

Prezentare proiect Python

Alegerea Proiectului:

Am ales acest proiect deoarece consider ca traim in era tehnologiei unde nu mai exista actiunea de "A fura" ci exista **a ataca cibernetica pe cineva**, mai exact, acum totul se rezuma la parole, 2FA (2 factor authentication), face ID, etc. Acum oricine are un calculator, telefon sau laptop poate invata sa sparga parole, sa fure identitatea sau poate chiar bunuri materiale din online. Astfel eu am ales sa fac un generator de parole folosind un randomizer, ceea ce maresc complexitatea parolei si te mentine protejat.

Sa incep cu inceputul:

```
pass_gen12.py > generate_password
1 import tkinter as tk
2 from tkinter import messagebox
3 import string
4 import random
5
```

Prima bucatie de cod a fost reprezentata de catre librarii:

- Tkinter – partea grafica a programului
- String – siruri folosite pentru a simplifica codul
- Random – randomizer ul folosit pentru a marii complexitatea parolei

Partea principala:

```
def generate_password():
    try:
        l = int(length_entry.get())
    except ValueError:
        messagebox.showerror("Eroare", "Introduceți un număr valid pentru lungimea parolei.")
        return

    lc = ""
    if letters_var.get():
        lc += string.ascii_letters
    if numbers_var.get():
        lc += string.digits
    if special_var.get():
        lc += string.punctuation

    if not lc:
        messagebox.showerror("Eroare", "Selectați cel puțin un tip de caracter.")
        return

    password = ''.join(random.choice(lc) for _ in range(l))
    result_label.config(text=f"Parola random este: {password}")
```

Partea principala a insemnat inima si creierul programului.

Acestea sunt compuse din 4 parti:

1: Initializarea lungimii si verificarea erorilor

```
def generate_password():  
    try:  
        l = int(length_entry.get())  
    except ValueError:  
        messagebox.showerror("Eroare", "Introduceți un număr valid pentru lungimea parolei.")  
    return
```

Aceasta bucatica de cod a fost scrisa doar pentru initializare si verificarea erorilor ce pot aparea.

2: Conditonarea si optiunile programului

```
lc = ""  
if letters_var.get():  
    lc += string.ascii_letters  
if numbers_var.get():  
    lc += string.digits  
if special_var.get():  
    lc += string.punctuation
```

Aceasta este folosita strict pentru a verifica ce doreste utilizatorul sa foloseasca in crearea parolei.

3: Verificarea Integritatii:

```
if not lc:  
    messagebox.showerror("Eroare", "Selectați cel puțin un tip de caracter.")  
    return
```

Aceasta bucatica verifica daca am selectat macar o optiune valabila iar daca nu ne avertizeaza.

4: Randomizer si afisarea parolei

```
password = ''.join(random.choice(lc) for _ in range(l))  
result_label.config(text=f"Parola random este: {password}")
```

Aici ne folosim de randomizer si conditiile anterioare pentru a crea parola in sine. Mai apoi o afisam.

Ulterior a aparut si partea Grafica a programului, sau partea frumoasa.

```
# Creează fereastra principală
root = tk.Tk()
root.title("Generator de Parole")

# Lungimea parolei
tk.Label(root, text="Lungime dorită parola:").pack()
length_entry = tk.Entry(root)
length_entry.pack()

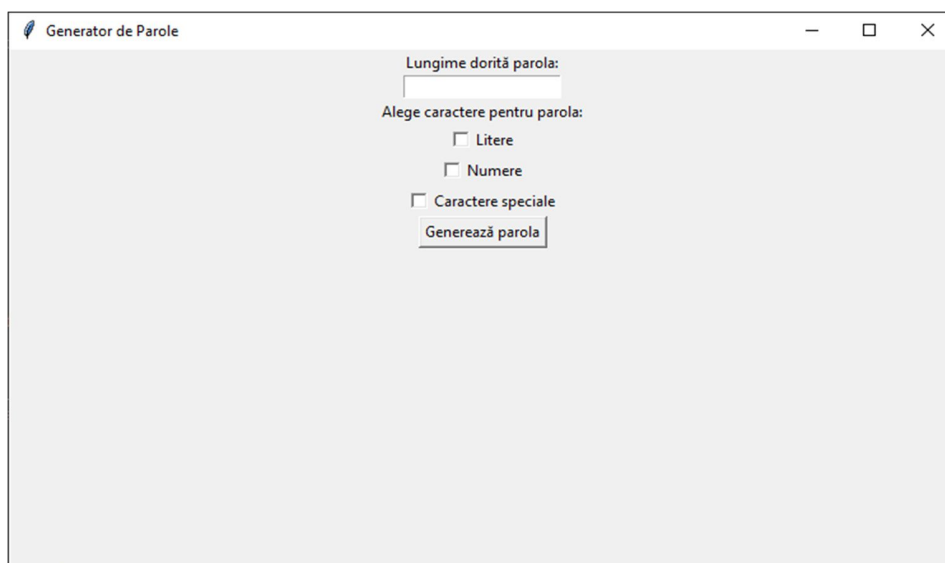
# Opțiuni de caractere
tk.Label(root, text="Alege caractere pentru parola:").pack()
letters_var = tk.BooleanVar()
tk.Checkbutton(root, text="Litere", variable=letters_var).pack()
numbers_var = tk.BooleanVar()
tk.Checkbutton(root, text="Numere", variable=numbers_var).pack()
special_var = tk.BooleanVar()
tk.Checkbutton(root, text="Caractere speciale", variable=special_var).pack()

# Buton pentru generarea parolei
generate_button = tk.Button(root, text="Generează parola", command=generate_password)
generate_button.pack()

# Etichetă pentru afișarea rezultatului
result_label = tk.Label(root, text="")
result_label.pack()

# Rulează bucla aplicației
root.mainloop()
```

Folosind Tkinter, o librerie grafica din Python. Am reusit sa cream aplicatia in sine.



Partea grafica, la randul ei, are 4 bucatele de cod importante.

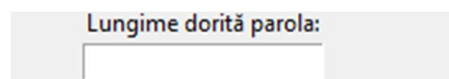
1: Initializarea ferestrei principale.

```
# Creează fereastra principală
root = tk.Tk()
root.title("Generator de Parole")
```

Root definește baza paginii iar root.title marchează titlul dat aplicației în sine.

2: Crearea widget-ului pentru lungime.

```
# Lungimea parolei
tk.Label(root, text="Lungime dorită parola:").pack()
length_entry = tk.Entry(root)
length_entry.pack()
```

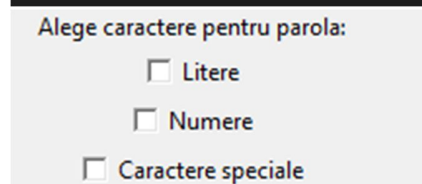


Lungime dorită parola:

Această bucată creează a 2-a cea mai importantă bucată din aplicație, și anume text box-ul pentru alegerea lungimii.

3: Check Box-urile pentru opțiuni

```
# Opțiuni de caractere
tk.Label(root, text="Alege caractere pentru parola:").pack()
letters_var = tk.BooleanVar()
tk.Checkbutton(root, text="Litere", variable=letters_var).pack()
numbers_var = tk.BooleanVar()
tk.Checkbutton(root, text="Numere", variable=numbers_var).pack()
special_var = tk.BooleanVar()
tk.Checkbutton(root, text="Caractere speciale", variable=special_var).pack()
```



Alege caractere pentru parola:

☐ Litere

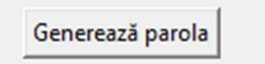
☐ Numere

☐ Caractere speciale

Aceasta este cea mai importantă bucată de cod pentru grafica proiectului, ea afișează check box-urile ce ajută utilizatorul la sporirea complexității parolei.

4: Butonul principal

```
# Buton pentru generarea parolei
generate_button = tk.Button(root, text="Generează parola", command=generate_password)
generate_button.pack()
```



Acesta este butonul fara de care nu ar merge aplicatia, toat bucatica asta de cod ajuta la afisarea lui iar urmatoarele la functionalitatea lui.

```
# Etichetă pentru afişarea rezultatului
result_label = tk.Label(root, text="")
result_label.pack()
```

```
3
4 # Rulează bucla aplicației
5 root.mainloop()
```

Multumesc!