Quản lý sinh viên (Spring MVC + JSP/JSTL)

Mục tiêu bài tập

Xây dựng ứng dụng web theo mô hình Spring Web MVC (Controller–Service–Repository/DAO) có:

* CRUD đầy đủ (list, chi tiết, tạo, sửa, xóa).
* Tìm kiếm, phân trang, sắp xếp.
* Ràng buộc & validate dữ liệu form với @ModelAttribute.
* Flash message khi redirect.
* (Tùy chọn) Lưu dữ liệu bằng JDBC; nếu chưa kịp, dùng in-memory (ArrayList/HashMap) rồi thay thế.
* View dùng JSP + JSTL với InternalResourceViewResolver.

Tính năng bắt buộc

1) Danh sách sinh viên (GET /students)

* Hiển thị bảng gồm: Mã số, Họ tên, Điểm tổng kết, Hạng (xếp loại từ điểm).
* Thanh công cụ:
  + Ô tìm kiếm theo Mã số/Họ tên: q (RequestParam).
  + Sắp xếp: sort ∈ {id,name,gpa} & dir ∈ {asc,desc}.
  + Phân trang: page (mặc định 1), size (mặc định 5 hoặc 10).
* Action:
  + Link “Thêm sinh viên” → /students/add
  + Link “Chi tiết” → /students/{id}
  + Link “Sửa” → /students/{id}/edit
  + Nút “Xóa” (POST/DELETE giả lập) → /students/{id}/delete
* Dùng JSTL <c:forEach>, <c:out> để render; phần phân trang hiển thị số trang & đang ở trang nào.
* Flash message: hiển thị khi thêm/sửa/xóa thành công (ví dụ message trong RedirectAttributes).
  + Trong view (file .JSP), cung cấp thêm vùng hiển thị thông báo (message) được gán flash-attribute

Controller gợi ý

|  |
| --- |
| @GetMapping("/students") public ModelAndView list(   @RequestParam(defaultValue = "") String q,   @RequestParam(defaultValue = "id") String sort,   @RequestParam(defaultValue = "asc") String dir,   @RequestParam(defaultValue = "1") int page,   @RequestParam(defaultValue = "10") int size) {     // gọi service: findAll(q, sort, dir, page, size)     return new ModelAndView("students/index", model); // model đặt list, pageInfo, q, sort, dir } |

2) Chi tiết sinh viên (GET /students/{id})

* Hiển thị thông tin đầy đủ 1 sinh viên.
* Có nút quay lại danh sách; link “Sửa”, “Xóa”.

Controller gợi ý

|  |
| --- |
| @GetMapping("/students/{id}") public String detail(@PathVariable String id, Model model) { ... } |

3) Thêm sinh viên (GET/POST /students/add)

* GET: trả form rỗng qua @ModelAttribute("studentForm") StudentForm.
* POST: nhận @ModelAttribute + @RequestParam (nếu cần), validate, nếu OK:
  + Lưu qua Service → Redirect /students + flash message “Thêm thành công”.
  + Nếu lỗi, quay lại form và hiển thị lỗi ngay bên dưới từng input.

Validation yêu cầu (kiểm tra thủ công):

* id (String): không rỗng, độ dài 3–20, không trùng.
* name (String): không rỗng, loại bỏ khoảng trắng dư thừa.
* gpa (float/double): 0.0 ≤ gpa ≤ 10.0.

Controller gợi ý

|  |
| --- |
| @GetMapping("/students/add") public String showAddForm(Model model) {   model.addAttribute("studentForm", new Student());   return "students/add"; }  @PostMapping("/students/add") public String doAdd(@ModelAttribute() Student studentForm,                     BindingResult binding,                     RedirectAttributes ra) {   if (binding.hasErrors()) return "students/add";   studentService.create(studentForm);   ra.addFlashAttribute("message", "Thêm sinh viên thành công!");   return "redirect:/students"; } |

4) Sửa sinh viên (GET/POST /students/{id}/edit)

* GET: load dữ liệu hiện có → bind vào studentForm.
* POST: validate tương tự thêm; id là khóa (không đổi) hoặc cho đổi nhưng phải kiểm tra trùng.
* Redirect về /students + flash message.

5) Xóa sinh viên (POST /students/{id}/delete)

* Xác nhận xóa (có thể làm modal hoặc trang xác nhận đơn giản).
* Redirect /students + flash “Xóa thành công”.