+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

🧠 Practical Task: "Student Performance Analyzer"

Scenario:

You have a list of student objects. Each student has a name, scores for different subjects, and a boolean indicating if they have submitted all assignments.

```
const students = [

 { name: "Alice", math: 85, english: 78, science: 92, submitted: true },

 { name: "Bob", math: 45, english: 52, science: 58, submitted: false },

 { name: "Charlie", math: 95, english: 88, science: 91, submitted: true },

 { name: "David", math: 66, english: 70, science: 60, submitted: true },

 { name: "Eva", math: 50, english: 49, science: 45, submitted: false },

];
```

✅ Tasks

1. Filter out only students who submitted all assignments

Use `filter()` to get only those students who have `submitted: true`.

```
const submittedStudents = students.filter(({ submitted }) => submitted);

console.log(submittedStudents);
```

---

2. Map the filtered students to calculate their average score

Use `map()` and **object destructuring** to calculate the average of `math`, `english`, and `science` scores.

```
const studentAverages = submittedStudents.map(({ name, math, english, science }) => {
```

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

```
  const average = ((math + english + science) / 3).toFixed(2);

  return { name, average: Number(average) };

});

console.log(studentAverages);
```

---

3. Filter top performers (average ≥ 80)

```
const topPerformers = studentAverages.filter(({ average }) => average >= 80);

console.log(topPerformers);
```

🧪 Bonus Task: Create a report summary

Use `map()` and destructuring to generate a string like:

"Alice scored an average of 85.00"

```
const summary = studentAverages.map(({ name, average }) => `${name} scored an average of ${average}`);

console.log(summary);
```

🧩 Concepts Covered

- `filter()` for condition-based selection
- `map()` for transforming objects
- Object destructuring in parameters
- Template literals

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

🔰 Basic Level

1. Filter students who passed all subjects (>= 40 marks each).

- Use `filter`.
- Return array of student names who passed all subjects.

---

2. List names of students who did NOT submit their assignment.

- Use `filter` and `map`.

---

3. Print average marks of each student.

- Use `map` to return name and average.
- Output example: `{ name: "Alice", average: 85 }`

---

4. Find the topper in science subject.

- Use `reduce` to find the student with highest science marks.

---

5. Count how many students submitted their assignments.

- Use `filter` or `reduce`.

⚙️ Intermediate Level

6. Sort students by total marks (descending).

- Use `sort`.
- Show names with total marks.

---

7. Find names of students who failed in at least one subject.

- Use `filter`.
- Failed = marks < 40 in any subject.

---

8. Group students into 'Passed' and 'Failed' categories.

Use `reduce` to build an object like:

```
{
  passed: ["Alice", "Charlie", "David"],
  failed: ["Bob", "Eva"]
}
```

9. Get list of students with subject-wise grades (A, B, C, F).

- Use `map`.
- Assign grades per subject:
  - A: 80+, B: 60–79, C: 40–59, F: <40

---

10. Generate a result summary report:

- Total students
- Students submitted
- Students failed
- Average marks per subject