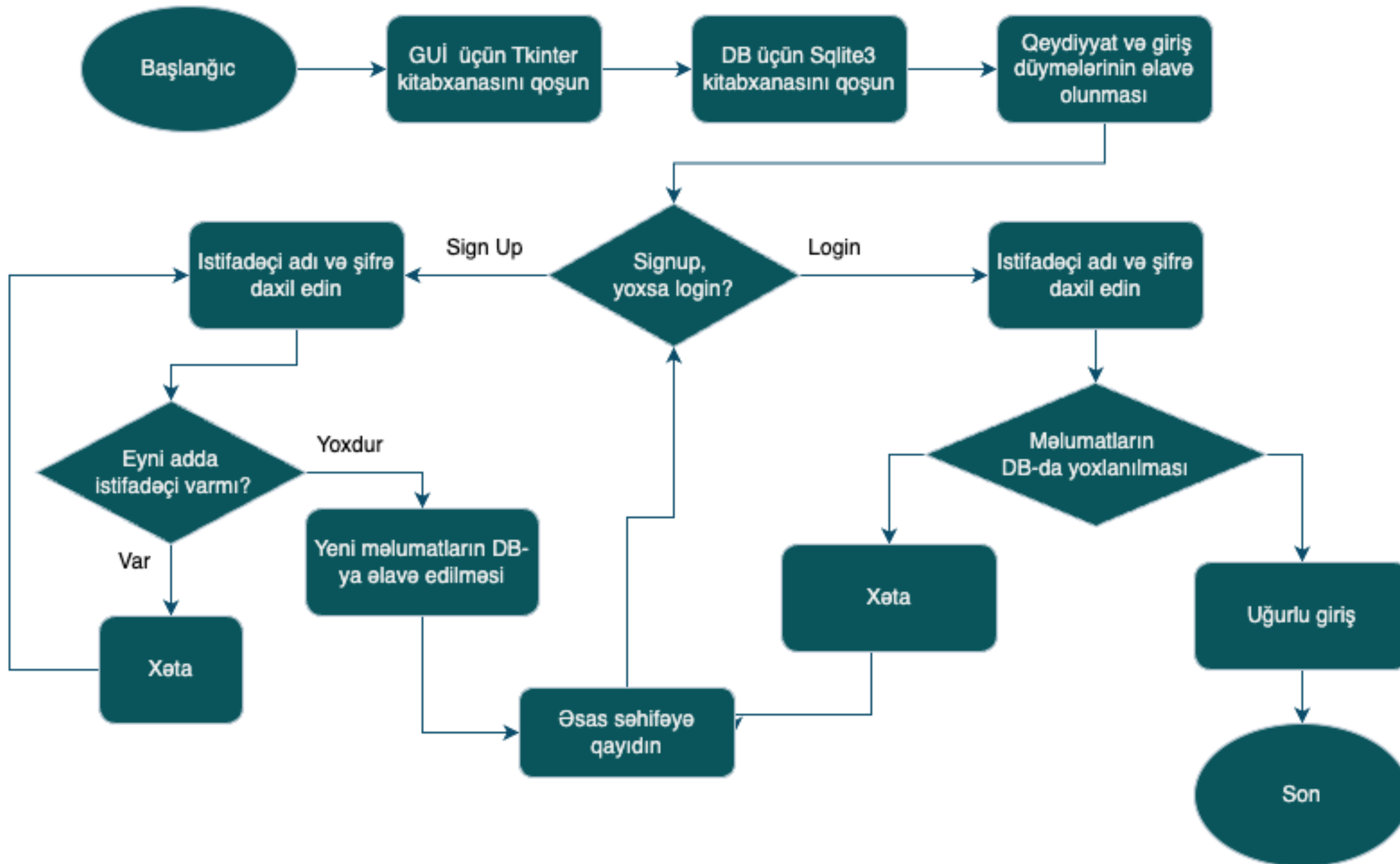


**İstifadəçi qruplarının Verilənlər
Bazası ilə yaradılması və idarə
olunması üçün proqram**



Alqoritmin blok-sxemlərlə təsviri



Kitabxanaların əlavə edilməsi və databazanın qurulması

```
import tkinter as tk
from tkinter import messagebox, simpledialog
import sqlite3

# Set up the SQLite database
def setup_database():
    conn = sqlite3.connect('users.db')
    cursor = conn.cursor()
    cursor.execute('''
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY,
    username TEXT NOT NULL UNIQUE,
    password TEXT NOT NULL
)
''')
    # Create default admin user
    cursor.execute('''
INSERT OR IGNORE INTO users (username, password) VALUES ('admin', 'admin')
''')
    conn.commit()
    conn.close()
```

Sqlite3 kitabxanasından istifadə edilərək lokalda **db faylı** yaradılır.

Default admin profili əlavə edilir.

Qeydiyyat panelinin qurulması

```
# Function to handle user sign-up
def signup():
    def submit_signup():
        username = signup_username_entry.get()
        password = signup_password_entry.get()

        if not username or not password:
            messagebox.showerror("Error", "All fields are required")
            return

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

        try:
            cursor.execute('INSERT INTO users (username, password) VALUES (?, ?)', (username, password))
            conn.commit()
            messagebox.showinfo("Success", "User signed up successfully!")
            main_page()
        except sqlite3.IntegrityError:
            messagebox.showerror("Error", "Username already exists")
        finally:
            conn.close()

    clear_frame()
    tk.Label(app, text="Sign Up", font=("Arial", 14)).pack(pady=10)
    tk.Label(app, text="Username").pack()
    signup_username_entry = tk.Entry(app, bg="black", bd=2, fg="white")
    signup_username_entry.pack(pady=5)
    tk.Label(app, text="Password").pack()
    signup_password_entry = tk.Entry(app, show="*", bg="black", bd=2, fg="white")
    signup_password_entry.pack(pady=5)
    tk.Button(app, text="Sign Up", command=submit_signup).pack(pady=20)
    tk.Button(app, text="Back to Main Page", command=main_page).pack(pady=5)
```

Daxil edilən **inputlar**
DB-ya əlavə edilir.

Əgər istifadəçi DB-da
mövcuddursa, ekrana
xəta çıxarır.

Tkinter ilə **GUI**
yaxşılaşdırılır.

Giriş panelinin qurulması

Daxil edilən **inputlar**
DB-ya sorğulanır.
Məlumatlar doğrudursa,
giriş uğurlu alınır.

Əgər istifadəçi
admindirsə, admin
panelinə daxil olur.

GUI yaxşılaşdırılır.

```
# Function to handle user login
def login():
    def submit_login():
        username = login_username_entry.get()
        password = login_password_entry.get()

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

        cursor.execute('SELECT * FROM users WHERE username = ? AND password = ?', (username, password))
        result = cursor.fetchone()

        if result:
            if username == "admin":
                admin_panel()
            else:
                messagebox.showinfo("Success", "Login successful!")
                main_page()
        else:
            messagebox.showerror("Error", "Invalid username or password")

        conn.close()

    clear_frame()
    tk.Label(app, text="Login", font=("Arial", 14)).pack(pady=10)
    tk.Label(app, text="Username").pack()
    login_username_entry = tk.Entry(app, bg="black", bd=2, fg="white")
    login_username_entry.pack(pady=5)
    tk.Label(app, text="Password").pack()
    login_password_entry = tk.Entry(app, show="*", bg="black", bd=2, fg="white")
    login_password_entry.pack(pady=5)
    tk.Button(app, text="Login", command=submit_login).pack(pady=20)
    tk.Button(app, text="Back to Main Page", command=main_page).pack(pady=5)
```

Admin panelinin qurulması

Qeydiyyatdakı istifadəçiləri görmək üçün
DB-dan məlumatlar sorğulanır.

```
# Function to handle the admin panel
def admin_panel():
    def view_users():
        clear_frame()
        tk.Label(app, text="Admin Panel - View Users", font=("Arial", 14)).pack(pady=10)

        conn = sqlite3.connect('users.db')
        cursor = conn.cursor()
        cursor.execute('SELECT id, username FROM users WHERE username != "admin"')
        users = cursor.fetchall()
        conn.close()

        for user in users:
            user_frame = tk.Frame(app)
            user_frame.pack(pady=5)
            tk.Label(user_frame, text=user[1]).pack(side=tk.LEFT)
            tk.Button(user_frame, text="Edit", command=lambda u=user: edit_user(u[0])).pack(side=tk.LEFT, padx=5)
            tk.Button(user_frame, text="Delete", command=lambda u=user: delete_user(u[0])).pack(side=tk.LEFT, padx=5)

    tk.Button(app, text="Back to Admin Panel", command=admin_panel).pack(pady=20)
```


Admin panelinə funksionallıqların əlavə edilməsi

İstifadəçiləri redaktə etmək və silmək üçün funksiyalar yaradılır.

```
def edit_user(user_id):
    new_password = simpledialog.askstring("Edit User", "Enter new password")
    if new_password:
        conn = sqlite3.connect('users.db')
        cursor = conn.cursor()
        cursor.execute('UPDATE users SET password = ? WHERE id = ?', (new_password, user_id))
        conn.commit()
        conn.close()
        messagebox.showinfo("Success", "Password updated successfully")
        view_users()

def delete_user(user_id):
    conn = sqlite3.connect('users.db')
    cursor = conn.cursor()
    cursor.execute('DELETE FROM users WHERE id = ?', (user_id,))
    conn.commit()
    conn.close()
    messagebox.showinfo("Success", "User deleted successfully")
    view_users()
```

Programın çalışdırılması

Login & Registration App

Main Page

Sign Up

Login

Login & Registration App

Sign Up


Username

elonmusk

Password

Sign Up

Back to Main Page



User signed up successfully!

OK

Admin panelinə giriş

Login & Registration App

Admin Panel

View Users

Back to Main Page

Login & Registration App

Login

Username

admin

Password

Login

Back to Main Page

Login & Registration App

Admin Panel - View Users

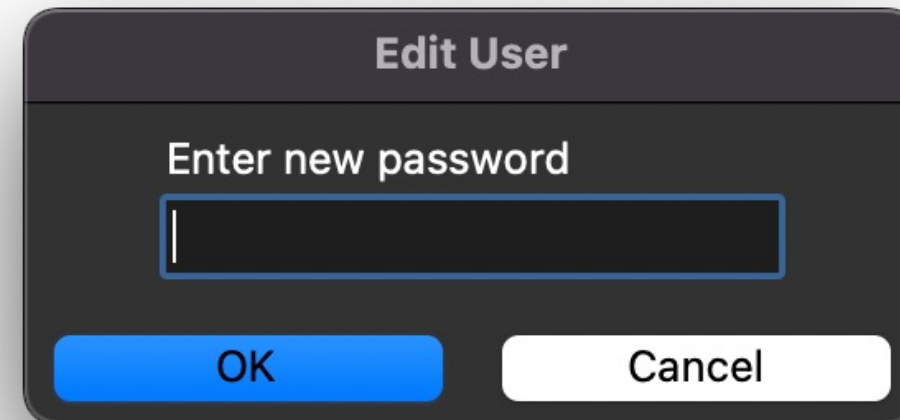
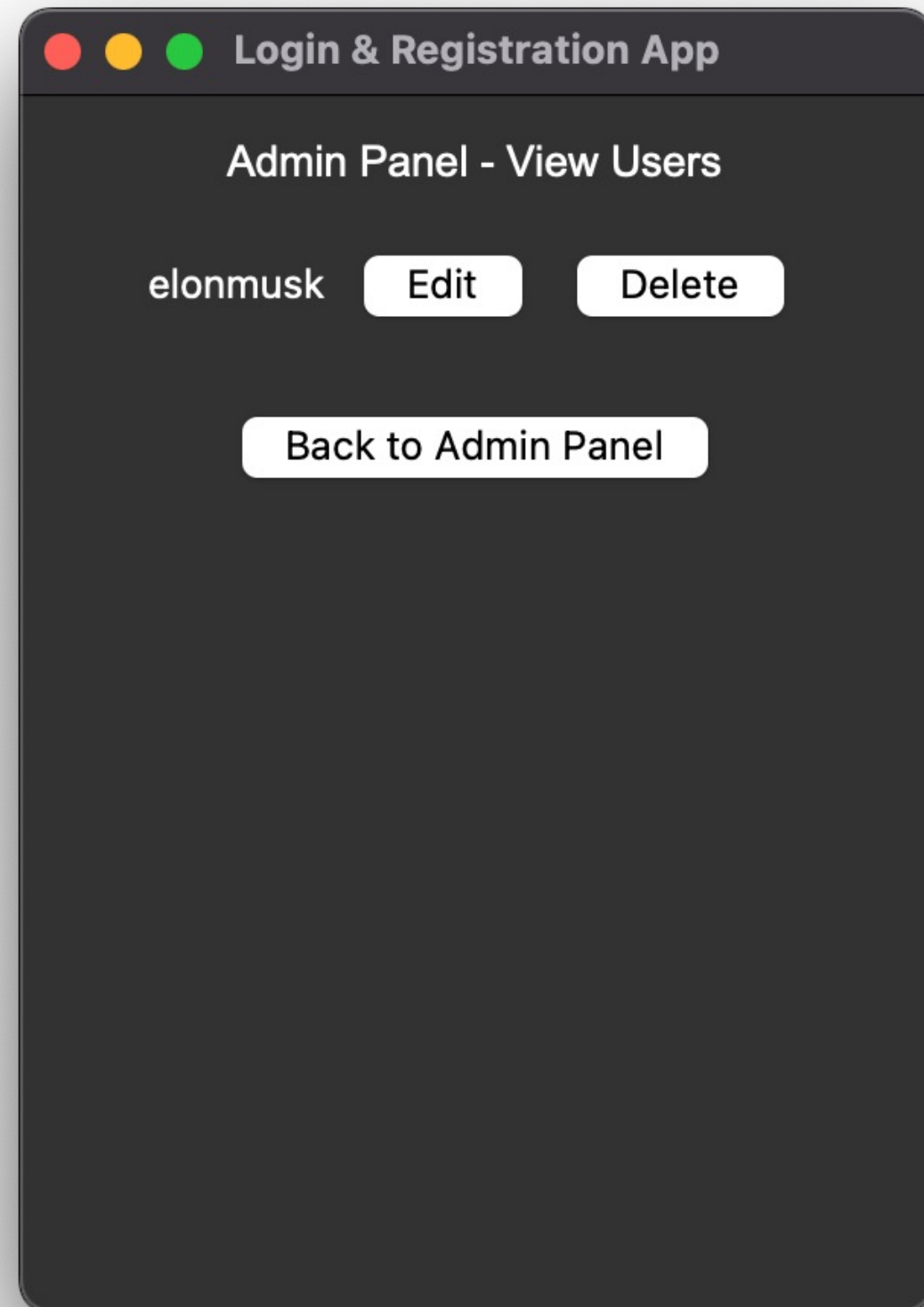
elonmusk

Edit

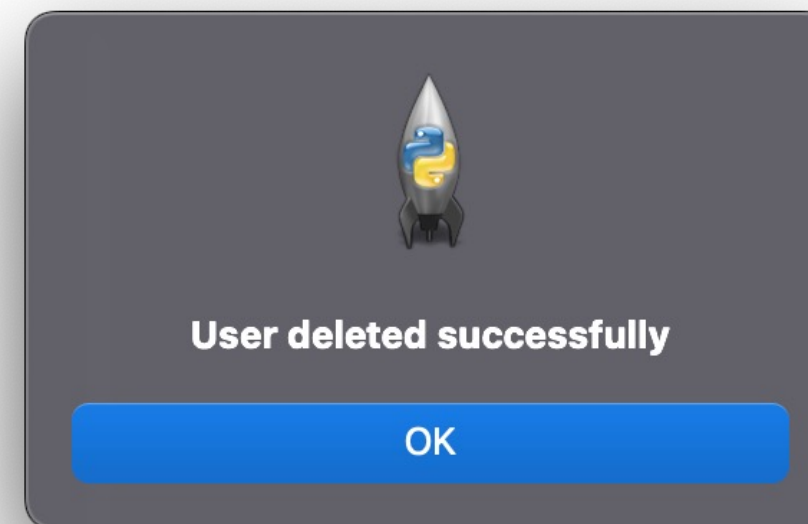
Delete

Back to Admin Panel

Redaktə və silmə prosesi



İstifadəçiyə yeni şifrə təyin olunması



İstifadəçinin DB-dan silinməsi