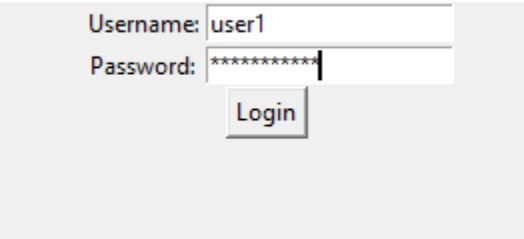
Login and Logout example with mysql database using Tkinter :



To match a username and password from a MySQL database in a Tkinter-based login form, you'll need to connect to a MySQL database using Python's `mysql-connector` library. Here's an example of how to do it:

1. Install the MySQL connector for Python:

   ```
   Copy code
   pip install mysql-connector-python
   ```

2. Create a MySQL database and a table for storing user credentials. For instance, the database might be named `login_db` and the table `users` might look like this:

   Sql command:-

   ```
   CREATE DATABASE login_db;

   USE login_db;

   CREATE TABLE users (
       id INT AUTO_INCREMENT PRIMARY KEY,
       username VARCHAR(50) NOT NULL,
       password VARCHAR(255) NOT NULL
   );

   INSERT INTO users (username, password) VALUES ('user1', 'password123');
   ```

3. Below is the full Tkinter application with a login form that checks the username and password against a MySQL database:

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

# Database connection setup
def connect_to_db():
    # Replace with your own MySQL credentials
    return mysql.connector.connect(
        host="localhost",
        user="root",          # Your MySQL username
        password="",  # Your MySQL password
        database="login_db"   # Your database name
    )

class App:
    def __init__(self, root):
        self.root = root
        self.root.title("Login Example")

        # Frames for login and logged in views
        self.login_frame = tk.Frame(self.root)
        self.logged_in_frame = tk.Frame(self.root)

        # Create login view widgets
        self.username_label = tk.Label(self.login_frame, text="Username:")
        self.username_entry = tk.Entry(self.login_frame)

        self.password_label = tk.Label(self.login_frame, text="Password:")
        self.password_entry = tk.Entry(self.login_frame, show="*")

        self.login_button = tk.Button(self.login_frame, text="Login",
command=self.login)

        self.username_label.grid(row=0, column=0)
        self.username_entry.grid(row=0, column=1)
        self.password_label.grid(row=1, column=0)
        self.password_entry.grid(row=1, column=1)
        self.login_button.grid(row=2, columnspan=2)

        # Create logged-in view widgets
        self.logged_in_label = tk.Label(self.logged_in_frame, text="Welcome!")
        self.logout_button = tk.Button(self.logged_in_frame, text="Logout",
command=self.logout)

        self.logged_in_label.grid(row=0, column=0)
        self.logout_button.grid(row=1, columnspan=2)
```

```
        # Start with login screen
        self.login_frame.pack()

    def login(self):
        username = self.username_entry.get()
        password = self.password_entry.get()

        # Connect to MySQL database and validate credentials
        db_connection = connect_to_db()
        cursor = db_connection.cursor()

        query = "SELECT * FROM users WHERE username = %s AND password = %s"
        cursor.execute(query, (username, password))

        result = cursor.fetchone()

        if result:
            # Switch to logged-in view
            self.login_frame.pack_forget()
            self.logged_in_frame.pack()
        else:
            # Show error message
            messagebox.showerror("Login Failed", "Incorrect username or
password.")

        # Close the database connection
        cursor.close()
        db_connection.close()

    def logout(self):
        # Switch back to login view
        self.logged_in_frame.pack_forget()
        self.login_frame.pack()

# Create the main window
root = tk.Tk()
app = App(root)
root.mainloop()
```

How this works:

1. **MySQL Database Connection**:

- The function `connect_to_db()` establishes a connection to a MySQL database (`login_db`) using MySQL credentials.

2. **Login Validation**:
   - When the user clicks the login button, it checks the database using the SQL query:

     ```
     sql
     Copy code
     SELECT * FROM users WHERE username = %s AND password = %s
     ```

   - If the result is found (meaning the username and password match), the user is logged in and the GUI switches to a logged-in view.
   - If the credentials are incorrect, a message box is shown with an error.
3. **Logout**:
   - Clicking the logout button switches the user back to the login screen.

Steps to Set Up:

1. Make sure you have MySQL installed and a database named `login_db` with a table `users` that has at least `username` and `password` fields.
2. Replace the connection details in `connect_to_db()` with your actual MySQL credentials.
3. Run the Python script, and try logging in with the credentials stored in the database.