# PASSWORD GENERATOR AND MANAGER

**DONE BY:**
**ARUL LINCY.A(RA2211030010016)**
**G.GAYATHRISRI(RA2211030010029)**
**RAAGHASHREE(RA2211030010040)**

# OVERVIEW

The password Generator application consists of 2 components:

- Password Generator:

  This helps generate strong, randomized password as per user request, i.e. strength and length of password.

- Password Manager:

  This helps store the passwords along with the username according to the websites in a database through the application.

Together, these tools facilitate a comprehensive approach to password management and security.

Some tools used to create the application:

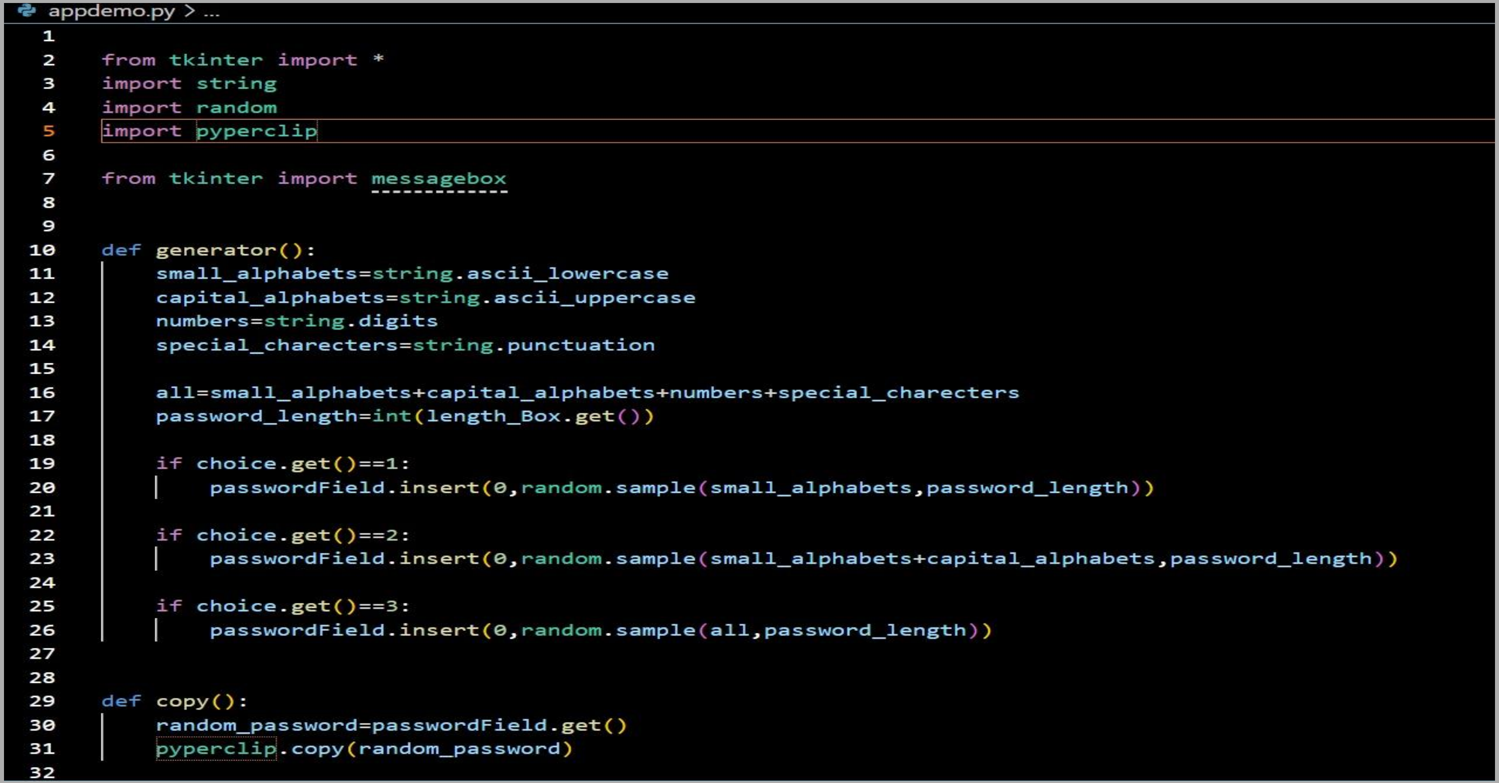
- Python Tkinter:

  This is a built-in GUI developing tool of python. It houses  various attributes like frames ,buttons ,labels ,etc.

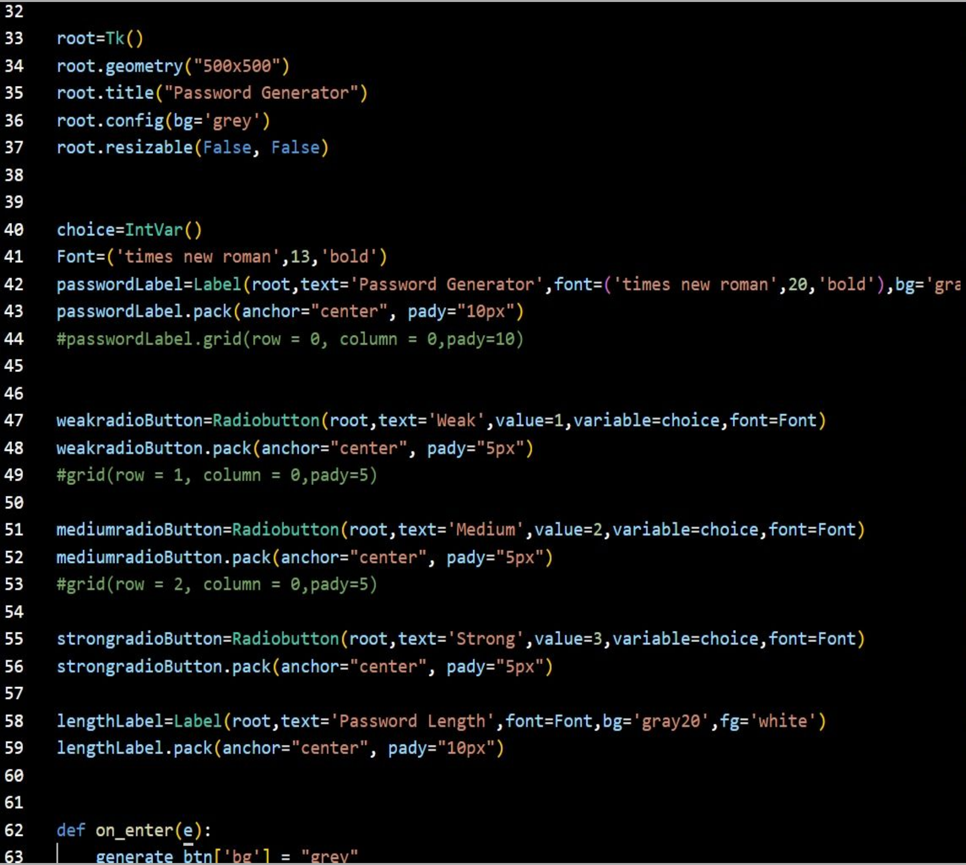
- SQLite3:

  Python module to execute SQL queries and update databases in realtime.

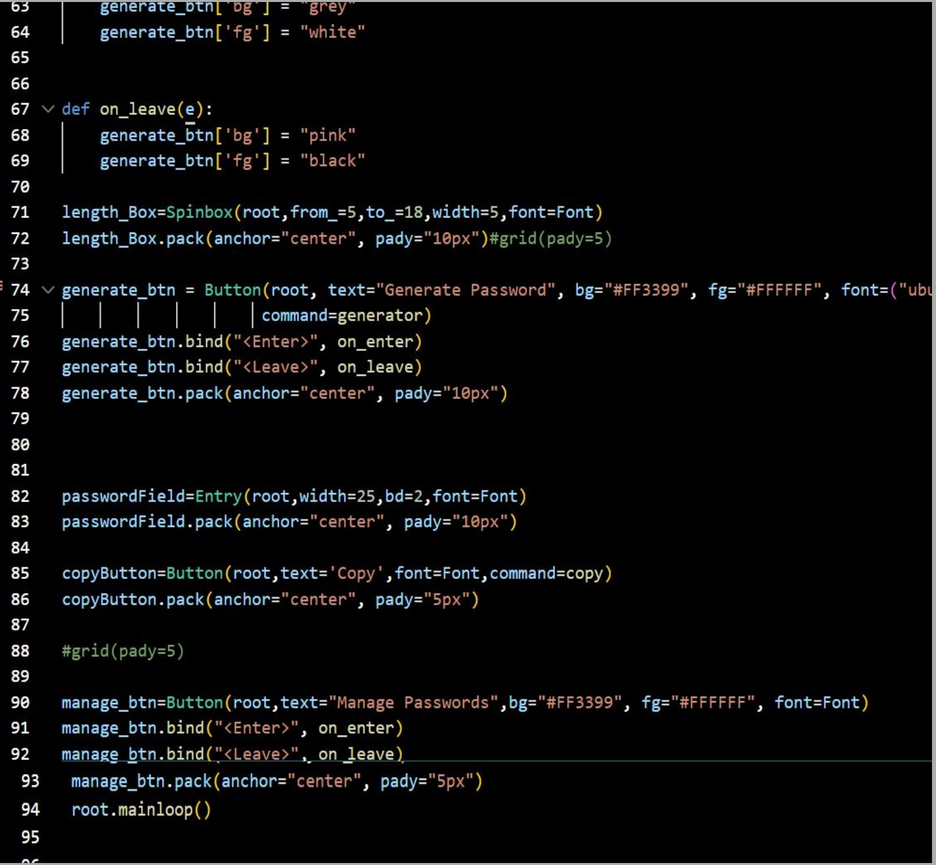
# SOURCE CODE



```python

from tkinter import *
import string
import random
import pyperclip

from tkinter import messagebox


def generator():
    small_alphabets=string.ascii_lowercase
    capital_alphabets=string.ascii_uppercase
    numbers=string.digits
    special_charecters=string.punctuation

    all=small_alphabets+capital_alphabets+numbers+special_charecters
    password_length=int(length_Box.get())

    if choice.get()==1:
        passwordField.insert(0,random.sample(small_alphabets,password_length))

    if choice.get()==2:
        passwordField.insert(0,random.sample(small_alphabets+capital_alphabets,password_length))

    if choice.get()==3:
        passwordField.insert(0,random.sample(all,password_length))


def copy():
    random_password=passwordField.get()
    pyperclip.copy(random_password)

```

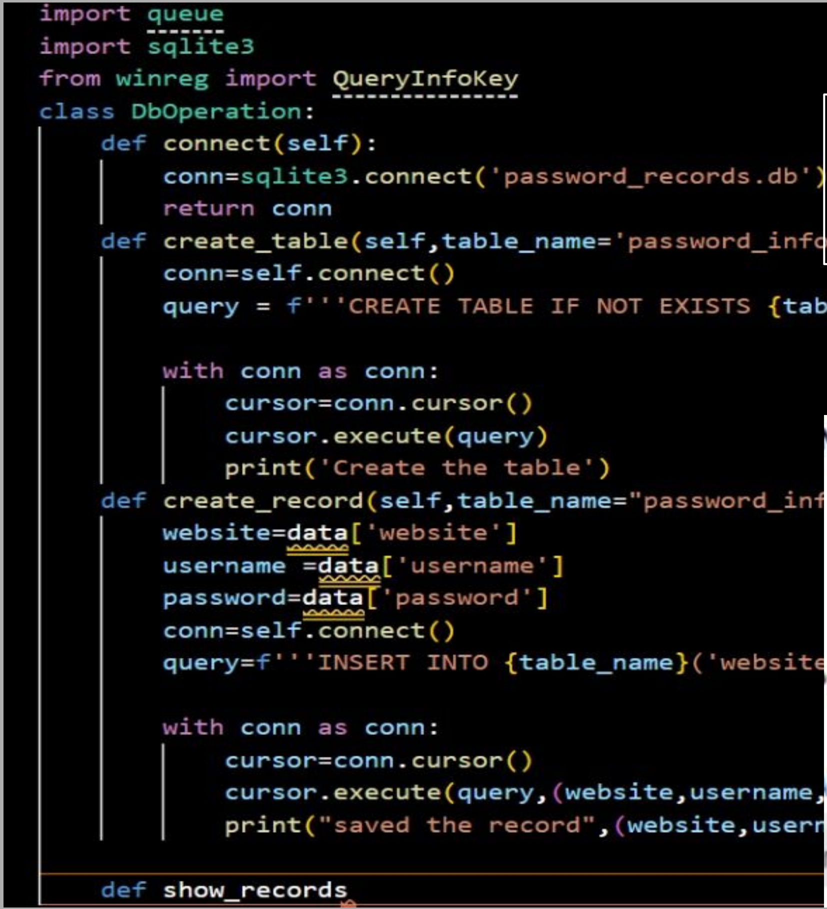


```python

root=Tk()
root.geometry("500x500")
root.title("Password Generator")
root.config(bg='grey')
root.resizable(False, False)


choice=IntVar()
Font=('times new roman',13,'bold')
passwordLabel=Label(root,text='Password Generator',font=('times new roman',20,'bold'),bg='gra
passwordLabel.pack(anchor="center", pady="10px")
#passwordLabel.grid(row = 0, column = 0,pady=10)


weakradioButton=Radiobutton(root,text='Weak',value=1,variable=choice,font=Font)
weakradioButton.pack(anchor="center", pady="5px")
#grid(row = 1, column = 0,pady=5)

mediumradioButton=Radiobutton(root,text='Medium',value=2,variable=choice,font=Font)
mediumradioButton.pack(anchor="center", pady="5px")
#grid(row = 2, column = 0,pady=5)

strongradioButton=Radiobutton(root,text='Strong',value=3,variable=choice,font=Font)
strongradioButton.pack(anchor="center", pady="5px")

lengthLabel=Label(root,text='Password Length',font=Font,bg='gray20',fg='white')
lengthLabel.pack(anchor="center", pady="10px")


def on_enter(e):
    generate_btn['bg'] = "grey"
```

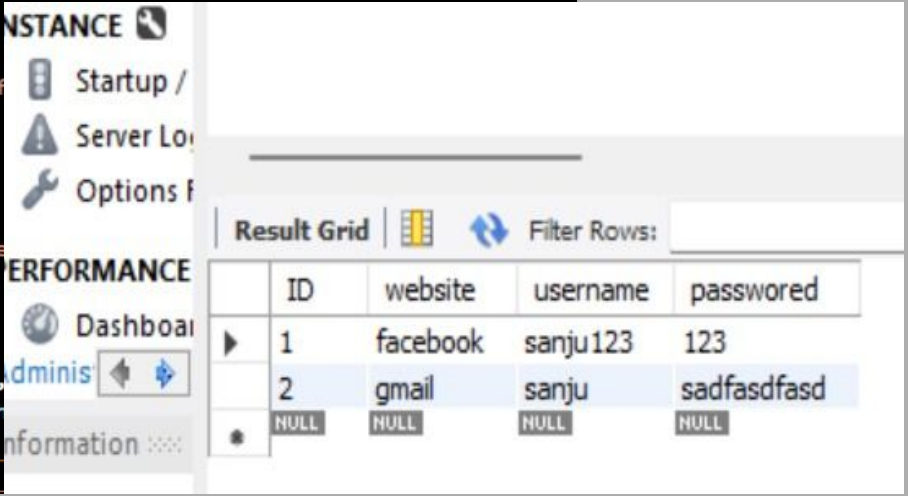


```python
    generate_btn['bg'] = "grey"
    generate_btn['fg'] = "white"


def on_leave(e):
    generate_btn['bg'] = "pink"
    generate_btn['fg'] = "black"

length_Box=Spinbox(root,from_=5,to_=18,width=5,font=Font)
length_Box.pack(anchor="center", pady="10px")#grid(pady=5)

generate_btn = Button(root, text="Generate Password", bg="#FF3399", fg="#FFFFFF", font=("ubu
                     command=generator)
generate_btn.bind("<Enter>", on_enter)
generate_btn.bind("<Leave>", on_leave)
generate_btn.pack(anchor="center", pady="10px")



passwordField=Entry(root,width=25,bd=2,font=Font)
passwordField.pack(anchor="center", pady="10px")

copyButton=Button(root,text='Copy',font=Font,command=copy)
copyButton.pack(anchor="center", pady="5px")

#grid(pady=5)

manage_btn=Button(root,text="Manage Passwords",bg="#FF3399", fg="#FFFFFF", font=Font)
manage_btn.bind("<Enter>", on_enter)
manage_btn.bind("<Leave>", on_leave)
manage_btn.pack(anchor="center", pady="5px")
root.mainloop()

```

# BACKEND

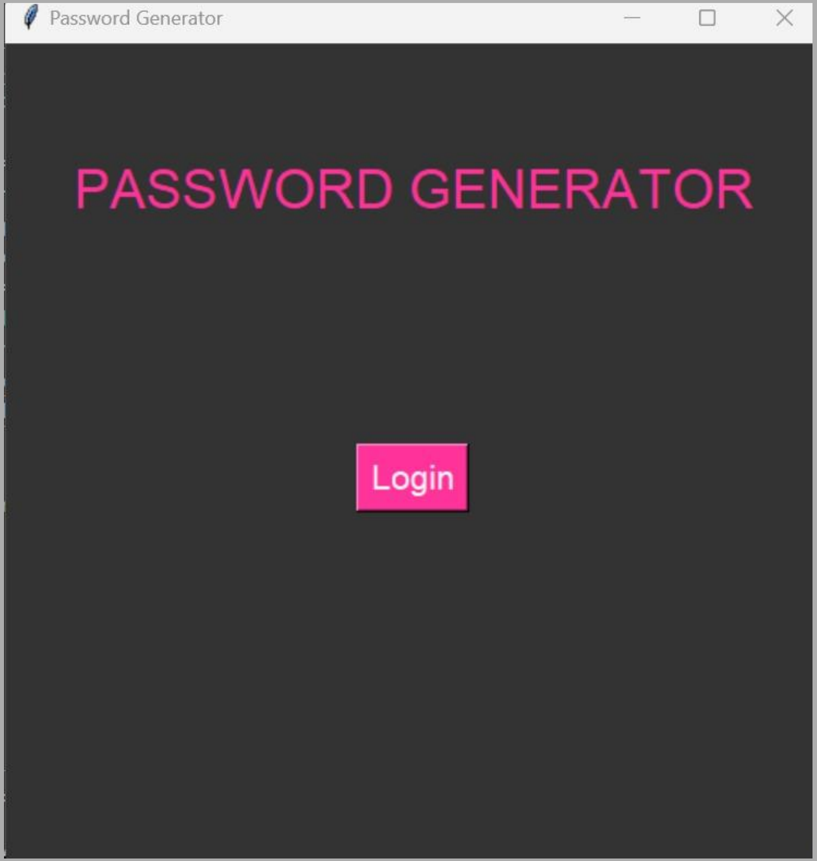


```
import queue
import sqlite3
from winreg import QueryInfoKey
class DbOperation:
    def connect(self):
        conn=sqlite3.connect('password_records.db')
        return conn
    def create_table(self,table_name='password_info'):
        conn=self.connect()
        query = f'''CREATE TABLE IF NOT EXISTS {table_name} (ID INT PRIMARY KEY AUTOINCR

        with conn as conn:
            cursor=conn.cursor()
            cursor.execute(query)
            print('Create the table')
    def create_record(self,table_name="password_inf
        website=data['website']
        username =data['username']
        password=data['password']
        conn=self.connect()
        query=f'''INSERT INTO {table_name}('website

        with conn as conn:
            cursor=conn.cursor()
            cursor.execute(query,(website,username,
            print("saved the record",(website,usern

    def show_records
```

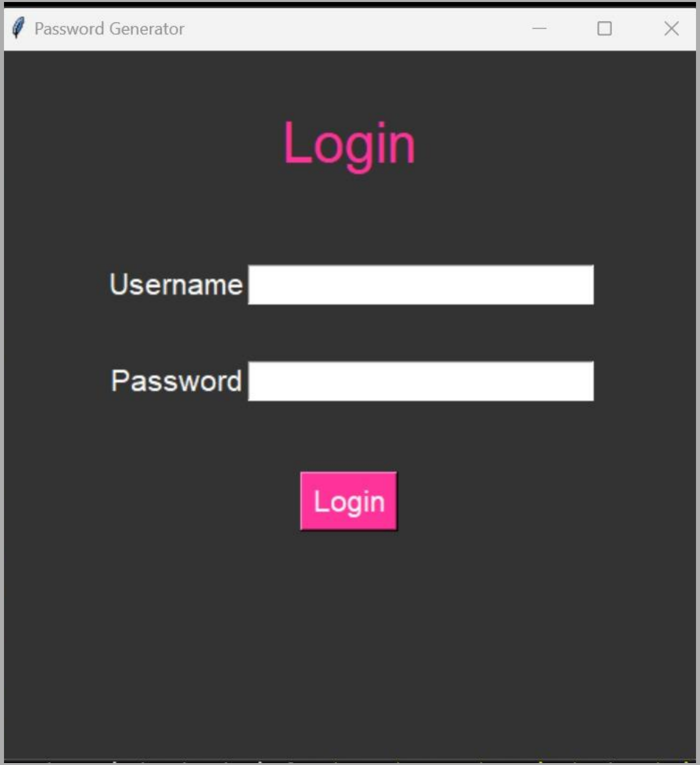


| | ID | website | username | passwored |
|---|---|---|---|---|
| ▶ | 1 | facebook | sanju123 | 123 |
| | 2 | gmail | sanju | sadfasdfasd |
| * | NULL | NULL | NULL | NULL |

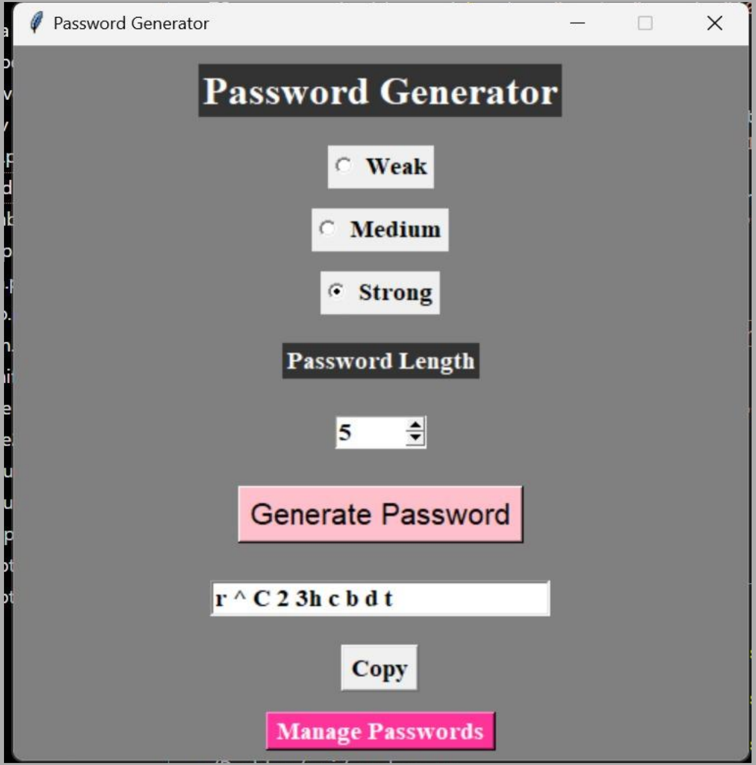
# OUTPUT



-MAIN PAGE



USER LOGIN PAGE



○ Weak

○ Medium

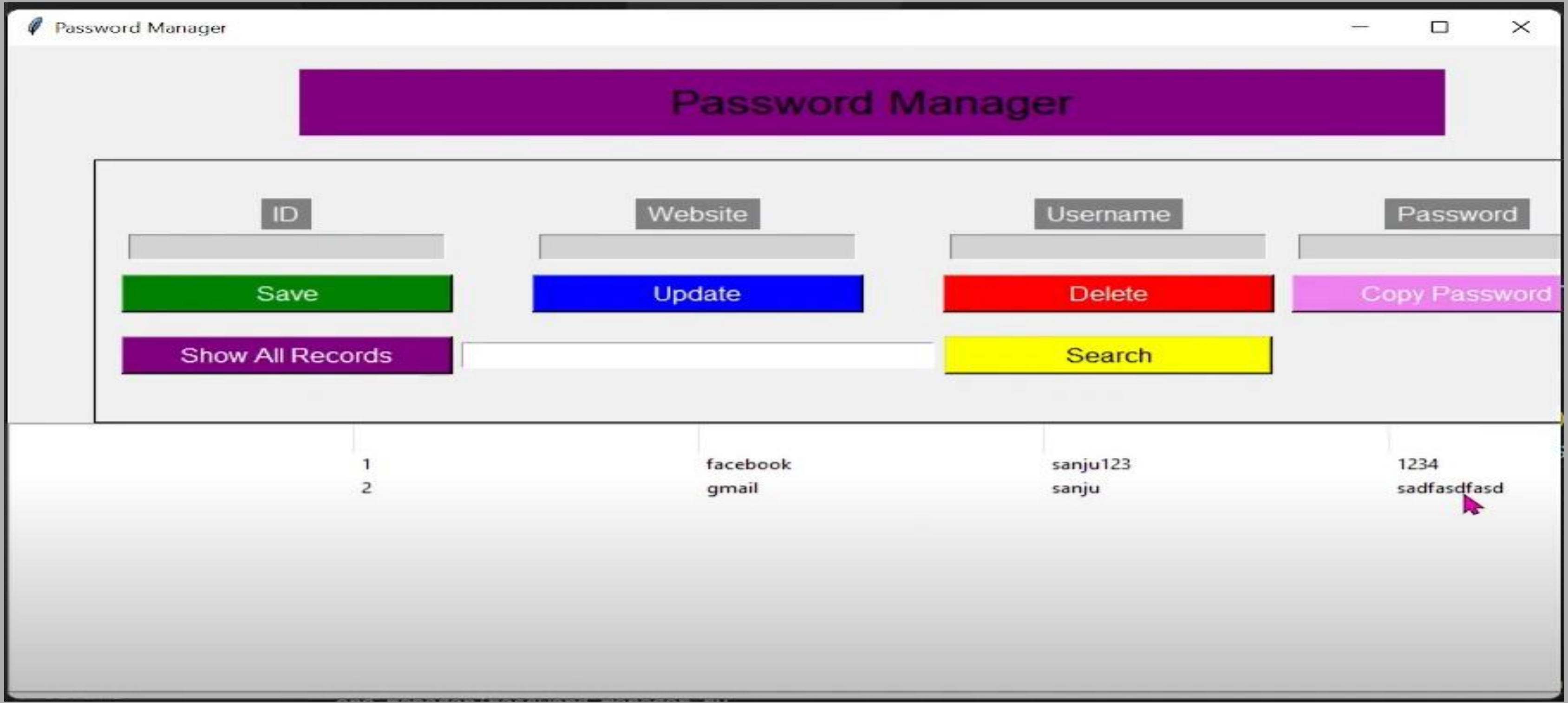
● Strong

Password Length

5

r ^ C 2 3h c b d t

PASSWORD GENERATOR PAGE



| | | | |
|---|---|---|---|
| 1 | facebook | sanju123 | 1234 |
| 2 | gmail | sanju | sadfasdfasd |

PASSWORD MANAGER PAGE

# CONCLUSION

This Password Generator is a user-friendly application designed in such a way to cater to a wide range of users and individuals seeking stronger personal passwords.

This promotes the adoption of secure password practices, contributing to a safer digital environment for all.

As we look to the future, the application stands as a testament to the importance of proactive cybersecurity measures.

The project’s potential for enhancing digital security and privacy is limitless.

# THANK YOU