Web Application Development

Lab 5: SERVLET & MVC PATTERN

Name: Dang Thanh Lam – ITITDK23039

Part A: In class exercises

Demonstrations:

Model Layer

1. Student JavaBean follows conventions
    - The Student class encapsulates all student properties with private fields (id, studentCode, fullName, email, major, createdAt).
    - It provides public no-arg and parameterized constructors.
    - Getters and setters allow controlled access.
    - toString() is overridden for meaningful object representation.
    - This ensures encapsulation and JavaBean compliance.
2. StudentDAO has all CRUD methods
    - DAO implements database operations:
        - getAllStudents() – fetches all students.
        - getStudentById(int id) – retrieves a single student.
        - insertStudent(Student s) – adds a new student.
        - updateStudent(Student s) – modifies an existing student.
        - deleteStudent(int id) – removes a student.
    - Uses try-with-resources for safe database connection handling.
    - Exceptions are caught and logged, ensuring robust error handling.

3. Database operations work correctly
    - All CRUD operations interact with MySQL database successfully.
    - Test sequence demonstrates creation, retrieval, update, and deletion of student records.
    - Timestamps (createdAt) are automatically recorded for new students.

Controller Layer

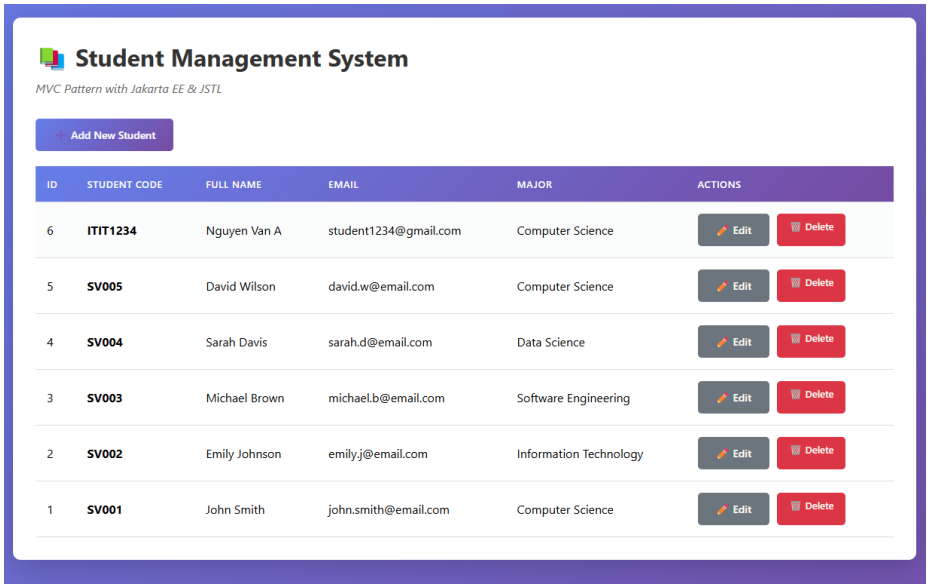
1. Servlet properly annotated
    - StudentController is annotated with @WebServlet("/student"), making it accessible at the specified URL.
2. Routes requests correctly
    - doGet() routes based on the action parameter (list, new, edit, delete).
    - doPost() handles insert and update actions.
3. Calls DAO methods
    - Controller uses StudentDAO to perform all data operations, keeping business logic separate from presentation.
4. Sets request attributes
    - Attributes like studentList and student are set before forwarding to JSP.
    - This allows JSP to access data through Expression Language (EL).
5. Forwards/redirects appropriately
    - GET actions like list, new, and edit are forwarded to JSP views.
    - POST actions like insert and update are redirected to /student?action=list to avoid form resubmission.

View Layer

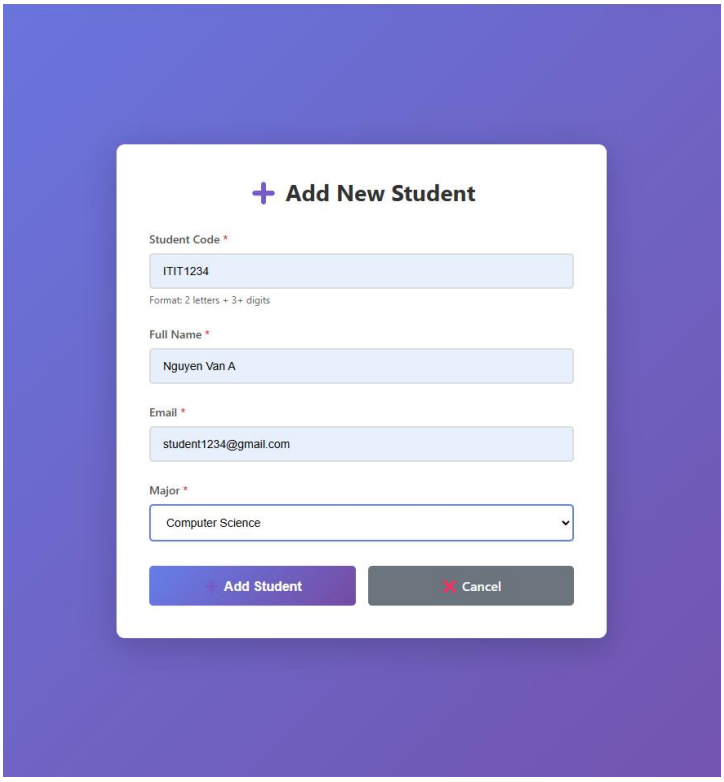
1. No scriptlets in JSP
    - All JSP pages are scriptlet-free, using JSTL and EL to render dynamic content.
2. JSTL tags used correctly
    - Tags like <c:forEach> iterate over student lists.
    - Conditional rendering uses <c:if> to display messages or empty states.
3. EL for all data access
    - All dynamic content is accessed via ${studentList} or ${student.field}.
    - This separates presentation from logic, adhering to MVC.
4. Single form for add/edit works
    - student-form.jsp is used for both adding and editing students.
    - Form fields are pre-filled when editing.
    - Action changes dynamically (insert or update) based on context.

Functionality

1. **List students**
    - Navigating to /student displays all students in a table.
    - If no students exist, shows "No students found."



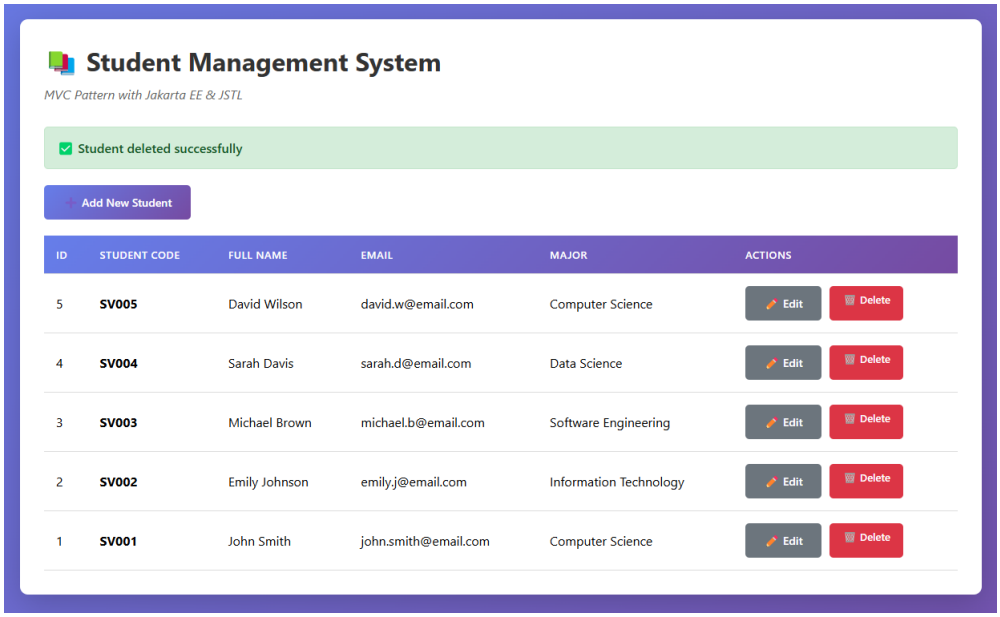
2. **Add student**
    - Clicking "Add New Student" opens the form.
    - Submitting inserts a new record and redirects to the list.
    - Success message is displayed.



3. **Edit student**

- Clicking "Edit" on a student pre-fills the form.
- Modifying and submitting updates the record.
- Update message is displayed.

4. **Delete student**
    - Clicking "Delete" removes the record.
    - Redirects to the list with a deletion message.
    - Handles empty state gracefully when all students are removed.



5. **Messages display**
    - Confirmation, success, and empty state messages are displayed using JSTL and EL.
    - Ensures clear user feedback for all operations.