

FINAL PROJECT ON JAVASCRIPT

Project Title

Student Management System

Project Description

You are required to create a **console-based Student Management System** using **JavaScript**. The system will keep track of multiple students' details and allow the user to perform operations such as adding, viewing, searching, updating, deleting, and evaluating student performance.

The project must demonstrate the use of **Variables, Functions, If-Else statements, Switch statements, Objects, and Try-Catch error handling**.

Project Requirements

1. Add a Student

- The user should be able to enter details such as:
 - Student ID
 - Name
 - Age
 - Grade (numeric)
- The program should **prevent duplicate IDs** and handle this error using **try-catch**.

2. View All Students

- Display a list of all students with their details in a clear format.
- If no students are found, display an appropriate message.

3. Search for a Student

- Allow the user to search by Student ID.
- Display the student's details if found, otherwise show a message that the student is not found.

4. Update Student Details

- Allow updating of one or more fields (name, age, grade) of an existing student.
- If the student is not found, display an error message.

5. Delete a Student

- Remove a student's record using their ID.
- Confirm if deletion was successful.

6. Evaluate Student Performance

- Using **if-else**:
 - $\text{Grade} \geq 90 \rightarrow \text{Excellent}$
 - $\text{Grade} \geq 75 \text{ and } < 90 \rightarrow \text{Good}$
 - $\text{Grade} \geq 50 \text{ and } < 75 \rightarrow \text{Pass}$
 - $\text{Grade} < 50 \rightarrow \text{Fail}$
 - Using **switch**:
 - Show the same evaluation in a different format (e.g., with emojis or descriptive words).
-

Technical Requirements

- **Variables**: Store information like arrays of students, IDs, grades, etc.
 - **Functions**: Each operation (add, view, search, update, delete, evaluate) should be a separate function.
 - **If-Else Statements**: For decision-making, especially grade evaluation and checking if a student exists.
 - **Switch Statements**: For alternative grade evaluation messages.
 - **Objects**: Store each student's details as an object with properties such as `id`, `name`, `age`, `grade`.
 - **Try-Catch**: Handle potential errors such as duplicate IDs or invalid inputs.
-

Example Workflow

1. Add three students to the system.
2. View all students to confirm they were added.
3. Search for one student by ID.
4. Update that student's grade and name.
5. Evaluate the student's performance using both **if-else** and **switch**.
6. Delete one student.
7. View all students to confirm deletion.

Expected Output

- Clear console messages showing the results of each action.
- Proper handling of errors (e.g., duplicate ID, student not found).
- Organized display of student records.

DYNASOFTY JAVASCRIPT