Assignment on Advance javascript Function

Basic function :

➢ Write a function named `greet` that takes a name as an argument and returns a greeting message.

Default Parameters

➢ Write a function `multiply` that takes two numbers as arguments. The second argument should have a default value of 1. The function should return the product of the two numbers.

Function Returning Values

➢ Write a function `calculateArea` that takes the radius of a circle as an argument and returns the area of the circle. Use the formula: Area = $\pi * r^2$.

Function with Multiple Returns

➢ Create a function `getMinMax` that takes an array of numbers and returns both the minimum and maximum numbers from the array.

Anonymous Function Expression :

➢ Create an anonymous function expression that takes two numbers as arguments and returns their sum. Assign this function to a variable named `add`.
➢ Write an anonymous function expression that takes a string and returns the number of vowels in the string. Assign this function to a variable named `countVowels`.

Arrow function :

➢ Create an arrow function that takes two numbers as arguments and returns their sum. Assign this function to a variable named `add`.

Callback Functions

➢ Write a function `calculate` that takes two numbers and a callback function. The callback function should perform an operation (like addition, subtraction, multiplication, or division) on the two numbers.

Higher-Order Functions

➢ Write a function `higherOrderFunction` that takes a function `fn` and a value `x` as arguments. The function should apply `fn` to `x` and return the result.

Immediately Invoked Function Expression

➢ Create an IIFE that takes two numbers as arguments and returns their sum. Log the result to the console.

Call by Value with Primitive Data Types

➢ Write a function `modifyNumber` that takes a number as an argument, modifies it inside the function, and logs the value both inside and outside the function to demonstrate call by value.

Call by Value with Reference Data Types

➢ Write a function `modifyArray` that takes an array, modifies its elements inside the function, and logs the array both inside and outside the function.