Name : Madhina Banu N

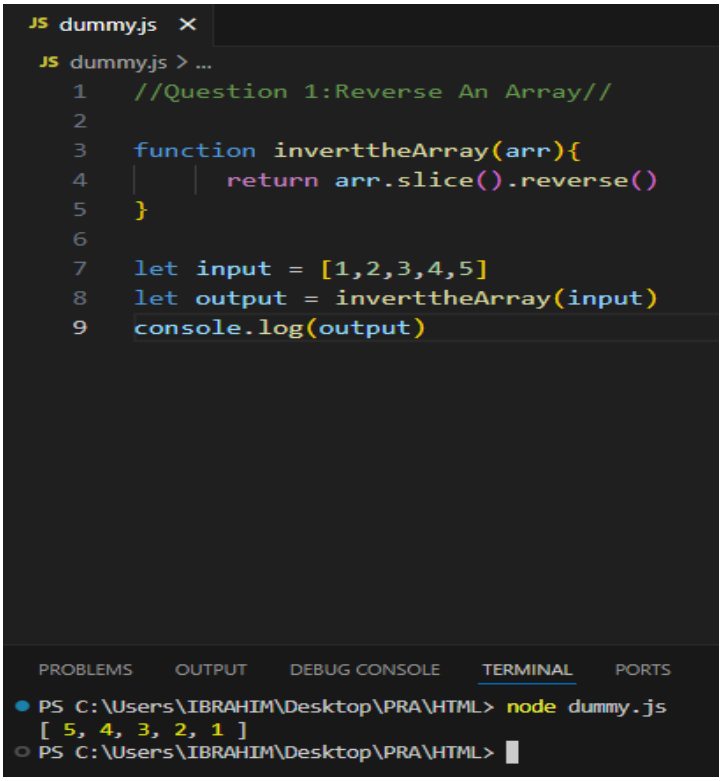
Week – 6 JS Assignment Questions And Solutions

Question 1: Reverse an Array

Problem: Write a function that takes an array and returns a new array with the elements in reverse order.

Let input = [1,2,3,4,5]

Let output = [5,4,3,2,1]



Solution - Brief Explanation:

- Function Definition: The function inverttheArray is defined to take one parameter, arr, which is the input array.
- Loop: It uses a loop that starts from the last element of the array and goes to the first element.
- Pushing Elements: Each element is pushed into a new array called reversedArr.
- Return: Finally, the function returns the new array that contains the elements in reverse order.

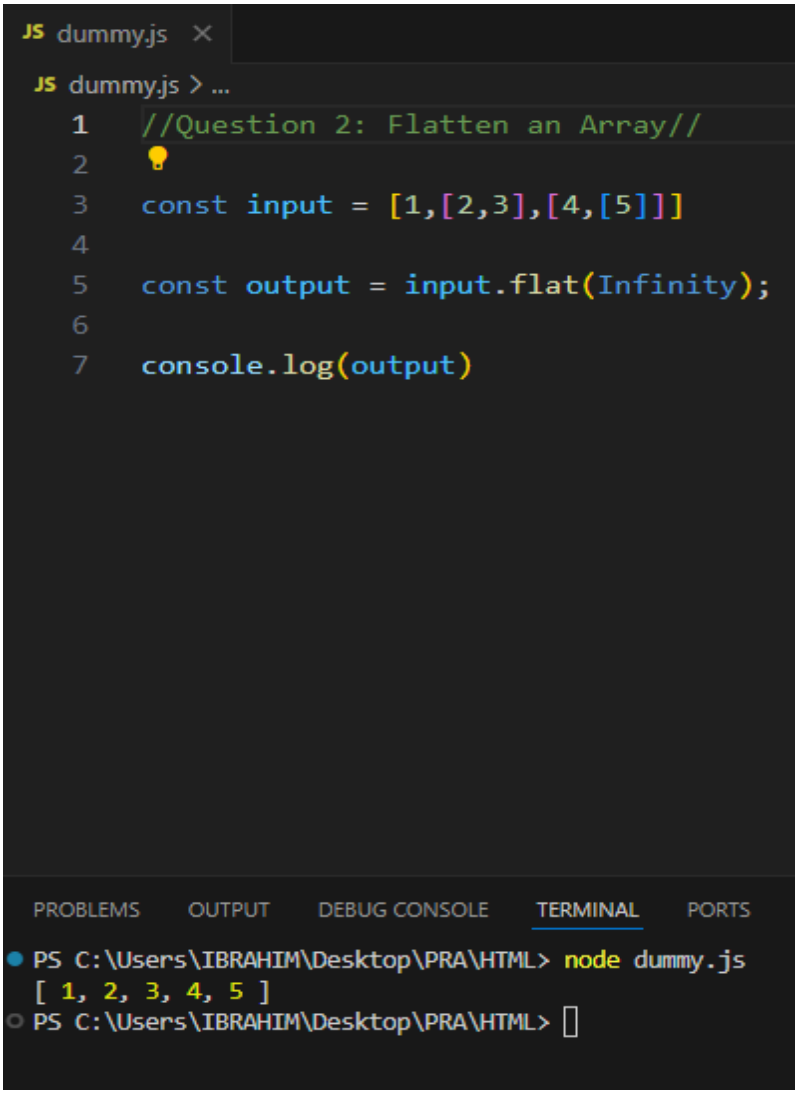
  Output: [5, 4, 3, 2, 1]

Question 2: Flatten an Array

Problem: Write a function that takes a nested array and flattens it to a single-level array.

Input: [1, [2, 3], [4, [5]]]

Output: [1, 2, 3, 4, 5]



Solution - Brief Explanation:

- Using flat(): The flat() method simplifies the process by flattening the array to the specified depth (in this case, Infinity means all levels).

  Output: [1, 2, 3, 4, 5]

Question 3: Check for Duplicates

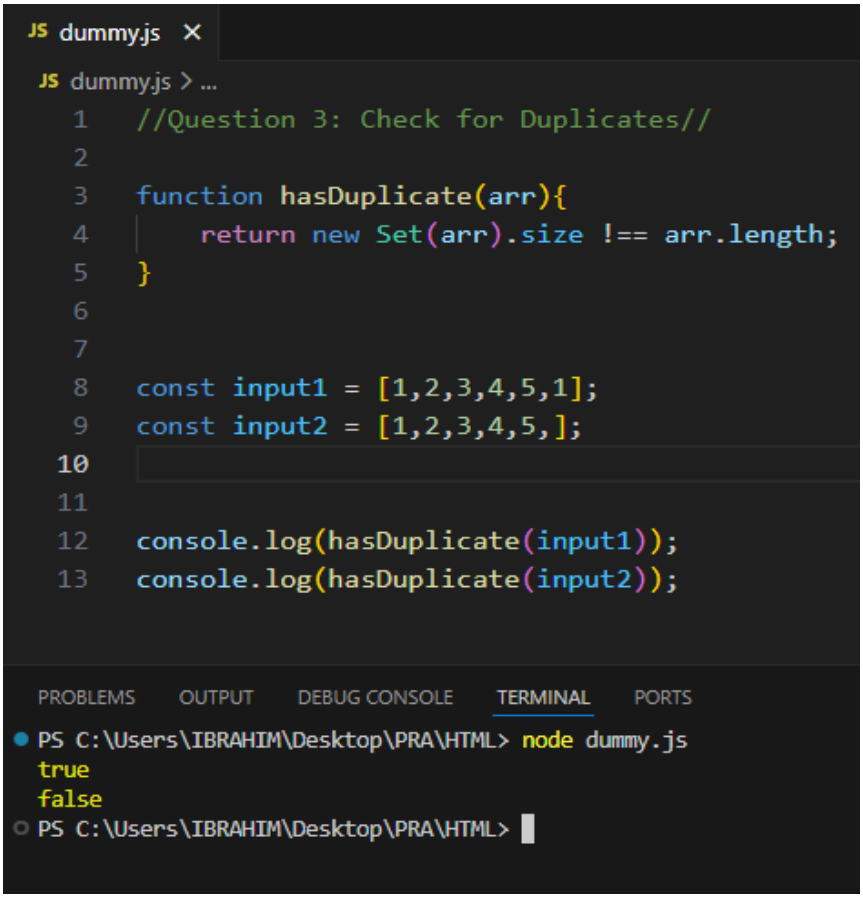
Problem: Write a function that checks if an array contains duplicates.

Input: [1, 2, 3, 4, 5, 1]

Output: true

Input: [1, 2, 3, 4, 5]

Output: false



Solution - Brief Explanation:

- new Set(arr) creates a Set from the array arr. A Set only stores unique values, so if there are duplicates in arr, the Set will have fewer elements than arr.
- Set.size returns the number of unique elements in the Set.
- arr.length returns the number of elements in the original array.
- By comparing new Set(arr).size with arr.length, you can determine if there are duplicates:

- If the sizes are different, there are duplicates, and the function returns true.
- If the sizes are the same, there are no duplicates, and the function returns false.

Output: true

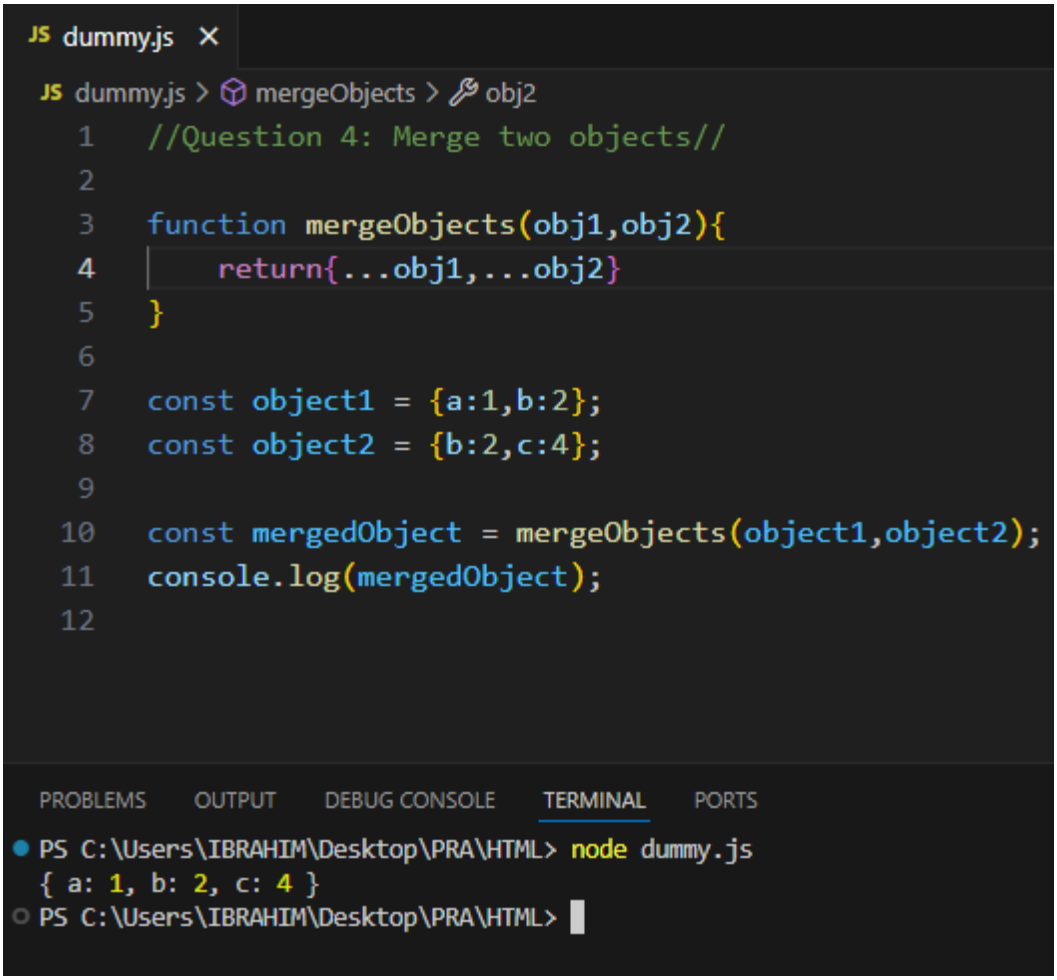
Output: true

Question 4: Merge Two Objects

Problem: Write a function that merges two objects into one.

Input: { a: 1, b: 2 }, { b: 2, c: 4 }

Output: { a: 1, b: 2, c: 4 }



Solution - Brief Explanation:

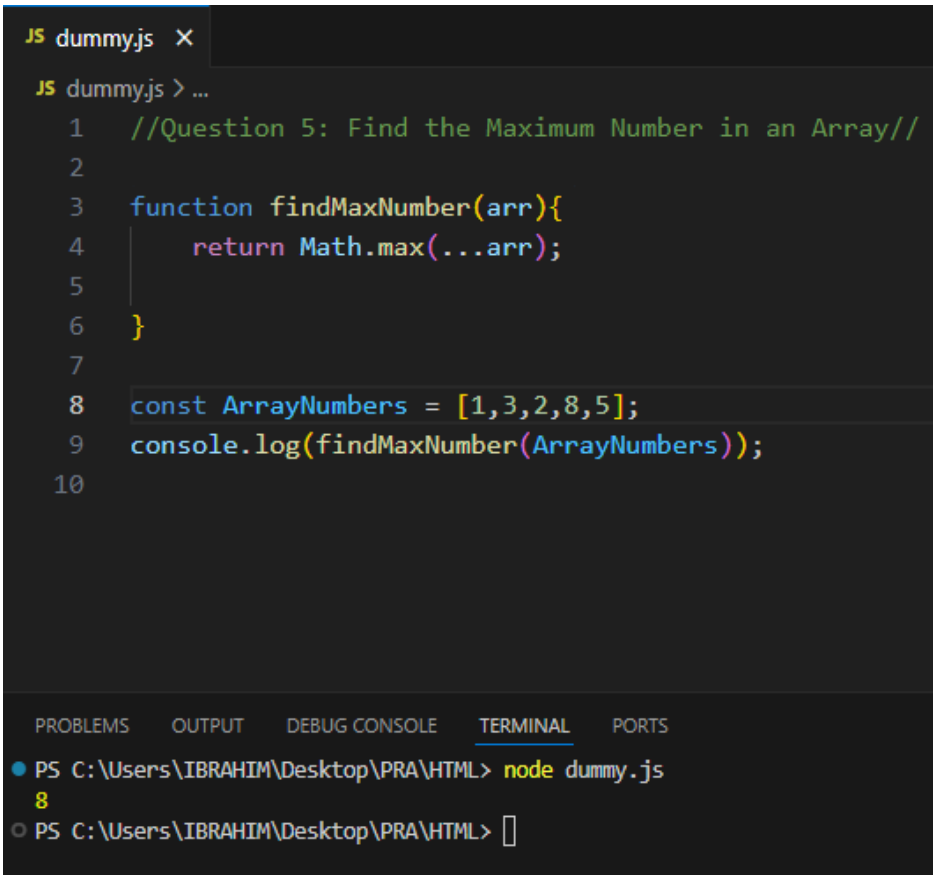
- The ... (spread) operator is used to merge the properties of obj1 and obj2.
- If there are overlapping keys (like b in this example), the value from obj2 will override the value from obj1.
- Output: { a: 1, b: 2, c: 4 }

Question 5: Find the Maximum Number in an Array

Problem: Write a function that finds the maximum number in an array.

Input: [1, 3, 2, 8, 5]

Output : 8



Solution - Brief Explanation:

- Math.max(...arr) uses the spread operator ... to spread the elements of arr and pass them individually to Math.max(), which returns the largest value.
- const ArrayNumbers = [1,3,2,8,5]; defines the array of numbers.
- console.log(findMaxNumber(ArrayNumbers)); calls the findMaxNumber function and logs the maximum number to the console.

Output: 8

Question 6: Group Array of Objects by Property

Problem: Write a function that groups an array of objects by a specific property.

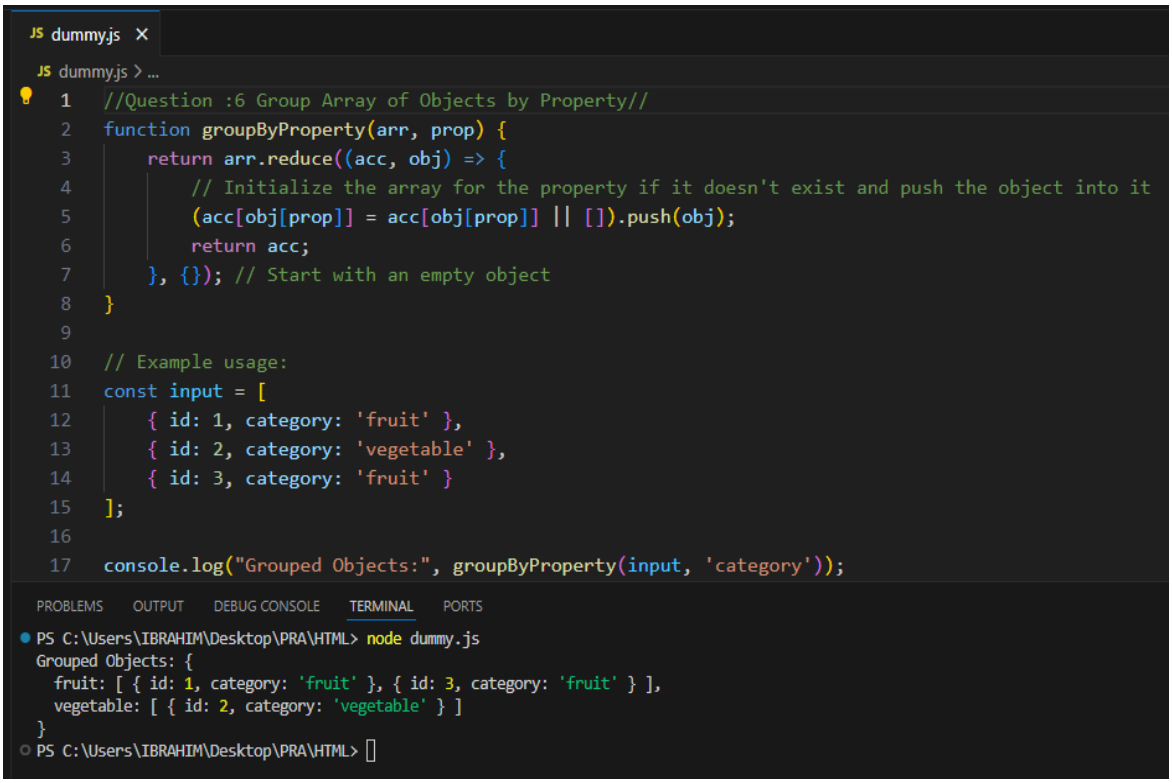
Input: [{ id: 1, category: 'fruit' }, { id: 2, category: 'vegetable' }, { id: 3, category: 'fruit' }]

Output: {

fruit: [{ id: 1, category: 'fruit' }, { id: 3, category: 'fruit' }],

vegetable: [{ id: 2, category: 'vegetable' }]

}



Solution - Brief Explanation:

Purpose: The function groups an array of objects based on a specified property.

Parameters: arr: The array of objects to be grouped.

prop: The property name used for grouping.

Using reduce:

reduce is a method used to iterate over each object in arr and accumulate results in an object.

Logic:

(acc[obj[prop]] = acc[obj[prop]] || []):

This line checks if acc[obj[prop]] (the array for the specific property) already exists.

If it doesnt, it initializes it as an empty array ([]).

.push(obj): Adds the current object obj to the appropriate array.

5. Result:

The objects are grouped into arrays based on the value of prop, and the final grouped object is returned.

Output:

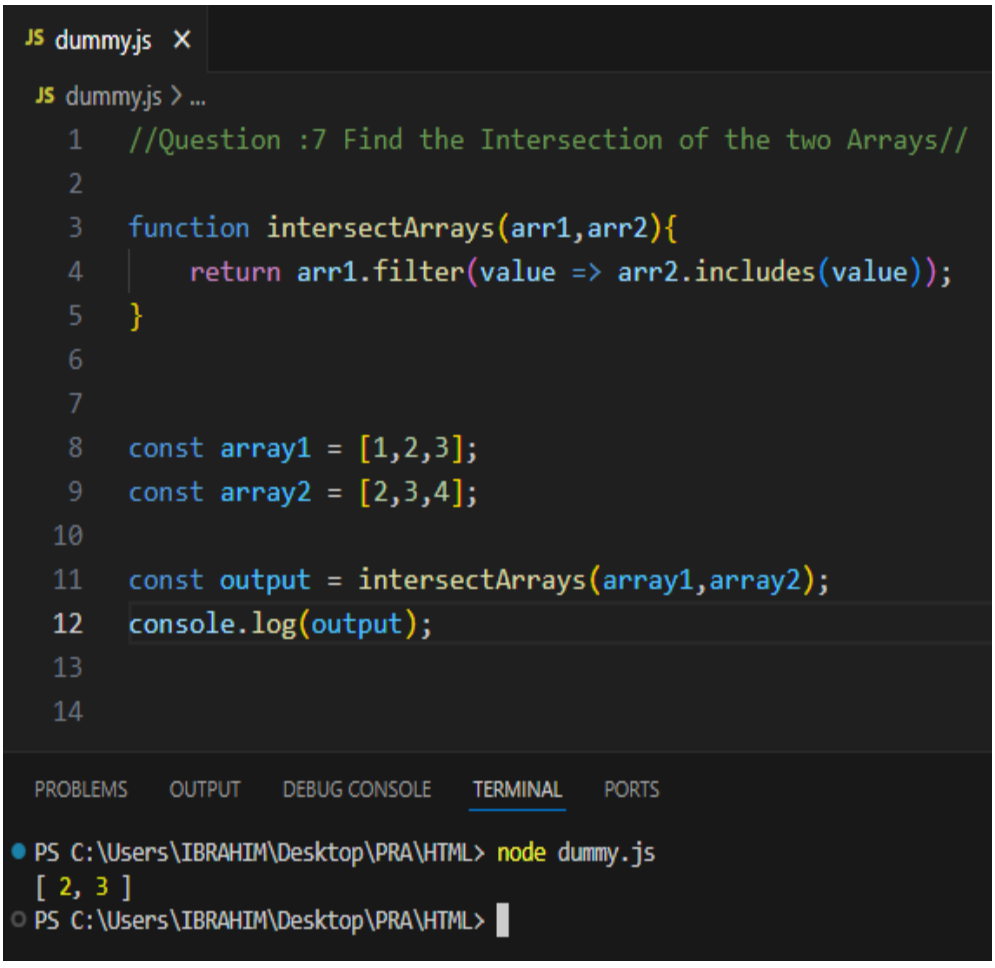
```
{
  fruit: [ { id: 1, category: 'fruit' }, { id: 3, category: 'fruit' } ],
    vegetable: [ { id: 2, category: 'vegetable' } ]
    }
```

Question 7: Find the Intersection of Two Arrays

Problem: Write a function that returns the intersection of two arrays.

Input: [1, 2, 3], [2, 3, 4]

Output: [2, 3]



Solution - Brief Explanation:

- arr1 is [1, 2, 3].
- filter starts by looking at the first element of arr1, which is 1.

It checks arr2.includes(1), which is false because 1 is not in arr2 ([2, 3, 4]). So, 1 is not included in the new array.

- Then it checks the second element of arr1, which is 2.

It checks arr2.includes(2), which is true because 2 is in arr2. So, 2 is included in the new array.

- Finally, it checks the third element of arr1, which is 3.

It checks arr2.includes(3), which is true because 3 is also in arr2. So, 3 is included in the new array.

- The filter method then returns the new array [2, 3].

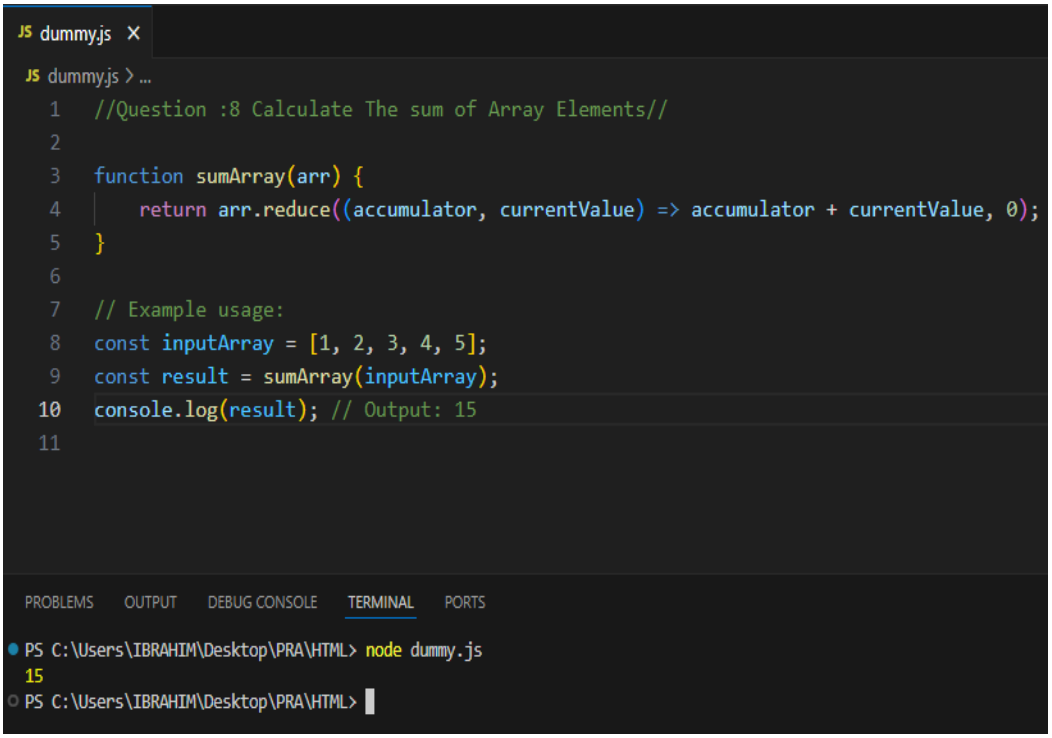
Output :[2,3]

Question 8: Calculate the Sum of Array Elements

Problem: Write a function that calculates the sum of all numbers in an array.

Input: [1, 2, 3, 4, 5]

Output: 15



Solution - Brief Explanation:

- Function Definition:The function sumArray takes one parameter, arr, which is the array of numbers.
- Using reduce:arr.reduce(...) goes through each element of the array and applies the provided function to accumulate a result.

- The function (accumulator, currentValue) => accumulator + currentValue takes two parameters:
- accumulator: This keeps track of the ongoing sum.
- currentValue: This represents the current element being processed from the array.The 0 at the end is the initial value for the accumulator, starting the sum from zero.

Step-by-Step for the Example Input

1. Start with accumulator = 0.

2. Add the first element (1): 0 + 1 = 1.

3. Add the second element (2): 1 + 2 = 3.

4. Add the third element (3): 3 + 3 = 6.

5. Add the fourth element (4): 6 + 4 = 10.

6. Add the fifth element (5): 10 + 5 = 15.

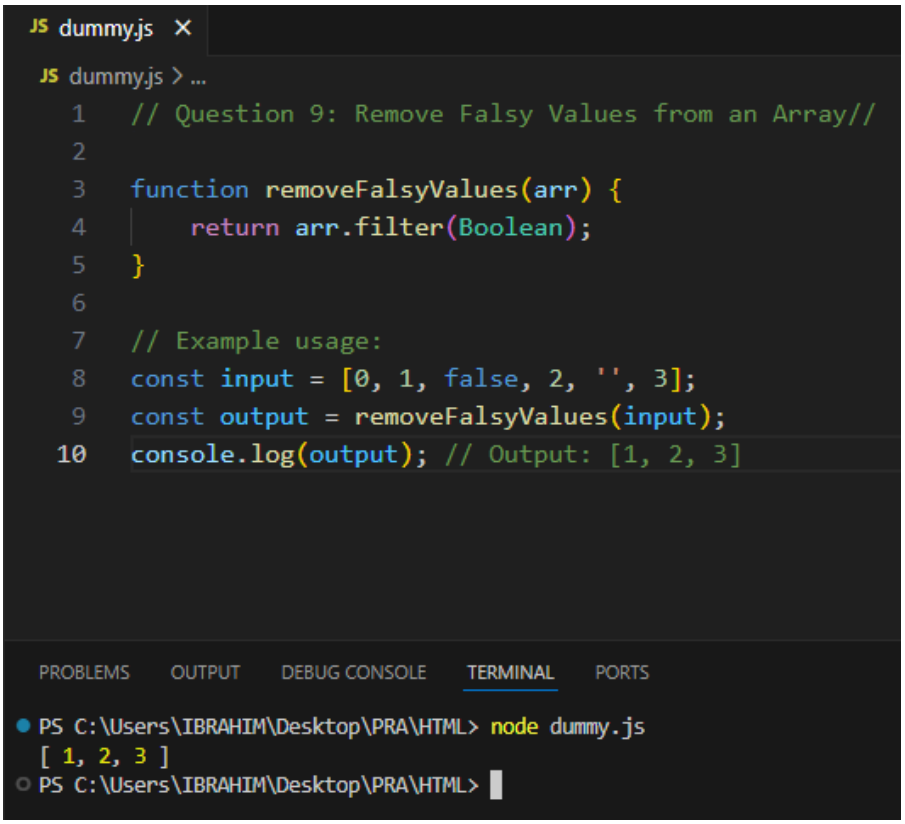
7. The final result is 15.

Question 9: Remove Falsy Values from an Array

Problem: Write a function that removes all falsy values from an array.

Input: [0, 1, false, 2, '', 3]

Output: [1, 2, 3]



Solution - Brief Explanation:

- Step 1: We have an input array with values like 0, 1, false, 2, '', and 3.
- Step 2: We use the filter method to check each element in the input array. By passing Boolean as the function, it removes all falsy values.
- Step 3: We print the output array, which now only includes the truthy values: [1, 2, 3].

Boolean

Boolean is a built-in function in JavaScript that checks whether each value in the array is truthy or falsy.

Meaning of Boolean in This Code

- When we use Boolean inside the filter method:
- It checks each element in the array to determine if it is truthy or falsy.
- If Boolean(element) returns true, the element stays in the array.
- If Boolean(element) returns false, the element is removed from the array.

Example Explanation

- Boolean(0) → false (0 is falsy, so it's removed)

- Boolean(1) → true (1 is truthy, so it's kept)

- Boolean(false) → false (false is falsy, so it's removed)
- Boolean(2) → true (2 is truthy, so it's kept)
- Boolean('') → false ('' is falsy, so it's removed)
- Boolean(3) → true (3 is truthy, so it's kept)

Falsy Value

Falsy Value is a value in JavaScript that evaluates to false when converted to a Boolean.

List of Falsy Values

Here are the common falsy values in JavaScript:

1. false – The Boolean value false.
2. 0 – The number zero.

3. -0 — Negative zero (though its still zero, its considered falsy).

4. '' or "" — An empty string (both single or double quotes).

5. null — The absence of any value or object.

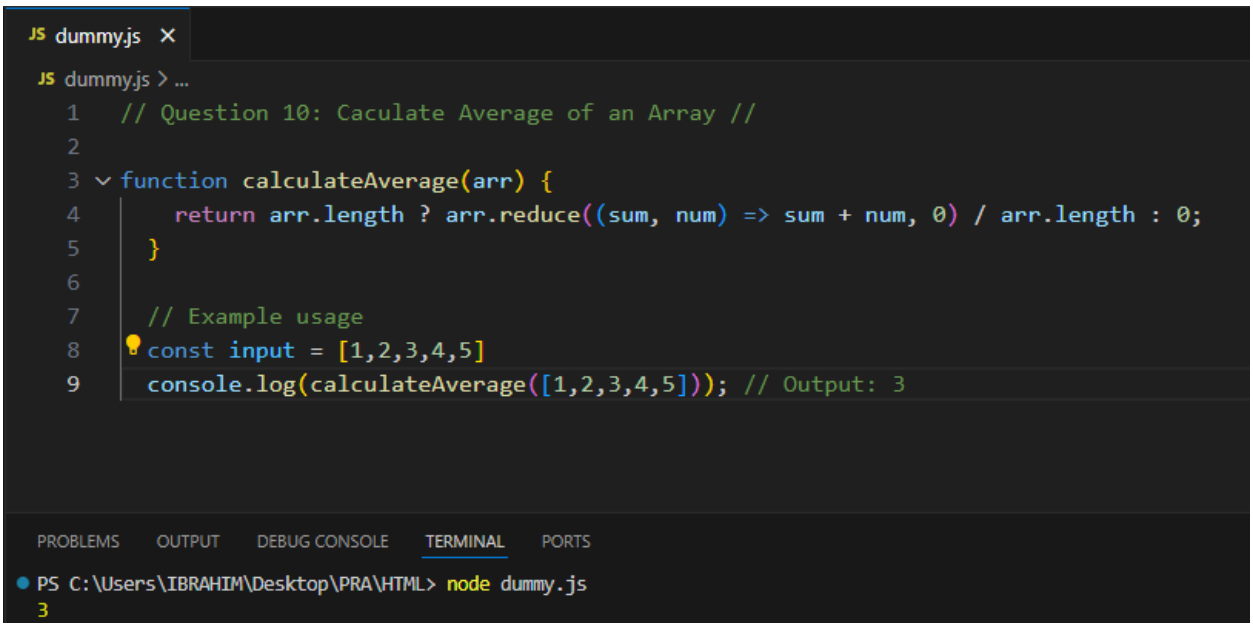
6. undefined — A variable that has been declared but not assigned a value.

7. NaN — The special "Not-a-Number" value.

Question 10: Calculate Average of an Array

Problem: Write a function that calculates the average of all numbers in an array.

Input: [1, 2, 3, 4, 5]



Solution - Brief Explanation:

- Function Definition: calculateAverage(arr) takes an array of numbers as input.
- 2. Check for Empty Array: Uses a conditional (arr.length ? ... : 0) to return 0 if the array is empty.
- 3. Sum Calculation: Utilizes reduce to accumulate the sum of the numbers, starting from an initial value of 0.
- 4. Average Calculation: Divides the total sum by the length of the array to compute the average.
- 5. Return Value: Returns the calculated average or 0 if the array is empty.
- 6. Example Usage:
- Input: [1, 2, 3, 4, 5]

  Output: 3