

Đề Bài: Hệ Thống Quản Lý Sinh Viên - StudentHub

Mô tả bài toán:

Trường Đại học ABC cần xây dựng hệ thống quản lý thông tin sinh viên và điểm số đơn giản. Hệ thống sử dụng Java Servlet, JSP và MySQL với 2 bảng chính đã được thiết kế sẵn:

Bảng 1: STUDENTS (Sinh viên)

```
sql
CREATE TABLE STUDENTS (
  student_id VARCHAR(8) PRIMARY KEY,
  full_name VARCHAR(100) NOT NULL,
  email VARCHAR(50) NOT NULL UNIQUE,
  phone VARCHAR(12),
  class_name VARCHAR(20) NOT NULL,
  major VARCHAR(50) NOT NULL,
  enrollment_year INT NOT NULL,
  status VARCHAR(20) DEFAULT 'ACTIVE',
  created_date DATE DEFAULT CURRENT_DATE
);
```

Bảng 2: SCORES (Điểm số)

```
sql
CREATE TABLE SCORES (
  score_id INT AUTO_INCREMENT PRIMARY KEY,
  student_id VARCHAR(8) NOT NULL,
  subject_name VARCHAR(50) NOT NULL,
  score DECIMAL(3,1) NOT NULL,
  semester VARCHAR(10) NOT NULL,
  exam_date DATE NOT NULL,
  score_type VARCHAR(20) DEFAULT 'FINAL',
  notes TEXT,
  FOREIGN KEY (student_id) REFERENCES STUDENTS(student_id)
);
```

Dữ liệu mẫu đã được chuẩn bị sẵn với các ngành: IT, Business, Design, Marketing

Yêu cầu thực hiện:

Phần A: Backend - Servlet và Database (5.0 điểm)

1. StudentServlet (2.5 điểm)

URL Pattern: `/students`

Chức năng:

- **GET /students?action=list** (0.8 điểm)
 - Hiển thị danh sách tất cả sinh viên với phân trang (10 sinh viên/trang)
 - Hỗ trợ tìm kiếm theo tên hoặc mã sinh viên
 - Lọc theo ngành học (major)
 - Forward đến `student-list.jsp`
- **GET /students?action=detail&id=xxx** (0.7 điểm)
 - Hiển thị thông tin chi tiết 1 sinh viên
 - Kèm theo danh sách điểm các môn học
 - Forward đến `student-detail.jsp`
- **POST /students?action=add** (1.0 điểm)
 - Thêm sinh viên mới
 - Validate: email không trùng, enrollment_year >= 2020
 - Tự động generate student_id theo format: năm nhập học + 4 số (VD: 20240001)
 - Redirect về danh sách sau khi thêm thành công

2. ScoreServlet (2.5 điểm)

URL Pattern: `/scores`

Chức năng:

- **GET /scores?action=form&studentId=xxx** (0.8 điểm)
 - Hiển thị form nhập điểm cho sinh viên được chọn
 - Load thông tin sinh viên để hiển thị
 - Forward đến `score-form.jsp`
- **POST /scores?action=add** (1.2 điểm)
 - Thêm điểm mới cho sinh viên
 - Validate: điểm từ 0.0 đến 10.0

- Kiểm tra sinh viên có tồn tại không
- Insert vào database
- Redirect với thông báo thành công
- **GET /scores?action=transcript&studentId=xxx (0.5 điểm)**
 - Hiển thị bảng điểm của 1 sinh viên
 - Tính GPA trung bình
 - Forward đến `transcript.jsp`

Phần B: Frontend - JSP và Giao diện (5.0 điểm)

1. student-list.jsp (1.5 điểm)

Yêu cầu:

- Header với title và menu navigation (0.3 điểm)
- Form tìm kiếm: textbox tên/MSSV và dropdown ngành học (0.4 điểm)
- Bảng hiển thị: MSSV, Họ tên, Lớp, Ngành, Năm nhập học, Trạng thái (0.5 điểm)
- Nút "Xem chi tiết" và "Nhập điểm" cho mỗi sinh viên (0.3 điểm)

2. student-detail.jsp (1.0 điểm)

Yêu cầu:

- Hiển thị đầy đủ thông tin sinh viên trong card đẹp (0.4 điểm)
- Bảng điểm các môn đã học (nếu có) (0.3 điểm)
- Nút "Nhập điểm mới" và "Xem bảng điểm" (0.3 điểm)

3. score-form.jsp (1.0 điểm)

Yêu cầu:

- Form nhập điểm: Tên môn, Điểm, Học kỳ, Ngày thi, Loại điểm, Ghi chú (0.5 điểm)
- Hiển thị thông tin sinh viên (readonly) (0.2 điểm)
- Validation JavaScript: điểm 0-10, ngày không được tương lai (0.3 điểm)

4. add-student.jsp (1.0 điểm)

Yêu cầu:

- Form thêm sinh viên: Tên, Email, SĐT, Lớp, Ngành, Năm nhập học (0.5 điểm)

- Dropdown ngành học và năm nhập học (0.3 điểm)
- Validation form với required fields (0.2 điểm)

5. transcript.jsp (0.5 điểm)

Yêu cầu:

- Bảng điểm đẹp với các môn học và điểm số (0.3 điểm)
- Hiển thị GPA ở cuối bảng (0.2 điểm)

Phần C: Báo cáo đơn giản (2.0 điểm)

Yêu cầu: Tạo trang thống kê cơ bản (statistics.jsp)

Câu truy vấn đơn giản cần thực hiện: (1.5 điểm)

```
sql

-- Thống kê số lượng sinh viên và điểm trung bình theo ngành
SELECT
    s.major,
    COUNT(DISTINCT s.student_id) as total_students,
    COUNT(sc.score_id) as total_scores,
    ROUND(AVG(sc.score), 2) as average_score,
    MAX(sc.score) as highest_score,
    MIN(sc.score) as lowest_score,
    COUNT(CASE WHEN sc.score >= 8.0 THEN 1 END) as excellent_count,
    COUNT(CASE WHEN sc.score >= 6.5 AND sc.score < 8.0 THEN 1 END) as good_count,
    COUNT(CASE WHEN sc.score < 5.0 THEN 1 END) as fail_count
FROM STUDENTS s
LEFT JOIN SCORES sc ON s.student_id = sc.student_id
WHERE s.status = 'ACTIVE'
GROUP BY s.major
ORDER BY total_students DESC;
```

Hiển thị trong JSP: (0.5 điểm)

- Tạo StatisticsServlet để xử lý truy vấn trên
- Hiển thị kết quả dưới dạng bảng thống kê
- Thêm biểu đồ cột đơn giản (có thể dùng Chart.js hoặc chỉ CSS)

Cấu hình và yêu cầu kỹ thuật:

Database: MySQL 8.0+

Application Server: Apache Tomcat 10.x

Technologies: Java Servlet 5.0, JSP 3.0, JSTL, Bootstrap 5

Cấu trúc project đơn giản:

```
src/
├── main/
│   ├── java/
│   │   ├── controller/
│   │   │   ├── StudentServlet.java
│   │   │   ├── ScoreServlet.java
│   │   │   └── StatisticsServlet.java
│   │   ├── model/
│   │   │   ├── Student.java
│   │   │   └── Score.java
│   │   ├── dao/
│   │   │   ├── StudentDAO.java
│   │   │   └── ScoreDAO.java
│   │   └── util/
│   │       └── DBConnection.java
│   └── webapp/
│       ├── WEB-INF/
│       │   └── web.xml
│       ├── css/
│       │   └── style.css
│       ├── js/
│       │   └── validation.js
│       └── views/
│           ├── student-list.jsp
│           ├── student-detail.jsp
│           ├── add-student.jsp
│           ├── score-form.jsp
│           ├── transcript.jsp
│           └── statistics.jsp
```

Sample Data:

- **Majors:** Information Technology, Business Administration, Graphic Design, Digital Marketing
- **Classes:** IT01, IT02, BUS01, BUS02, DES01, MKT01
- **Subjects:** Java Programming, Database, Web Development, Math, English
- **Score Types:** MIDTERM, FINAL, ASSIGNMENT, QUIZ

Validation Rules:

- Student ID: auto-generated (không cần nhập)
- Email: format hợp lệ và unique
- Score: từ 0.0 đến 10.0
- Enrollment Year: ≥ 2020

Thời gian: 150 phút (giảm 30 phút so với bài khó) **Nộp bài:** Source code + WAR file + Database script

Tiêu chí đánh giá:

- Backend servlet và database: 50%
- Frontend JSP: 40%
- Báo cáo thống kê: 10%

Điều kiện pass:

- Ít nhất 7/12 điểm để pass
- Phải có ít nhất 1 chức năng CRUD hoạt động
- Database connection thành công

Gợi ý cho sinh viên:

- Bắt đầu từ database connection
- Làm chức năng đơn giản trước (list students)
- Test từng chức năng trước khi làm tiếp
- Giao diện đơn giản nhưng phải rõ ràng, dễ sử dụng