Task 1: Dynamic Student Report Card Using Destructuring and Template Literals

Problem Statement:
You are given a student object that contains the details of a student including name, roll number, and marks in 5 subjects.

Using object and array destructuring and template literals, write a program to generate a formatted report card in the console.

Your program should:
- Calculate the total and percentage.
- Use destructuring to extract the marks.
- Use template literals to display a multi-line summary like:

Report Card for Riya (Roll No: 101)

Subjects: Math: 85, Science: 90, English: 78, History: 88, Art: 92
Total Marks: 433
Percentage: 86.6%

Hint: You must decide the structure of the object and choose how to destructure it effectively.

Task 2: Build a Custom Greeting Generator with Default Parameters and Arrow Functions
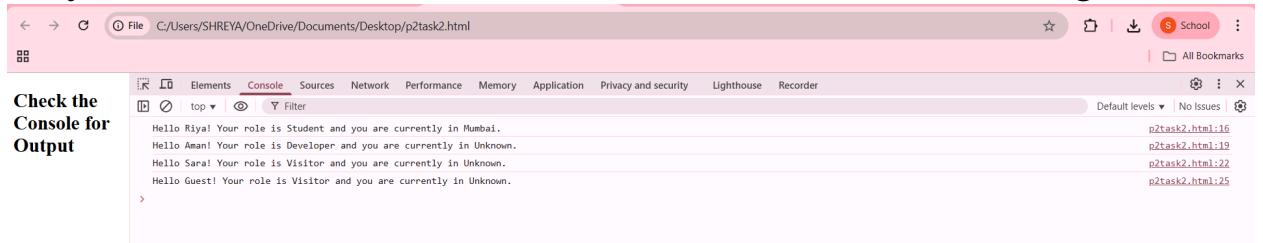
Problem Statement:
- Write a function named generateGreeting that:

- Takes in name, role, and location as parameters (all with default values).
- Uses an arrow function to return a greeting like:

"Hello [name]! Your role is [role] and you are currently in [location]."

Requirements:
- Do not use function keyword.
- Use template literals for message construction.
- Call the function with different combinations of arguments and no arguments.
- Only one function should handle all variations without breaking.

Task 3: Implement a Data Type Analyzer (without typeof)
Problem Statement:

Create a function analyzeData that takes any input and prints:
- What kind of primitive or reference type it is (without using typeof).
- Whether it's mutable or immutable.
- A summary message.

You cannot use typeof, but must still determine if it's:
- string, number, boolean, undefined, null, symbol, bigint
- array, object, function

You must figure out creative ways to determine the type, such as using constructor checks, Array.isArray, etc.

Elements   Console   Sources   Network   Performance   Memory   Application   Privacy and security   Lighthouse   Recorder   ⚙   ⋮   ✕

⊞ ⊘   top ▾   ⊙   ▽ Filter   Default levels ▾ | No Issues ⚙

```
Analysis Result:                                                                                                    p2task3.html:57
 - Type: string
 - Mutable: No
 - Summary: The input is a string and it is immutable.
Analysis Result:                                                                                                    p2task3.html:57
 - Type: number
 - Mutable: No
 - Summary: The input is a number and it is immutable.
Analysis Result:                                                                                                    p2task3.html:57
 - Type: boolean
 - Mutable: No
 - Summary: The input is a boolean and it is immutable.
Analysis Result:                                                                                                    p2task3.html:57
 - Type: undefined
 - Mutable: No
 - Summary: The input is a undefined and it is immutable.
Analysis Result:                                                                                                    p2task3.html:57
 - Type: null
 - Mutable: No
 - Summary: The input is a null and it is immutable.
Analysis Result:                                                                                                    p2task3.html:57
 - Type: symbol
 - Mutable: No
 - Summary: The input is a symbol and it is immutable.
Analysis Result:                                                                                                    p2task3.html:57
 - Type: bigint
 - Mutable: No
 - Summary: The input is a bigint and it is immutable.
Analysis Result:                                                                                                    p2task3.html:57
 - Type: array
 - Mutable: Yes
 - Summary: The input is a array and it is mutable.
Analysis Result:                                                                                                    p2task3.html:57
 - Type: object
 - Mutable: Yes
 - Summary: The input is a object and it is mutable.
```

Task 4: Color Mixer using Destructuring and Spread/Rest Operators

Problem Statement:
Write a program that:
- Takes two arrays of colors from the user (e.g., ["red", "green"] and ["blue", "yellow"]).
- Merges the arrays using the spread operator into a new array called palette.
- Extract the first and last colors from the palette using array destructuring.
- Collect the middle colors using the rest operator.
- Display a message like:

"Main colors: Red and Yellow. Others in the palette: Green, Blue."

You are not allowed to use loops or array indexing manually.

Main colors: Red and Yellow. Others in the palette: Green, Blue.