BÁO CÁO ASSIGNMENT PRJ302

**LIBRARY MANAGEMENT SYSTEM**

Tên Sinh Viên: Hoàng Nhật Phương Linh

MSSV: HE181087

Giảng Viên: Nguyễn Thanh Tú

MỤC LỤC

1. [**Chức năng của Web application 3**](#_d4y4obblsc4v)
2. [**Luồng đi cho từng chức năng chính trong Web application: 5**](#_bkp3kt5s8b7a)

[- Trang login tài khoản: 5](#_anre9qmz2e6p)

[● Admin ⇒ adminPage.jsp 6](#_51kd52e6dsx9)

[● Staff ⇒ staffPage.jsp 25](#_ki8fbct2w80f)

[● Reader ⇒ readerPage.jsp 26](#_lyf71ri29okw)

1. [**Thiết kế Database 28**](#_livlb6epvvpr)

# 

# I. Chức năng của Web application

* **Views**: chứa các tệp JSP, xây dựng giao diện cho từng chức năng trong ứng dụng:

1. **adminPage.jsp, readerPage.jsp, staffPage.jsp** - Các trang tổng quan dành riêng cho quản trị viên, độc giả, và nhân viên.
2. **borrowRecord.js**p - Hiển thị hoặc quản lý các bản ghi mượn sách chi tiết.
3. **editBook.jsp, editReader.jsp, editStaff.jsp** - Giao diện chỉnh sửa thông tin sách, độc giả và nhân viên.
4. **insertBook.jsp, insertReader.jsp, insertStaff.jsp** - Giao diện thêm mới cho sách, độc giả và nhân viên.
5. **listBook.jsp, listReader.jsp, listStaff.jsp** - Danh sách hiển thị trực quan cho sách, độc giả và nhân viên trong thư viện.
6. **Controllers:** bao gồm các Servlet đảm nhận vai trò xử lý yêu cầu từ người dùng và phân phối đến các chức năng cụ thể:
7. **AdminServlet** - Quản lý bảng điều khiển và các thao tác dành riêng cho quản trị viên.
8. **BorrowRecordServle**t - Quản lý thông tin mượn sách, cho phép người dùng xem hoặc xử lý chi tiết mượn trả.
9. **DeleteServlet** - Xử lý việc xóa các bản ghi quan trọng như sách, độc giả, hoặc nhân viên.
10. **EditBookServlet, EditReaderServlet, EditStaffServlet, EditServlet** - Cho phép chỉnh sửa thông tin chi tiết về sách, độc giả và nhân viên.
11. **InsertBookServlet, InsertReaderServlet, InsertStaffServlet** - Cung cấp chức năng thêm mới sách, độc giả và nhân viên vào hệ thống.
12. **ListBookServlet, ListReaderServlet, ListStaffServlet** - Truy xuất và hiển thị danh sách sách, độc giả, và nhân viên một cách dễ dàng.
13. **ReaderServlet, StaffServlet** - Điều hướng và cung cấp các chức năng dành riêng cho từng vai trò người đọc và nhân viên.
14. **LoginServle**t - Đảm nhận việc xác thực và quản lý phiên đăng nhập của người dùng.

* **Database Access Layer - DAL**

1. **DAO.java** - Định nghĩa các phương thức truy xuất dữ liệu, đảm bảo kết nối và tương tác hiệu quả với cơ sở dữ liệu.
2. **DBContext.java** - Quản lý kết nối đến cơ sở dữ liệu, tạo nền tảng cho các thao tác dữ liệu.

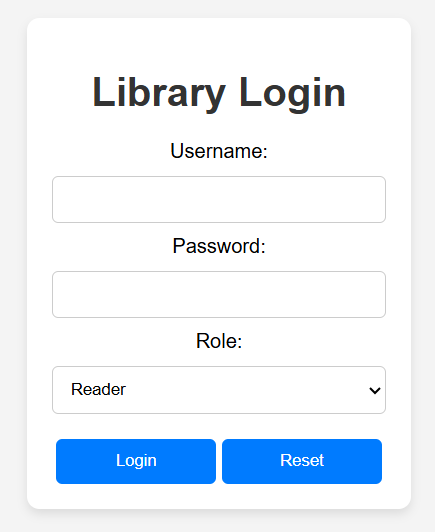
* **Models**: gồm các thư mục xây dựng các lớp dữ liệu, mô phỏng các đối tượng trong thư viện:

1. **Book, Reader, Staff** - Đại diện cho dữ liệu sách, độc giả và nhân viên, với các thuộc tính chi tiết.
2. **BorrowRecord, BorrowReturnDetail, ReturnRecord** - Ghi lại toàn bộ lịch sử mượn và trả sách, tạo nên sự minh bạch trong quản lý.
3. **Login, Role** - Xử lý thông tin đăng nhập và phân quyền người dùng cho các vai trò khác nhau trong hệ thống.

* **Session**: bao gồm **SessionLogin** và **SessionLoginDetails**, giúp duy trì trạng thái đăng nhập của người dùng, đảm bảo trải nghiệm liền mạch trong suốt quá trình sử dụng ứng dụng.

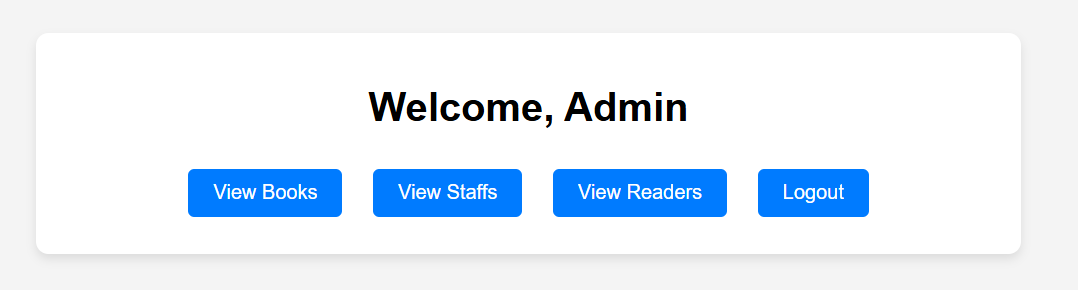
# II. Luồng đi cho từng chức năng chính trong Web application:

## Trang login tài khoản:



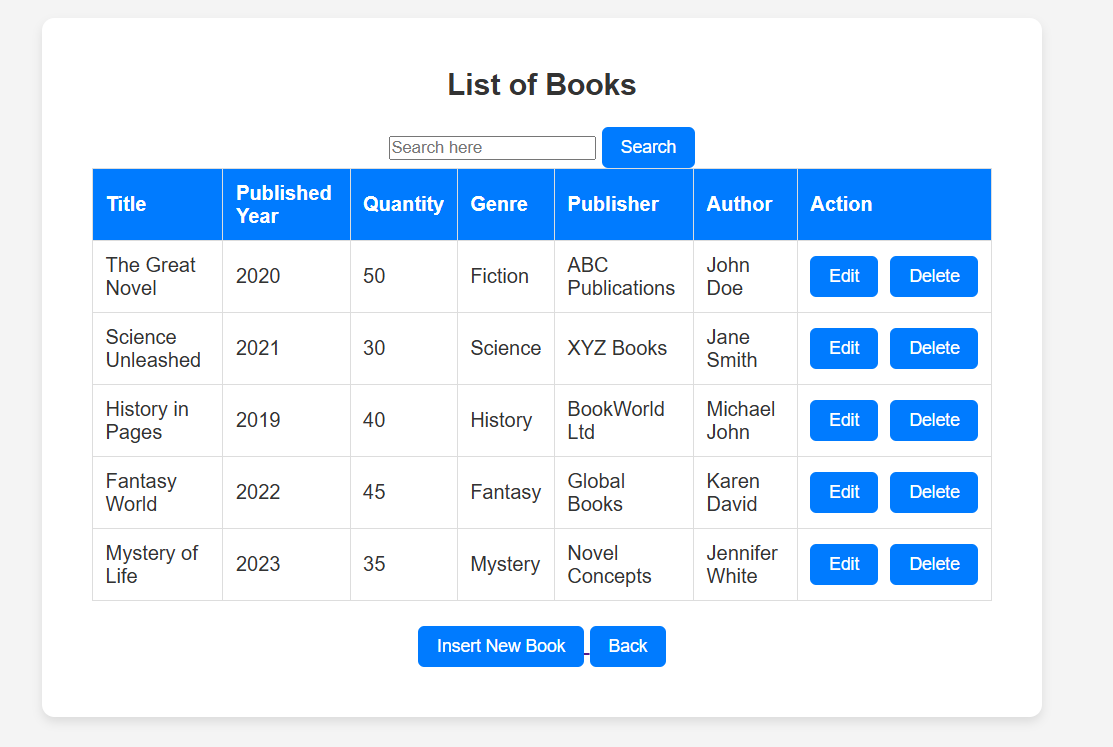
* **LoginServle**t nhận yêu cầu, kiểm tra tên đăng nhập và mật khẩu qua **DAO** trong cơ sở dữ liệu.
* Nếu thông tin hợp lệ, **SessionLogin** lưu thông tin đăng nhập, xác định quyền hạn (Admin, Reader, Staff).
* Người dùng được chuyển hướng đến trang chủ phù hợp với quyền hạn:

## **Admin ⇒ adminPage.jsp**

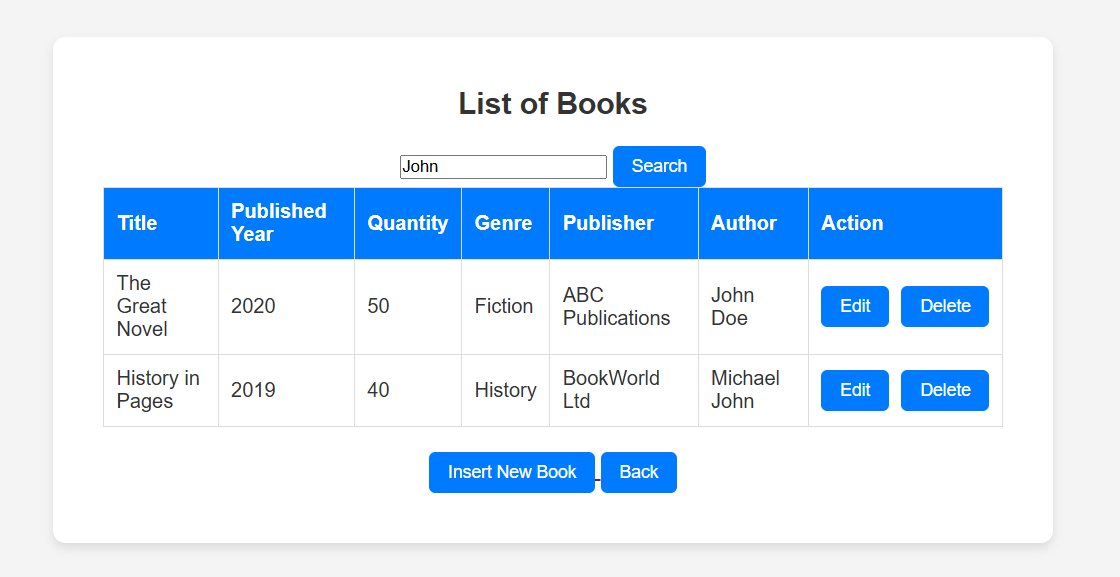


* Nếu người dùng chọn “View Books” =>  **ListBookServlet.**

**ListBookServlet** truy vấn cơ sở dữ liệu qua **DAO** để lấy danh sách sách và trả về **listBook.jsp** để hiển thị.

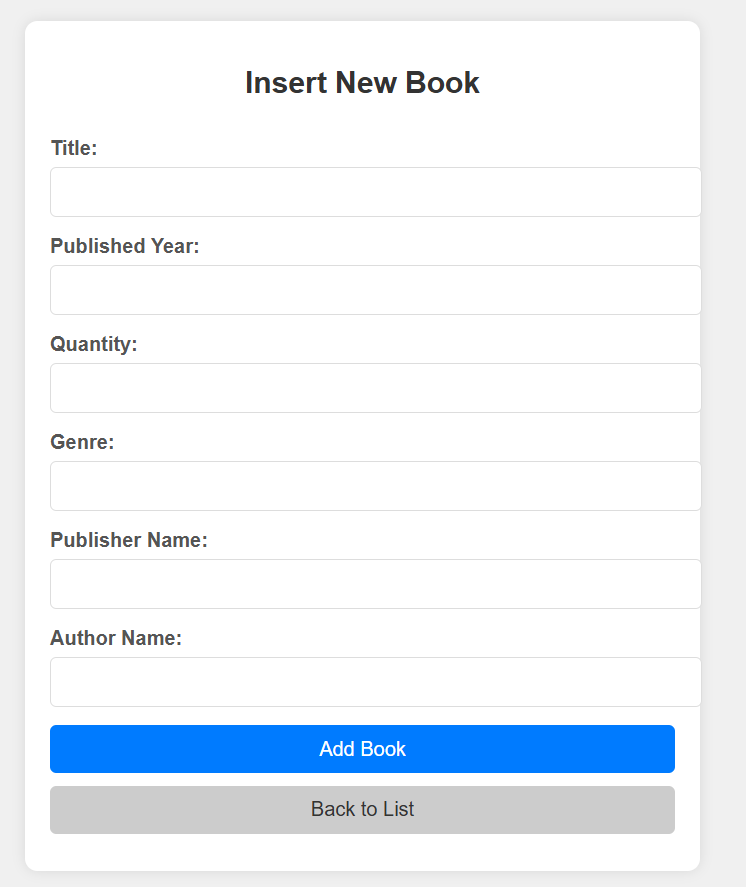


* Nếu người dùng muốn tìm kiếm, nhập vào “Search here” để tìm kiếm sách, năm xuất bản, thể loại, nhà xuất bản, tác giả. Danh sách sẽ hiển thị theo yêu cầu. Để trống ô “Search here” để quay trở lại danh sách đầy đủ.

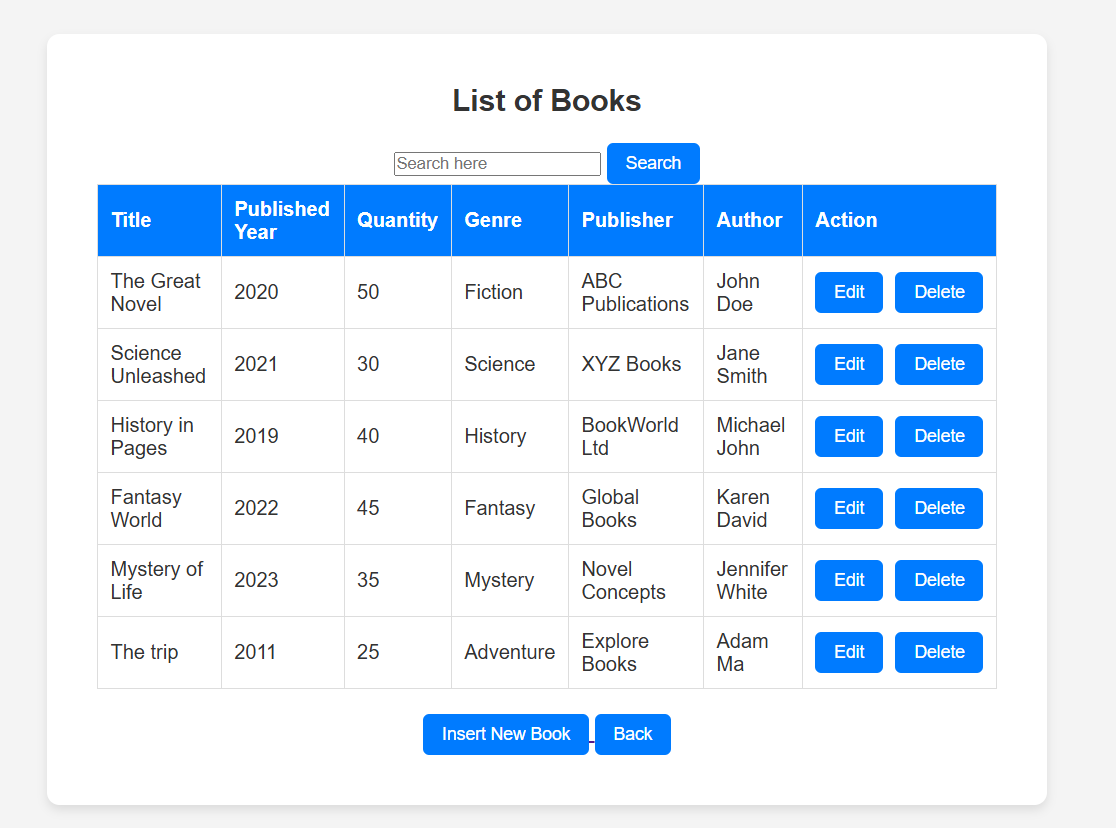


+ Nếu người dùng chọn "Insert New Book" -> **InsertBookServlet.**

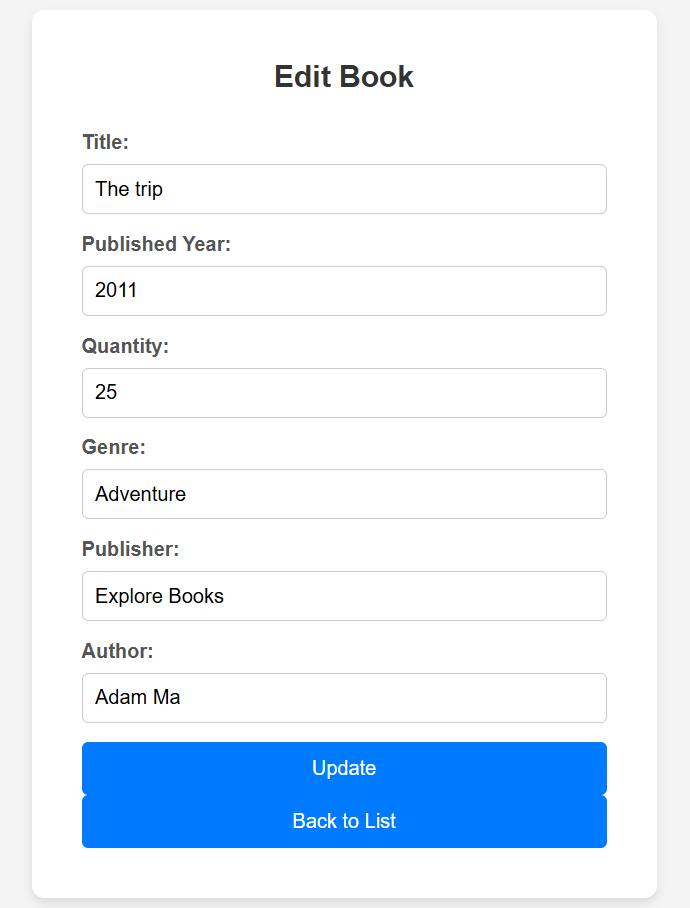
**InsertBookServlet** nhận thông tin sách mới từ giao diện **insertBook.jsp** và lưu vào cơ sở dữ liệu.



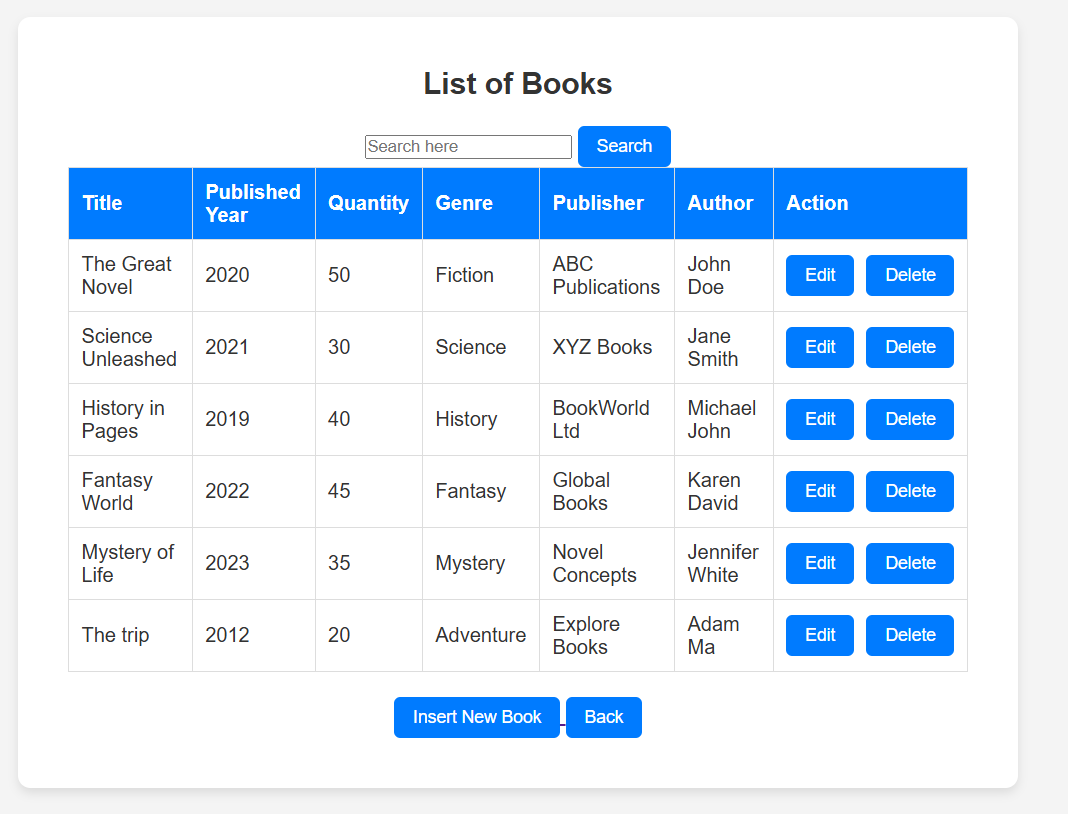
* Sau khi thêm thành công, hệ thống chuyển hướng về trang danh sách sách.



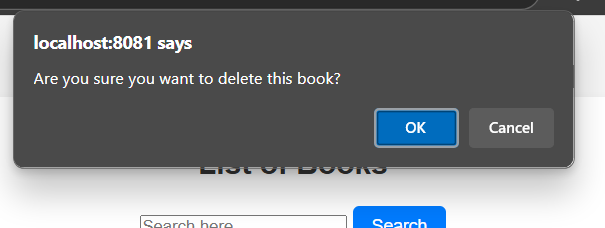
* Nếu người dùng chọn “Edit” từ danh sách, **EditBookServlet** hiển thị thông tin hiện tại trên **editBook.jsp.**



* Sau khi chỉnh sửa và gửi thông tin, **EditBookServlet** cập nhật dữ liệu trong cơ sở dữ liệu và điều hướng về trang danh sách.

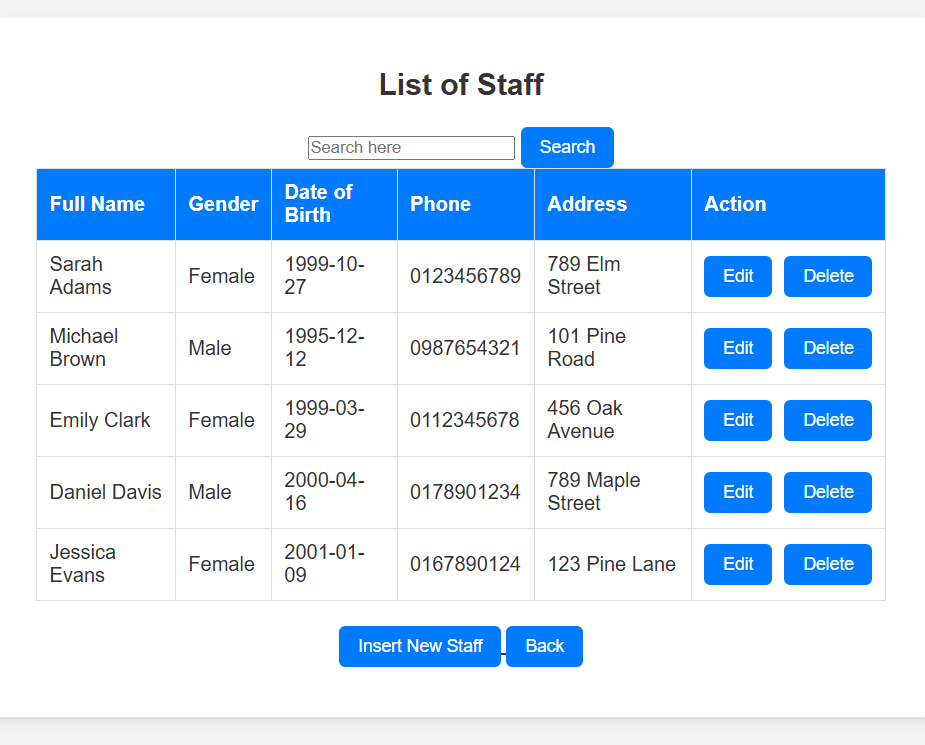


* Nếu người dùng muốn xoá sách, chọn “Delete” và xác nhận

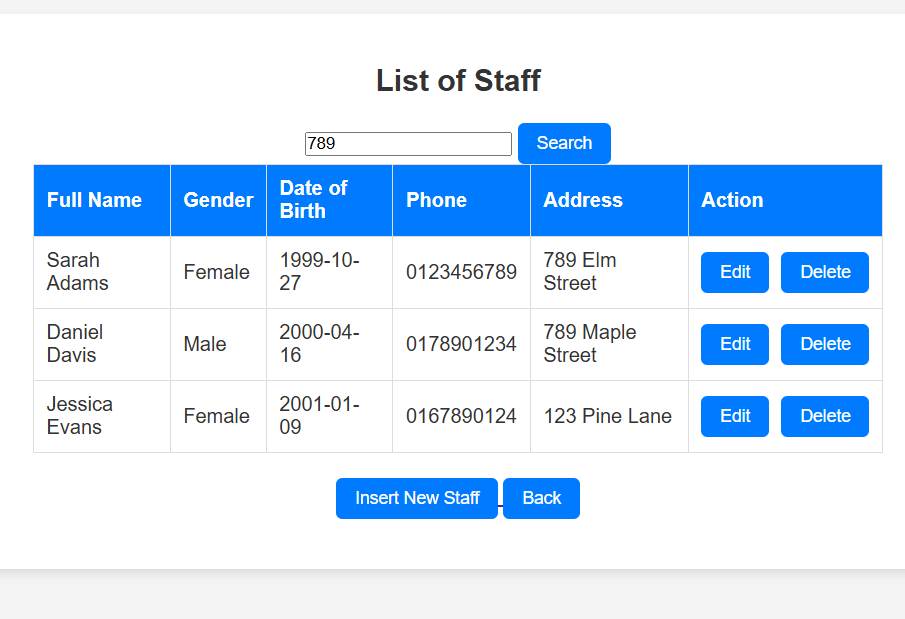


* Nếu người dùng chọn “View Staffs” => **ListStaffServlet.**

**ListStaffServlet** truy vấn cơ sở dữ liệu qua **DAO** để lấy danh sách sách và trả về **listStaff.jsp** để hiển thị.

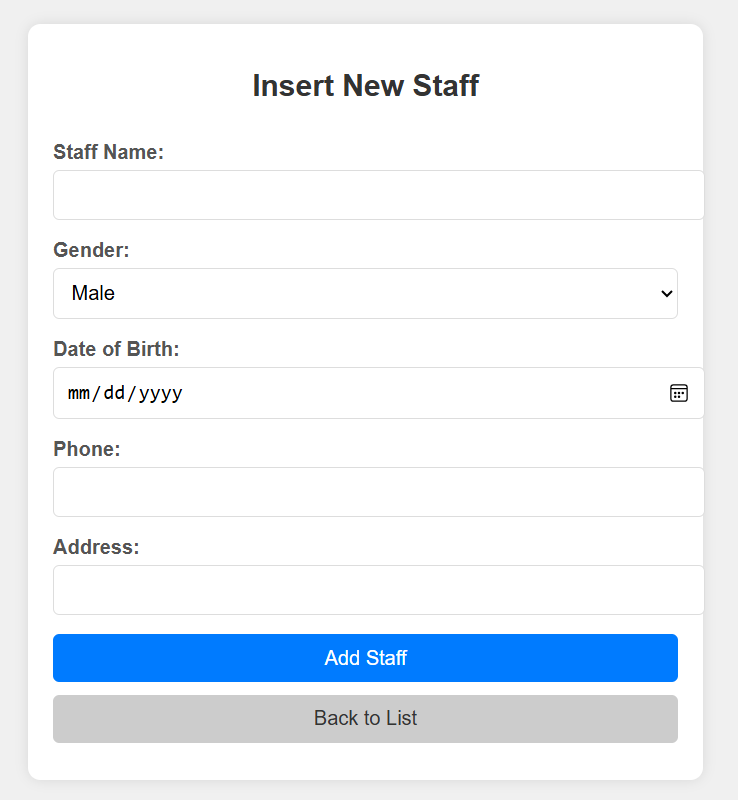


* Nếu người dùng muốn tìm kiếm, nhập vào “Search here” để tìm kiếm Tên, Ngày tháng năm sinh, Số điện thoại, Địa chỉ của Staff. Danh sách sẽ hiển thị theo yêu cầu. Để trống ô “Search here” để quay trở lại danh sách đầy đủ.

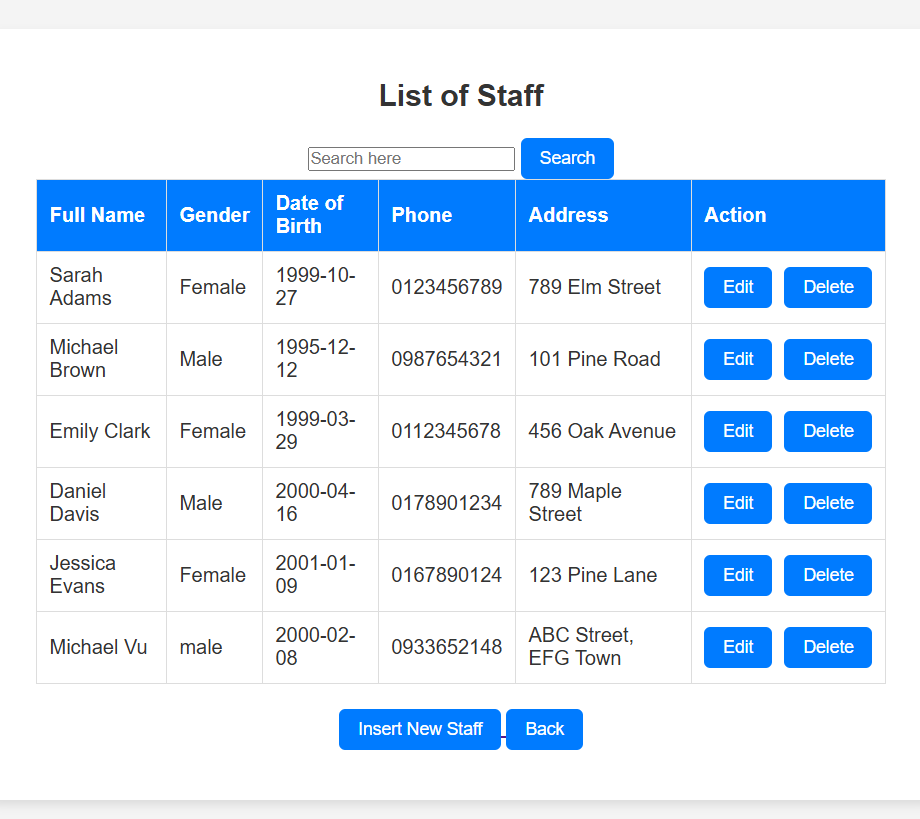


* Nếu người dùng chọn "Insert New Staff" -> **InsertStaffServlet.**

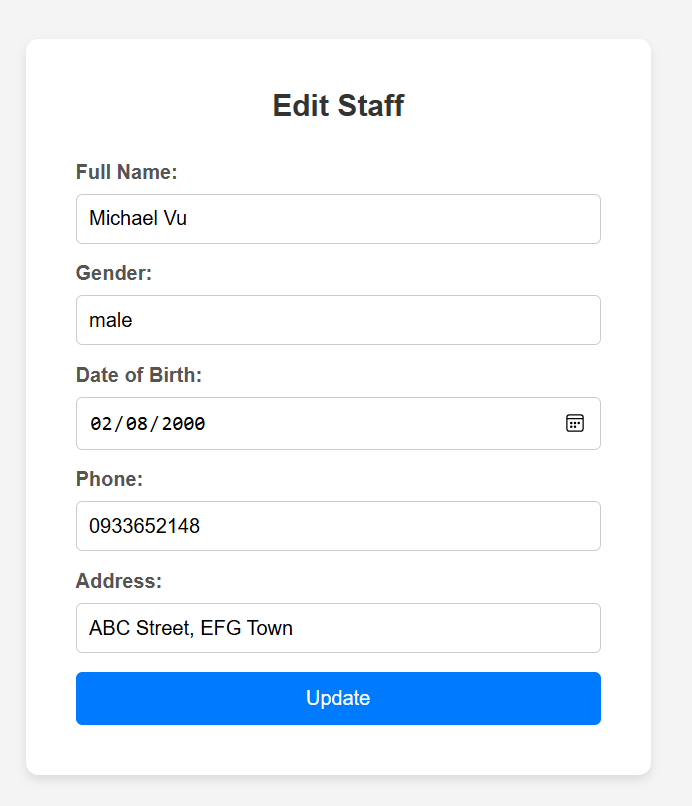
**InsertStaffServlet** nhận thông tin sách mới từ giao diện insertStaff.jsp và lưu vào cơ sở dữ liệu.



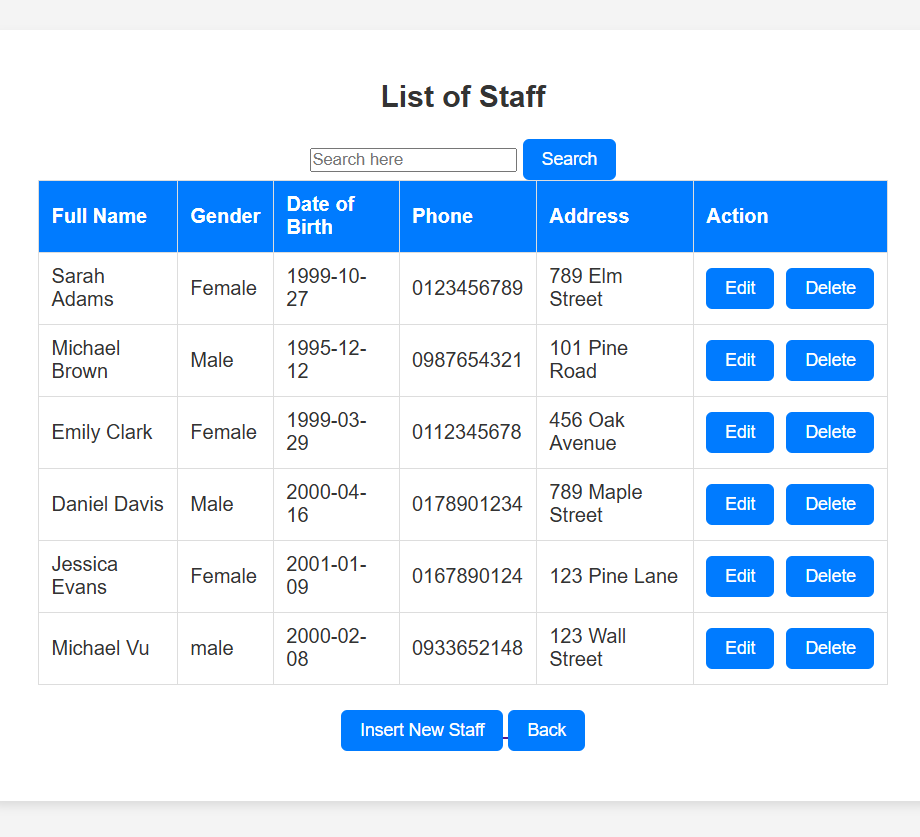
* Sau khi thêm thành công, hệ thống chuyển hướng về trang danh sách sách.



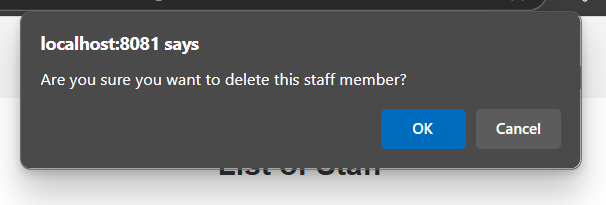
* Nếu người dùng chọn “Edit” từ danh sách, **EditStaffServlet** hiển thị thông tin hiện tại trên **editStaff.jsp**.



* Sau khi chỉnh sửa và gửi thông tin, **EditBookServlet** cập nhật dữ liệu trong cơ sở dữ liệu và điều hướng về trang danh sách.

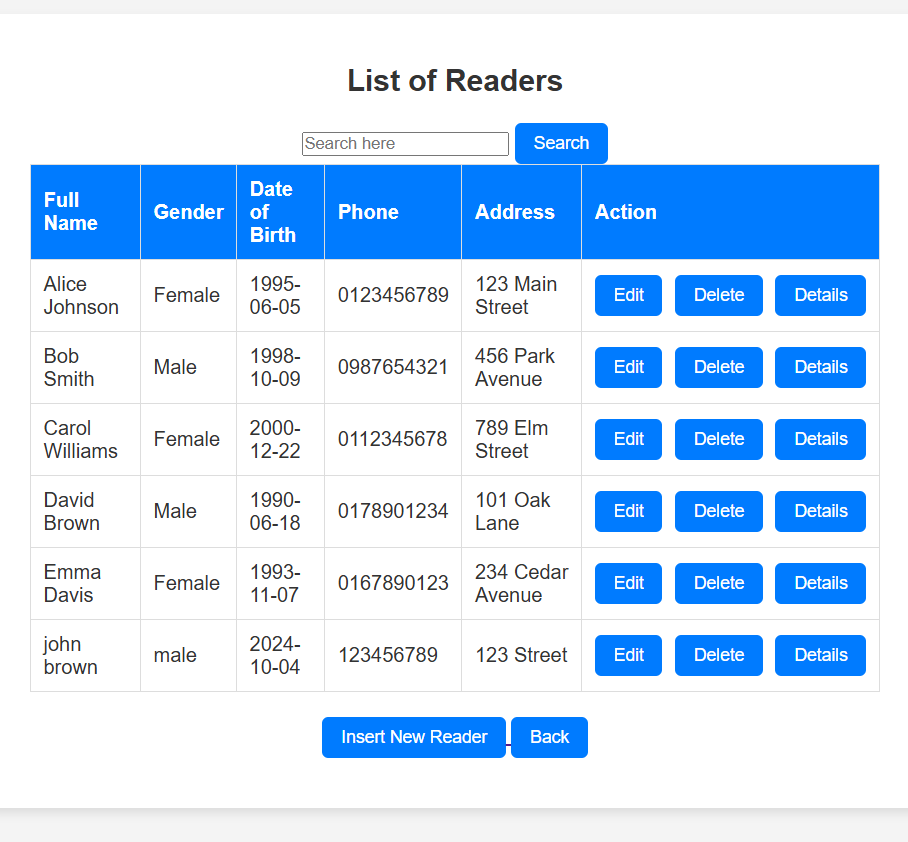


* Nếu người dùng muốn xóa nhân viên, chọn “Delete” và xác nhận

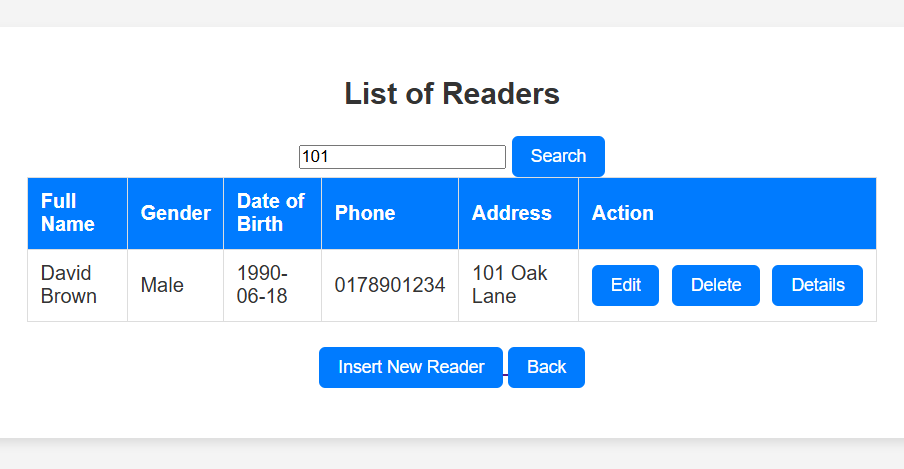


* Nếu người dùng chọn “View Readers” =>  **ListReaderServlet.**

**ListReaderServlet** truy vấn cơ sở dữ liệu qua **DAO** để lấy danh sách sách và trả về **listReader.jsp** để hiển thị.

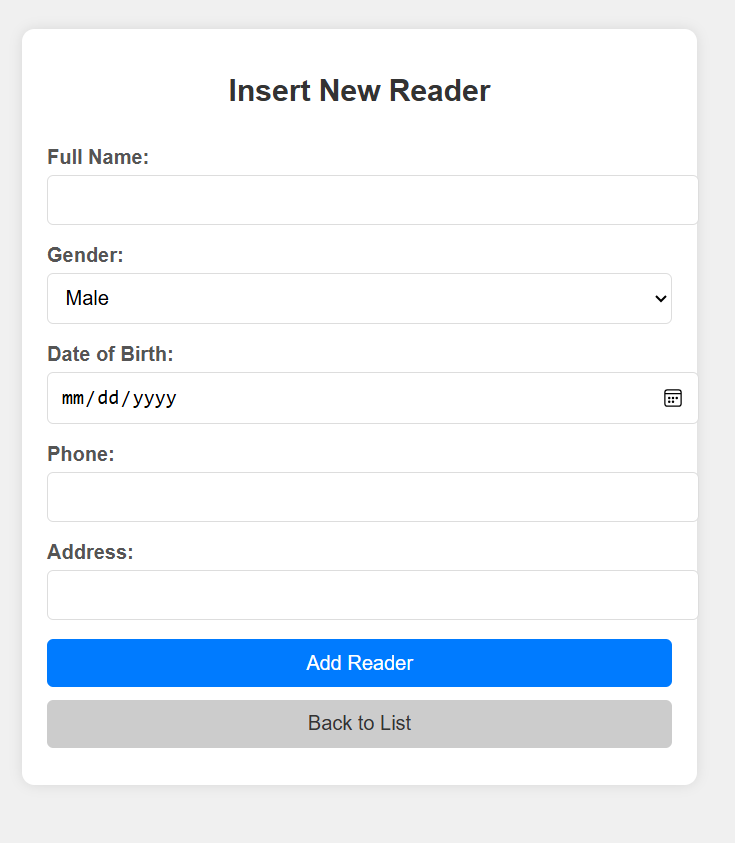


* Nếu người dùng muốn tìm kiếm, nhập vào “Search here” để tìm kiếm Tên, Ngày tháng năm sinh, Số điện thoại, Địa chỉ của Reader. Danh sách sẽ hiển thị theo yêu cầu. Để trống ô “Search here” để quay trở lại danh sách đầy đủ.

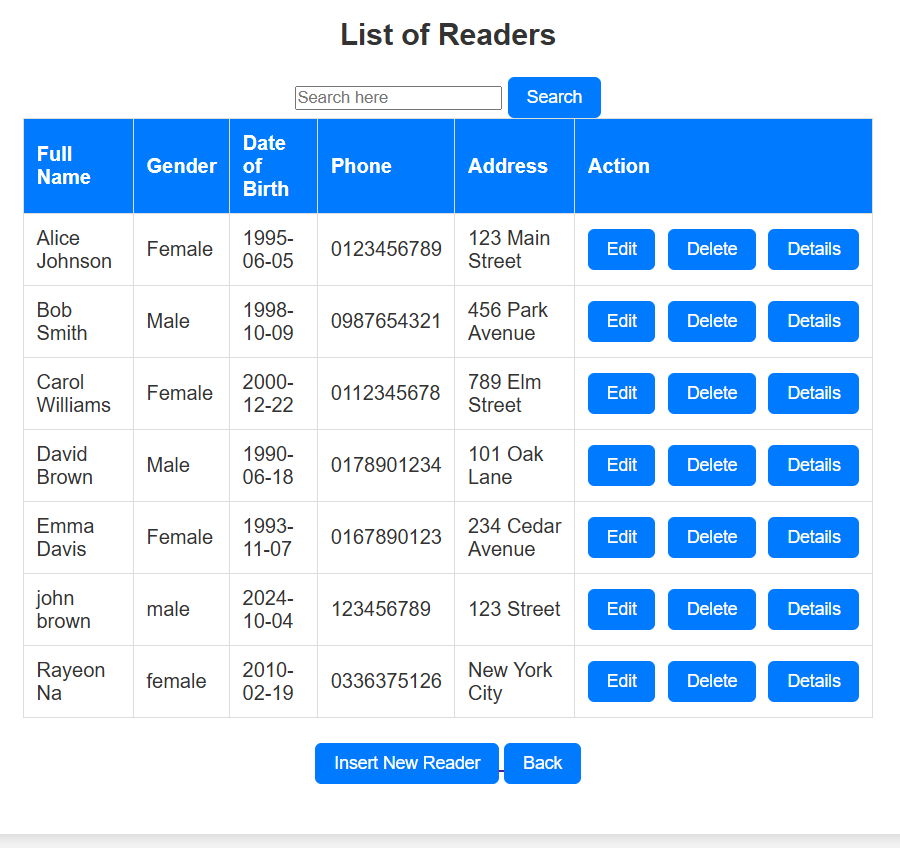


* Nếu người dùng chọn "Insert New Reader" -> **InsertReaderServlet.**

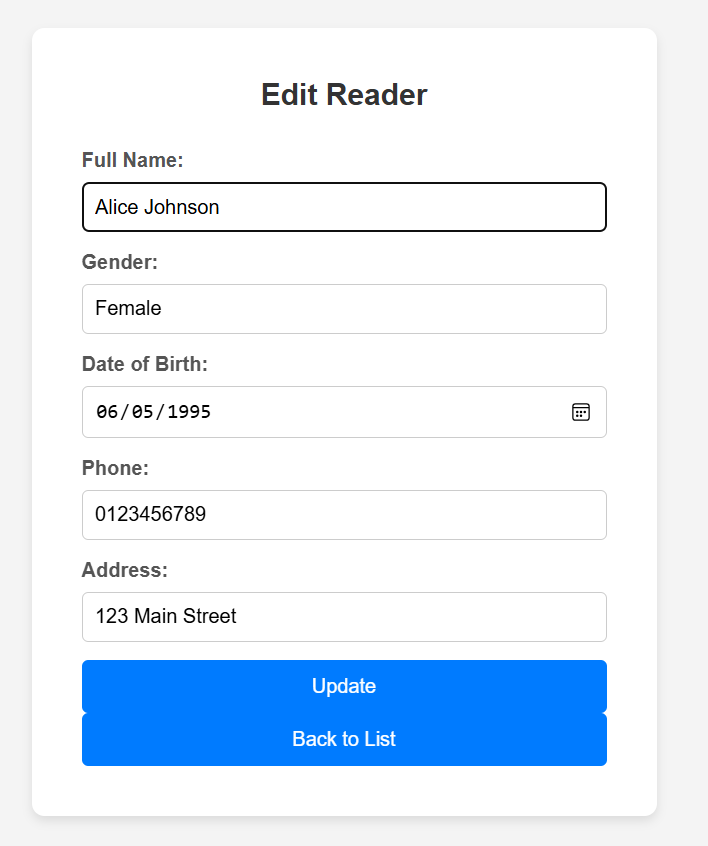
**InsertReaderServlet** nhận thông tin sách mới từ giao diện **insertReader.jsp** và lưu vào cơ sở dữ liệu.



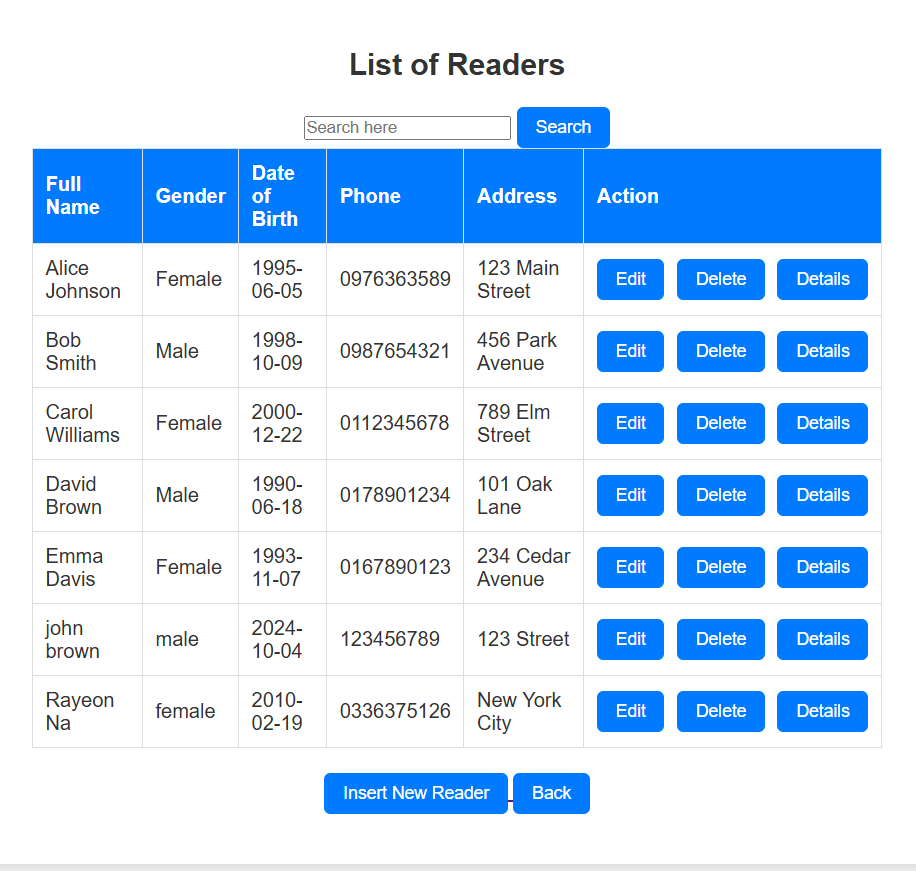
* Sau khi thêm thành công, hệ thống chuyển hướng về trang danh sách sách.



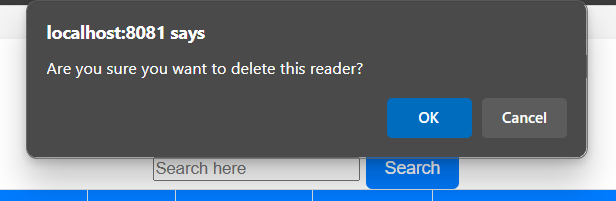
* Nếu người dùng chọn “Edit” từ danh sách, **EditReaderServlet** hiển thị thông tin hiện tại trên **editReader.jsp**.



* Sau khi chỉnh sửa và gửi thông tin, **EditBookServlet** cập nhật dữ liệu trong cơ sở dữ liệu và điều hướng về trang danh sách.

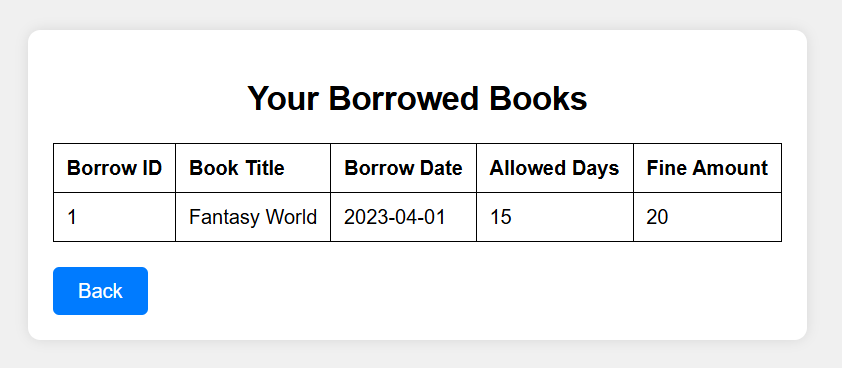


* Nếu người dùng muốn xoá Reader, chọn “Delete” và xác nhận



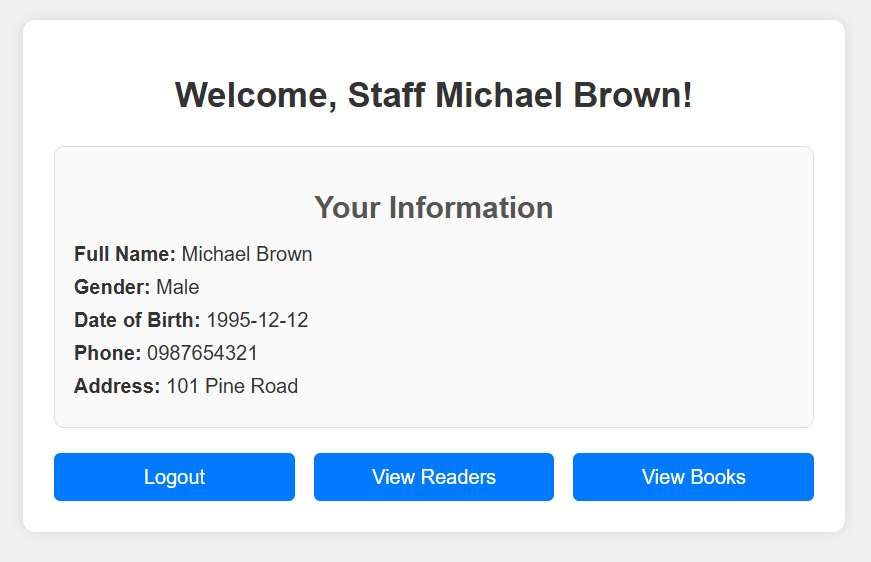
* Nếu người dùng muốn xem danh sách mượn sách của Reader, chọn “Details” => **BorrowRecordServlet.**

**BorrowRecordServlet** lấy thông tin từ cơ sở dữ liệu qua **DAO** và hiển thị trên **borrowRecord.jsp.**



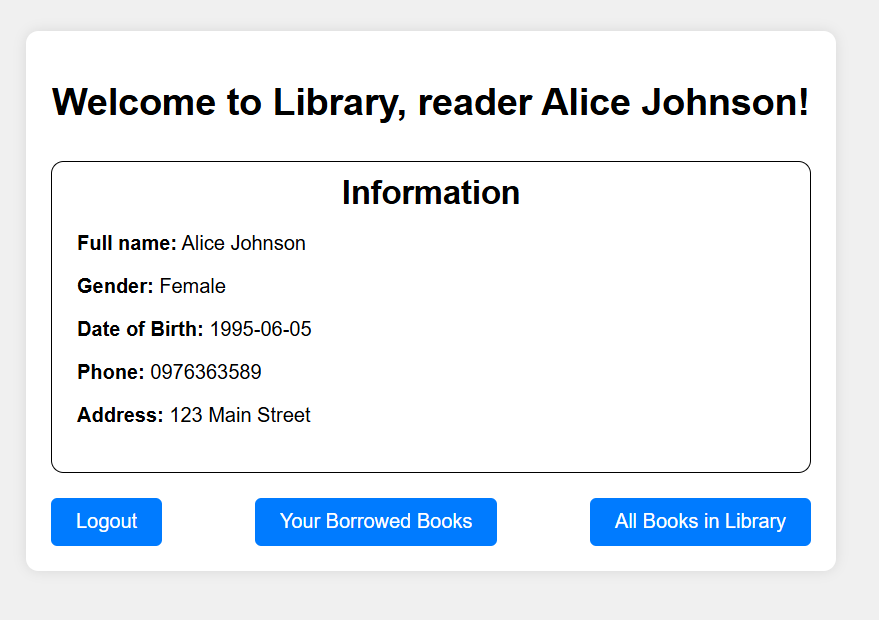
* Nếu người dùng muốn đăng xuất, chọn “Logout”

## **Staff ⇒ staffPage.jsp**

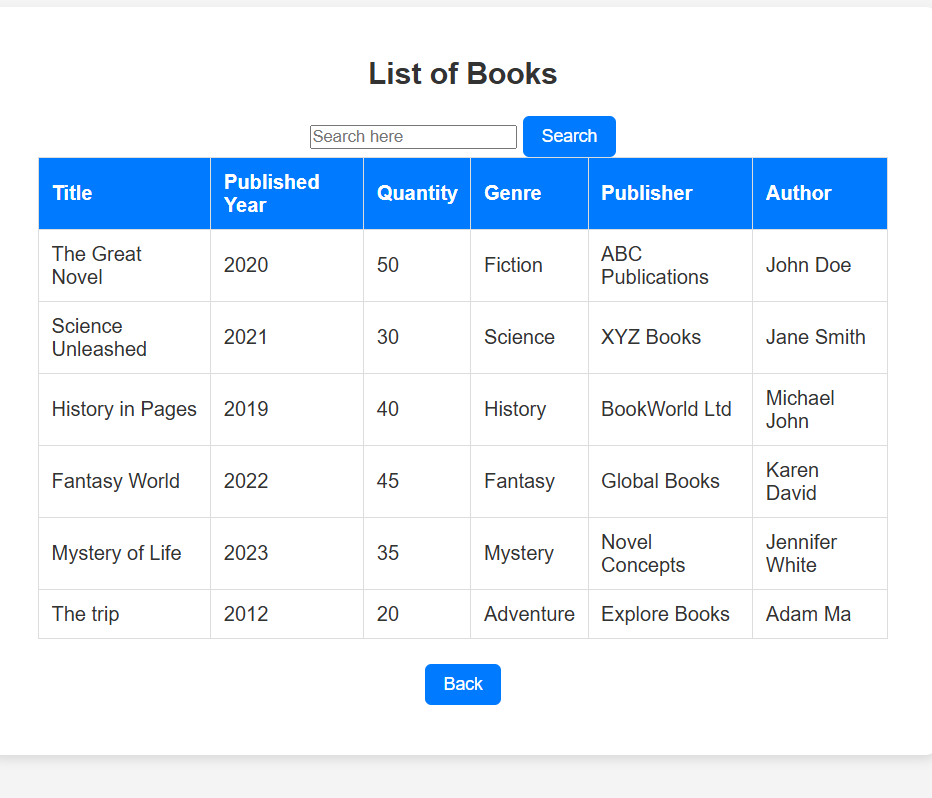


* Nếu người dùng chọn “View Books” hoặc “View Readers,” họ sẽ có các quyền như Admin: Edit, Delete, xem Details, và Insert. Tuy nhiên, người dùng với vai trò Staff sẽ không thể truy cập “View Staffs” để đảm bảo phân quyền hợp lý trong hệ thống.

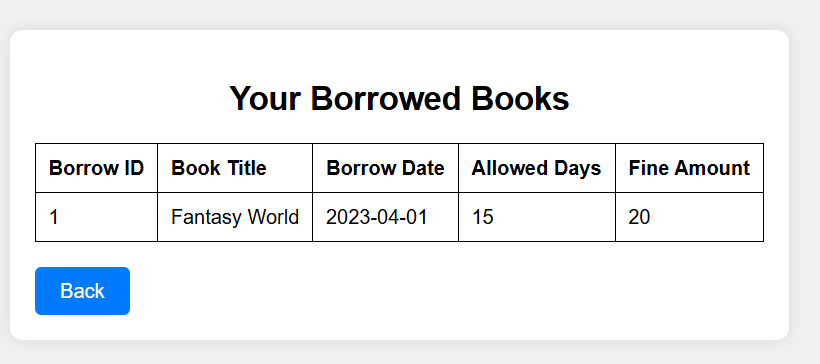
## **Reader ⇒ readerPage.jsp**



* Nếu người dùng chọn “View Books,” họ chỉ có thể xem danh sách sách và sử dụng chức năng tìm kiếm (Search)



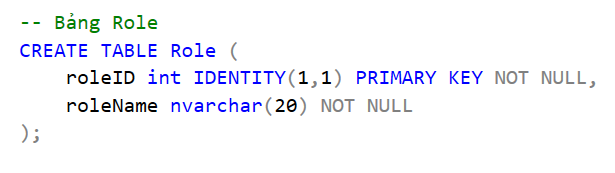
* Nếu người dùng muốn xem bản ghi mượn sách của mình, chọn “Your Borrowed Books”



# III. Thiết kế Database

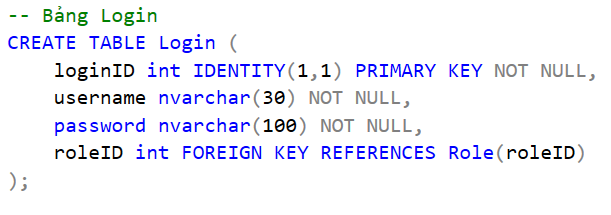
**1. Bảng Role**

* Mục đích: Lưu trữ các vai trò người dùng trong hệ thống.



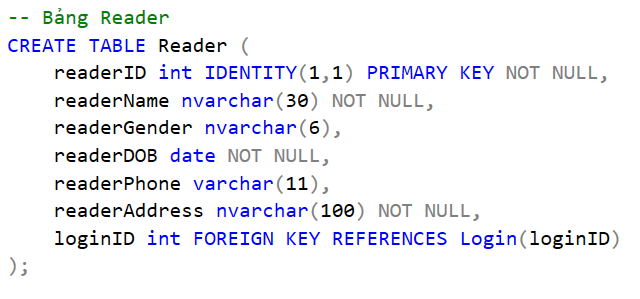
**2. Bảng Login**

* Mục đích: Lưu thông tin đăng nhập và phân quyền người dùng.



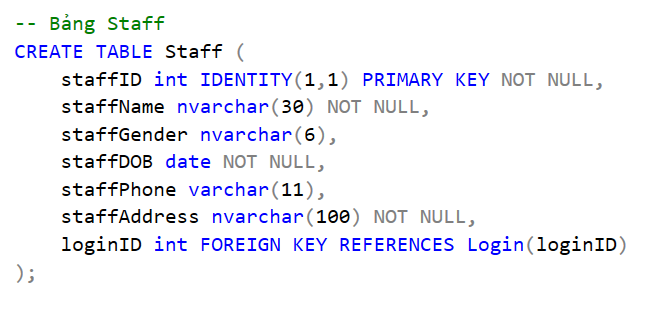
3. Bảng Reader

* Mục đích: Lưu trữ thông tin cá nhân của các độc giả.



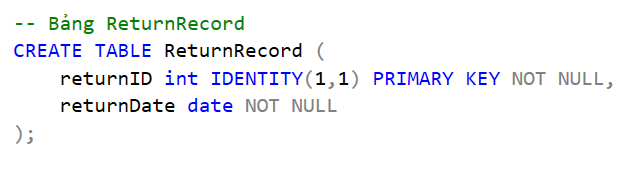
**4. Bảng Staff**

* Mục đích: Lưu thông tin của nhân viên thư viện.



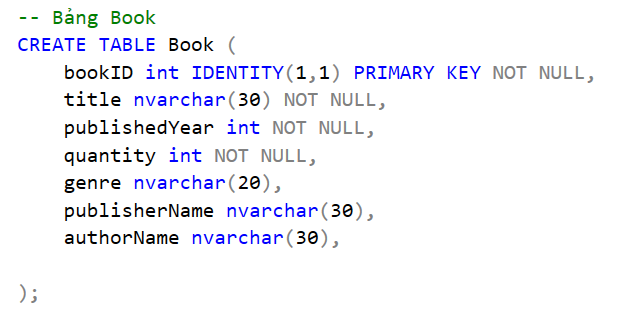
**5. Bảng ReturnRecord**

* Mục đích: Ghi nhận các lần trả sách của độc giả.



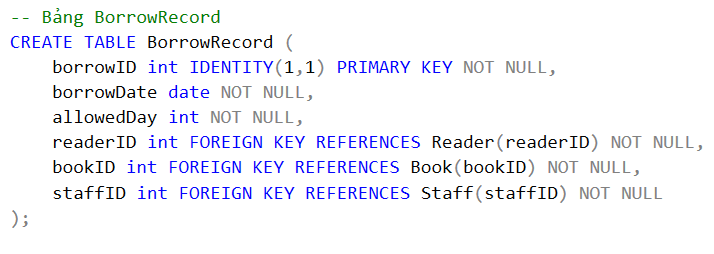
**6. Bảng Book**

* Mục đích: Lưu trữ thông tin về các cuốn sách trong thư viện.



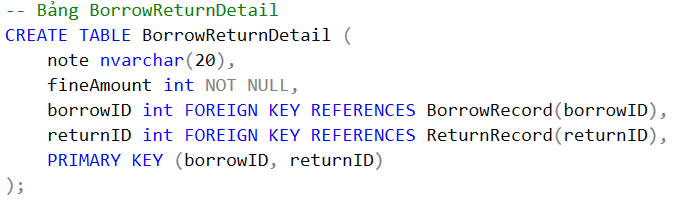
**7. Bảng BorrowRecord**

Mục đích: Ghi nhận thông tin mượn sách của độc giả.



**8. Bảng BorrowReturnDetail**

Mục đích: Quản lý chi tiết trả sách, bao gồm các khoản phạt nếu có.



* **Mối Quan Hệ Chính**

1. **Login - Role**: Mỗi tài khoản đăng nhập có một vai trò nhất định thông qua khóa ngoại roleID.
2. **Reader/Staff - Login**: Mỗi độc giả và nhân viên đều có một tài khoản đăng nhập liên kết qua loginID.
3. **BorrowRecord - Reader/Book/Staff**: Bảng BorrowRecord liên kết các bản ghi mượn với độc giả, sách và nhân viên.
4. **BorrowReturnDetail - BorrowRecord/ReturnRecord**: Bảng này liên kết các lần mượn với chi tiết trả, lưu lại các ghi chú và tiền phạt nếu có.

* **Database Diagram**

