A Simple Login System With Python & Tkinter

First we will Install Required Libraries

Tkinter — It is Pre Installed with Python, but for sometimes it does not come. It happened with me.

```
apt-get install python-tk
```

Now, Coming Towards Bcrypt and SQlite3.

```
pip install bcrypt
```

SQlite3 Comes Pre Installed

Structuring our main.py

main.py is Main Entry Point for our App.

We will be Using OOP Methodology to program our Login System

```
from tkinter import *
from login import Login, Register


class MainWindow:
    def __init__(self):
        self.app = Tk()
        self.app.title("Login with Python")
        self.app.geometry("300x250")
        self.label = Label(self.app, text="Welcome To App")
        self.label.place(x=95, y=40)
        self.login = Button(self.app, text="Login",
                            pady=5, padx=30, command=login)
        self.login.place(x=100, y=100)
        self.register = Button(self.app, text="Register",
```

```
16                                  pady=5, padx=20, command=register)
17          self.register.place(x=100, y=150)
18
19      def run(self):
20          self.app.mainloop()
21
22
23  def login():
24      loginTk = Login()
25      loginTk.run()
26
27
28  def register():
29      registerTk = Register()
30      registerTk.run()
31
32
33  app = MainWindow()
34  app.run()
```

Here,

Line 1, Import all functions from tkinter Library

Line 3, Defines a Class called MainWindow for our Program

Line 4, It is __init__() Special Method for Classes which help to define local variables for Classes

In __init__() method, We define app and make it an instance of Tk() class of Tkinter

And other stuff is,

Title — It is used to assign a Title to our Main Window

Geometry — It is a String used to define Height and Width of our GUI

Label — It is used to print text on Screen or GUI

Button — It is used to create a Button

Add these 2 lines below above code to run it..

```
app = MainWindow()
app.run()
```

And then go to terminal, and cd into that directory.

Use following command to run python file

```
cd YourDirectory
python3 main.py
```

On ***Windows,***

```
python main.py
```

This Code will produce error, as some things need to be changed,

Add a Login and register function in code

Code Should Look Something like this,

from tkinter import *
from login import Login, Register

```
class MainWindow:
 def __init__(self):
 self.app = Tk()
 self.app.title("Login with Python")
 self.app.geometry("300x250")
 self.label = Label(self.app, text="Welcome To App")
 self.label.place(x=95, y=40)
 self.login = Button(self.app, text="Login",
 pady=5, padx=30, command=login)
 self.login.place(x=100, y=100)
 self.register = Button(self.app, text="Register",
```

```
 pady=5, padx=20, command=register)
 self.register.place(x=100, y=150)

def run(self):
 self.app.mainloop()

def login():
 loginTk = Login()
 loginTk.run()

def register():
 registerTk = Register()
 registerTk.run()

app = MainWindow()
app.run()
```

Now, You need to create a file named “login.py” to store Login and Register Class.

login.py

```
from tkinter import *
from tkinter import messagebox
import bcrypt
from database import Database

db = Database()
db.createTable()

class Login:

“””
 Class for Login
 @param username
 @param password
 “””

def __init__(self):
 “””
 Class Init Method for GUI
 :params — loginWindow, label, username
 “””

# Variables for Tkinter
 self.loginWindow = Tk()
 self.loginWindow.title(“Login with Python”)
 self.loginWindow.geometry(“300x250”)
```

```
 self.label = Label(self.loginWindow, text=”Login”)
 self.label.place(x=95, y=40)

# Just Creepy Tkinter Stuff
 self.usernameS = StringVar()
 self.passwordS = StringVar()

self.usernameE = Entry(
 self.loginWindow, relief=FLAT, textvariable=self.usernameS)
 self.usernameE.place(x=70, y=80)
 self.passwordE = Entry(
 self.loginWindow, show=”*”, relief=FLAT,
textvariable=self.passwordS)
 self.passwordE.place(x=70, y=120)

# Actual Variales
 self.username = self.usernameS.get()
 self.password = self.passwordS.get()

self.submit = Button(self.loginWindow, text=”Submit”,
 pady=5, padx=20, command=self.validate)
 self.submit.place(x=100, y=150)

def validate(self):
 data = (self.username,)
 inputData = (self.username, self.password,)
 try:
 if (db.validateData(data, inputData)):
 messagebox.showinfo(“Successful”, “Login Was Successful”)
 else:
 messagebox.showerror(“Error”, “Wrong Credentials”)
 except IndexError:
 messagebox.showerror(“Error”, “Wrong Credentials”)

def run(self):
 self.loginWindow.mainloop()

class Register:

“””
 Class for Register
 @param username
 @param password
 “””

def __init__(self):

self.registerWindow = Tk()
 self.registerWindow.title(“Register with Python”)
 self.registerWindow.geometry(“300x250”)
 self.label = Label(self.registerWindow, text=”Register”)
 self.label.place(x=95, y=40)
```

```
# Just Creepy Tkinter Stuff
 self.usernameS = StringVar()
 self.passwordS = StringVar()

self.usernameE = Entry(self.registerWindow,
 relief=FLAT, textvariable=self.usernameS)
 self.usernameE.place(x=70, y=80)
 self.passwordE = Entry(self.registerWindow, show="*",
 relief=FLAT, textvariable=self.passwordS)
 self.passwordE.place(x=70, y=120)
 self.submit = Button(self.registerWindow,
 text="Submit", pady=5, padx=20, command=self.add)
 self.submit.place(x=100, y=150)

# Actual Variales
 self.username = self.usernameS.get()
 self.password = self.passwordS.get()

self.salt = bcrypt.gensalt()
 self.hashed = bcrypt.hashpw(self.password.encode(), self.salt)

def run(self):
 self.registerWindow.mainloop()

def add(self):
 data = (self.username,)

result = db.searchData(data)

print(result)

if result != 0:
 data = (self.username, self.hashed)
 db.insertData(data)
 messagebox.showinfo("Successful", "Username Was Added")
 else:
 messagebox.showwarning("Warning", "Username already Exists")
```

Explaining Code

First of all, We Import Libraries —

```
Bcrypt — for Encrypting Password

Tkinter — GUI library

database — It is our python file which has some SQlite Code in it
```

Then, We create Instance of Database Class present inside database.py

And Then, call method of that class — Which will create a Table for us

```
db = Database()
db.createTable()
```

We Defined Login Class, which will handle all Login related Stuff

Then we define **Dunder Method or Special Method of Python**

```
def __init__():
```

In that, we have some tkinter related stuff.

Second Method of Login Class:

```
def validate():
```

This method will Validate Login and match Password.

It arranges username — for finding user in database in a tuple

Then arranges input data in tuple

```
def validate(self):
 data = (self.username,)
 inputData = (self.username, self.password,)
```

After That, We check call a method from Database Class to ValidateData or check Data.

```
try:
 if (db.validateData(data, inputData)):
```

```
 messagebox.showinfo("Successful", "Login Was Successful")
 else:
 messagebox.showerror("Error", "Wrong Credentials")
```

If you look carefully, We Started a Try — Except Block.

We check if we have some return value and then show the user "They have Successfully Logged In"

Else we show them error

Except Block —

```
except IndexError:
 messagebox.showerror("Error", "Wrong Credentials")
```

If We Not found them in Database, then SQlite will throw **IndexError,** which we can tackle easily..

After that we defined a Register Class — Which will do the same but it adds it the to database and with

```
self.salt = bcrypt.gensalt()
self.hashed = bcrypt.hashpw(self.password.encode(), self.salt)
```

It used to encrypt Password.

database.py

```
# Importing Important Libraries

import sqlite3
import bcrypt

class Database:
 '''
 Database Class for sqlite3
```

```
 :params conn — sqlite3Connection
 :params curr — cursor
 '''

def __init__(self):
 try:
 self.conn = sqlite3.connect("test.db")
 print("Successfully Opened Database")
 self.curr = self.conn.cursor()
 except:
 print("Failed")

def createTable(self):
 '''
 Method for Creating Table in Database
 '''

create_table = '''
 CREATE TABLE IF NOT EXISTS cred(
 id Integer PRIMARY KEY AUTOINCREMENT,
 username TEXT NOT NULL,
 password TEXT NOT NULL
 );
 '''

self.curr.execute(create_table)
 self.conn.commit()

def insertData(self, data):
 '''
 Method for Insertig Data in Table in Database
 '''

insert_data = """
 INSERT INTO cred(username, password)
 VALUES(?, ?);
 """
 self.curr.execute(insert_data, data)
 self.conn.commit()

def searchData(self, data):
 '''
 Method for Searching Data in Table in Database
 '''

search_data = '''
 SELECT * FROM cred WHERE username = (?);
 '''

self.curr.execute(search_data, data)
```

```
rows = self.curr.fetchall()
 if rows == []:
 return 1
 return 0

def validateData(self, data, inputData):
 '''
 Method for Validating Data Table in Database
 '''

print(data)
 print(inputData)

validate_data = """
 SELECT * FROM cred WHERE username = (?);
 """

self.curr.execute(validate_data, data)
 row = self.curr.fetchall()

if row[0][1] == inputData[0]:
 return row[0][2] == bcrypt.hashpw(inputData[1].encode(), row[0][2])
```

These are some bunch of SQL Commands executed with Python.