🧠 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 }) => {
  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