

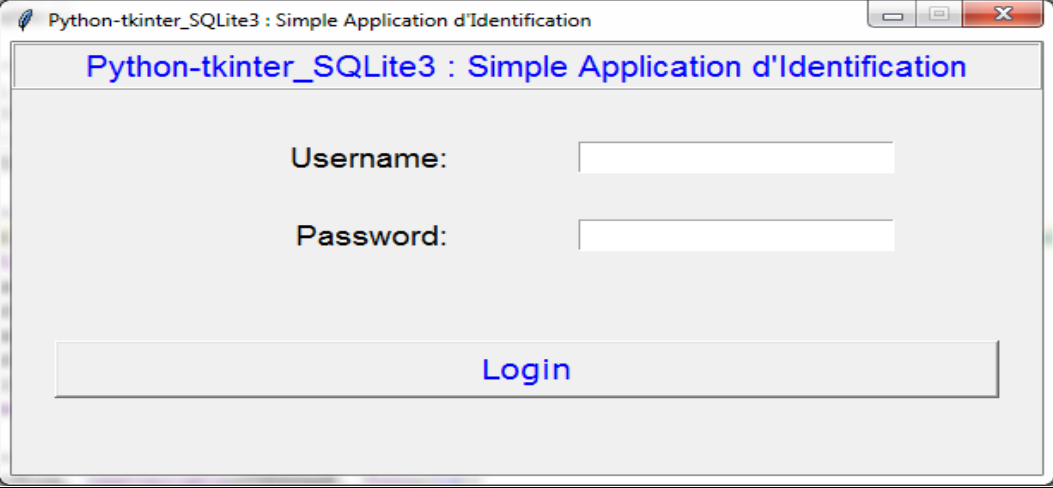
Interface Homme Machine Python-Tkinter-SQLite3 Travaux Pratiques N° 8

Objetif :

- Se connecter à une base de données SQLite3 en vue d'effectuer une identification.

Cette application utilise la base de données "filiere.db" créée lors du

Interface de l'application



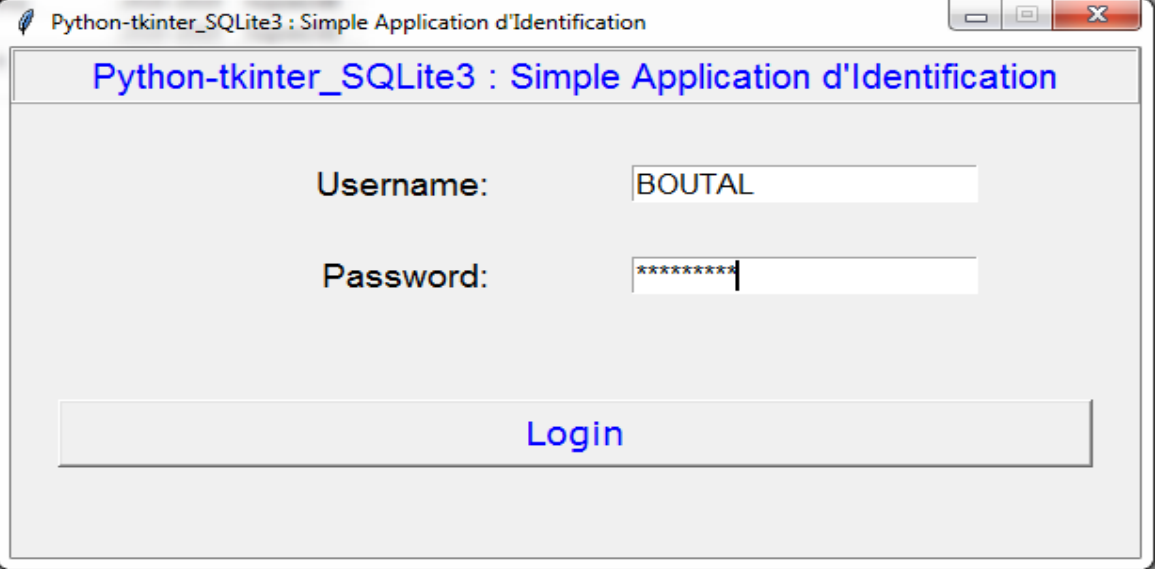
Python-tkinter_SQLite3 : Simple Application d'Identification

Username:

Password:

Login

L'étudiant doit fournir son nom et son password, puis cliquer sur le bouton "Login".



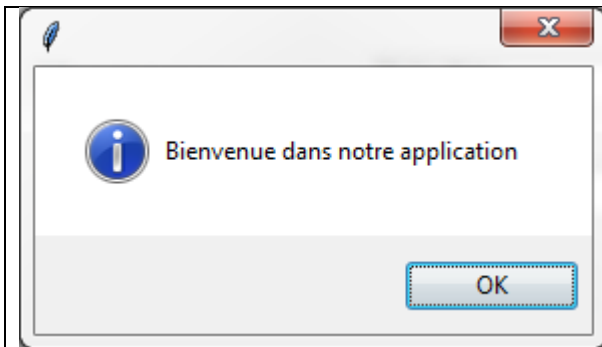
Python-tkinter_SQLite3 : Simple Application d'Identification

Username:

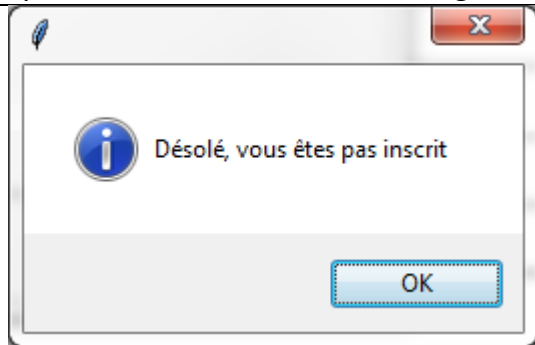
Password:

Login

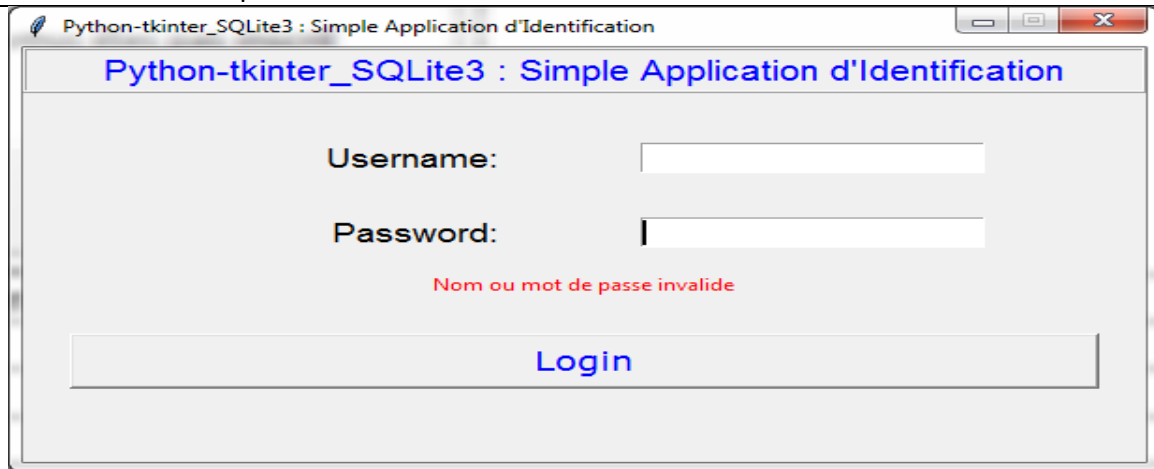
Si les informations correspondent à celles stockées dans la base de données (Table Etudiants), le système affiche une boîte de dialogue affichant un message de Bienvenue



Si les informations ne correspondent pas à celles stockées dans la base de données, le système affiche une boîte de dialogue affichant un message de désolation.



Il affiche également un message en rouge sur l'interface de l'application indiquant que le nom et le mot de passe fournis sont invalides



Code de l'application

```

from tkinter import *
from tkinter import messagebox
import sqlite3

#===== Fonction Login =====
def Login(event=None):
    global USERNAME
    global PASSWORD

    #1: Connexion à la base de données
    conn = sqlite3.connect("filiere.db")
    #2: Création du curseur
    cursor = conn.cursor()

    if USERNAME.get() == "" or PASSWORD.get() == "":

```

```

lbl_text.config(text="Prière de donner votre nom et votre mot de passe", fg="red")
messagebox.showinfo("", "Prière de donner votre nom et mot de passe!")
else:
    #==Exécuter l'instruction SELECT avec comme condition Nom=<<NomSaisi>> et passwor="mot de passe saisie
    cursor.execute("SELECT * FROM Etudiants WHERE NomEtudiant = ? AND Password = ?",
        (USERNAME.get(), PASSWORD.get()))

    #cursor.fetchone() retourne un tuple qui se trouve à la position courante du curseur
    #Cette ligne est un objet = None, si le curseur ne contient aucune ligne
    if cursor.fetchone() is not None:
        messagebox.showinfo("", "Bienvenue dans notre application")
        USERNAME.set("")
        PASSWORD.set("")
        lbl_text.config(text="")
    else:
        lbl_text.config(text="Nom ou mot de passe invalide", fg="red")
        messagebox.showinfo("", "Désolé, vous êtes pas inscrit")
        USERNAME.set("")
        PASSWORD.set("")
    cursor.close()
    conn.close()
#=====

root = Tk()
root.title("Python-tkinter_SQLite3 : Simple Application d'Identification")
width = 600
height = 300
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()
x = (screen_width/2) - (width/2)
y = (screen_height/2) - (height/2)
root.geometry("%dx%d+%d+%d" % (width, height, x, y))
root.resizable(0, 0)

# =====VARIABLES=====
USERNAME = StringVar()
PASSWORD = StringVar()

# =====FRAMES=====
Top = Frame(root, bd=2, relief=RIDGE)
Top.pack(side=TOP, fill=X)
Form = Frame(root, height=200)
Form.pack(side=TOP, pady=20)

# =====LABELS=====
lbl_title = Label(Top, text="Python-tkinter_SQLite3 : Simple Application d'Identification", font=('arial', 15), fg="blue")
lbl_title.pack(fill=X)
lbl_username = Label(Form, text="Username:", font=('arial', 14), bd=15)
lbl_username.grid(row=0, sticky="e")
lbl_password = Label(Form, text="Password:", font=('arial', 14), bd=15)
lbl_password.grid(row=1, sticky="e")
lbl_text = Label(Form)
lbl_text.grid(row=2, columnspan=2)

# =====ENTRY WIDGETS=====
username = Entry(Form, textvariable=USERNAME, font=(14))
username.grid(row=0, column=1)
password = Entry(Form, textvariable=PASSWORD, show="*", font=(14))
password.grid(row=1, column=1)

```

```

#=====

# =====BUTTON WIDGETS=====
btn_login = Button(Form, text="Login", width=45, fg="blue", font=("verdana", 14), command=Login)
btn_login.grid(pady=25, row=3, columnspan=2)
btn_login.bind('<Return>', Login)

#=====INITIALIATION=====
root.mainloop()

```

Annexe

Création des tables de la base de données "filiere.db"

#1: Importer le module sqlite3

```
import sqlite3
```

#2: création d'un objet Connection à la base de données(filiere dans ce cas)

```
conn = sqlite3.connect('filiere.db')
```

#3:Création d'un objet cursor qui permet d'exécuter les instructions SQL

```
c = conn.cursor()
```

4: Utilistion de la méthode execute de l'objet cursor pour exécuter une instruction SQL

#Création de la table Enseignants

```
c.execute("""CREATE TABLE IF NOT EXISTS Enseignants(
    CodeEnseignant TEXT CONSTRAINT pkEnseignants PRIMARY KEY,
    NomEnseignant TEXT,
    PrenomEnseignant TEXT,
    Genre TEXT,
    Statut TEXT,
    Username Text,
    Password TEXT)""")
```

#Création de la table Modules

```
c.execute("""CREATE TABLE IF NOT EXISTS Modules(
    CodeModule Text CONSTRAINT pkModules PRIMARY KEY,
    LibelleModule TEXT,
    CodeEnseignant TEXT,
    CONSTRAINT fkModEnseig FOREIGN KEY(CodeEnseignant) REFERENCES Enseignants(CodeEnseignant)""")
```

#Création de la Table ElementsModules

```
c.execute("""CREATE TABLE IF NOT EXISTS ElementsModules(
    CodeEltModule TEXT,
    LibelleEltModule TEXT,
    Coefficient REAL,
    VHCM INTEGER,
    VHTD INTEGER,
    VHTP INTEGER,
    VHEC INTEGER,
    CodeModule Text,
    CONSTRAINT pkEltModule PRIMARY KEY(CodeEltModule),
    CONSTRAINT fkEltModModule FOREIGN KEY(CodeModule) REFERENCES Modules(CodeModule),
```

```
CONSTRAINT ckCoefficient CHECK (Coefficient BETWEEN 0 AND 1)) """)
```

```
#Création de la table EnseignantsMatières
```

```
c.execute("""CREATE TABLE IF NOT EXISTS EnseignantsMatières(  
    CodeEnseignant TEXT,  
    CodeEltModule TEXT,  
    VHCM INTEGER,  
    VHTD INTEGER,  
    VHTP INTEGER,  
    VHEC INTEGER,  
    CONSTRAINT pkEnseigEltMod PRIMARY KEY (CodeEnseignant, CodeEltModule),  
    CONSTRAINT fkEnseigEltMod_Enseig FOREIGN KEY(CodeEnseignant) REFERENCES Enseignants(CodeEnseignant),  
    CONSTRAINT fkEnseigEltMod_EltMod FOREIGN KEY(CodeEltModule) REFERENCES EItModule(CodeModule)  
    )""")
```

```
#Création de la table TypesEpreuves
```

```
c.execute("""CREATE TABLE IF NOT EXISTS TypesEpreuves(  
    CodeTypeEpreuve TEXT,  
    LibelleTypeEpreuve TEXT,  
    CoefficientTypeEpreuve REAL,  
    CONSTRAINT pkTypesEpreuves PRIMARY KEY(CodeTypeEpreuve)  
    )""")
```

```
#Création de la table TypesEpreuves
```

```
c.execute("""CREATE TABLE IF NOT EXISTS Epreuves(  
    CodeEpreuve TEXT,  
    CodeTypeEpreuve TEXT,  
    CodeEltModule TEXT,  
    Date TEXT,  
    CONSTRAINT pkEpreuve PRIMARY KEY(CodeEpreuve),  
    CONSTRAINT fkEpreuveTypeEpreuve FOREIGN KEY(CodeTypeEpreuve) REFERENCES TypeEpreuve(CodeTypeEpreuve),  
    CONSTRAINT fkEpreuvesEltModule FOREIGN KEY (CodeEltModule) REFERENCES ElementsModules(CodeEltModule)  
    )""")
```

```
#Création de la table Etudiants
```

```
c.execute("""CREATE TABLE IF NOT EXISTS Etudiants(  
    NumInscription TEXT,  
    NomEtudiant TEXT,  
    PrenomEtudiant TEXT,  
    Genre TEXT,  
    Promotion Text,  
    Username Text,  
    Password Text,  
    CONSTRAINT pkEtudiants PRIMARY KEY(NumInscription)  
    )""")
```

```
#Création de la table Notes
```

```
c.execute("""CREATE TABLE IF NOT EXISTS Notes(  
    NumInscription TEXT,  
    CodeEpreuve TEXT,  
    Note REAL,  
    Remarque TEXT,  
    CONSTRAINT pkNotes PRIMARY KEY(NumInscription, CodeEpreuve),  
    CONSTRAINT fkNotesEpreuves FOREIGN KEY(CodeEpreuve) REFERENCES Epreuves(CodeEpreuve),  
    CONSTRAINT fkNotesEtudiants FOREIGN KEY(NumInscription) REFERENCES Etudiants(NumInscription),  
    CONSTRAINT ckNote CHECK(Note BETWEEN 0 AND 20)  
    )""")
```

```
#5: Valider les changements
conn.commit()
```

```
#6: Fermer la connexion.
conn.close()
```

Insertion des données dans les tables

```
#1: Importer le module sqlite3
```

```
import sqlite3
```

```
#2: création d'un objet Connection à la base de données(filiere dans ce cas)
```

```
conn = sqlite3.connect('filiere.db')
```

```
#3:Création d'un objet cursor qui permet d'exécuter les instructions SQL
```

```
c = conn.cursor()
```

```
# 4: Utilisation de la méthode execute de l'objet cursor pour exécuter une instruction SQL
```

```
#Insertion des données dans la table Enseignants
```

```
c.execute("""INSERT INTO Enseignants(CodeEnseignant, NomEnseignant, PrenomEnseignant, Genre, Statut, username,password) VALUES
('Ens001', 'AFRAOUI', 'Nadine', 'F', 'Permanent','AFRAOUI', 'topsecret'),
('Ens002', 'NAHLAOUI', 'Houyame', 'F', 'Permanent','NAHLAOUI', 'topsecret'),
('Ens003', 'BALGHOUTI', 'Hamid', 'H', 'Permanent','BALGHOUTI', 'topsecret'),
('Ens004', 'BOUCHAREB', 'Kamal', 'H', 'Permanent','BOUCHAREB', 'topsecret'),
('Ens005', 'FOULADI', 'Redouane', 'H', 'Permanent','FOUDALI', 'topsecret'),
('Ens006', 'HASSNAOUI', 'Adil', 'H', 'Permanent', 'HASNAOUI', 'topsecret'),
('Ens007', 'MOUZAD', 'Abdelilah', 'H', 'Permanent','MOUZAD', 'topsecret'),
('Ens008', 'FOUZZI', 'Omar', 'H', 'Permanent', 'FOUZZ', 'topsecret'),
('Ens009', 'ROUDANI', 'Ali', 'H', 'Permanent', 'ROUDANI', 'topsecret'),
('Ens010', 'TIZRAOUI', 'Hassan', 'H', 'Permanent', 'TIZRAOUI', 'topsecret'),
('Ens011', 'KAZANI', 'Abdelhamid', 'H', 'Vacataire', 'KAZANI', 'topsecret')
""")
```

```
#Insertion des données dans la table Modules
```

```
c.execute("""INSERT INTO Modules(CodeModule, LibelleModule, CodeEnseignant) VALUES
('M01', 'Génie Logiciel_1', 'Ens008'),
('M02', 'Programmation Orientée Objet Avancée', 'Ens006'),
('M03', 'Développement Web avec JavaEE et programmation Android', 'Ens010'),
('M04', 'Commutation de réseau local(CCNA3) et réseau étendu (CCNA4)', 'Ens009')
""")
```

```
#Insertion des données dans la table ElementsModules
```

```
c.execute("""INSERT INTO ElementsModules(CodeEltModule, LibelleEltModule, Coefficient, VHCM, VHTD, VHTP, VHEC, CodeModule)
VALUES
('M01A', 'Génie logiciel',0.5, 15, 6,0,3, 'M01' ),
('M01B', 'Méthodes agiles',0.5, 12, 9,0,3, 'M01' ),
('M02A', 'Programmation Orientée Objet avancée en Java', 0.7, 12, 8,12,3, 'M02' ),
('M02B', 'Programmation répartie en java : Java RMI', 0.3, 7.5, 3,3,1.5, 'M02' ),
('M03A', 'Développement web avec Java EE', 0.7, 12, 8, 12, 3, 'M03' ),
('M03B', 'Programmation Android', 0.3, 6, 3,4.5,1.5, 'M03' )
""")
```

```
c.execute("""INSERT INTO Etudiants(NumInscription, NomEtudiant, PrenomEtudiant, Promotion, username, password) VALUES
('Etu001', 'HABIT', 'Noura', '2018-2019', 'HABTI', 'topsecret'),
('Etu002', 'TALII', 'Houda', '2018-2019', 'TALII', 'topsecret'),
```

```
('Etu003', 'BALDI', 'Hassan', '2018-2019', 'BALDI', 'topsecret'),  
(Etu004', 'BOUTAL', 'Karim', '2018-2019', 'BOUTAL', 'topsecret'),  
(Etu005', 'FADLAN', 'Reda', '2018-2019', 'FADLAN', 'topsecret'),  
(Etu006', 'HARD', 'Driss', '2018-2019', 'HARD', 'topsecret'),  
(Etu007', 'MAZRAOUI', 'Farid', '2018-2019', 'MAZRAOUI', 'topsecret'),  
(Etu008', 'ZOUFIR', 'Oussama', '2018-2019', 'ZOUFIR', 'topsecret'),  
(Etu009', 'ROUTAD', 'Ali', '2018-2019', 'ROUTAD', 'topsecret'),  
(Etu010', 'ZAROUTI', 'Hassan', '2018-2019', 'ZAROUTI', 'topsecret'),  
(Etu011', 'KAZAOUI', 'Tarik', '2018-2019', 'KAZAOUI', 'topsecret')  
*****)
```

```
conn.commit()  
conn.close()
```