Laboratory Activity 4: JavaScript Fundamentals Part 2

Learning Outcomes

By the end of this activity, students will be able to:

1. Create and manipulate arrays in JavaScript (CRUD, search, filter, sort).
2. Define and use objects with properties and methods (including computed outputs).
3. Perform DOM manipulation to render lists, update content, and respond to user actions.
4. Integrate arrays/objects into a simple Information System (IS) user interface.
5. Document and demonstrate their solution through an unlisted YouTube video walkthrough.

Activity Instructions

- Choose one IS context for all problems:
    - A. Schedule IS (class scheduling system)
    - B. Library IS (books and borrowers' information system)
    - C. Product Inventory IS (item reservations system)
- Code in HTML + JavaScript, with visible output rendered in the browser.
- Use comments to explain code (purpose + student learning).
- Submit a text file containing the GitHub repo link and an unlisted YouTube link of your explanation.

Problem Set

Problem 1 – Arrays (Dynamic Records in IS Context)

1. Initialize an empty array to store records.
2. Allow the user (via form input) to add at least 5 records (e.g., classes, books, or items).
3. Implement the following operations:
    - Add a record (via form).
    - Update a record by searching for a unique field (e.g., code, title, or itemName).
    - Delete a record with confirmation.
    - Filter records by a meaningful rule (classes on Friday, books after 2015, items below stock threshold).
    - Sort records by one numeric and one string field.

4. Write a custom search function that accepts partial keywords (typing "Ma" matches "Math 101" or "Mango").
5. Show results both in the console and in the DOM (HTML list or table).

Problem 2 – Objects (Domain Model with Behavior)

Define two interconnected objects relevant to your chosen IS:

- Schedule IS Example:
    - student object { name, id, enrolledClasses: [] }.
    - class object { code, day, time }.
    - Methods:
        - addClass(classObj) rejects if class conflicts in time.
        - listClasses() returns all enrolled classes in readable format.
- Library IS Example:
    - borrower object { name, id, borrowed: [] }.
    - book object { title, author, year, isAvailable }.
    - Methods:
        - borrow(bookObj) rejects if isAvailable = false.
        - returnBook(title) removes from borrowed list and toggles availability.
- Inventory IS Example:
    - cart object { items: [] }.
    - item object { name, qty, price }.
    - Methods:
        - add(itemObj, qty) rejects if qty exceeds stock.
        - checkout() returns total with tax (12%) and applies discount if > ₱5000.

At least one method must return a human-readable report string, not just raw values.

Problem 3 – DOM (Interactive Information System UI)

Build an interactive web interface:

1. UI Elements:

- Heading (<h2>) showing your chosen IS.
- A form with at least 3 inputs for adding new records.
- Buttons: Add, Filter, Delete, Sort, Reset.
- A container (<div id="list"></div>) to display records.
- A summary section (<div id="summary"></div>) to show counts, totals, or reports.

2. Functional Requirements:
   - On Add, validate inputs, insert record into array, re-render list.
   - On Filter, apply chosen filter and re-render.
   - On Delete, confirm before removing record.
   - On Sort, toggle ascending/descending.
   - Implement live search with partial keywords.
   - Highlight special cases in red (e.g., overdue book, class after 6 PM, zero stock).
   - Provide a toggle view (table view vs. card view).

Expected Output

1. A working HTML + JS project (Lab4_<LastName>.html or folder with index.html + app.js).
2. Console logs for array/object operations.
3. A dynamic DOM-rendered list/table that updates with actions.
4. A text file containing:
   - GitHub repo link.
   - Link of Unlisted YouTube video (5–8 minutes) where you explain:
     - Your chosen IS and why,
     - How arrays and objects power your system,
     - How DOM events update the UI,
     - A demo of Add → Filter → Delete → Sort → Summary,
     - One challenge/bug you solved and how you fixed it.