**Hệ thống thông tin quản lý tuyển dụng**

**1. Thành phần chính của hệ thống**

Hệ thống quản lý tuyển dụng gồm các chức năng chính:

1. **Quản lý tài khoản**
   * Đăng ký
   * Đăng nhập
   * Sửa thông tin
   * Xóa tài khoản
2. **Quản lý CV**
   * Tạo CV
   * Chỉnh sửa CV
   * Xóa CV
3. **Ứng tuyển công việc**
   * Nộp CV
   * Theo dõi trạng thái hồ sơ
4. **Quản lý tin tuyển dụng**
   * Thêm tin tuyển dụng
   * Sửa tin
   * Xóa tin
5. **Thống kê – báo cáo**
   * Thống kê số lượng ứng viên
   * Thống kê số lượng tin tuyển dụng
   * Thống kê hồ sơ ứng tuyển
6. **Tìm kiếm – tra cứu**
   * Tìm việc theo từ khóa, ngành nghề
   * Nhà tuyển dụng tìm ứng viên
   * Tìm kiếm CV

**2. Quy trình hoạt động của hệ thống**

**2.1 Ứng viên**

* **Đăng ký / đăng nhập**
* **Quản lý hồ sơ & CV**
  + Tạo, chỉnh sửa, xóa CV
* **Ứng tuyển công việc**
  + Nộp hồ sơ ứng tuyển
  + Theo dõi trạng thái hồ sơ
  + Nhận phản hồi từ nhà tuyển dụng
* **Tìm kiếm việc làm**
  + Nhận gợi ý công việc
  + Tìm kiếm công việc theo ngành nghề, từ khóa

**2.2 Nhà tuyển dụng (Công ty / Doanh nghiệp)**

* **Đăng ký / đăng nhập**
* **Quản lý tin tuyển dụng**
  + Đăng, chỉnh sửa, xóa tin tuyển dụng
* **Quản lý ứng viên**
  + Xác nhận đăng ký, thông tin tài khoản
  + Danh sách CV ứng viên
  + Theo dõi hồ sơ ứng tuyển
* **Phản hồi kết quả ứng tuyển**
* **Tìm kiếm CV ứng viên**

**2.3 Admin (Quản trị hệ thống)**

* Quản lý người dùng (ứng viên & nhà tuyển dụng)
* Kiểm duyệt tin tuyển dụng và CV
* Quản lý báo cáo, log hệ thống
* Xuất thông kê toàn hệ thống
* Quản lý hệ thống, đảm bảo hoạt động ổn định

**3. Quy trình tổng thể**

1. **Ứng viên**:
   * Tạo tài khoản, quản lý CV, nộp hồ sơ ứng tuyển.
   * Nhận thông báo về trạng thái ứng tuyển.
   * Tra cứu việc làm và nhận gợi ý.
2. **Nhà tuyển dụng**:
   * Đăng ký tài khoản.
   * Quản lý tin tuyển dụng.
   * Xem danh sách ứng viên và CV.
   * Gửi phản hồi kết quả ứng tuyển.
3. **Admin**:
   * Quản lý người dùng, kiểm duyệt CV và tin tuyển dụng.
   * Quản lý báo cáo, thống kê dữ liệu hệ thống.
   * Xuất báo cáo tổng hợp và giám sát hệ thống.

**1️⃣ Cấu trúc cơ bản**

* index.html đặt trong src/main/resources/templates.
* Header nằm trong fragments/header.html (dùng Thymeleaf fragment để tái sử dụng).
* Controller chính HomeController để render index.html.
* Security config dùng Spring Security để quản lý login.

**2️⃣ Tạo header fragment**

**templates/fragments/header.html**

<header>

<nav>

<a href="/">Trang chủ</a>

<a href="/jobs">Tìm việc</a>

<a href="/create-cv">Tạo CV</a>

<a href="/support">Hỗ trợ</a>

<span th:if="${#authentication.principal?.username == null}">

<a href="/login">Đăng nhập</a>

<a href="/register">Đăng ký</a>

</span>

<span th:if="${#authentication.principal?.username != null}">

Xin chào, <b th:text="${#authentication.principal.username}">User</b>

<a href="/logout">Đăng xuất</a>

</span>

</nav>

</header>

* #authentication là object của Spring Security, có thể check xem user đã login hay chưa.

**3️⃣ Include header trong index.html**

**templates/index.html**

<!DOCTYPE html>

<html lang="vi" xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

<title>Trang chủ</title>

</head>

<body>

<div th:replace="fragments/header :: header"></div>

<h1>Chào mừng đến với website</h1>

</body>

</html>

* th:replace sẽ nhúng header fragment vào.

**4️⃣ Controller cơ bản**

@Controller

public class HomeController {

@GetMapping("/")

public String home() {

return "index";

}

}

**5️⃣ Spring Security cơ bản**

@Configuration

@EnableWebSecurity

public class SecurityConfig {

@Bean

public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {

http

.csrf(AbstractHttpConfigurer::disable)

.authorizeHttpRequests(auth -> auth

.requestMatchers("/", "/login", "/register", "/css/\*\*", "/js/\*\*").permitAll()

.anyRequest().authenticated()

)

.formLogin(form -> form

.loginPage("/login")

.defaultSuccessUrl("/", true)

.permitAll()

)

.logout(logout -> logout

.logoutUrl("/logout")

.logoutSuccessUrl("/")

);

return http.build();

}

}

* Sau khi login thành công, user được redirect về / → index.html.
* Header tự động thay đổi dựa vào #authentication.principal.username.

✅ **Tóm tắt workflow:**

1. Người dùng chưa login → index.html hiển thị header có button **Đăng nhập/Đăng ký**.
2. Người dùng login thành công → Spring Security redirect về /.
3. index.html được render lại → header check login → hiển thị tên user + **Đăng xuất**.