TRƯỜNG ĐẠI HỌC VĂN LANG KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN 2

MÔN HỌC LẬP TRÌNH PYTHON NÂNG CAO

Chủ đề:

BÀI TẬP 2

SVTH: Nguyễn Thành Nhân

MSSV: 2274802010518

LỚP: 241_71ITSE31003_0201

GVHD: HUỲNH THÁI HỌC

TP. Hồ Chí Minh – 10/2024

Bai Tap 2

```
import tkinter as tk
from tkinter import messagebox
import psycopg2
from psycopg2 import sql

class DatabaseApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Database App")

        # Database connection fields
        self.db_name = tk.StringVar(value='dbtest')
        self.user = tk.StringVar(value='postgres')
        self.password = tk.StringVar(value='131206')
        self.host = tk.StringVar(value='localhost')
        self.port = tk.StringVar(value='5432')
        self.table_name = tk.StringVar(value='sinhvien')
        # Self connect
        self.connect_db()
        # Create the GUI elements
        self.create_widgets()

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

        tk.Label(connection_frame, text="DB Name:").grid(row=0, column=0, padx=5,
pady=5)
        tk.Entry(connection_frame, textvariable=self.db_name).grid(row=0,
column=1, padx=5, pady=5)

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

        tk.Label(connection_frame, text="Password:").grid(row=2, column=0,
padx=5, pady=5)
```

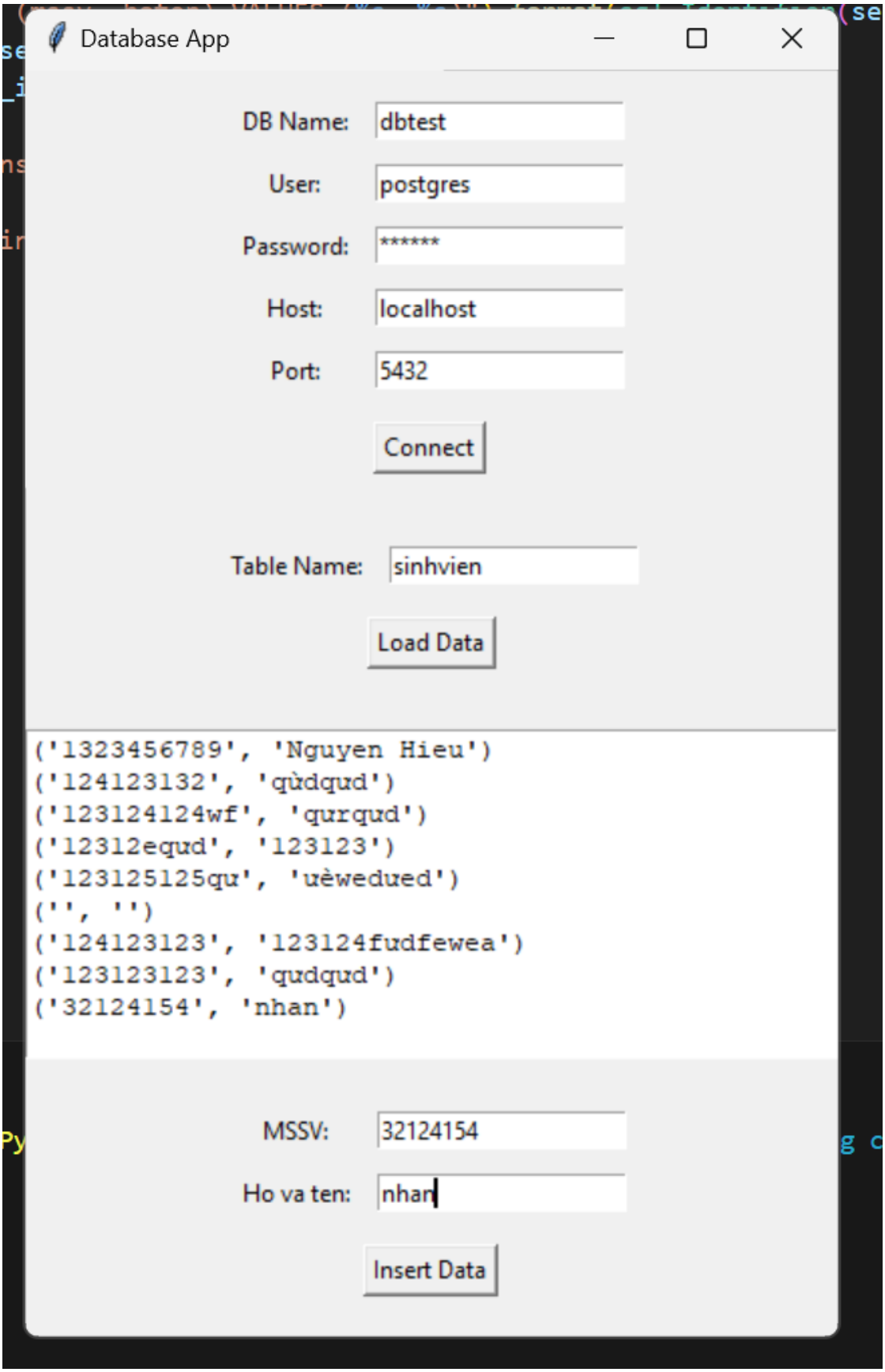
```
        tk.Entry(connection_frame, textvariable=self.password,
show="*").grid(row=2, column=1, padx=5, pady=5)

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

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

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

        # Query section
        query_frame = tk.Frame(self.root)
        query_frame.pack(pady=10)

        tk.Label(query_frame, text="Table Name:").grid(row=0, column=0, padx=5,
pady=5)
        tk.Entry(query_frame, textvariable=self.table_name).grid(row=0, column=1,
padx=5, pady=5)

        tk.Button(query_frame, text="Load Data",
command=self.load_data).grid(row=1, columnspan=2, pady=10)

        self.data_display = tk.Text(self.root, height=10, width=50)
        self.data_display.pack(pady=10)

        # Insert section
        insert_frame = tk.Frame(self.root)
        insert_frame.pack(pady=10)

        self.column1 = tk.StringVar()
        self.column2 = tk.StringVar()

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

        tk.Label(insert_frame, text="Ho va ten:").grid(row=1, column=0, padx=5,
pady=5)
```

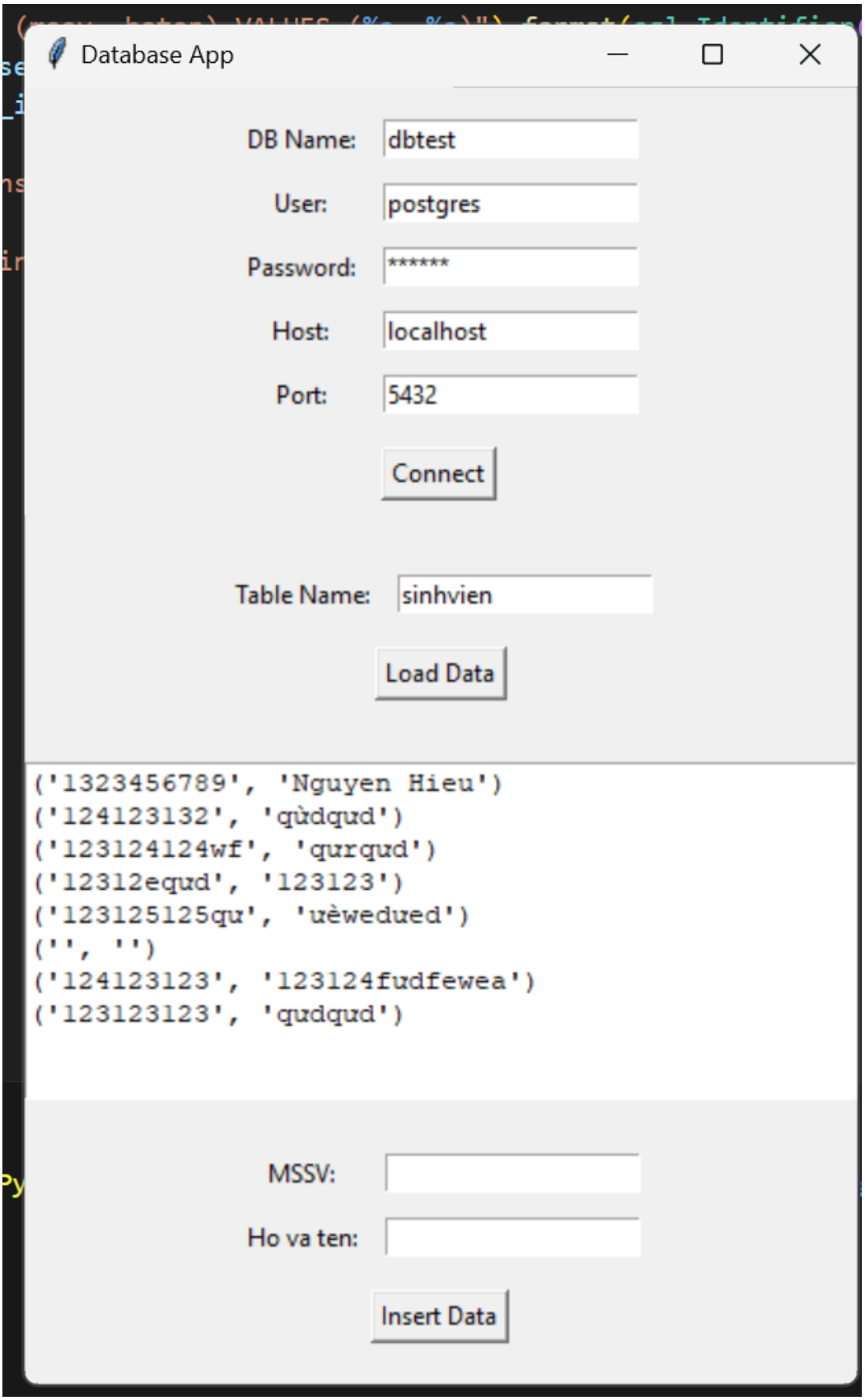
```
        tk.Entry(insert_frame, textvariable=self.column2).grid(row=1, column=1,
padx=5, pady=5)

        tk.Button(insert_frame, text="Insert Data",
command=self.insert_data).grid(row=2, 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!")
        except Exception as e:
            messagebox.showerror("Error", f"Error connecting to the database:
{e}")

    def load_data(self):
        try:
            query = sql.SQL("SELECT * FROM
{}").format(sql.Identifier(self.table_name.get()))
            self.cur.execute(query)
            rows = self.cur.fetchall()
            self.data_display.delete(1.0, tk.END)
            for row in rows:
                self.data_display.insert(tk.END, f"{row}\n")
        except Exception as e:
            messagebox.showerror("Error", f"Error loading data: {e}")

    def insert_data(self):
        try:
            insert_query = sql.SQL("INSERT INTO {} (mssv, hoten) VALUES (%s,
%s)").format(sql.Identifier(self.table_name.get()))
            data_to_insert = (self.column1.get(), self.column2.get())
            self.cur.execute(insert_query, data_to_insert)
            self.conn.commit()
            messagebox.showinfo("Success", "Data inserted successfully!")
        except Exception as e:
```

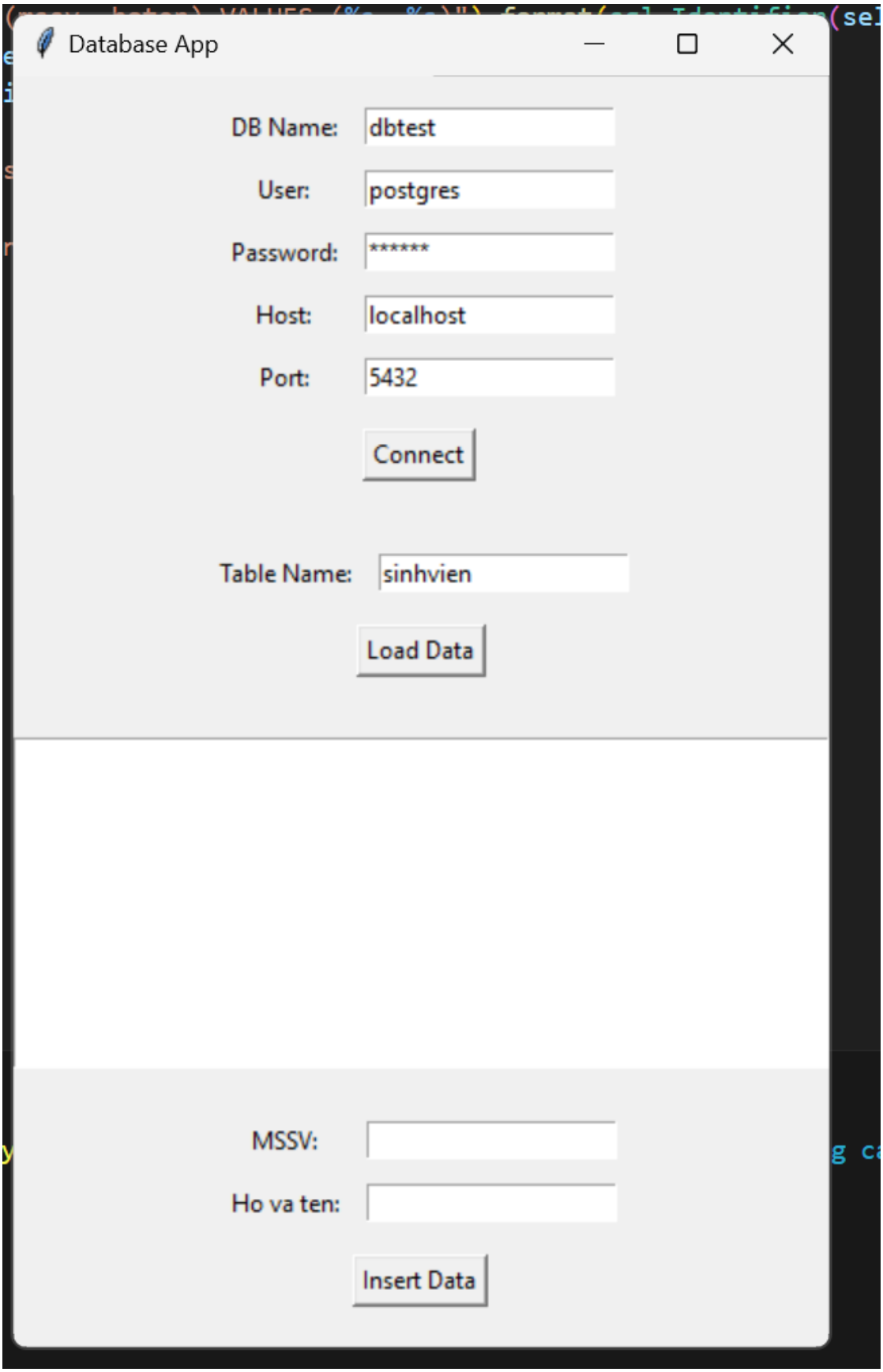
```
            messagebox.showerror("Error", f"Error inserting data: {e}")

if __name__ == "__main__":
    root = tk.Tk()
    app = DatabaseApp(root)
    root.mainloop()
```

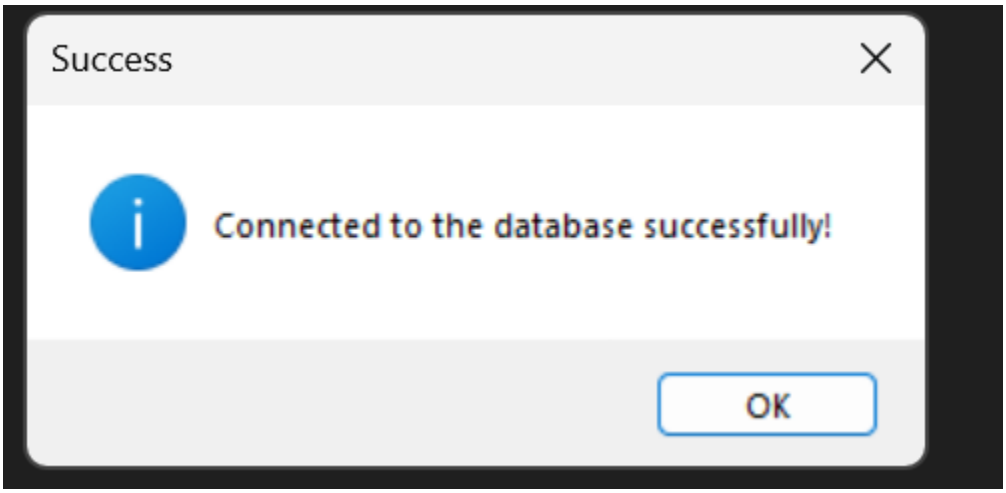
Data Output | Messages | Notifications

	mssv [PK] character varying (13)	hoten character varying (200)
1		
2	123123123	qưdqưd
3	123124124wf	qưrqưd
4	123125125qư	ưèwedưed
5	12312equd	123123
6	124123123	123124fưdfewea
7	124123132	qừdqưd
8	1323456789	Nguyen Hieu
9	32124154	nhan


Database App
DB Name:
dbtest
User:
postgres
Password:
******
Host:
localhost
Port:
5432
Connect
Table Name:
sinhvien
Load Data
('1323456789', 'Nguyen Hieu')
('124123132', 'qùdqud')
('123124124wf', 'qurqud')
('12312equd', '123123')
('123125125qu', 'uèwedued')
('', '')
('124123123', '123124fudfewea')
('123123123', 'qudqud')
('32124154', 'nhan')
MSSV:
32124154
Ho va ten:
nhan
Insert Data


Database App
DB Name:
dbtest
User:
postgres
Password:
******
Host:
localhost
Port:
5432
Connect
Table Name:
sinhvien
Load Data
('1323456789', 'Nguyen Hieu')
('124123132', 'qùdqud')
('123124124wf', 'qurqud')
('12312equd', '123123')
('123125125qu', 'uèwedued')
('', '')
('124123123', '123124fudfewea')
('123123123', 'qudqud')
MSSV:
Ho va ten:
Insert Data


Database App
DB Name:
dbtest
User:
postgres
Password:
******
Host:
localhost
Port:
5432
Connect
Table Name:
sinhvien
Load Data
MSSV:
Ho va ten:
Insert Data



Link Github: