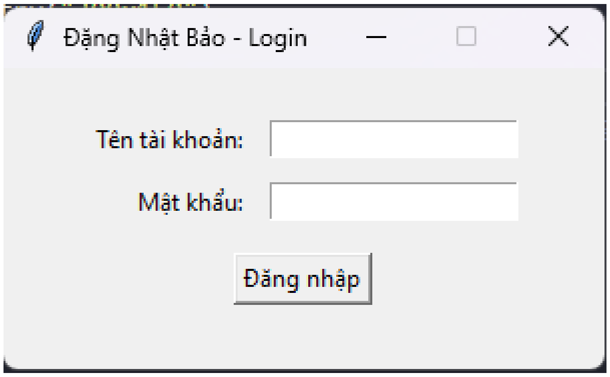
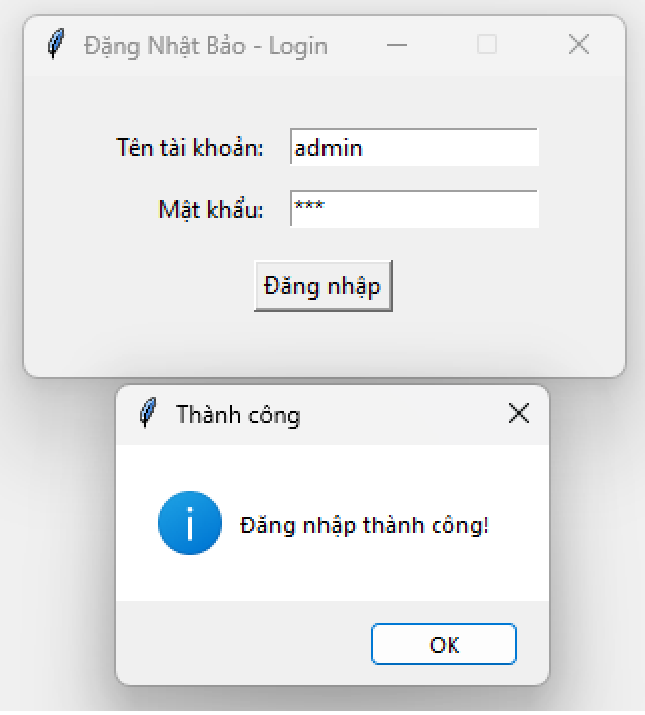
Cài thư viện và tạo form đăng nhập:

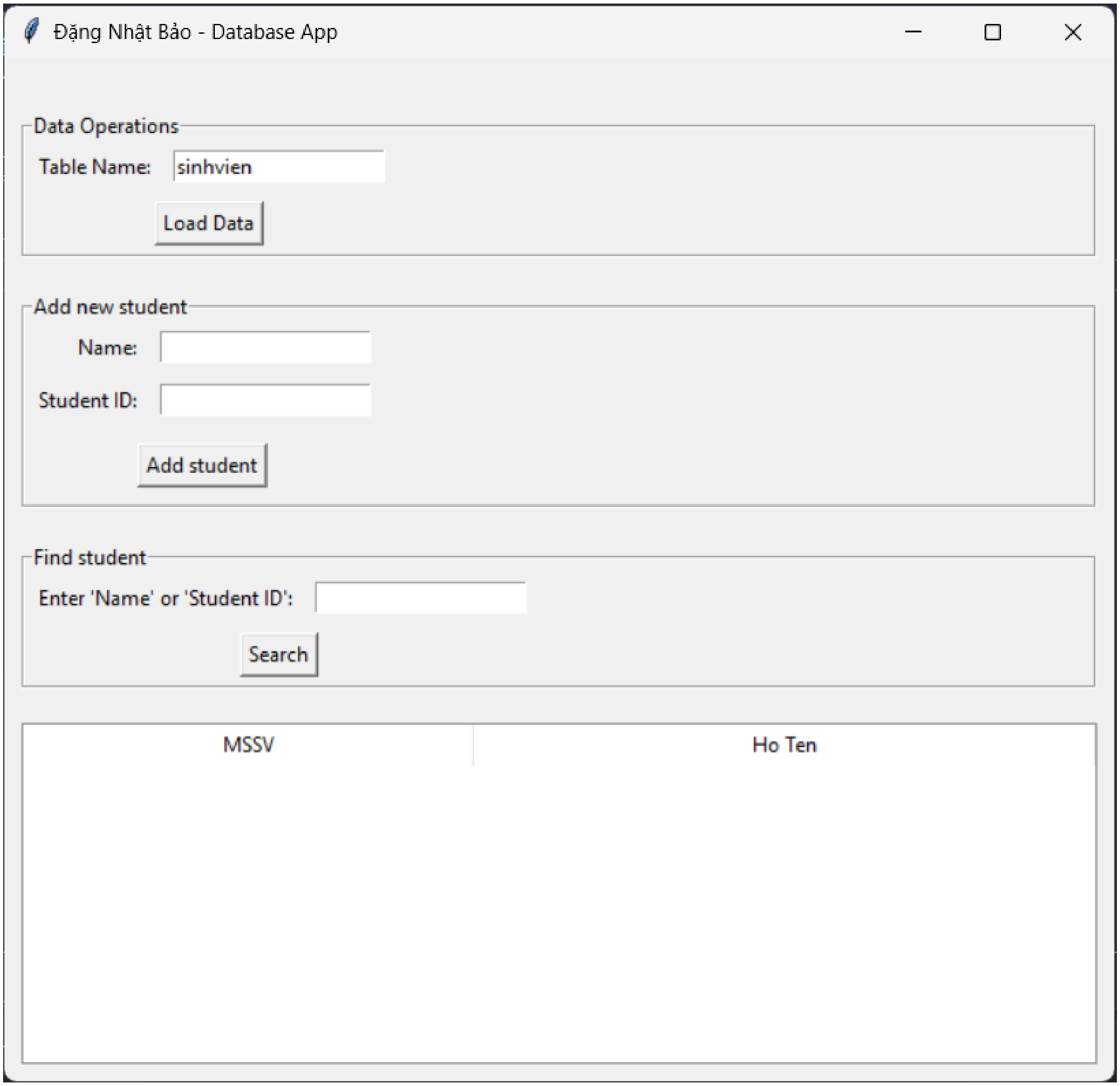




Tên tài khoản: admin

Mật khẩu: ***

Đăng nhập thành công!



Đặng Nhật Bảo - Database App

Data Operations

Table Name: sinhvien

Add new student

Name:

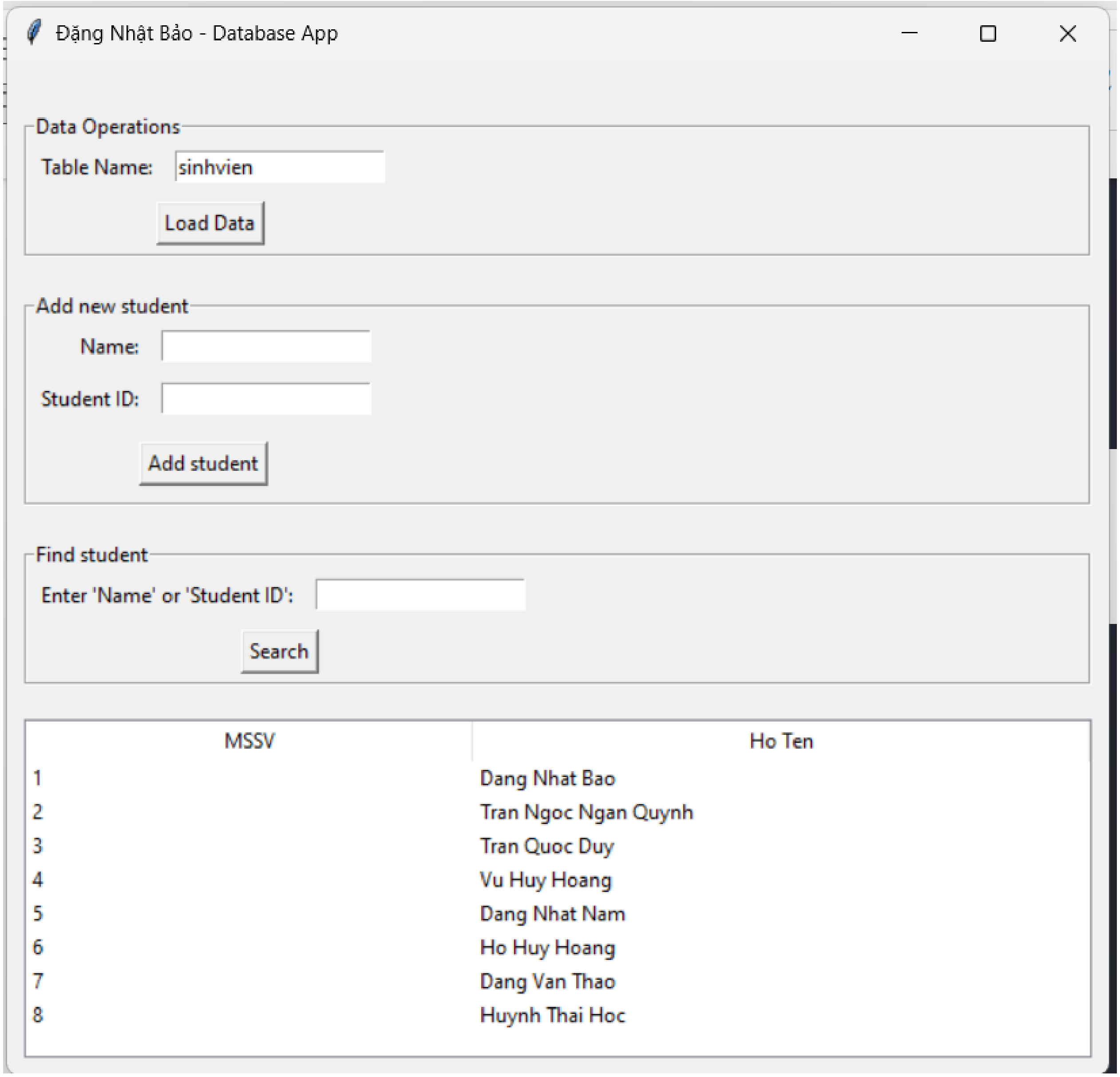
Student ID:

Find student

Enter 'Name' or 'Student ID':

MSSV	Ho Ten

Nhập tên bảng cần truy xuất và tải dữ liệu lên:



Đặng Nhật Bảo - Database App

Table Name: sinhvien

Name:

Student ID:

Enter 'Name' or 'Student ID':

MSSV	Ho Ten
1	Dang Nhat Bao
2	Tran Ngoc Ngan Quynh
3	Tran Quoc Duy
4	Vu Huy Hoang
5	Dang Nhat Nam
6	Ho Huy Hoang
7	Dang Van Thao
8	Huynh Thai Hoc

Thêm mới dữ liệu và tải lại dữ liệu:

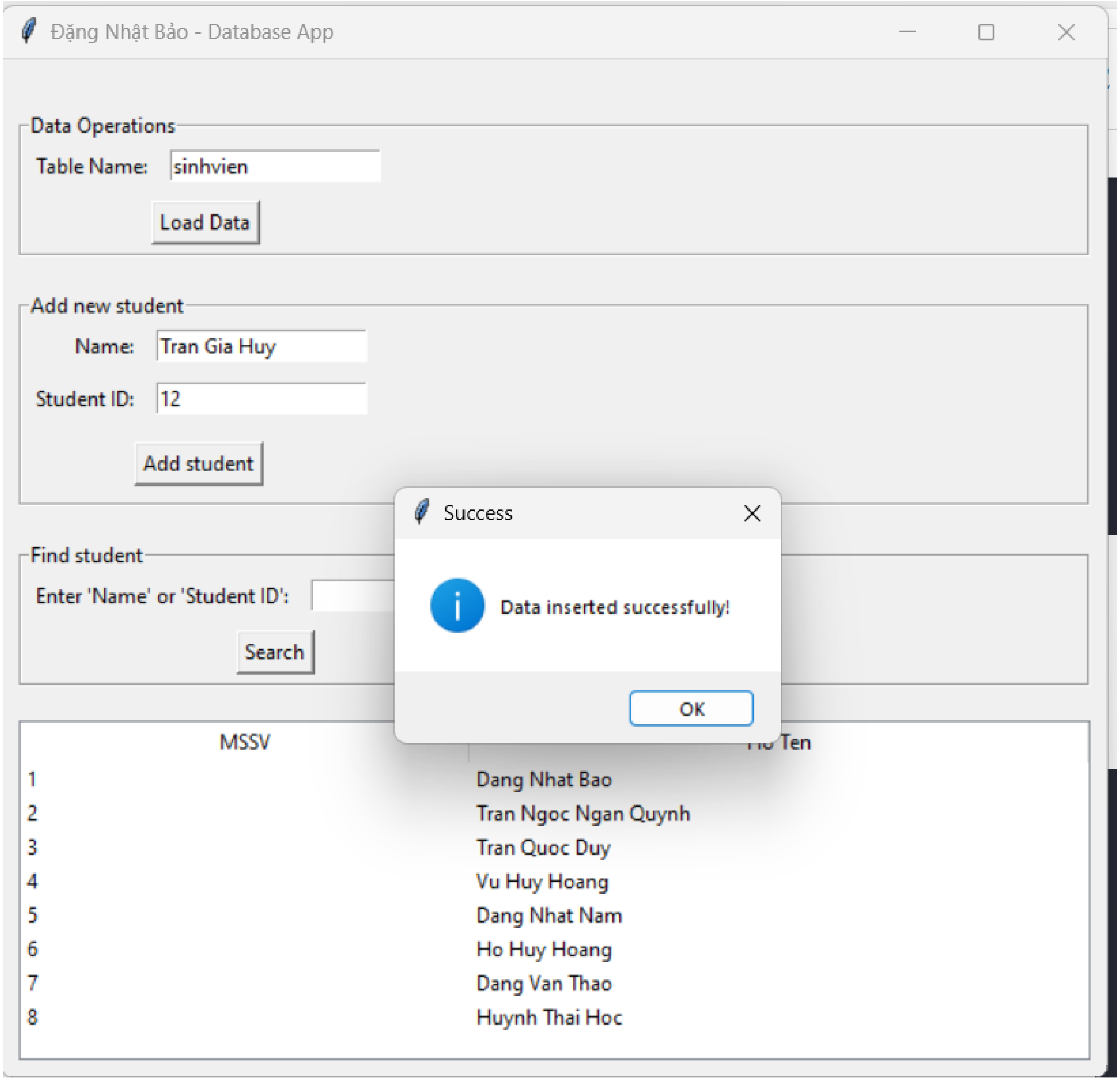


Table Name: sinhvien

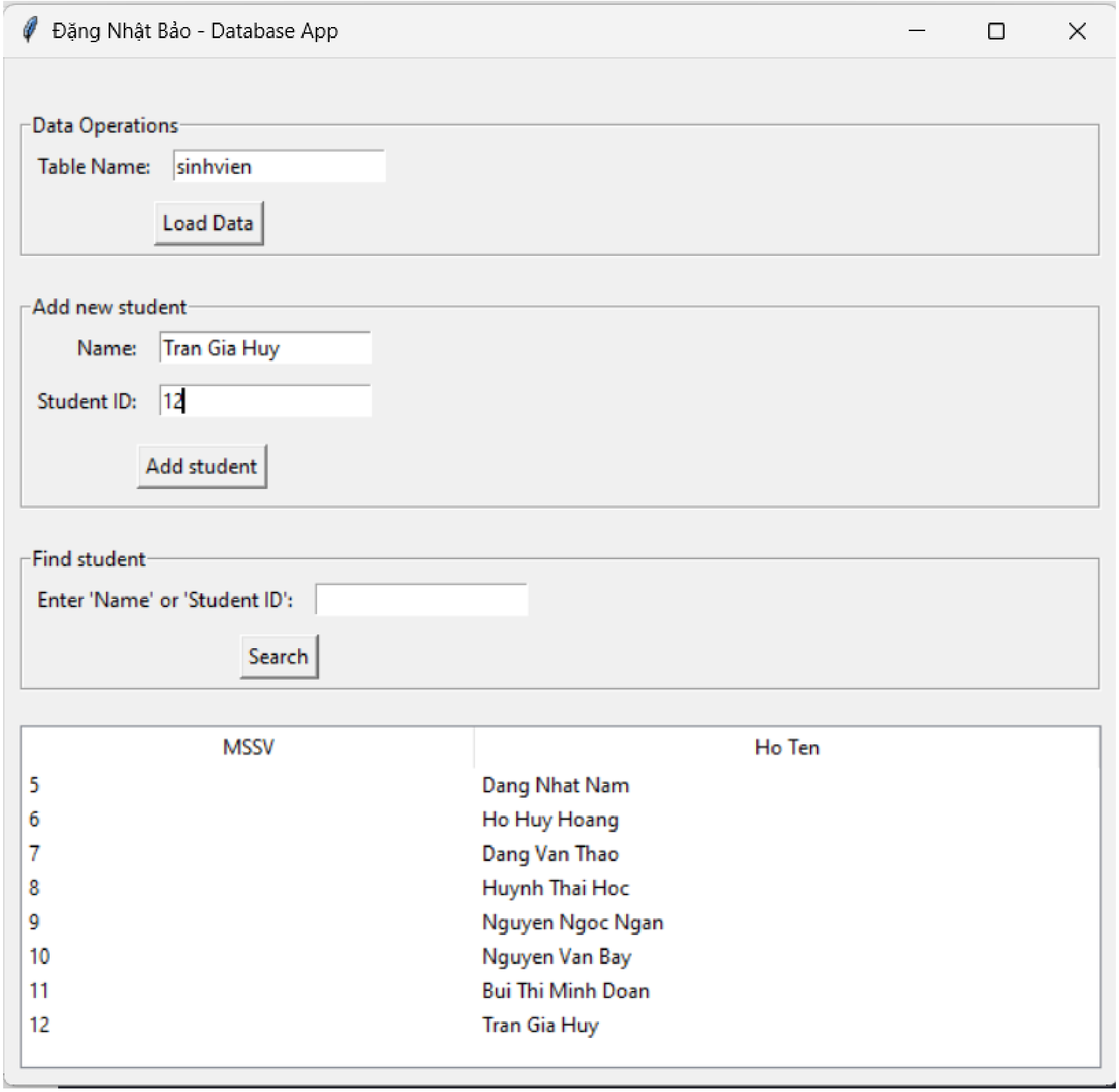
Name: Tran Gia Huy

Student ID: 12

Enter 'Name' or 'Student ID':

Data inserted successfully!

MSSV	Ho Ten
1	Dang Nhat Bao
2	Tran Ngoc Ngan Quynh
3	Tran Quoc Duy
4	Vu Huy Hoang
5	Dang Nhat Nam
6	Ho Huy Hoang
7	Dang Van Thao
8	Huynh Thai Hoc



Đặng Nhật Bảo - Database App

Table Name: sinhvien

Name: Tran Gia Huy

Student ID: 12

Enter 'Name' or 'Student ID':

MSSV	Ho Ten
5	Dang Nhat Nam
6	Ho Huy Hoang
7	Dang Van Thao
8	Huynh Thai Hoc
9	Nguyen Ngoc Ngan
10	Nguyen Van Bay
11	Bui Thi Minh Doan
12	Tran Gia Huy

Tìm kiếm dữ liệu:

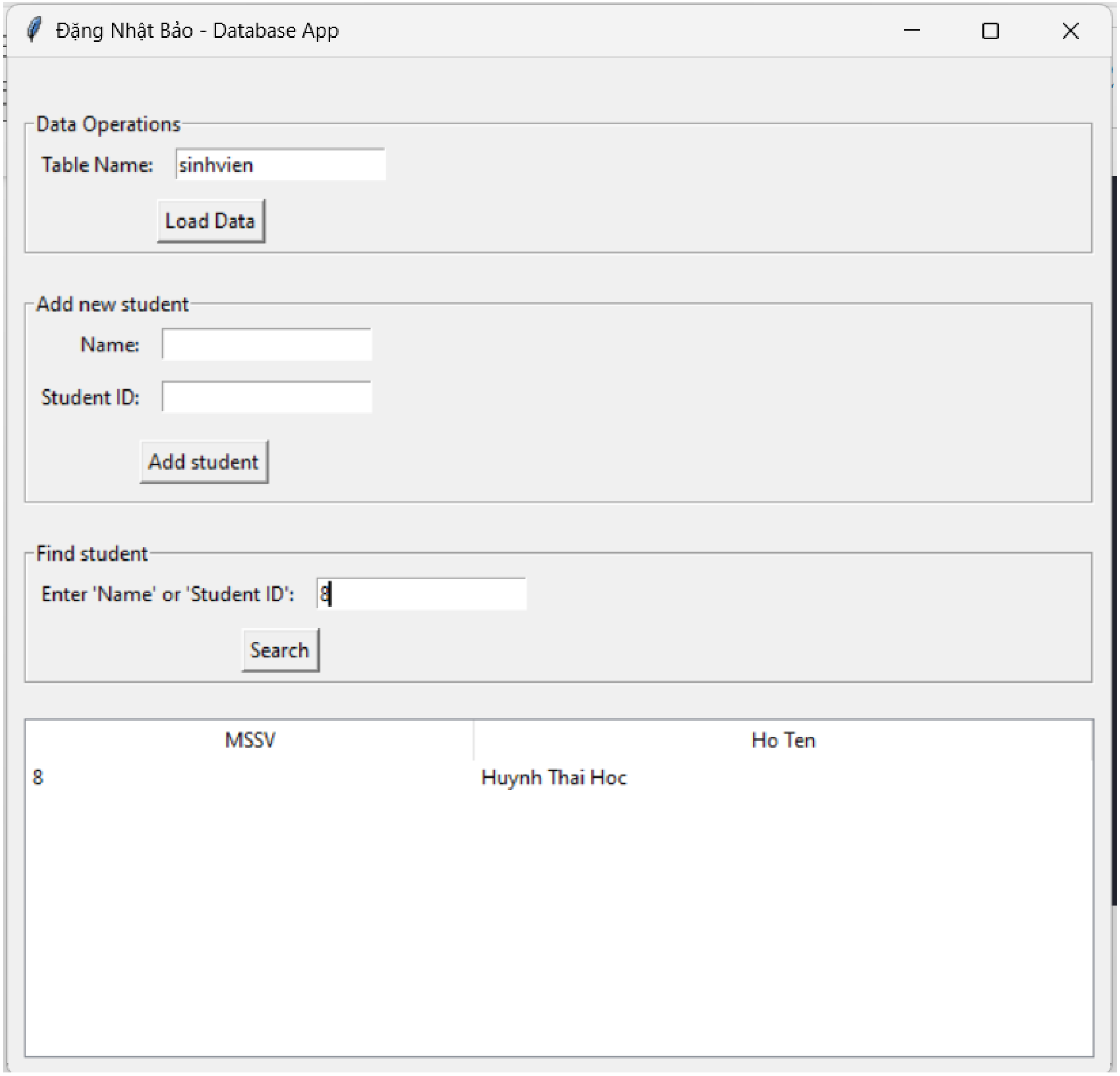


Table Name: sinhvien

Enter 'Name' or 'Student ID': 8

MSSV	Ho Ten
8	Huynh Thai Hoc

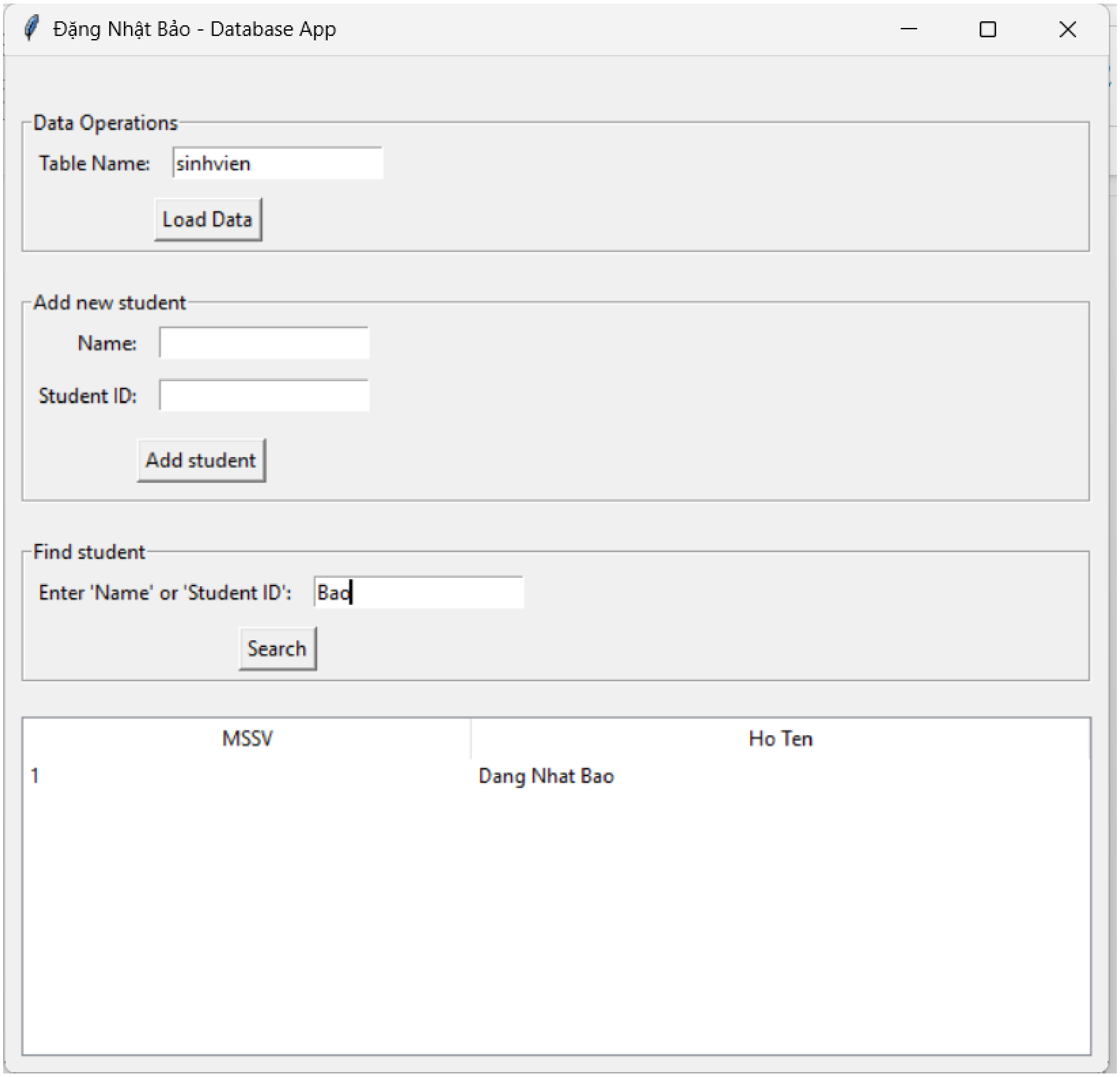


Table Name: sinhvien

Name:

Student ID:

Enter 'Name' or 'Student ID': Bao

MSSV	Ho Ten
1	Dang Nhat Bao

Code:

main.py

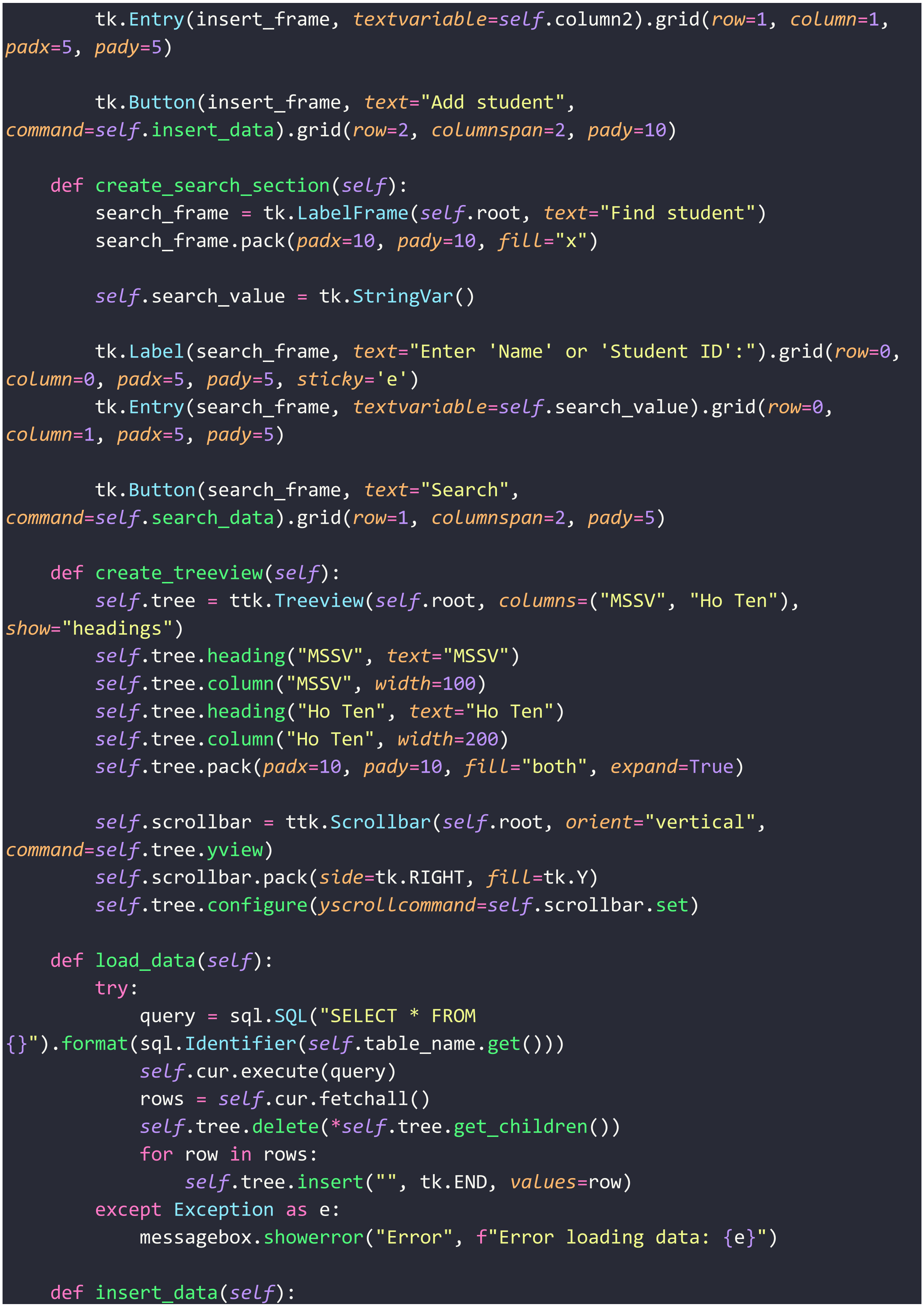
```python
import tkinter as tk
from login_window import LoginWindow

if __name__ == "__main__":
    root = tk.Tk()
    login = LoginWindow(root)
    root.mainloop()
```

login_window.py

```python
import tkinter as tk
from tkinter import messagebox
import psycopg2
from psycopg2 import sql
from main_app import open_main_app  # Import từ mô-đun khác

class LoginWindow:
    def __init__(self, root):
        self.root = root
        self.root.title("Đặng Nhật Bảo - Login")
        self.root.geometry("300x150")
        self.root.resizable(False, False)

        # Login credentials
        self.valid_username = "admin"
        self.valid_password = "123"

        # Database connection info
        self.db_name = 'dbtest'
        self.user = 'postgres'
        self.password = '123456'
        self.host = 'localhost'
        self.port = '5432'

        # Tạo các trường nhập liệu
        self.username = tk.StringVar()
        self.password_var = tk.StringVar()

        self.create_widgets()

    def create_widgets(self):
        frame = tk.Frame(self.root)
        frame.pack(pady=20)

        tk.Label(frame, text="Tên tài khoản:").grid(row=0, column=0, padx=5, pady=5, sticky='e')
        tk.Entry(frame, textvariable=self.username).grid(row=0, column=1, padx=5, pady=5)

        tk.Label(frame, text="Mật khẩu:").grid(row=1, column=0, padx=5, pady=5, sticky='e')
        tk.Entry(frame, textvariable=self.password_var, show="*").grid(row=1, column=1, padx=5, pady=5)
        tk.Button(frame, text="Đăng nhập", command=self.authenticate).grid(row=2,
columnspan=2, pady=10)

    def authenticate(self):
        user = self.username.get()
        pwd = self.password_var.get()

        if user == self.valid_username and pwd == self.valid_password:
            try:
                self.conn = self.connect_to_db()
                messagebox.showinfo("Thành công", "Đăng nhập thành công!")
                self.root.destroy()
                open_main_app(self.conn)
            except Exception as e:
                messagebox.showerror("Lỗi", f"Không thể kết nối cơ sở dữ liệu:
{e}")
        else:
            messagebox.showerror("Lỗi", "Thông tin đăng nhập không hợp lệ!")

    def connect_to_db(self):
        conn = psycopg2.connect(
            dbname=self.db_name,
            user=self.user,
            password=self.password,
            host=self.host,
            port=self.port
        )
        return conn

if __name__ == "__main__":
    root = tk.Tk()
    login = LoginWindow(root)
    root.mainloop()
```

main_app.py

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

class DatabaseApp:
    def __init__(self, root, conn):
        self.root = root
        self.conn = conn
        self.cur = self.conn.cursor()
        self.root.title("Đặng Nhật Bảo - Database App")
        self.root.geometry("650x600")

        self.create_widgets()

    def create_widgets(self):
        # Main frame
        main_frame = tk.Frame(self.root)
        main_frame.pack(padx=10, pady=10, fill="both", expand=True)

        # Query section
        query_frame = tk.LabelFrame(self.root, text="Data Operations")
        query_frame.pack(padx=10, pady=10, fill="x")

        self.table_name = tk.StringVar(value='sinhvien')

        tk.Label(query_frame, text="Table Name:").grid(row=0, column=0, padx=5,
pady=5, sticky='e')
        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=5)

        # Insert section
        self.create_insert_section()

        # Search section
        self.create_search_section()

        # Data Treeview
        self.create_treeview()

    def create_insert_section(self):
        insert_frame = tk.LabelFrame(self.root, text="Add new student")
        insert_frame.pack(padx=10, pady=10, fill="x")

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

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

        tk.Label(insert_frame, text="Student ID:").grid(row=1, column=0, padx=5,
pady=5, sticky='e')
```



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

        tk.Button(insert_frame, text="Add student", command=self.insert_data).grid(row=2, columnspan=2, pady=10)

    def create_search_section(self):
        search_frame = tk.LabelFrame(self.root, text="Find student")
        search_frame.pack(padx=10, pady=10, fill="x")

        self.search_value = tk.StringVar()

        tk.Label(search_frame, text="Enter 'Name' or 'Student ID':").grid(row=0, column=0, padx=5, pady=5, sticky='e')
        tk.Entry(search_frame, textvariable=self.search_value).grid(row=0, column=1, padx=5, pady=5)

        tk.Button(search_frame, text="Search", command=self.search_data).grid(row=1, columnspan=2, pady=5)

    def create_treeview(self):
        self.tree = ttk.Treeview(self.root, columns=("MSSV", "Ho Ten"), show="headings")
        self.tree.heading("MSSV", text="MSSV")
        self.tree.column("MSSV", width=100)
        self.tree.heading("Ho Ten", text="Ho Ten")
        self.tree.column("Ho Ten", width=200)
        self.tree.pack(padx=10, pady=10, fill="both", expand=True)

        self.scrollbar = ttk.Scrollbar(self.root, orient="vertical", command=self.tree.yview)
        self.scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
        self.tree.configure(yscrollcommand=self.scrollbar.set)

    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.tree.delete(*self.tree.get_children())
            for row in rows:
                self.tree.insert("", tk.END, values=row)
        except Exception as e:
            messagebox.showerror("Error", f"Error loading data: {e}")

    def insert_data(self):
        try:
            insert_query = sql.SQL("INSERT INTO {} (hoten, mssv) 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!")
            self.load_data()
        except Exception as e:
            self.conn.rollback()
            messagebox.showerror("Error", f"Error inserting data: {e}")

    def search_data(self):
        try:
            search_value = self.search_value.get()
            search_query = sql.SQL("SELECT * FROM {} WHERE hoten ILIKE %s OR CAST(mssv AS TEXT) ILIKE %s").format(sql.Identifier(self.table_name.get()))
            search_pattern = f"%{search_value}%"
            self.cur.execute(search_query, (search_pattern, search_pattern))
            rows = self.cur.fetchall()
            self.tree.delete(*self.tree.get_children())
            for row in rows:
                self.tree.insert("", tk.END, values=row)
        except Exception as e:
            self.conn.rollback()
            messagebox.showerror("Error", f"Error searching data: {e}")

def open_main_app(conn):
    root = tk.Tk()
    app = DatabaseApp(root, conn)
    root.mainloop()
```