Database Integration in python with Tkinter

Database integration in Python allows applications to interact with databases, enabling data storage, retrieval, and manipulation. Python supports various databases, including MySQL, PostgreSQL, and SQLite, through libraries like **`mysql-connector-python`**

Step 1: Install MySQL Connector

If you haven't installed the MySQL connector, use this below command in CMD command prompt :

```
pip install mysql-connector-python
```

Step 2: Create a MySQL Database and Table step by step

```
CREATE DATABASE mydatabase;
```

```
USE mydatabase;
```

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100)
);
```

Run these above commands in CMD

Basic Login & Register page

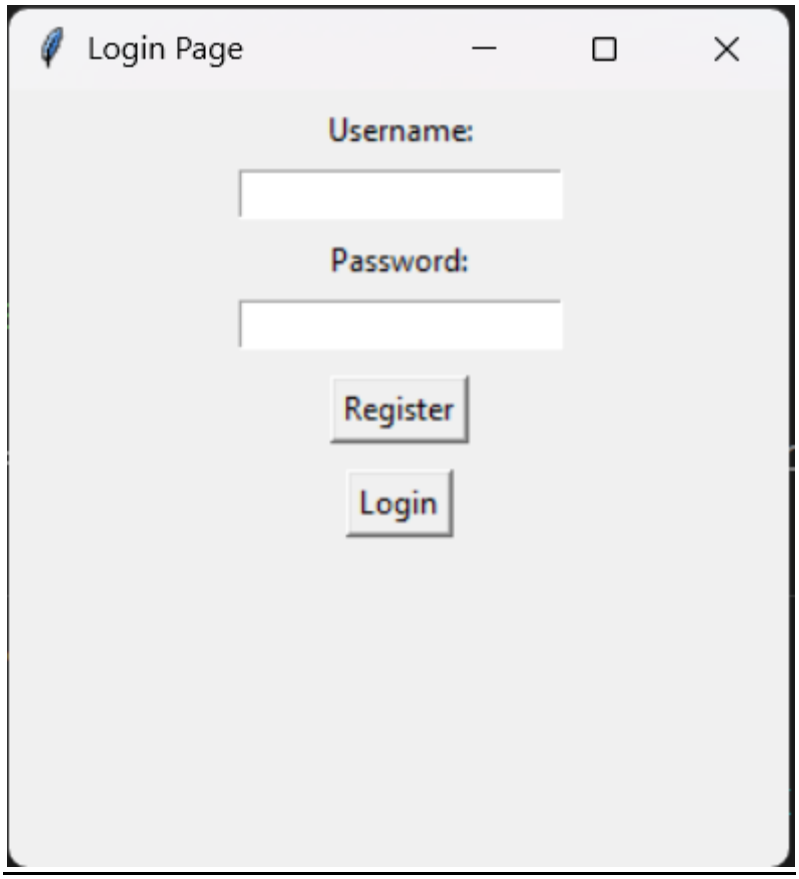
Step 3: Python Tkinter Code with MySQL Integration

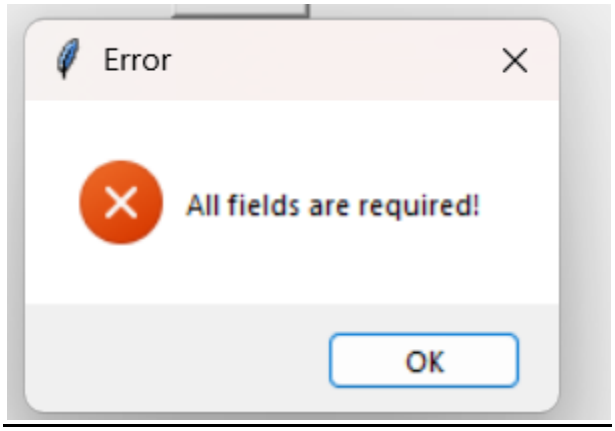
- **Import Tkinter :** used to import Tkinter library to run the code written using Tkinter attributes in python
- **Import mysql.connector :** This header file is used to establish connection to database.
- **Import bcrypt : This** is specified library used to encrypt the password and convert it into hash code
- **From Tkinter import messagebox :** This header file helps to display the dialog-box for "Errors" or to show "info"

```python
import tkinter as tk
from tkinter import messagebox
import mysql.connector
import bcrypt

def connect_db():
    return mysql.connector.connect(host="localhost", user="root", password="root",database="login_page")

                                #insert the username and password accordingly as set in your mysql appliction

def hash_password(password):
    return bcrypt.hashpw(password.encode(),bcrypt.gensalt())

def check_pass(entered_password,hash_code):
    return bcrypt.checkpw(entered_password.encode(),hash_code.encode() )
def register():
    username = Entry_username.get()
    password = Entry_password.get()
    hashed_password= hash_password(password).decode()

    if username == "" or password == "":
        messagebox.showerror("Error","All  Fields are required..!")
        return
    try:

       conn = connect_db()
       cursor=conn.cursor()
       cursor.execute("INSERT INTO users (username, password) values(%s,%s)",(username, hashed_password))

       conn.commit()
       conn.close()
       messagebox.showinfo("success","Registered successfully")
    except mysql.connector.IntegrityError:
        messagebox.showerror("Error.. username already exists..")
    except mysql.connector.Error as err:
        messagebox.showerror("Error",str(err))

def login():
    username = Entry_username.get()
    password = Entry_password.get()



    try:

       conn = connect_db()
       cursor=conn.cursor()
       cursor.execute("SELECT password FROM users WHERE username=%s",[username])
       result = cursor.fetchone()



       conn.close()

       if result and check_pass(password, result[0]):
           messagebox.showinfo("success","Login Successfull..!")
       else:
           messagebox.showerror("ERROR Invalid username or password..!")

    except mysql.connector.Error as err:
       messagebox.showerror("Data base error", str(err))
```

```
 #Basic user interface

window = tk.Tk()
window.title("Login Page")
window.geometry("300x300")


tk.Label(window, text="Username :").pack(pady=5)
Entry_username = tk.Entry(window)
Entry_username.pack()

tk.Label(window, text="password :").pack(pady=5)
Entry_password = tk.Entry(window, show="*")
Entry_password.pack()

tk.Button(window, text="Register", command=register).pack(pady=5)
tk.Button(window, text="Login", command=login).pack(padx =10,pady=5)



window.mainloop()
```

OUTPUT:

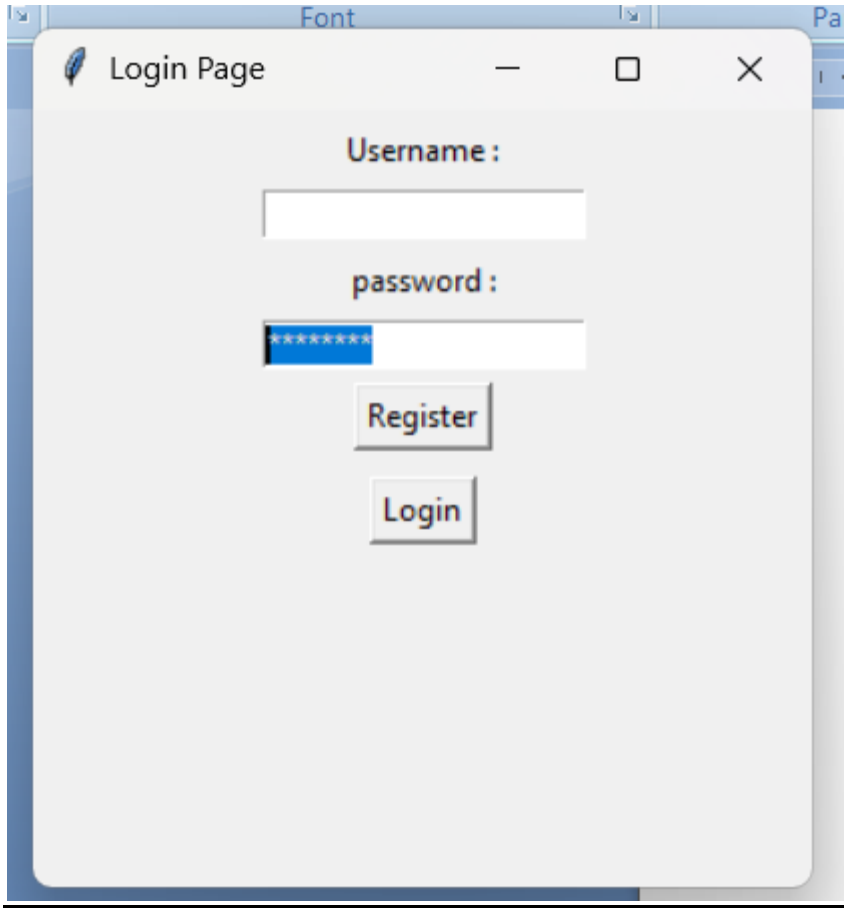


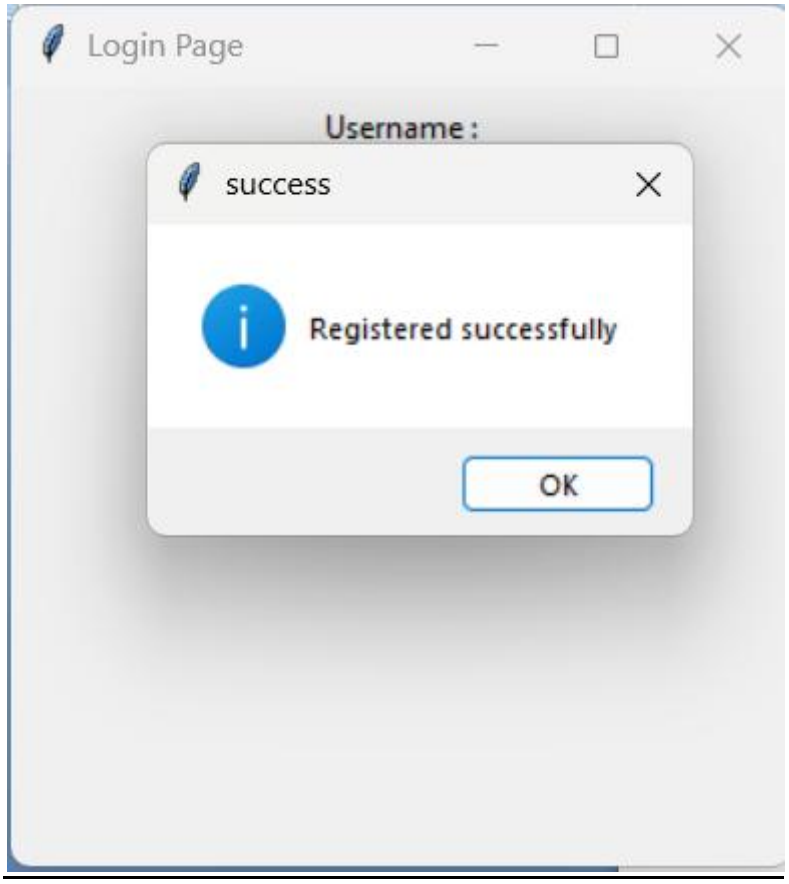


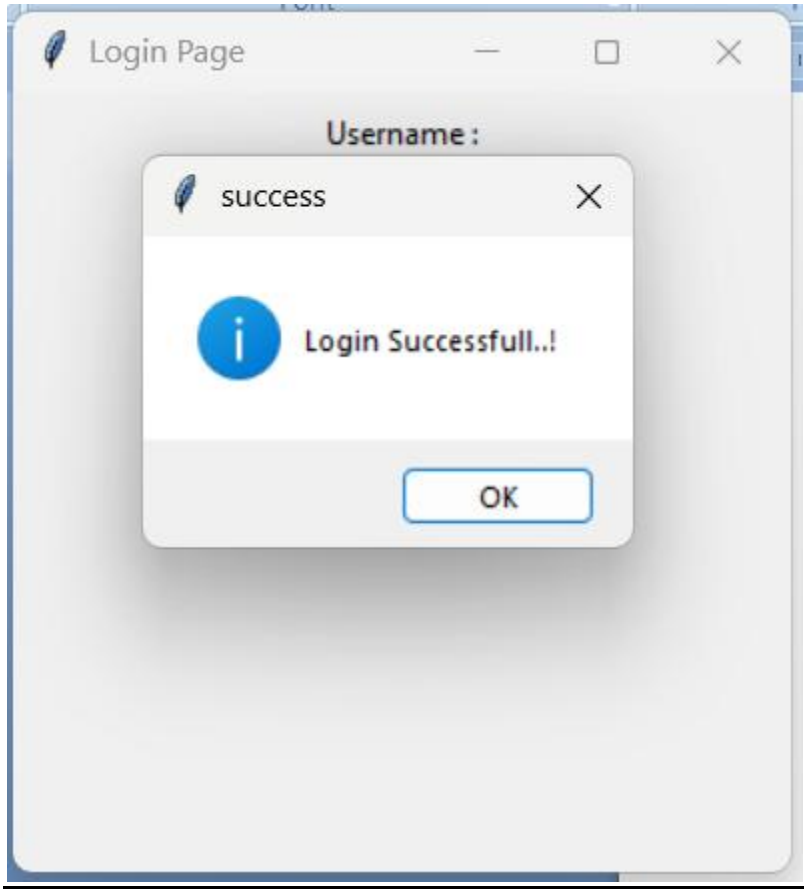
Error encounters when both fields are empty :

Password will be in **** for privacy.. using this below code for lable

```
tk.Label(window, text="password :").pack(pady=5)
Entry_password = tk.Entry(window, show="*")
Entry_password.pack()
```







How to see the stored Data in DataBase

Select that specified Database used in the code and run the below commands in mysql : (CMD)

- Show Databases;

- Use Database_name; (enter your database name in the place of Database_name)
- Select *from Table_name; (enter the table in which you have inserted the data in python code)