**Giao diện:**

* Đăng nhập:

A screenshot of a computer

Description automatically generated

* Menu sau khi đăng nhập thành công:

A screenshot of a computer

Description automatically generated

* Giao diện tìm kiếm sản phẩm:

A screenshot of a computer

Description automatically generated

* Giao diện thêm sản phẩm:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Mã nguồn:**

import psycopg2

from psycopg2 import sql

import tkinter as tk

from tkinter import messagebox, ttk

# Hàm kết nối đến cơ sở dữ liệu

def connect\_to\_db(username, password):

    try:

        connection = psycopg2.connect(

            dbname="quanly\_quanao",

            user=username,

            password=password,

            host="localhost"

        )

        messagebox.showinfo("Thành công", "Đăng nhập thành công!")

        return connection

    except psycopg2.OperationalError as e:

        messagebox.showerror("Lỗi", f"Không thể kết nối đến cơ sở dữ liệu: {e}")

        return None

# Chức năng tìm kiếm sản phẩm

def search\_product(connection, product\_name):

    try:

        with connection.cursor() as cursor:

            query = "SELECT \* FROM products WHERE product\_name ILIKE %s"

            cursor.execute(query, ('%' + product\_name + '%',))

            return cursor.fetchall()

    except Exception as e:

        messagebox.showerror("Lỗi", f"Không thể tìm sản phẩm: {e}")

        return None

# Chức năng thêm sản phẩm mới

def add\_product(connection, product\_name, product\_price, category\_id):

    try:

        with connection.cursor() as cursor:

            query = """

                INSERT INTO products (product\_name, product\_price, category\_id)

                VALUES (%s, %s, %s)

            """

            cursor.execute(query, (product\_name, product\_price, category\_id))

            connection.commit()

            messagebox.showinfo("Thành công", "Thêm sản phẩm thành công!")

    except Exception as e:

        messagebox.showerror("Lỗi", f"Không thể thêm sản phẩm: {e}")

# Hàm cài đặt kiểu dáng cho ứng dụng

def set\_style(widget):

    style = ttk.Style(widget)

    style.theme\_use("clam")

    style.configure(

        "TButton",

        font=("Arial", 12, "bold"),

        background="#42a5f5",

        foreground="#ffffff",

        padding=8,

    )

    style.map(

        "TButton",

        background=[("active", "#1565c0")],

    )

# Giao diện đăng nhập

def login\_form():

    window = tk.Tk()

    window.title("Đăng Nhập Hệ Thống")

    window.geometry("500x300")

    window.configure(bg="#e3f2fd")

    set\_style(window)

    tk.Label(

        window, text="ĐĂNG NHẬP HỆ THỐNG", font=("Arial", 16, "bold"), bg="#e3f2fd", fg="#1565c0"

    ).pack(pady=20)

    login\_frame = tk.Frame(window, bg="#e3f2fd")

    login\_frame.pack(pady=20)

    tk.Label(login\_frame, text="Tên đăng nhập:", font=("Arial", 12), bg="#e3f2fd", fg="#333333").grid(

        row=0, column=0, padx=10, pady=10

    )

    username\_entry = ttk.Entry(login\_frame, width=30)

    username\_entry.grid(row=0, column=1, padx=10, pady=10)

    tk.Label(login\_frame, text="Mật khẩu:", font=("Arial", 12), bg="#e3f2fd", fg="#333333").grid(

        row=1, column=0, padx=10, pady=10

    )

    password\_entry = ttk.Entry(login\_frame, width=30, show="\*")

    password\_entry.grid(row=1, column=1, padx=10, pady=10)

    def login\_action():

        username = username\_entry.get()

        password = password\_entry.get()

        if username.strip() == "" or password.strip() == "":

            messagebox.showwarning("Cảnh báo", "Vui lòng nhập đủ thông tin!")

        else:

            connection = connect\_to\_db(username, password)

            if connection:

                window.destroy()

                show\_menu(connection)

    ttk.Button(window, text="Đăng Nhập", command=login\_action).pack(pady=15)

    window.mainloop()

# Giao diện menu sau khi đăng nhập

def show\_menu(connection):

    menu\_window = tk.Tk()

    menu\_window.title("Menu Quản Lý")

    menu\_window.geometry("400x300")

    menu\_window.configure(bg="#e3f2fd")

    set\_style(menu\_window)

    tk.Label(

        menu\_window, text="CHỨC NĂNG QUẢN LÝ", font=("Arial", 16, "bold"), bg="#e3f2fd", fg="#1565c0"

    ).pack(pady=20)

    ttk.Button(menu\_window, text="Tìm kiếm sản phẩm", command=lambda: search\_product\_form(connection)).pack(

        pady=10

    )

    ttk.Button(menu\_window, text="Thêm sản phẩm", command=lambda: add\_product\_form(connection)).pack(pady=10)

    menu\_window.mainloop()

# Giao diện tìm kiếm sản phẩm

def search\_product\_form(connection):

    search\_window = tk.Tk()

    search\_window.title("Tìm kiếm sản phẩm")

    search\_window.geometry("500x400")

    search\_window.configure(bg="#e3f2fd")

    set\_style(search\_window)

    tk.Label(

        search\_window, text="Tìm Kiếm Sản Phẩm", font=("Arial", 16, "bold"), bg="#e3f2fd", fg="#1565c0"

    ).pack(pady=20)

    search\_frame = tk.Frame(search\_window, bg="#e3f2fd")

    search\_frame.pack(pady=10)

    tk.Label(search\_frame, text="Nhập tên sản phẩm:", font=("Arial", 12), bg="#e3f2fd", fg="#333333").grid(

        row=0, column=0, padx=10, pady=10

    )

    search\_entry = ttk.Entry(search\_frame, width=30)

    search\_entry.grid(row=0, column=1, padx=10, pady=10)

    result\_text = tk.Text(search\_window, width=60, height=10, wrap="word")

    result\_text.pack(pady=10)

    def search\_action():

        product\_name = search\_entry.get()

        result = search\_product(connection, product\_name)

        result\_text.delete(1.0, tk.END)

        if result:

            for row in result:

                result\_text.insert(tk.END, f"ID: {row[0]}, Tên: {row[1]}, Giá: {row[2]}, Loại: {row[3]}\n")

        else:

            result\_text.insert(tk.END, "Không tìm thấy sản phẩm nào.\n")

    ttk.Button(search\_window, text="Tìm kiếm", command=search\_action).pack(pady=10)

    search\_window.mainloop()

# Giao diện thêm sản phẩm

def add\_product\_form(connection):

    add\_window = tk.Tk()

    add\_window.title("Thêm sản phẩm mới")

    add\_window.geometry("500x400")

    add\_window.configure(bg="#e3f2fd")

    set\_style(add\_window)

    tk.Label(

        add\_window, text="Thêm Sản Phẩm Mới", font=("Arial", 16, "bold"), bg="#e3f2fd", fg="#1565c0"

    ).pack(pady=20)

    add\_frame = tk.Frame(add\_window, bg="#e3f2fd")

    add\_frame.pack(pady=10)

    tk.Label(add\_frame, text="Tên sản phẩm:", font=("Arial", 12), bg="#e3f2fd", fg="#333333").grid(

        row=0, column=0, padx=10, pady=10

    )

    product\_name\_entry = ttk.Entry(add\_frame, width=30)

    product\_name\_entry.grid(row=0, column=1, padx=10, pady=10)

    tk.Label(add\_frame, text="Giá sản phẩm:", font=("Arial", 12), bg="#e3f2fd", fg="#333333").grid(

        row=1, column=0, padx=10, pady=10

    )

    product\_price\_entry = ttk.Entry(add\_frame, width=30)

    product\_price\_entry.grid(row=1, column=1, padx=10, pady=10)

    tk.Label(add\_frame, text="Mã loại sản phẩm:", font=("Arial", 12), bg="#e3f2fd", fg="#333333").grid(

        row=2, column=0, padx=10, pady=10

    )

    category\_id\_entry = ttk.Entry(add\_frame, width=30)

    category\_id\_entry.grid(row=2, column=1, padx=10, pady=10)

    def add\_action():

        product\_name = product\_name\_entry.get()

        product\_price = product\_price\_entry.get()

        category\_id = category\_id\_entry.get()

        if product\_name and product\_price and category\_id:

            add\_product(connection, product\_name, float(product\_price), int(category\_id))

    ttk.Button(add\_window, text="Thêm sản phẩm", command=add\_action).pack(pady=20)

    add\_window.mainloop()

# Khởi chạy ứng dụng

login\_form()