

TRƯỜNG ĐẠI HỌC HỌC VĂN LANG  
KHOA CÔNG NGHỆ THÔNG TIN



# **Báo Cáo**

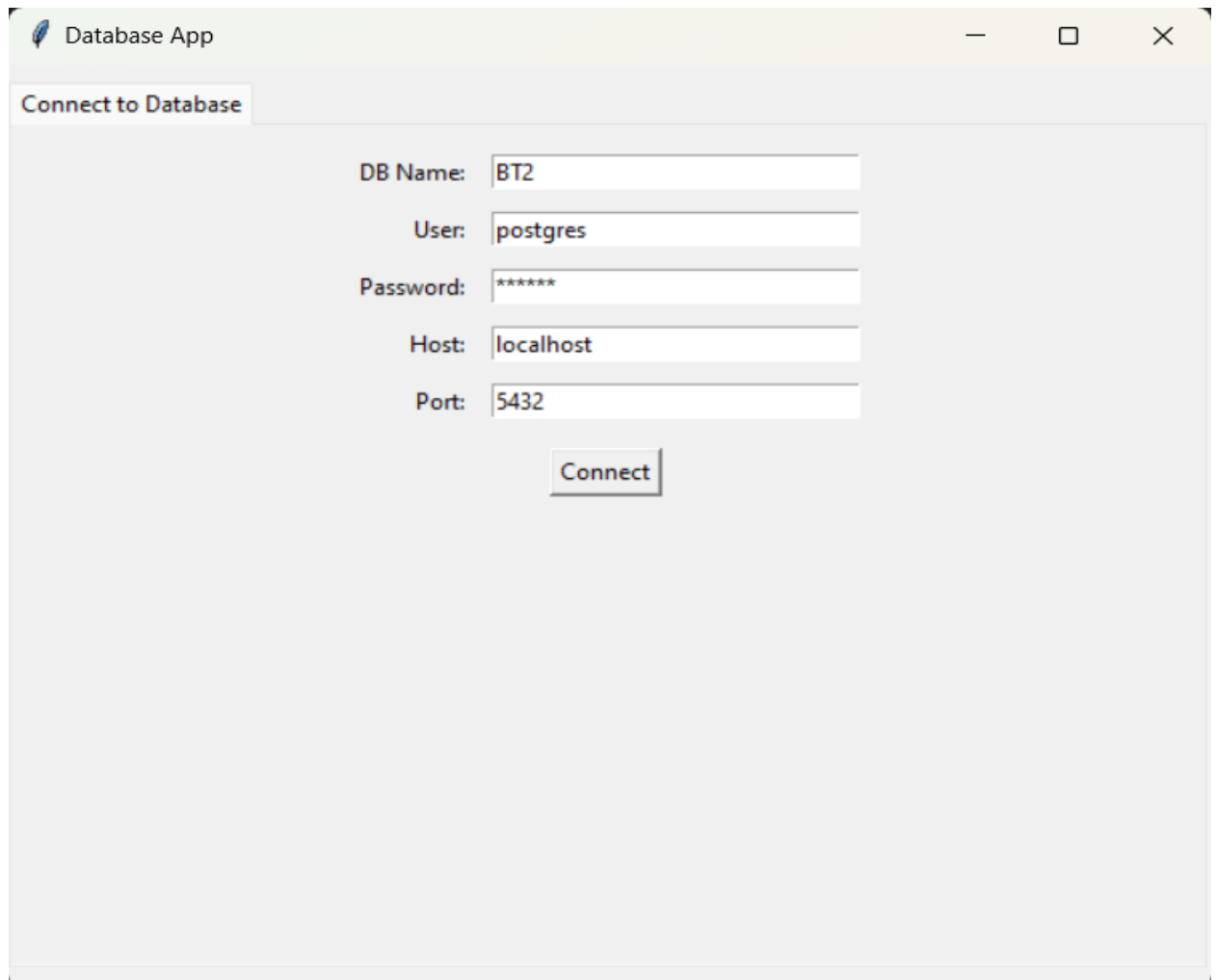
## **Bài Tập: DATABASE**

**SVTH:** Lâm Quang Tùng  
**MSSV:** 2274802010973

**GVHD:** Huỳnh Thái Học

2024

## 1. Giao diện



The image shows a window titled "Database App" with a standard macOS-style title bar (yellow, with minimize, maximize, and close buttons). Below the title bar is a tab labeled "Connect to Database". The main area of the window contains a form with the following fields:

- DB Name:
- User:
- Password:
- Host:
- Port:

Below these fields is a "Connect" button.

Database App

—

□

×

Connect to Database

Load Data

MSSV:

Họ và Tên:

Lớp:

Địa chỉ:

ID	MSSV	HOTEN	LOP	DIACHI
----	------	-------	-----	--------

Add

Update

Delete

Load Data

## 2. Chức năng

- Login vào database
- Thêm database
- Xóa database
- Chỉnh sửa database

## 3. Mã nguồn

```
1  import tkinter as tk
2  from tkinter import messagebox
3  from tkinter import ttk
4  import psycopg2
5  from psycopg2 import sql
6
7  class DatabaseApp:
8      def __init__(self, root):
9          self.root = root
10         self.root.title("Database App")
11         self.root.geometry("600x500") # Set kích thước cửa sổ chính
12
13         # Tạo notebook để chứa các tab
14         self.notebook = ttk.Notebook(self.root)
15         self.notebook.pack(pady=10, expand=True, fill="both")
16
17         # Tạo frame cho tab kết nối
18         self.connection_tab = tk.Frame(self.notebook)
19         self.notebook.add(self.connection_tab, text='Connect to Database')
20
21         # Tạo frame cho tab load data (ban đầu ẩn)
22         self.query_tab = tk.Frame(self.notebook, bg="lightgray")
23
24         # Database connection fields
25         self.db_name = tk.StringVar(value='BT2')
26         self.user = tk.StringVar(value='postgres')
27         self.password = tk.StringVar(value='123456')
28         self.host = tk.StringVar(value='localhost')
29         self.port = tk.StringVar(value='5432')
30         self.table_name = tk.StringVar(value='danh sach')
31
32         # Tạo các widget cho tab kết nối
33         self.create_widgets()
34
```

```

def create_widgets(self):
    # Connection section
    connection_frame = tk.Frame(self.connection_tab)
    connection_frame.pack(pady=10, padx=10)

    # Labels và Entries cho kết nối
    tk.Label(connection_frame, text="DB Name:").grid(row=0, column=0, padx=5, pady=5, sticky="e")
    tk.Entry(connection_frame, textvariable=self.db_name, width=30).grid(row=0, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(connection_frame, text="User:").grid(row=1, column=0, padx=5, pady=5, sticky="e")
    tk.Entry(connection_frame, textvariable=self.user, width=30).grid(row=1, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(connection_frame, text="Password:").grid(row=2, column=0, padx=5, pady=5, sticky="e")
    tk.Entry(connection_frame, textvariable=self.password, show="*", width=30).grid(row=2, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(connection_frame, text="Host:").grid(row=3, column=0, padx=5, pady=5, sticky="e")
    tk.Entry(connection_frame, textvariable=self.host, width=30).grid(row=3, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(connection_frame, text="Port:").grid(row=4, column=0, padx=5, pady=5, sticky="e")
    tk.Entry(connection_frame, textvariable=self.port, width=30).grid(row=4, column=1, padx=5, pady=5, sticky="ew")

    tk.Button(connection_frame, text="Connect", command=self.connect_db).grid(row=5, columnspan=2, pady=10)

```

```

def connect_db(self):
    try:
        self.conn = psycopg2.connect(
            dbname=self.db_name.get(),
            user=self.user.get(),
            password=self.password.get(),
            host=self.host.get(),
            port=self.port.get()
        )
        self.cur = self.conn.cursor()
        messagebox.showinfo("Success", "Connected to the database successfully!")

        # Sau khi kết nối thành công, thêm tab Load data
        self.create_query_tab()

        # Chuyển sang tab Load data
        self.notebook.add(self.query_tab, text='Load Data')
        self.notebook.select(self.query_tab)

    except Exception as e:
        messagebox.showerror("Error", f"Error connecting to the database: {e}")

```

```

def create_query_tab(self):
    # Query section
    query_frame = tk.Frame(self.query_tab, bg="lightblue")
    query_frame.pack(pady=10, padx=10, expand=True, fill="both")

    # Insert section
    insert_frame = tk.Frame(query_frame)
    insert_frame.pack(pady=10, fill=tk.X)

    self.column1 = tk.StringVar() # Mã số sinh viên
    self.column2 = tk.StringVar() # Họ và tên
    self.column3 = tk.StringVar() # Lớp
    self.column4 = tk.StringVar() # Địa chỉ

    tk.Label(insert_frame, text="MSSV:").grid(row=0, column=0, padx=5, pady=5, sticky="e")
    tk.Entry(insert_frame, textvariable=self.column1, width=30).grid(row=0, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(insert_frame, text="Họ và Tên:").grid(row=1, column=0, padx=5, pady=5, sticky="e")
    tk.Entry(insert_frame, textvariable=self.column2, width=30).grid(row=1, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(insert_frame, text="Lớp:").grid(row=2, column=0, padx=5, pady=5, sticky="e")
    tk.Entry(insert_frame, textvariable=self.column3, width=30).grid(row=2, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(insert_frame, text="Địa chỉ:").grid(row=3, column=0, padx=5, pady=5, sticky="e")
    tk.Entry(insert_frame, textvariable=self.column4, width=30).grid(row=3, column=1, padx=5, pady=5, sticky="ew")

    # Treeview setup
    danhsach = ('ID', 'MSSV', 'HOTEN', 'LOP', 'DIACHI')
    self.listBox = ttk.Treeview(query_frame, columns=danhsach, show='headings', height=7)

    for col in danhsach:
        self.listBox.heading(col, text=col)
        self.listBox.column(col, width=100)
    self.listBox.pack(pady=10, padx=10, expand=True, fill="both")

    # Bắt sự kiện khi người dùng nhấn đúp chuột vào một dòng
    self.listBox.bind('<Double-1>', self.get_value)

    # Thêm các nút Add, Update, Delete
    button_frame = tk.Frame(query_frame, bg="lightblue")
    button_frame.pack(pady=10)

    tk.Button(button_frame, text="Add", command=self.add_data, width=10, height=3).grid(row=0, column=0, padx=20)
    tk.Button(button_frame, text="Update", command=self.update_data, width=10, height=3).grid(row=0, column=1, padx=20)
    tk.Button(button_frame, text="Delete", command=self.delete_data, width=10, height=3).grid(row=0, column=2, padx=20)
    tk.Button(button_frame, text="Load Data", command=self.load_data, width=10, height=3).grid(row=0, column=3, padx=20)

```

```

def get_value(self, event):
    try:
        # Lấy id của dòng được chọn trong Treeview
        selected_row = self.listBox.selection()[0]
        # Lấy giá trị của dòng được chọn
        selected_values = self.listBox.item(selected_row, 'values')
        # Đổ dữ liệu vào các Entry
        self.column1.set(selected_values[1]) # MSSV
        self.column2.set(selected_values[2]) # Họ và tên
        self.column3.set(selected_values[3]) # Lớp
        self.column4.set(selected_values[4]) # Địa chỉ

    except IndexError:
        # Nếu không có dòng nào được chọn
        messagebox.showwarning("Warning", "No item selected")

def load_data(self):
    try:
        # Xóa tất cả các dòng hiện có trong Treeview
        for item in self.listBox.get_children():
            self.listBox.delete(item)
        # Truy vấn dữ liệu từ cơ sở dữ liệu
        query = sql.SQL("SELECT * FROM {}").format(sql.Identifier(self.table_name.get()))
        self.cur.execute(query)
        rows = self.cur.fetchall()
        # Thêm từng dòng dữ liệu vào Treeview
        for row in rows:
            self.listBox.insert("", "end", values=row)

    except Exception as e:
        messagebox.showerror("Error", f"Error loading data: {e}")

def add_data(self):
    try:
        query = sql.SQL("INSERT INTO {} (MSSV, HOTEN, LOP, DIACHI) VALUES (%s, %s, %s, %s)").format(sql.Identifier(self.table_name.get()))
        values = (self.column1.get(), self.column2.get(), self.column3.get(), self.column4.get())
        self.cur.execute(query, values)
        self.conn.commit()
        messagebox.showinfo("Success", "Data added successfully!")
        self.load_data() # Load lại dữ liệu sau khi thêm

    except Exception as e:
        messagebox.showerror("Error", f"Error adding data: {e}")

```

```

def update_data(self):
    try:
        selected_row = self.listBox.selection()[0]
        selected_values = self.listBox.item(selected_row, 'values')
        record_id = selected_values[0]
        query = sql.SQL("UPDATE {} SET MSSV=%s, HOTEN=%s, LOP=%s, DIACHI=%s WHERE id=%s").format(sql.Identifier(self.table_name.get()))
        values = (self.column1.get(), self.column2.get(), self.column3.get(), self.column4.get(), record_id)
        self.cur.execute(query, values)
        self.conn.commit()
        messagebox.showinfo("Success", "Data updated successfully!")
        self.load_data() # Load Lại dữ liệu sau khi cập nhật

    except Exception as e:
        messagebox.showerror("Error", f"Error updating data: {e}")

def delete_data(self):
    try:
        selected_row = self.listBox.selection()[0]
        selected_values = self.listBox.item(selected_row, 'values')
        record_id = selected_values[0]
        query = sql.SQL("DELETE FROM {} WHERE id=%s").format(sql.Identifier(self.table_name.get()))
        self.cur.execute(query, (record_id,))
        self.conn.commit()
        messagebox.showinfo("Success", "Data deleted successfully!")
        self.load_data() # Load Lại dữ liệu sau khi xóa

    except Exception as e:
        messagebox.showerror("Error", f"Error deleting data: {e}")

# Tạo cửa sổ ứng dụng
root = tk.Tk()
app = DatabaseApp(root)
root.mainloop()

```

#### 4. Github

[https://github.com/Tql211104/Python\\_NangCao](https://github.com/Tql211104/Python_NangCao)