

pyperclip.copy(password)

# Create the GUI

root = tk.Tk()

root.title("Password Generator")

# Length input

tk.Label(root, text="Password Length:").pack()

length\_entry = tk.Entry(root)

length\_entry.pack()

# Checkboxes for character sets

uppercase\_var = tk.BooleanVar()

lowercase\_var = tk.BooleanVar()

digits\_var = tk.BooleanVar()

symbols\_var = tk.BooleanVar()

tk.Checkbutton(root, text="Include Uppercase Letters", variable=uppercase\_var).pack()

tk.Checkbutton(root, text="Include Lowercase Letters", variable=lowercase\_var).pack()

tk.Checkbutton(root, text="Include Digits", variable=digits\_var).pack()

tk.Checkbutton(root, text="Include Symbols", variable=symbols\_var).pack()

# Generate button

generate\_button = tk.Button(root, text="Generate Password", command=generate\_password)

generate\_button.pack()

# Result label

result\_label = tk.Label(root, text="")

result\_label.pack()

# Run the application

root.mainloop()

**Running the GUI Version**

python password\_generator\_gui.py