

TASK3-PASSWORD GENERATOR APPLICATION

PROJECT TITLE:

Password Generator GUI App

DESCRIPTION:

A graphical user interface application built using CustomTkinter that allows users to generate strong, random passwords of user-specified length using letters, digits, and punctuation characters.

FEATURES:

- 1. CustomTkinter-based dark-themed UI.**
- 2. Input for password length.**
- 3. Validates for positive integers.**
- 4. Generates secure passwords using random and string modules.**

TECHNOLOGIES USED:

- 1. Python**
- 2. CustomTkinter**
- 3. random, string modules**

TARGET USERS:

- 1. General users seeking strong password generation.**
- 2. Security and privacy-conscious individuals.**

APPENDIX:

```
import customtkinter as ctk  
import random  
import string
```

#Creating the class for Password Generation

```
class PasswordGeneratorApp(ctk.CTk):
```

```
    def __init__(self):
```

```
        super().__init__()
```

```
        self.title("PassWord Generator")
```

```
        self.geometry("800x500")
```

```
        self.resizable(False,False)
```

```
        ctk.set_appearance_mode("dark")
```

```
        ctk.set_default_color_theme("dark-blue")
```

```
        self.create_widgets()
```

#Labeling all the elements in the page

```
    def create_widgets(self):
```

```
        self.title_label = ctk.CTkLabel(self,text="PASSWORD  
GENERATOR",font=("Arial",28,"bold"),text_color="#5e35b1")
```

```
        self.title_label.pack(pady=20)
```

```
        self.lenght_frame = ctk.CTkFrame(self,fg_color="transparent")
```

```
        self.lenght_frame.pack(pady=20)
```

```
        self.lenght_label=ctk.CTkLabel(self.lenght_frame,text="Enter  
Password Length",font=("Arial",18),text_color="#5e35b1")
```

```
        self.lenght_label.grid(row=0,column=0,padx=10)
```

```
        self.lenght_entry =
```

```
ctk.CTkEntry(self.lenght_frame,width=100,font=("Arial",18))
```

```
        self.lenght_entry.grid(row=1,column=0)
```

```
self.gen_button= ctk.CTkButton(self,text="Generate  
Password",fg_color="#5e35b1",command=self.generate_password,  
width=200,)
```

```
self.gen_button.pack(pady=20)
```

```
self.password_out =  
ctk.CTkEntry(self,width=400,font=("Arial",18),justify="center")  
self.password_out.pack(pady=10)
```

```
#Function for Password generation using random module.
```

```
def generate_password(self):
```

```
    try:
```

```
        length = int(self.lenght_entry.get())
```

```
        if length <=0:
```

```
            self.password_out.delete(0,'end')
```

```
            self.password_out.insert(0,"Enter positive number!")
```

```
            return
```

```
        chars = string.ascii_letters + string.digits + string.punctuation
```

```
        password = ''.join(random.choice(chars) for _ in  
range(length))
```

```
        self.password_out.delete(0,'end')
```

```
        self.password_out.insert(0,password)
```

```
    except ValueError:
```

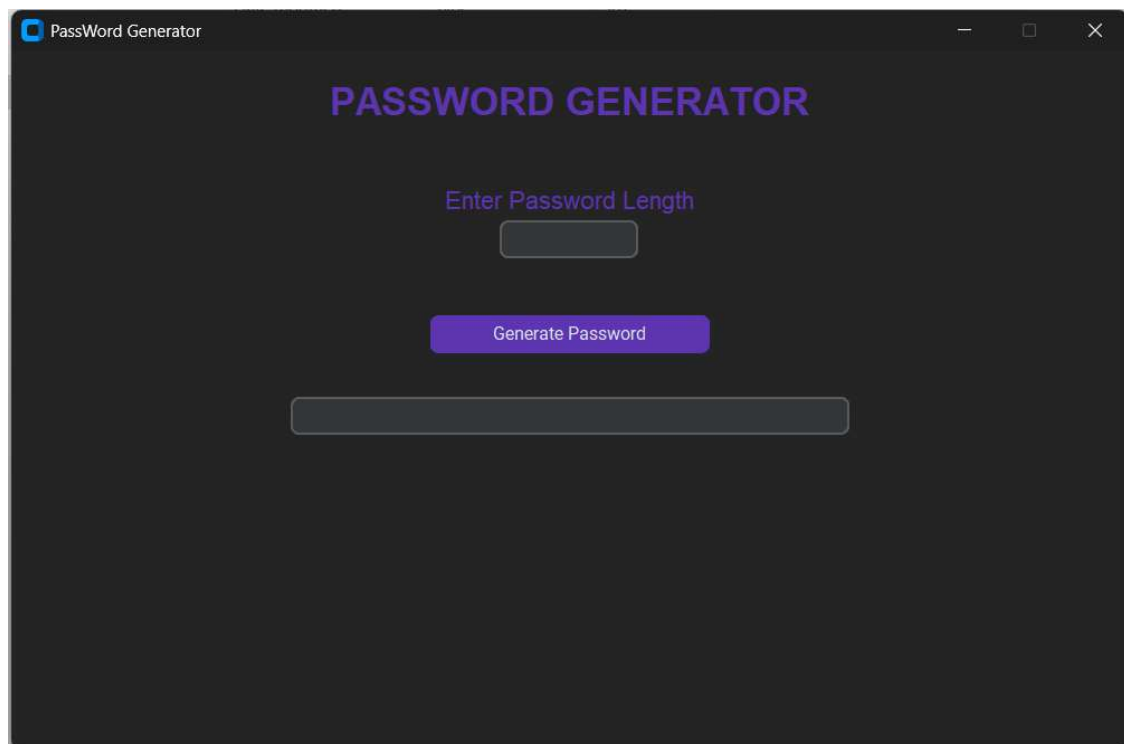
```
        self.password_out.delete(0,'end')
```

```
        self.password_out.insert(0,"Please Enter a Valid Number!!")
```

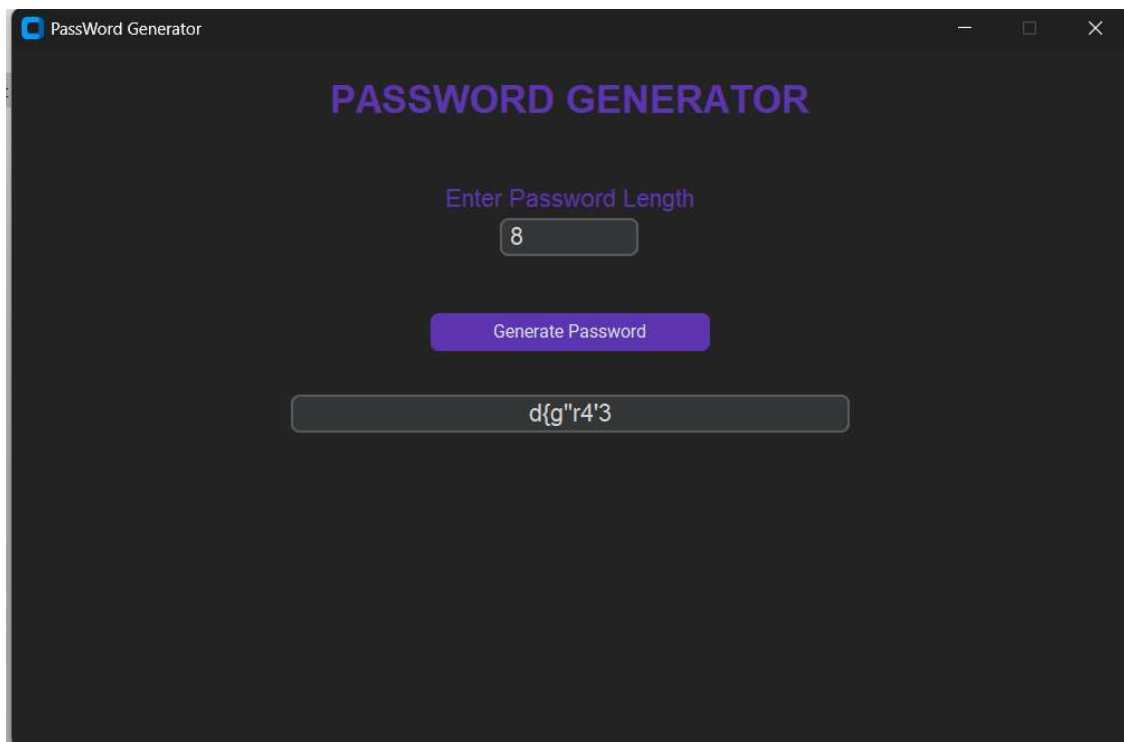
```
app=PasswordGeneratorApp()
```

```
app.mainloop()
```

OUTPUTS:



The screenshot shows a window titled "PassWord Generator" with a dark background. The title bar includes standard window controls (minimize, maximize, close). The main content area features the text "PASSWORD GENERATOR" in large, bold, purple letters. Below this, the text "Enter Password Length" is displayed in a smaller purple font. Underneath is a text input field. A purple button labeled "Generate Password" is positioned below the input field. At the bottom, there is a long, empty text input field for the generated password.



This screenshot shows the same "PassWord Generator" window after a password has been generated. The text "PASSWORD GENERATOR" and the "Enter Password Length" label remain. The text input field now contains the number "8". The "Generate Password" button is still present. The long text input field at the bottom now displays the generated password "d{g"r4'3".