

BÀI BÁO CÁO

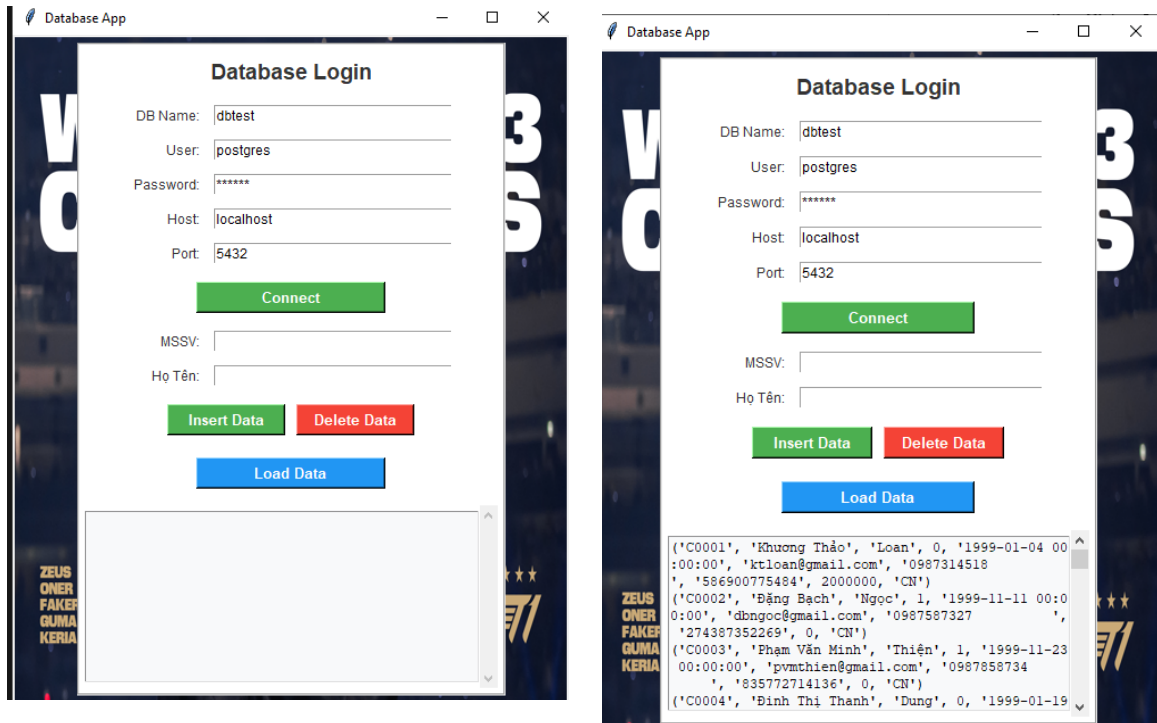
TÊN BÀI: KẾT NỐI GUI VỚI POSTGREE

MSSV: 2274802010083

HỌ TÊN: ĐOÀN TRUNG CƯỜNG

Nội dung:

1. **Giao diện:** Là 1 giao diện lọc các dữ liệu data trong PostgreSQL
2. **Chức năng:**



Kết nối cơ sở dữ liệu:

- Cho phép người dùng kết nối với cơ sở dữ liệu PostgreSQL bằng thông tin nhập vào (DB Name, User, Password, Host, Port).

Tải dữ liệu (Load Data):

- Hiển thị toàn bộ dữ liệu từ bảng được chỉ định trong cơ sở dữ liệu và hiển thị trong khung văn bản có thanh cuộn.

Chèn dữ liệu (Insert Data):

- Cho phép người dùng thêm dữ liệu mới (MSSV và Họ Tên) vào bảng cơ sở dữ liệu đã chọn.

Xóa dữ liệu (Delete Data):

- Xóa một bản ghi khỏi bảng dựa trên MSSV mà người dùng nhập vào.

Hiển thị dữ liệu với thanh cuộn:

- Dữ liệu được hiển thị trong khung văn bản có thể cuộn nếu lượng dữ liệu nhiều hơn kích thước khung.

3. Mã chương trình:

Import thư viện

- tkinter: Thư viện giúp tạo giao diện đồ họa (GUI) trong Python.
- messagebox: Một tiện ích của tkinter để hiển thị các hộp thoại thông báo.

- `psycopg2`: Thư viện giúp kết nối và thực hiện các thao tác với cơ sở dữ liệu PostgreSQL.
- `PIL.Image` và `PIL.ImageTk`: Thư viện từ PIL (Python Imaging Library) giúp xử lý và hiển thị hình ảnh trên giao diện.

Lớp DatabaseApp: Đây là phần chính của chương trình, quản lý toàn bộ giao diện người dùng và xử lý các chức năng liên quan đến cơ sở dữ liệu.

Phương thức `__init__`: Đây là phương thức khởi tạo của lớp, thực hiện các nhiệm vụ sau:

- **Khởi tạo cửa sổ GUI:** Đặt tiêu đề và kích thước cho cửa sổ ứng dụng.
- **Tải ảnh nền:** Cố gắng tải và đặt hình ảnh làm nền giao diện. Nếu không tải được ảnh, sử dụng màu nền mặc định.
- **Tạo khung giao diện chính:** Tạo `main_frame` để chứa tất cả các phần tử (widgets) khác như nút, nhãn, và ô nhập liệu.
- **Khai báo biến dữ liệu:** Tạo các biến lưu trữ thông tin cơ sở dữ liệu (DB Name, User, Password, Host, Port, Table Name) và các biến dữ liệu đầu vào (MSSV, Họ Tên).
- **Gọi hàm `create_widgets`:** Hàm này sẽ tạo ra các thành phần giao diện (widgets) cho ứng dụng.

Phương thức `create_widgets`: Phương thức này xây dựng giao diện người dùng với các thành phần chính sau:

- **Tiêu đề (`title_label`):** Hiển thị tiêu đề "Database Login" phía trên giao diện.
- **Các trường nhập liệu (DB Name, User, Password, Host, Port):** Tạo các trường nhập liệu cho thông tin kết nối cơ sở dữ liệu.
- **Nút "Connect":** Nút này sẽ gọi hàm `connect_db` khi nhấn để kết nối với cơ sở dữ liệu.
- **Các trường nhập liệu MSSV và Họ Tên:** Cho phép người dùng nhập mã số sinh viên và họ tên.
- **Nút "Insert Data" và "Delete Data":** Cho phép chèn và xóa dữ liệu từ cơ sở dữ liệu khi nhấn.
- **Nút "Load Data":** Nút này sẽ gọi hàm `load_data` để tải và hiển thị dữ liệu từ bảng trong cơ sở dữ liệu.
- **Khung hiển thị dữ liệu (Text box):** Khung hiển thị dữ liệu với thanh cuộn, hiển thị dữ liệu đã tải từ bảng.

Phương thức `connect_db`:

- Kết nối với cơ sở dữ liệu PostgreSQL bằng cách sử dụng các thông tin mà người dùng nhập vào (DB Name, User, Password, Host, Port).
- Nếu kết nối thành công, một thông báo thành công sẽ hiện ra, nếu không thì báo lỗi.

Phương thức `load_data`:

- Thực hiện câu lệnh SQL để truy xuất tất cả các bản ghi từ bảng đã chỉ định.
- Hiển thị dữ liệu đó trong khung văn bản (textbox), làm mới nội dung hiện tại.

Phương thức `insert_data`:

- Chèn một bản ghi mới vào bảng trong cơ sở dữ liệu với thông tin về MSSV và Họ Tên mà người dùng đã nhập.
- Sau khi chèn thành công, bảng dữ liệu sẽ được làm mới để hiển thị bản ghi mới được thêm vào.

Phương thức `delete_data`:

- Xóa một bản ghi trong bảng dựa trên MSSV được nhập vào.
- Sau khi xóa thành công, dữ liệu sẽ được làm mới để phản ánh sự thay đổi trong bảng.

Phần main:

- Đây là nơi ứng dụng được khởi chạy. Khi bạn chạy chương trình, một đối tượng của lớp DatabaseApp sẽ được tạo ra và giao diện người dùng sẽ được hiển thị thông qua `root.mainloop()`.

4. Github:

Link GitHub: https://github.com/Cuong2k42004/BaiCuoiKy_PythonNangCao/tree/main