**BẢNG ĐÁNH GIÁ ĐIỂM THUYẾT TRÌNH**

**MÔN LẬP TRÌNH JAVA NÂNG CAO**

**LỚP CTK44-PM**

|  |  |
| --- | --- |
| **Thông tin đề tài thuyết trình** | **Thông tin sinh viên thực hiện đánh giá** |
| Tên đề tài: Thymeleaf Template | MSSV của sinh viên đánh giá: 2012395 |
| Điểm tổng kết: Ngày trình bày: 23/5/2023 | Họ tên sinh viên & ký xác nhận: Nguyễn Hữu Trọng Vỹ |
| ***Thông tin sinh viên thuyết trình*** |
| MSSV: 2012378  Họ tên: Đặng Ngọc Thắng |

1. **ĐÁNH GIÁ THUYẾT TRÌNH**

**Thang điểm: Kém 0-5; Trung bình: 5-7; Khá 7-8; Tốt 8-9; Xuất sắc: 9-10**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bảng chấm điểm nhóm thuyết trình** | | | | | | |
| Các chủ đề chính có được xác định? (A) | Slide thuyết trình có thiết kế tốt (B) | Giao tiếp với người nghe bằng ánh mắt (C) | Giao tiếp với người nghe bằng giọng nói (D) | Khả năng trình bày nội dung (E) | Thảo luận & Hỏi đáp (F) | Điểm trung bình = (A+B+C+D+E+F)/6 |
| **9.5** | **9.5** | **9.5** | **9.5** | **9.5** | **9.5** | **9.5** |

1. **BÀI TẬP NẮM KIẾN THỨC**

Dựa vào nội dung vừa trình bày, trả lời các câu hỏi sau:

1. Trình bày tóm tắt nội dung anh/ chị đã nắm được về chủ đề vừa trình bày?

**Khái niệm**

Thymeleaf là một công cụ mạnh mẽ và linh hoạt được sử dụng để phát triển giao diện người dùng (UI) trong các ứng dụng web Java.

Nó là một công nghệ template engine, cho phép xây dựng các template HTML, XML hoặc văn bản thông qua việc kết hợp các mã thymeleaf vào trong các file template này.

Thymeleaf được tích hợp vào ứng dụng web Java thông qua một template engine. Nhiệm vụ của template engine là xử lý các template và thực hiện thay thế các đoạn mã thymeleaf bằng dữ liệu thực tế trong quá trình render.

Thymeleaf templates là một phần quan trọng trong việc phát triển giao diện người dùng trong các ứng dụng web Java.

**Ưu điểm**

* Cú pháp dễ đọc và hiểu
* Tích hợp tốt với Spring Framework
* Linh hoạt và mạnh mẽ
* Render trực tiếp và xem trước
* Hỗ trợ quốc tế hóa
* Cộng đồng phát triển và tài liệu phong phú

**Nhược điểm**

* Hiệu suất chậm hơn so với các công nghệ template khác
* Phụ thuộc vào Spring Framework
* Đòi hỏi kiến thức HTML và CSS
* Khả năng mở rộng có hạn
* Khó khăn trong việc kiểm tra và gỡ lỗi

**Đặc điểm chính**

* Tích hợp dữ liệu
* Xử lý sự kiện
* Lặp lại dữ liệu
* Quản lý layout template
* Tích hợp với Spring Framework

**Cú pháp**

* Thymeleaf sử dụng cú pháp đơn giản và dễ hiểu, tương tự với cú pháp của HTML.
* Cú pháp của Thymeleaf sẽ là một attributes của thẻ HTML và bắt đầu bằng chữ “th:”.
* Với cách tiếp cận này, chỉ cần sử dụng các thẻ HTML cơ bản đã biết mà không cần bổ sung thêm syntax hay thẻ mới như JSP truyền thống.

|  |  |  |
| --- | --- | --- |
|  | Cú pháp | Ý nghĩa |
| Thymeleaf Expression | **- th:text** | - Thêm nội dung văn bản vào một phần tử HTML. |
| **- th:utext** | - Thêm nội dung văn bản không được xử lý HTML vào một phần tử HTML |
| **- th:each** | - Lặp qua một danh sách và áp dụng cú pháp cho từng phần tử |
| **- th:if, th:unless** | - Điều kiện if/else để điều khiển hiển thị các phần tử HTML |
| **- th:switch, th:case** | - Switch/case để xác định trường hợp hiển thị phù hợp |
| **- th:attr** | - Đặt giá trị thuộc tính HTML |
| **- th:href, th:src** | - Đặt giá trị cho thuộc tính href và src. |
| **- th:value** | - Đặt giá trị cho giá trị mặc định của một trường input |
| **- th:class, th:style** | - Đặt giá trị cho thuộc tính class và style |
| Biểu thức Thymeleaf | **- ${expression}** | - Đánh giá biểu thức và hiển thị giá trị |
| **- \*{expression}** | - Ràng buộc dữ liệu và biểu thức với một đối tượng trong Model |
| **- @{expression}** | - Tạo URL tương đối dựa trên biểu thức |
| **- #{expression}** | - Lấy thông tin từ tệp tin ngôn ngữ (properties file) |
| Fragment | **- <th:block>** | - Tạo một khối chứa các phần tử HTML |
| **- <th:fragment>** | - Định nghĩa một đoạn mã có thể được sử dụng lại |
| Môi trường | **- th:object** | - Đặt một đối tượng làm môi trường hiện tại |
| **- th:with** | - Định nghĩa một biến mới trong môi trường hiện tại |
|  | Cú pháp | Ý nghĩa |
| Quản lý biểu mẫu | **- th:action** | - Đặt URL hoạt động cho một biểu mẫu |
| **- th:field** | - Liên kết một trường biểu mẫu với một trường đối tượng |
| **- th:errors** | - Hiển thị thông báo lỗi liên quan đến một trường biểu mẫu |

**Tích hợp Spring Framework**

Thymeleaf được thiết kế đặc biệt để tích hợp tốt với Spring Framework và cung cấp tích hợp sẵn với nhiều tính năng của Spring:

* Dependency
* Cấu hình Thymeleaf
* Sử dụng Thymeleaf trong Spring MVC
* Sử dụng Thymeleaf trong templates

**Dependency**

Cần thêm các dependency cần thiết vào file pom.xml để sử dụng Thymeleaf và Spring Framework.

**Cấu hình Thymeleaf**:

Cần cấu hình Thymeleaf trong file cấu hình của Spring (thông thường là file application.properties hoặc application.yml).

**Sử dụng Thymeleaf trong Spring MVC**:

Có thể trả về tên template và dữ liệu cho template trong model của Spring.

**Sử dụng Thymeleaf trong template**:

Có thể sử dụng cú pháp Thymeleaf để truy cập và hiển thị dữ liệu được truyền từ controller.

**Tích hợp CSS**

* Đảm bảo rằng đã định nghĩa các tệp CSS và đặt chúng trong thư mục tài nguyên của dự án (thông thường là thư mục resources/static/css/).
* Trong template Thymeleaf, sử dụng thuộc tính *th:href* để chỉ định đường dẫn đến tệp CSS và thuộc tính rel để chỉ định kiểu liên kết.

**Tích hợp JS**

* Đảm bảo rằng bạn đã định nghĩa các tệp JavaScript của bạn và đặt chúng trong thư mục tài nguyên của dự án (thông thường là thư mục resources/static/js/).
* Trong template Thymeleaf, sử dụng thuộc tính th:src để chỉ định đường dẫn đến tệp JavaScript và thuộc tính type để chỉ định kiểu tệp.

**Truyền dữ liệu từ Controller vào Template**

Để truyền dữ liệu từ Controller vào Template trong Thymeleaf, có thể sử dụng đối tượng Model để chứa và truyền dữ liệu.

**Hiển thị dữ liệu trong Template**

Trong Template Thymeleaf, có thể truy cập và hiển thị dữ liệu từ Model bằng cách sử dụng cú pháp Thymeleaf.

**Binding dữ liệu đối tượng**

Ngoài việc binding dữ liệu cơ bản, Thymeleaf cũng hỗ trợ binding dữ liệu từ đối tượng (object). Có thể truy cập các thuộc tính và phương thức của đối tượng trong template.

**Xử lý sự kiện**

Trong Thymeleaf, có thể xử lý sự kiện trên các thành phần HTML bằng cách sử dụng các thuộc tính và các cú pháp cung cấp bởi Thymeleaf.

**Template Layouts**

Trong Thymeleaf, template layout được sử dụng để tạo cấu trúc và giao diện chung cho nhiều trang trong ứng dụng web. Template layout cho phép xác định các phần chung như header, footer, thanh điều hướng và áp dụng chúng cho các trang cụ thể. Điều này giúp giảm sự trùng lặp và quản lý giao diện một cách hiệu quả.

**So sánh Thymeleaf Template và JSP Template Engine**

Tùy thuộc vào yêu cầu và môi trường phát triển, cả Thymeleaf và JSP đều có ưu điểm và giới hạn riêng. Thymeleaf thường được ưa chuộng hơn trong các dự án phát triển web Java hiện đại nhờ cú pháp HTML dễ đọc, tích hợp tốt với HTML và khả năng xử lý trên máy chủ linh hoạt. Việc chọn template engine phụ thuộc vào nhiều yếu tố như kỹ năng của nhóm phát triển, yêu cầu dự án, và sự tương thích với công nghệ hiện có.

|  |  |  |
| --- | --- | --- |
|  | Thymeleaf Template | JSP Template Engine |
| Cú pháp | Sử dụng cú pháp HTML thuần, giúp dễ đọc và hiểu. Cú pháp Thymeleaf được chèn vào các thuộc tính HTML thông qua các thuộc tính Thymeleaf như th:, th:text, th:if,... | Sử dụng cú pháp riêng (JSP Syntax) với các thẻ đặc biệt như <% %>, <%= %>, <%@ %>,... Cú pháp này có thể khó hiểu và khó bảo trì hơn cú pháp HTML thuần. |
| Tích hợp với HTML | Cú pháp Thymeleaf được chèn vào các thuộc tính HTML, nhưng cú pháp HTML vẫn hợp lệ mà không cần sửa đổi. | Cú pháp JSP phải được chèn vào các thẻ đặc biệt và có thể làm cho mã HTML trở nên khó đọc và khó bảo trì hơn. |
| Xử lý tại máy chủ | Mã Thymeleaf được xử lý tại máy chủ và được chuyển thành mã HTML trước khi gửi đến trình duyệt. Điều này có nghĩa là trình duyệt nhận được các tệp HTML thuần túy mà không biết gì về Thymeleaf. | Mã JSP được biên dịch thành Servlet và thực thi trên máy chủ web. Mã HTML được tạo ra và gửi trực tiếp đến trình duyệt. |
|  | Thymeleaf Template | JSP Template Engine |
| Hiệu suất | Tính năng caching mạnh mẽ, cho phép lưu trữ các trang đã được xử lý trước đó để cải thiện hiệu suất. | Hiệu suất tốt hơn trong môi trường phát triển và thử nghiệm cục bộ, nhưng có thể không hiệu quả trong các ứng dụng có tải cao do việc thực thi mã trên máy chủ. |
| Độ phổ biến và hỗ trợ cộng đồng | Đang ngày càng trở nên phổ biến và có được sự hỗ trợ từ cộng đồng ngày càng lớn. Thymeleaf là một công cụ phát triển web được sử dụng rộng rãi trong cộng đồng Java. | JSP đã tồn tại trong một thời gian dài và có cộng đồng hỗ trợ đông đảo. Tuy nhiên, JSP đã trở nên ít phổ biến hơn trong những năm gần đây, và nhiều dự án mới đang chuyển sang các công nghệ template khác như Thymeleaf. |

**Điểm hay của Thymeleaf Template**

* Cú pháp thuần HTML
* Tích hợp tốt với HTML
* Xử lý tại máy chủ (Server-side rendering)
* Hỗ trợ tích hợp tốt
* Công cụ mạnh mẽ
* Hiệu suất và caching

1. Anh/ chị hãy đưa ra các nhận xét và góp ý cho sinh viên vừa trình bày?

Nội dung đầy đủ, hoàn hiện Demo tốt hơn.