1. Write a simple Node.js HTTP server that listens on port 3000 and responds with a JSON Object when accessed using api call.

```
const http = require('http');

const server = http.createServer((req, res) => {

  res.writeHead(200, { 'Content-Type': 'application/json' });

  const responseObject = {
    message: 'Hello, World!',
    status: 'success',
    timestamp: new Date().toISOString()
  };

  res.end(JSON.stringify(responseObject));
});

const PORT = 3000;
server.listen(PORT, () => {
  console.log(`Server is listening on port ${PORT}`);
});
```

1.create mathematical operation using let, var and const methods using javascript and print the values in console accordingly.

```
const PI = 3.14159;
console.log("Value of PI:", PI);

let radius = 5;
let area = PI * radius * radius;
console.log("Area of the circle with radius", radius, "is:", area);

var length = 10;
var width = 4;
var perimeter = 2 * (length + width);
console.log("Perimeter of the rectangle with length", length, "and width", width, "is:", perimeter);


let a = 10, b = 5;
const sum = a + b;
let difference = a - b;
var product = a * b;
let quotient = a / b;

console.log("Sum:", sum);
console.log("Difference:", difference);
console.log("Product:", product);
console.log("Quotient:", quotient);

a = 15;
radius = 7;
length = 12;
console.log("Updated radius:", radius);
console.log("Updated length:", length);
```

2.write a function to implement map, reduce, filter, and flatmap using javascript.

2.1
```
function customMap(array, callback) {
  let result = [];
  for (let i = 0; i < array.length; i++) {
    result.push(callback(array[i], i, array));
  }
  return result;
}

console.log(customMap([1, 2, 3], x => x * 2)); // [2, 4, 6]
```

2.2
```
function customReduce(array, callback, initialValue) {
  let accumulator = initialValue !== undefined ? initialValue : array[0];
  let startIndex = initialValue !== undefined ? 0 : 1;

  for (let i = startIndex; i < array.length; i++) {
    accumulator = callback(accumulator, array[i], i, array);
  }
  return accumulator;
}

console.log(customReduce([1, 2, 3, 4], (acc, x) => acc + x, 0)); // 10
```

2.3
```
function customFilter(array, callback) {
  let result = [];
  for (let i = 0; i < array.length; i++) {
    if (callback(array[i], i, array)) {
      result.push(array[i]);
    }
  }
  return result;
}

console.log(customFilter([1, 2, 3, 4], x => x % 2 === 0)); // [2, 4]
```

2.4
```
function customFlatMap(array, callback) {
  let result = [];
  for (let i = 0; i < array.length; i++) {
    let mappedValue = callback(array[i], i, array);
    if (Array.isArray(mappedValue)) {
      result = result.concat(mappedValue);
    } else {
      result.push(mappedValue);
    }
  }
  return result;
}

console.log(customFlatMap([1, 2, 3], x => [x, x * 2])); // [1, 2, 2, 4, 3, 6]
```

3.Give an example of creating a callback function in javascript.

```
function processUserInput(name, callback) {
  console.log("Processing user input...");
  callback(name);
}

function greetUser(userName) {
  console.log("Hello, " + userName + "!");
}

processUserInput("Alice", greetUser);
```

4.write a program to reverse a string using arrays in javascript.

```
function reverseString(str) {
  let charArray = str.split('');

  let reversedArray = charArray.reverse();

  let reversedString = reversedArray.join('');

  return reversedString;
}

let originalString = "Hello, World!";
let reversedString = reverseString(originalString);

console.log("Original String:", originalString);
console.log("Reversed String:", reversedString);


function reverseStringManual(str) {
  let reversed = [];
  for (let i = str.length - 1; i >= 0; i--) {
    reversed.push(str[i]);
  }
  return reversed.join('');
}

console.log(reverseStringManual("Hello, World!")); // Output: !dlroW ,olleH
```