**Iteration 3**

import tkinter as tk

from tkinter import messagebox

import sqlite3

import hashlib

# Database setup

conn = sqlite3.connect('users.db')

c = conn.cursor()

c.execute('''CREATE TABLE IF NOT EXISTS users (

                username TEXT PRIMARY KEY,

                password TEXT

             )''')

# Adding admin account with hashed password

admin\_username = 'admin'

admin\_password = hashlib.sha256('adminpass'.encode()).hexdigest()

c.execute('''INSERT OR IGNORE INTO users (username, password) VALUES (?, ?)''', (admin\_username, admin\_password))

conn.commit()

class App:

    def \_\_init\_\_(self, root):

        self.root = root

        self.root.title("Login Simulation")

        self.main\_frame = tk.Frame(root)

        self.main\_frame.pack()

        self.login\_frame = tk.Frame(self.main\_frame)

        self.login\_frame.pack()

        self.username\_label = tk.Label(self.login\_frame, text="Username")

        self.username\_label.pack(pady=5)

        self.username\_entry = tk.Entry(self.login\_frame)

        self.username\_entry.pack(pady=5)

        self.password\_label = tk.Label(self.login\_frame, text="Password")

        self.password\_label.pack(pady=5)

        self.password\_entry = tk.Entry(self.login\_frame, show='\*')

        self.password\_entry.pack(pady=5)

        self.login\_button = tk.Button(self.login\_frame, text="Login", command=self.login)

        self.login\_button.pack(pady=5)

        self.register\_button = tk.Button(self.login\_frame, text="Register", command=self.register)

        self.register\_button.pack(pady=5)

        self.logged\_in\_user = None

    def login(self):

        username = self.username\_entry.get()

        password = self.password\_entry.get()

        hashed\_password = hashlib.sha256(password.encode()).hexdigest()

        c.execute("SELECT \* FROM users WHERE username=? AND password=?", (username, hashed\_password))

        user = c.fetchone()

        if user:

            self.logged\_in\_user = username

            self.show\_user\_menu()

        else:

            messagebox.showerror("Error", "Incorrect username or password")

    def register(self):

        username = self.username\_entry.get()

        password = self.password\_entry.get()

        hashed\_password = hashlib.sha256(password.encode()).hexdigest()

        try:

            c.execute("INSERT INTO users (username, password) VALUES (?, ?)", (username, hashed\_password))

            conn.commit()

            messagebox.showinfo("Success", "Account created successfully")

        except sqlite3.IntegrityError:

            messagebox.showerror("Error", "Username already exists")

    def show\_user\_menu(self):

        self.login\_frame.pack\_forget()

        self.user\_menu\_frame = tk.Frame(self.main\_frame)

        self.user\_menu\_frame.pack()

        self.welcome\_label = tk.Label(self.user\_menu\_frame, text=f"Welcome, {self.logged\_in\_user}")

        self.welcome\_label.pack(pady=5)

        self.logout\_button = tk.Button(self.user\_menu\_frame, text="Logout", command=self.logout)

        self.logout\_button.pack(pady=5)

        self.delete\_button = tk.Button(self.user\_menu\_frame, text="Delete Account", command=self.delete\_account)

        self.delete\_button.pack(pady=5)

    def logout(self):

        self.user\_menu\_frame.pack\_forget()

        self.logged\_in\_user = None

        self.username\_entry.delete(0, tk.END)

        self.password\_entry.delete(0, tk.END)

        self.login\_frame.pack()

    def delete\_account(self):

        if self.logged\_in\_user:

            c.execute("DELETE FROM users WHERE username=?", (self.logged\_in\_user,))

            conn.commit()

            self.logout()

            messagebox.showinfo("Success", "Account deleted successfully")

root = tk.Tk()

app = App(root)

root.mainloop()

# Close the database connection when the GUI window is closed

conn.close()