**# Báo cáo Bài tập lớn: Xây dựng ứng dụng phân chia công việc**

**## I. Giới thiệu**

Bài tập lớn của chúng tôi là xây dựng một ứng dụng phân chia công việc. Ứng dụng này cho phép các thành viên trong nhóm nhận các công việc chưa được làm và cập nhật trạng thái công việc (đang làm, hoàn thành, không làm được/trả lại cho người khác làm) về hệ thống.

**## II. Công nghệ sử dụng**

- Ngôn ngữ lập trình: Java

//Giới thiệu về java

+Giới thiệu về java swing

- Mô hình: Client-Server

//Giới thiệu về mô hình này

- Công nghệ: Socket Server trong Java

//giới thiệu về cái này

- Cơ sở dữ liệu: MySQL

//Giới thiệu về cái này

- Giao thức truyền tin: TCP

//giới thiệu về cái này

**## III. Tổng quan về ứng dụng**

Ứng dụng được xây dựng dựa trên mô hình client-server, sử dụng socket trong Java để tạo kết nối giữa client và server. Dữ liệu được lưu trữ trong cơ sở dữ liệu MySQL. Ứng dụng cũng hỗ trợ quản lý đa luồng để tăng hiệu suất và đảm bảo ứng dụng hoạt động mượt mà. Giao thức truyền tin TCP được sử dụng để đảm bảo tính tin cậy và an toàn trong quá trình truyền dữ liệu.

**## IV. Cách thức hoạt động của ứng dụng**

(Ở đây, bạn có thể mô tả chi tiết về cách thức hoạt động của ứng dụng, từ việc khởi tạo kết nối giữa client và server, cách thức client nhận và cập nhật công việc, đến việc dữ liệu được lưu trữ và quản lý trong MySQL.)

**## V. Kết luận**

(Ở đây, bạn có thể tổng kết lại quá trình thực hiện bài tập lớn, những khó khăn và thách thức mà bạn đã gặp phải, cũng như những kinh nghiệm và kiến thức mà bạn đã học được.)

**## VI. Tài liệu tham khảo**

(Ở đây, bạn có thể liệt kê các nguồn tài liệu mà bạn đã tham khảo để thực hiện bài tập lớn.)

**Báo cáo bài tập lớn: Ứng dụng phân chia công việc**

**1. Giới thiệu:**

* Đề tài: Xây dựng ứng dụng phân chia công việc.
* Mục tiêu: Phát triển một ứng dụng giúp nhóm phân chia và quản lý công việc hiệu quả.
* Mô tả: Ứng dụng gồm hai phần: máy chủ (server) và client.
  + Máy chủ lưu trữ danh sách công việc, thông tin thành viên và quản lý kết nối với client.
  + Client cho phép thành viên nhận công việc, cập nhật trạng thái và giao tiếp với server.
* Công nghệ: Java Swing, mô hình client-server, socket, MySQL,.

**2. Phân tích yêu cầu:**

* Hệ thống gồm hai thành phần chính: server và client.
* Server:
  + Lưu trữ danh sách công việc, thông tin thành viên.
  + Quản lý kết nối với client.
  + Cập nhật trạng thái công việc theo hành động của client.
* Client:
  + Hiển thị danh sách công việc cho thành viên.
  + Cho phép thành viên đăng nhập để nhận công việc
  + Cho phép thành viên chọn và nhận công việc.
  + Cập nhật trạng thái công việc (đang làm, hoàn thành, trả lại).
  + Giao tiếp với server để gửi/nhận dữ liệu.

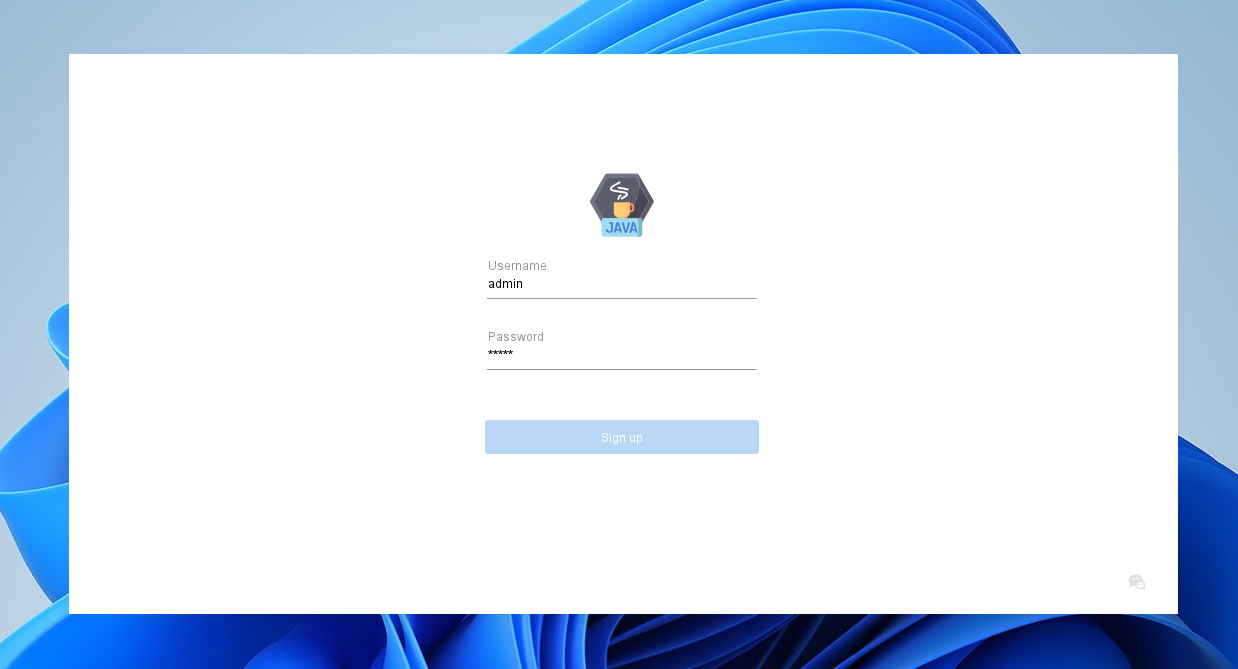
**3. Thiết kế hệ thống:**

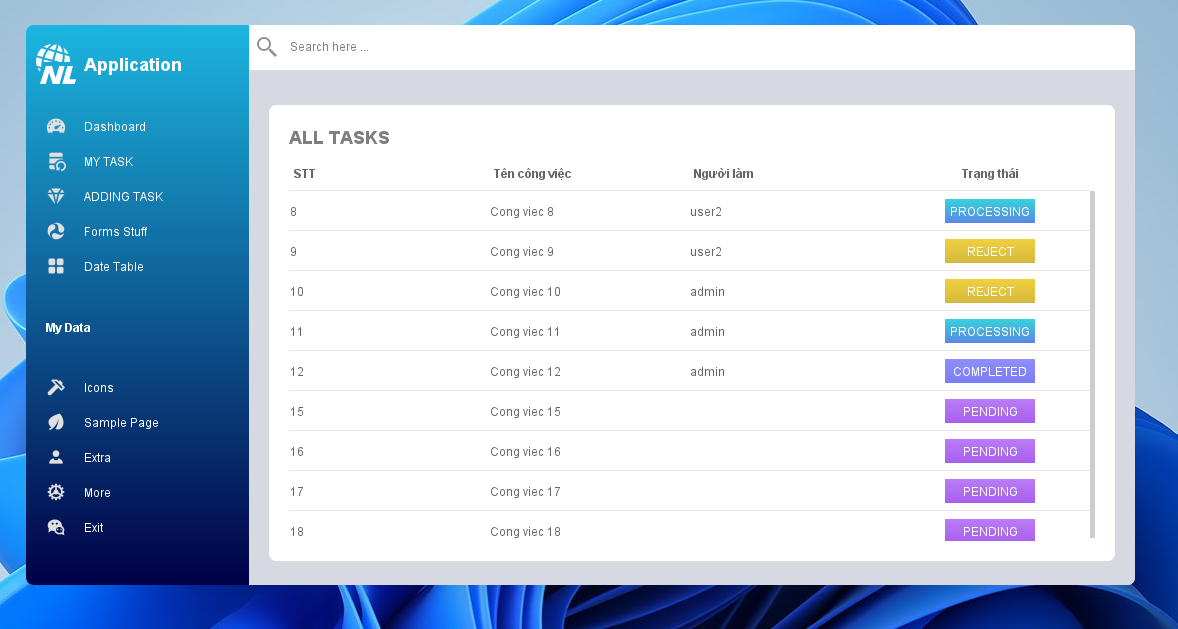
* Mô hình client-server:
  + Server: Chạy trên nền tảng máy chủ, chịu trách nhiệm quản lý dữ liệu và kết nối.
  + Client: Chạy trên máy tính cá nhân của thành viên, tương tác với server để thực hiện các thao tác.
* Kiến trúc phần mềm:
  + Server:
    - Gồm các lớp: Quản lý kết nối, Quản lý công việc, Quản lý thành viên.
    - Sử dụng socket server để lắng nghe và xử lý yêu cầu từ client.
    - Kết nối với database MySQL để lưu trữ dữ liệu.
  + Client:
    - Gồm các lớp: Giao diện người dùng, Giao tiếp mạng, Xử lý nghiệp vụ.
    - Giao diện người dùng: Hiển thị danh sách công việc, nút thao tác, thông tin trạng thái.
    - Giao tiếp mạng: Gửi/nhận dữ liệu từ server thông qua socket.
    - Xử lý nghiệp vụ: Thực hiện các chức năng như nhận công việc, cập nhật trạng thái.
* Cơ sở dữ liệu:
  + Sử dụng MySQL để lưu trữ thông tin công việc, thành viên và trạng thái công việc.
  + Thiết kế các bảng dữ liệu:
    - TaiKhoan(**tenDangNhap**,MatKhau)
    - CongViec(**id**,tenCongViec,nguoiThucHien,trangThai,version)
* Quản lý đa luồng:
  + Phía client
    - Sử dụng thư viện SwingWorker trong Java để xử lý các tác vụ tốn thời gian (như cập nhật database)
    - Đảm bảo giao diện người dùng mượt mà, không bị gián đoạn.
  + Phía server
    - Chia làm nhiều luồng. Để tránh xung đột khi thao tác với cơ sở dữ liệu thì ở đấy sẽ sử dụng một khái niệm là “optimistic lock” và kết hợp thêm Lock table trên Mysql.
* Cách thức hoạt động của “optimistic lock”
  + Khi một giao dịch muốn cập nhật dữ liệu, nó sẽ đọc giá trị phiên bản hiện tại của dữ liệu đó.
  + Giao dịch thực hiện các thay đổi cần thiết cho dữ liệu.
  + Khi giao dịch cố gắng ghi dữ liệu đã thay đổi, nó sẽ ghi lại cả giá trị phiên bản mới.
  + Hệ thống sẽ kiểm tra xem giá trị phiên bản mới có khớp với giá trị phiên bản hiện tại hay không.
  + Nếu khớp: Giao dịch thành công và dữ liệu được cập nhật.
  + Nếu không khớp: Giao dịch thất bại do dữ liệu đã bị thay đổi bởi giao dịch khác. Giao dịch có thể được thử lại hoặc thông báo lỗi cho người dùng.

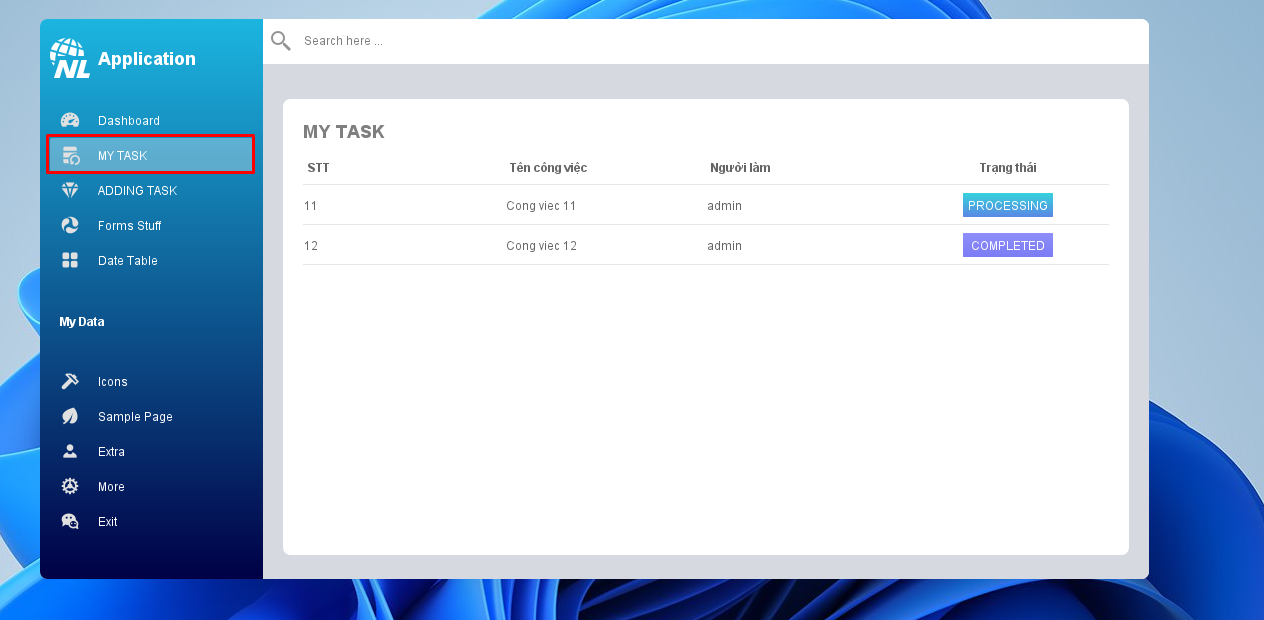
**4. Triển khai hệ thống:**

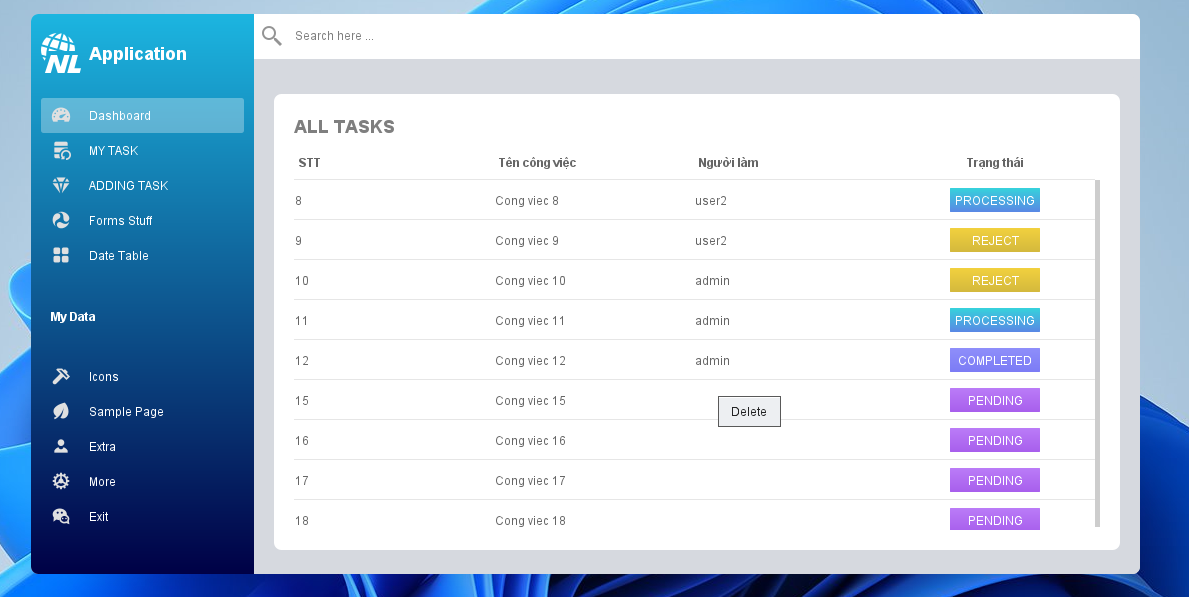
* Lựa chọn IDE: Sử dụng Netbean để phát triển ứng dụng Java (JDK 17).
* Thư viện kết nối cơ sở dữ liệu **JDBC (Java Database Connectivity)**
  + Cung cấp các lớp và hàm cho phép lập trình viên Java kết nối với cơ sở dữ liệu SQL.
  + Cho phép thực hiện các tác vụ như truy vấn, thêm, sửa, xóa dữ liệu trong cơ sở dữ liệu.
  + Không cung cấp cấu trúc ứng dụng hay quy trình phát triển cụ thể.
* Thư viện GSON dùng để chuyển đổi cấu trúc dữ liệu sang dạng chuỗi và ngược lại
* Code:
  + Server:
    - Class Server: Đây là class chính khi khi chương trình chạy. Class này sẽ lắng nghe tất cả các kết nối từ bên ngoài đến khi chương trình tắt. Mỗi khi có một client kết nối, class này sẽ chấp nhận kết nối và khởi tạo một luồng riêng biệt cho kết nối này. Lớp này có một thuộc tính static là một List chứa tất cả các client kết nối đến để kiểm soát kết nối.
    - Class DatabaseHelper: class này phục vụ cho việc kết nối và thao tác với cơ sở dữ liệu. Gồm các phương thức khởi tạo, kết nối, ngắt kết nối, thêm, sửa và xóa trên cơ sở dư liệu.
    - Class ClientHandler: Đây là class lắng nghe và xử lý các thông tin khi client gửi thông tin đến. Class này implements Runnable, mỗi một kết nốt đến sẽ khởi tạo class này và làm việc trên class này với một luồng riêng biệt.
    - Class TaiKhoan: Class này dùng cho việc chuyển đổi cấu trúc dữ liệu của tài khoản sang một chuôi JSON để gửi cho client và cũng như là khi nhận một chuỗi JSON từ client thì sẽ chuyển về thành đối tượng TaiKhoan để dễ dàng xử lý. Class này có phương thức kiểm tra xem tài khoản người dùng đăng nhập có tồn tại hay có chính xác thông tin trong cơ sở dữ liệu không.
    - Class CongViec: Tương đương với class TaiKhoan
    - Interface ClientHandlerDisconnected: Interface này sẽ có một phương thức ngắt kết nối. Khi phía client ngắt kết nối thì phương thức này sẽ xóa client đó khỏi List gồm các kết nối.
  + Client:
    - Thiết kế giao diện người dùng với Swing.
      * Package com.acer.component: Gói này chứa các thiết kế là Header và Menu.
      * Package com.acer.event: Chứa các interface để các class có thể đăng ký sự kiện.
        + EventMenuSelected: khi người dùng click vào menu thì sự kiện tương đương sẽ được kích hoạt bởi hàm selected do interface này cung cấp.
        + ServerMessageListener: Interface này cung cấp phương thức để đăng ký sự kiện lắng nghe với luồng lắng nghe.
      * Package com.acer.form: gồm các form chính người dùng sẽ thao tác.
      * Package com.acer.glasspanepopup: Đây là gói cung cấp các thư viện xử lý về popup.
      * Package com.acer.icon: đây là gói chứa các icon cho chương trình.
      * Package com.acer.model: gói này cung cấp các class để xử lý dữ liệu cho các class khác.
      * Package com.acer.socket: gói này có class ClientSocket, dùng để khởi tạo và kết nối với socket server
      * Package com.acer.swing: gói này cung cấp các thiết kế và logic để xây dựng giao diện người dùng.
    - các lớp giao tiếp mạng và xử lý nghiệp vụ.
      * Lớp ClientSocket: lớp này chịu trách nhiệm kết nối với server, khởi tạo luồng vào và luồng ra để gửi dữ liệu qua lại.
      * Lớp ServerListener nghe lắng nghe thông tin từ server gửi đến một cách liên tục không ngừng. Đối tượng của lớp này sẽ chạy trên một luồng riêng biệt. Các lớp nào cần nhận thông tin từ client sẽ đăng ký sự kiện nhận thông tin từ lớp này. Mỗi khi có thông tin đến thì lớp này sẽ cung cấp thông tin đó cho các lớp đã đăng ký nhận thông tin với nó.
* Cấu hình database:
  + Cơ sở dữ liệu gồm có 2 bảng là bảng tài khoản và bảng Công việc
    - TaiKhoan(**tenDangNhap**,MatKhau)
    - CongViec(**id**,tenCongViec,nguoiThucHien,trangThai,version)

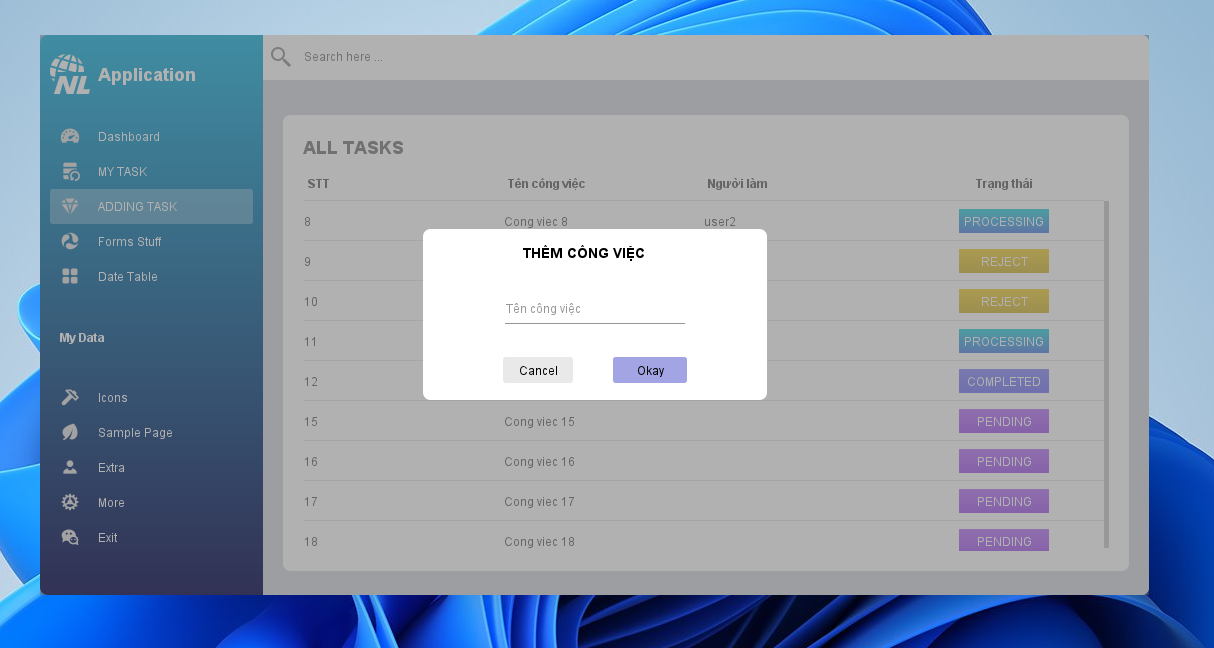
**5. Chương trình demo:**

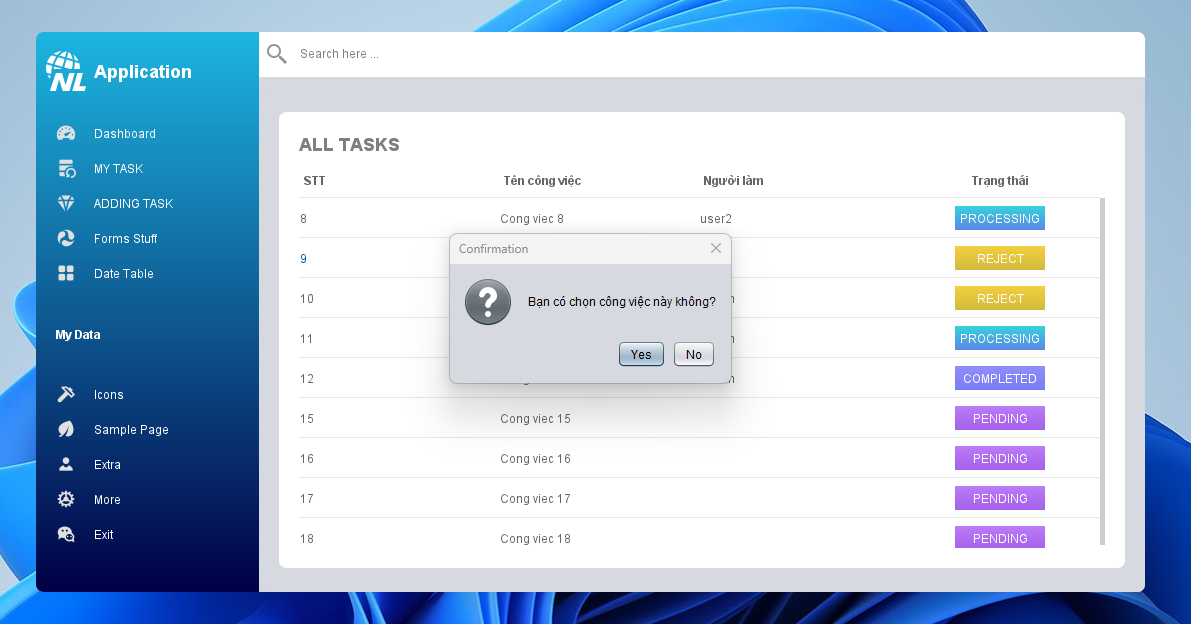






****





**6. Kết luận:**

(Ở đây, bạn có thể tổng kết lại quá trình thực hiện bài tập lớn, những khó khăn và thách thức mà bạn đã gặp phải, cũng như những kinh nghiệm và kiến thức mà bạn đã học được.)

**7. Tài liệu tham khảo:**

(Ở đây, bạn có thể liệt kê các nguồn tài liệu mà bạn đã tham khảo để thực hiện bài tập lớn.)