1. Write a function that prints the numbers from 1 to 100. But for multiples of three, print "Fizz" instead of the number, and for the multiples of five, print "Buzz". For numbers that are multiples of both three and five, print "FizzBuzz".

**Sol)**

function fizzBuzz(){

    for(let i=0;i<=100;i++){

        if(i%3===0 && i%5 ==0 ){

            console.log('FizzBuzz');

        }else if(i%3===0){

            console.log('Fizz');

        }else if(i%5===0){

            console.log('Buzz');

        }else{

            console.log(i);

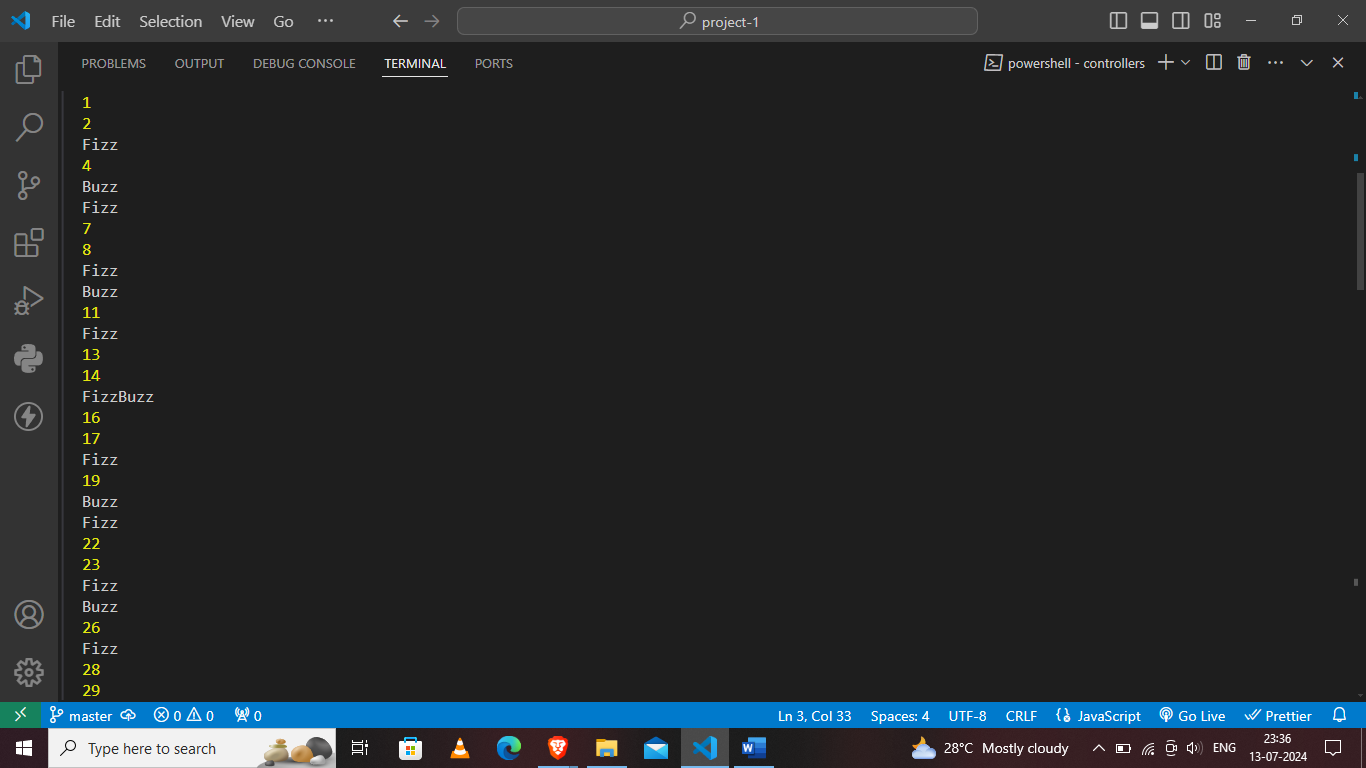
        }

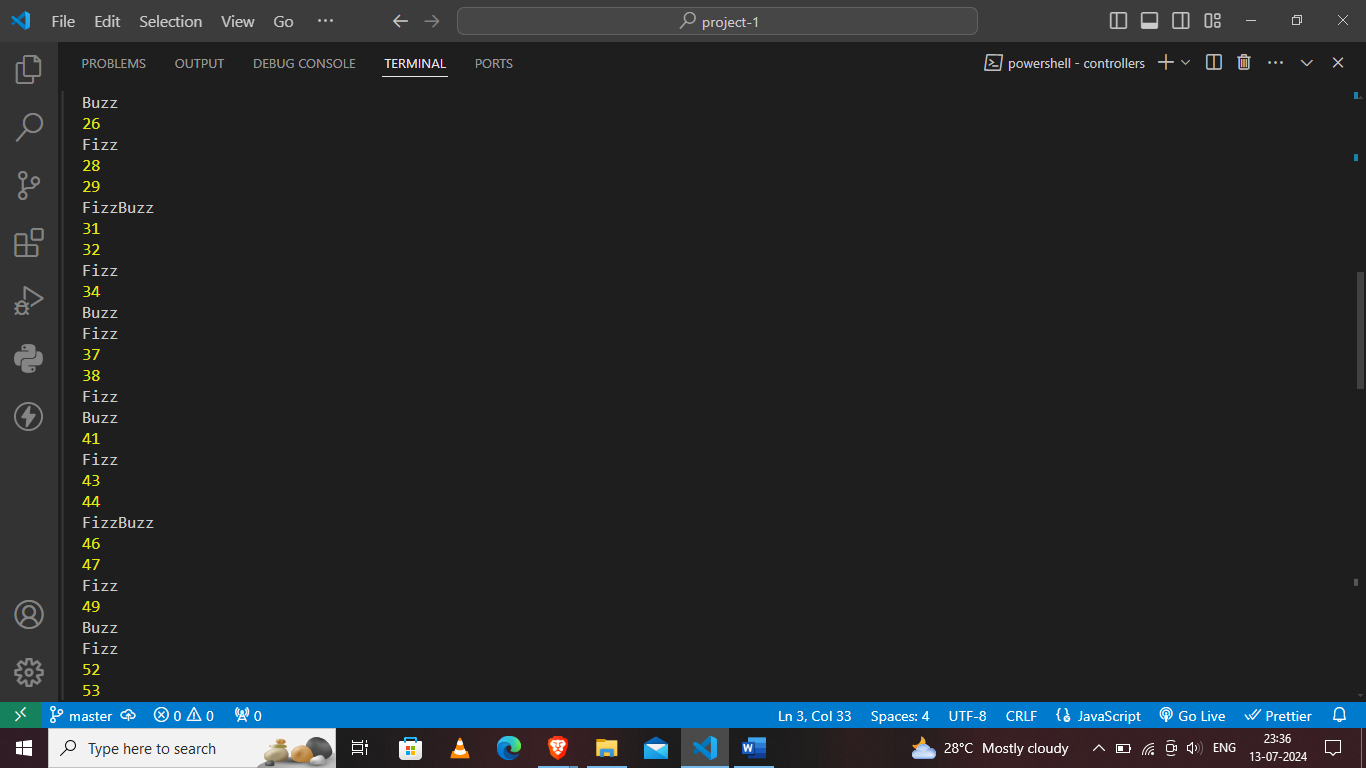
    }

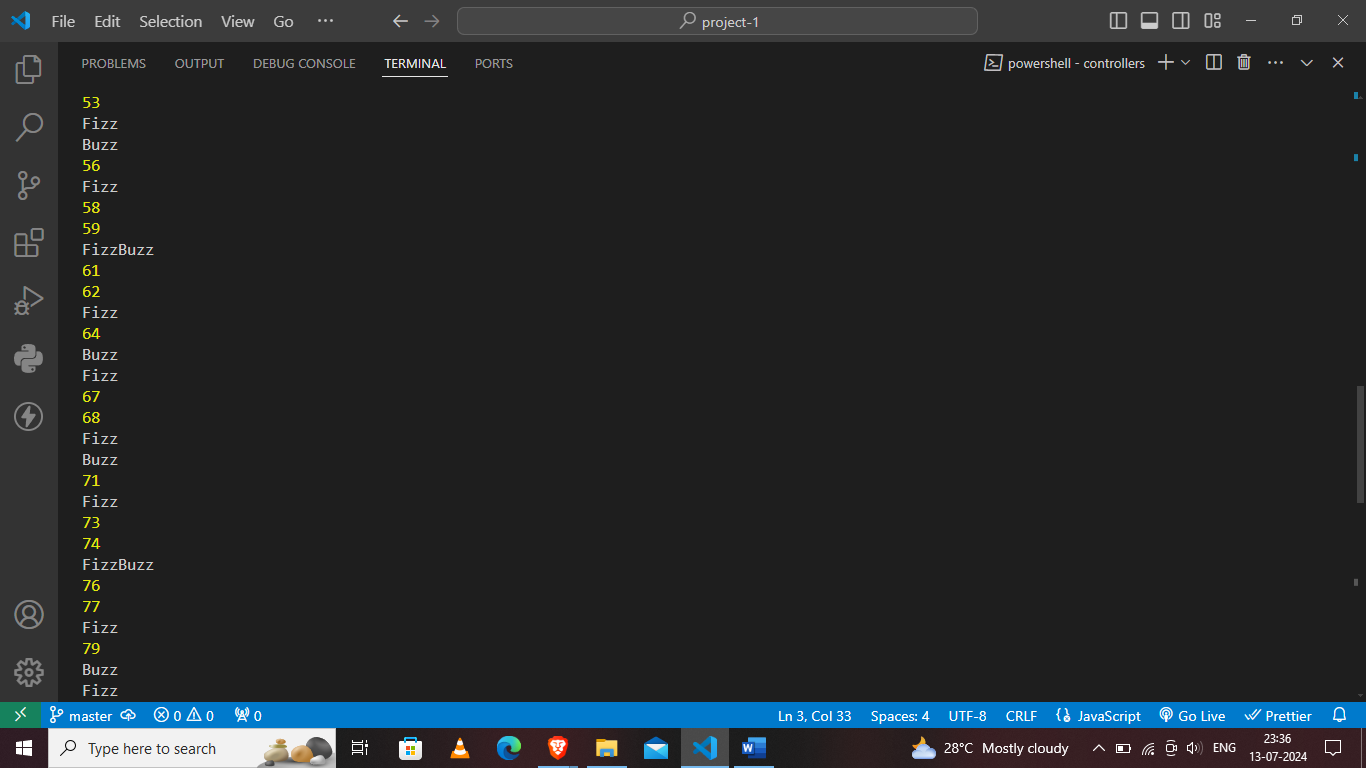
}

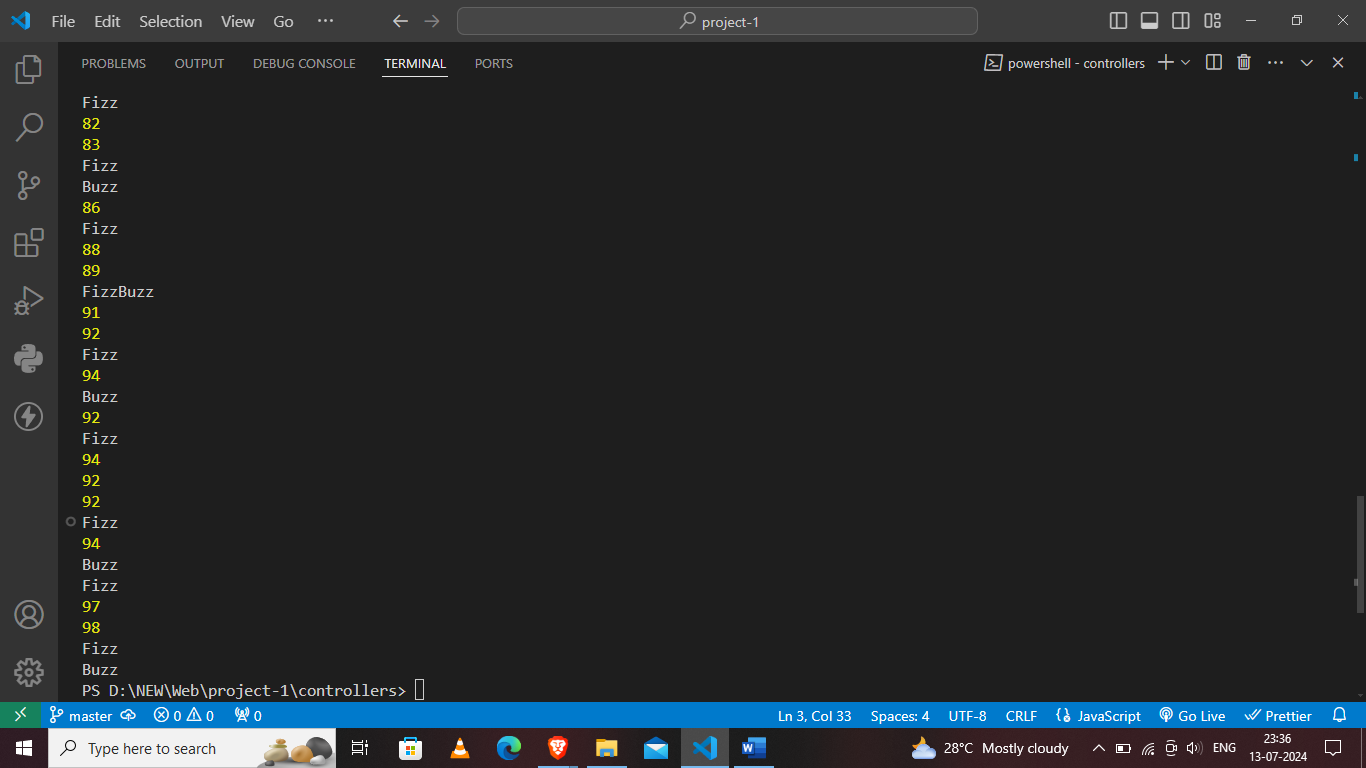
fizzBuzz();

**Output:**

****

****

****

****

**2.** Write a function that takes a string input representing a simple arithmetic expression (only addition and subtraction) and returns the result.

**Sol:-**

function evaluateExpression(expression) {

    expression = expression.replace(/\s+/g, '');

    let result = 0;

    let currentNumber = '';

    let currentOperator = '+';

    function applyOperator(operator, number) {

        if (operator === '+') {

            result += number;

        } else if (operator === '-') {

            result -= number;

        }

    }

    for (let i = 0; i < expression.length; i++) {

        let char = expression[i];

        if (/\d/.test(char)) {

            currentNumber += char;

        }

        if (/[+\-]/.test(char) || i === expression.length - 1) {

            applyOperator(currentOperator, parseInt(currentNumber));

            currentOperator = char;

            currentNumber = '';

        }

    }

    return result;

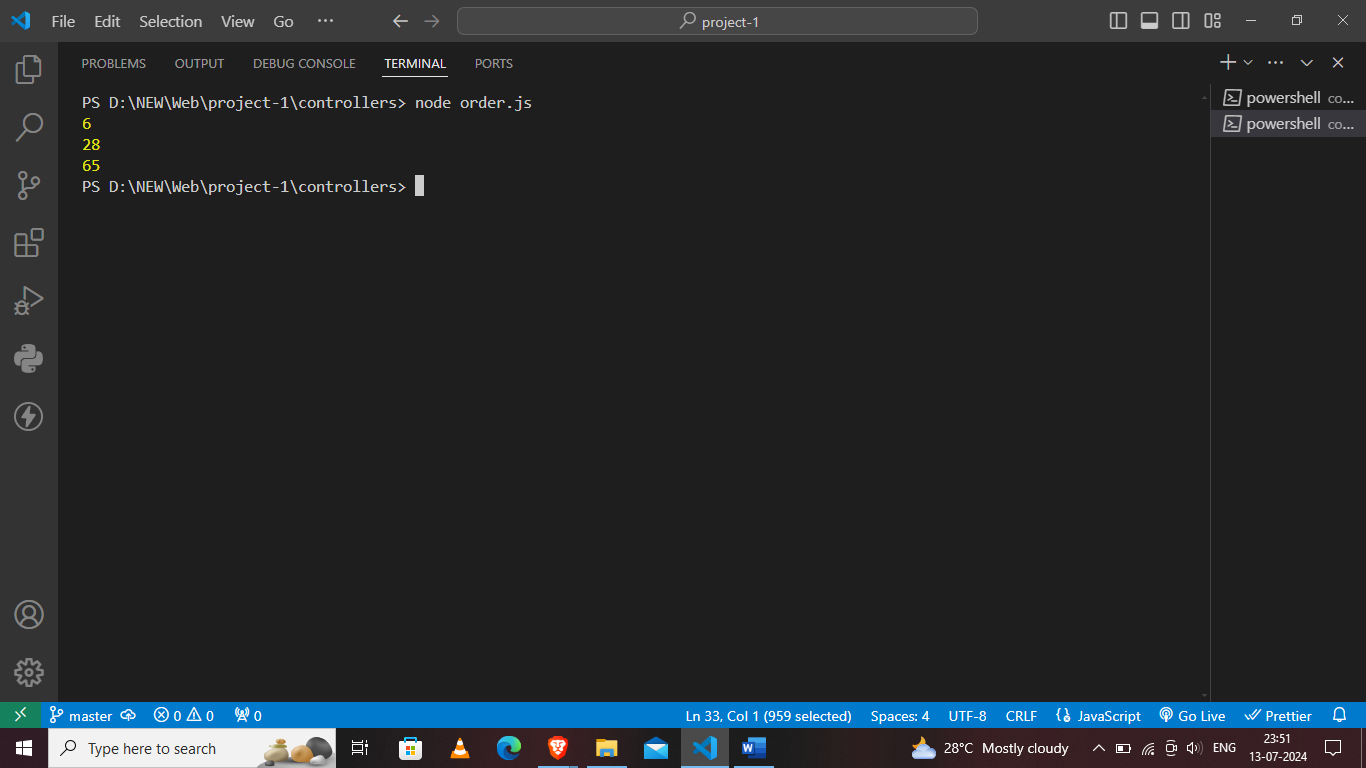
}

console.log(evaluateExpression("3 + 5 - 2"));

console.log(evaluateExpression("10 + 20 - 5 + 3"));

console.log(evaluateExpression("100 - 50 + 25 - 10"));

**Output:**

****

**3.** Write a function that takes a nested array and returns a flattened array.

**Sol:-**

function flattenArray(nestedArray) {

    let flattened = [];

    function flatten(element) {

        if (Array.isArray(element)) {

            for (let item of element) {

                flatten(item);

            }

        } else {

            flattened.push(element);

        }

    }

    flatten(nestedArray);

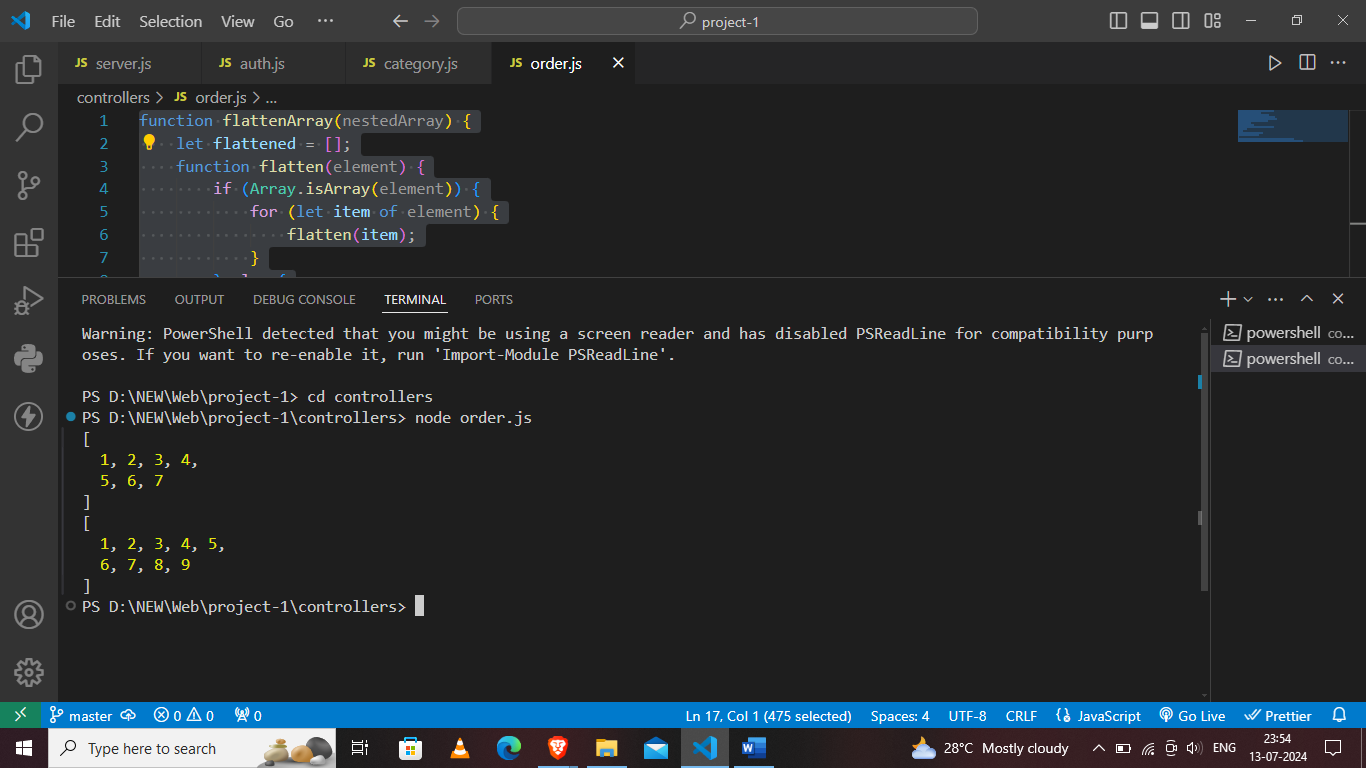
    return flattened;

}

console.log(flattenArray([1, [2, [3, [4, 5]], 6], 7]));

console.log(flattenArray([[1, 2, [3]], 4, [5, 6], 7, [8, [9]]]));

**Output:**

****

**4.** Write a function that checks if two given strings are anagrams of each other.

function areAnagrams(str1, str2) {

    const normalize = str => str.replace(/[^\w]/g, '').toLowerCase();

    const normalizedStr1 = normalize(str1);

    const normalizedStr2 = normalize(str2);

    if (normalizedStr1.length !== normalizedStr2.length) {

        return false;

    }

    const sortedStr1 = normalizedStr1.split('').sort().join('');

    const sortedStr2 = normalizedStr2.split('').sort().join('');

    return sortedStr1 === sortedStr2;

}

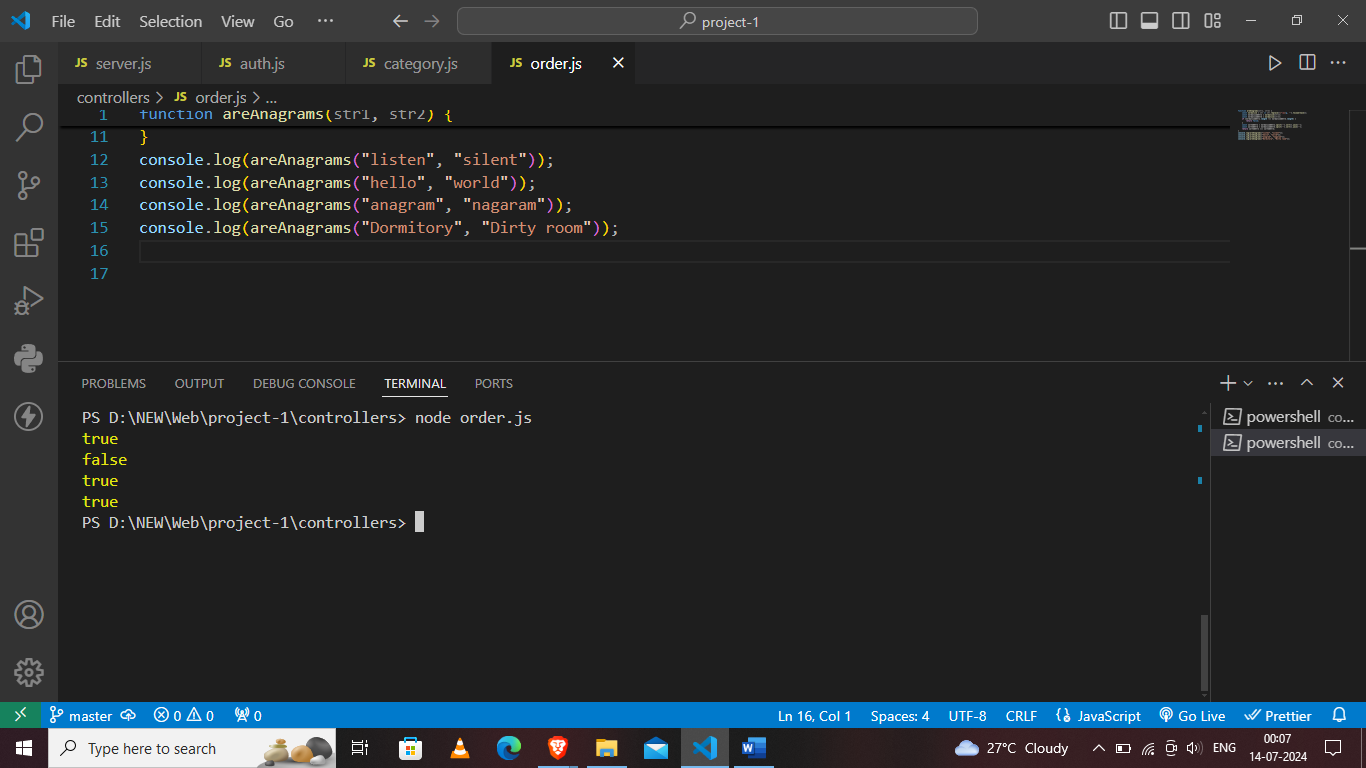
console.log(areAnagrams("listen", "silent"));

console.log(areAnagrams("hello", "world"));

console.log(areAnagrams("anagram", "nagaram"));

console.log(areAnagrams("Dormitory", "Dirty room"));

**Ouput:**

****

**5.** Write a function that takes an array and returns a new array with duplicates removed.

**Sol:-**

function removeDuplicates(arr) {

    return [...new Set(arr)];

}

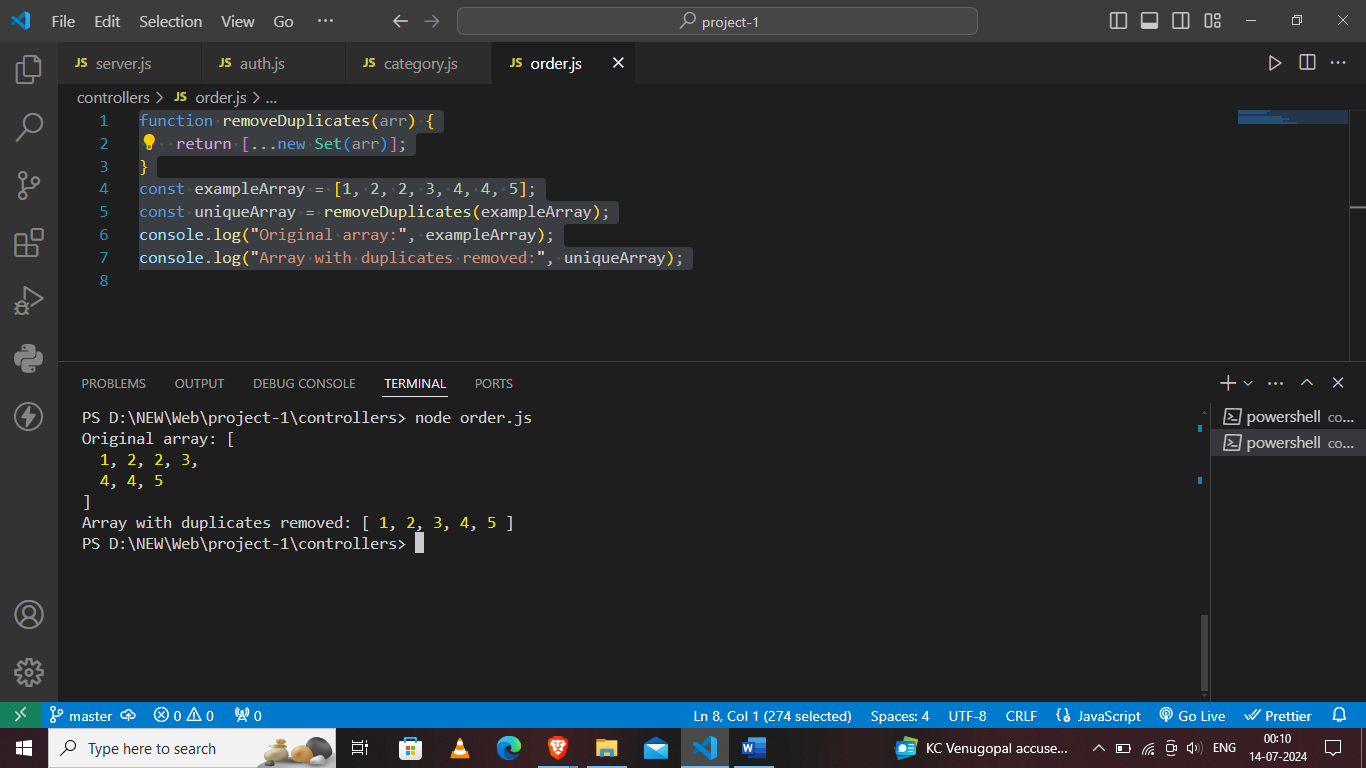
const exampleArray = [1, 2, 2, 3, 4, 4, 5];

const uniqueArray = removeDuplicates(exampleArray);

console.log("Original array:", exampleArray);

console.log("Array with duplicates removed:", uniqueArray);

**Output:**

****

**6.** Write a function that takes a string and capitalizes the first letter of each word in the string.

**Sol:-**

function capitalizeFirstLetters(str) {

    const words = str.split(' ');

    const capitalizedWords = words.map(word => {

        if (word === '') return word;

        return word[0].toUpperCase() + word.slice(1);

    });

    return capitalizedWords.join(' ');

}

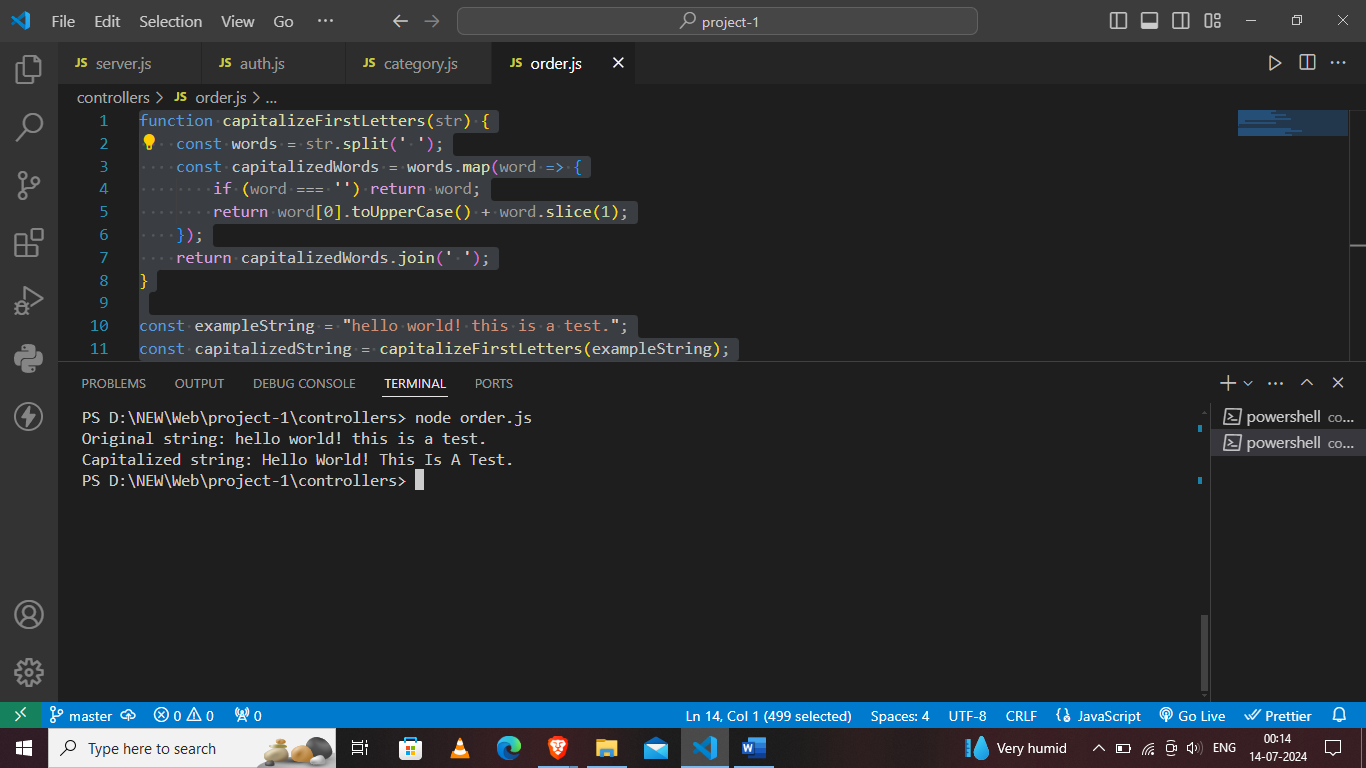
const exampleString = "hello world! this is a test.";

const capitalizedString = capitalizeFirstLetters(exampleString);

console.log("Original string:", exampleString);

console.log("Capitalized string:", capitalizedString);

**Output:**

****

**7.** Write a function that generates the first n numbers of the Fibonacci sequence

Sol:-

function generateFibonacci(n) {

    if (n <= 0) return [];

    if (n === 1) return [0];

    const fibSequence = [0, 1];

    for (let i = 2; i < n; i++) {

        fibSequence.push(fibSequence[i - 1] + fibSequence[i - 2]);

    }

    return fibSequence;

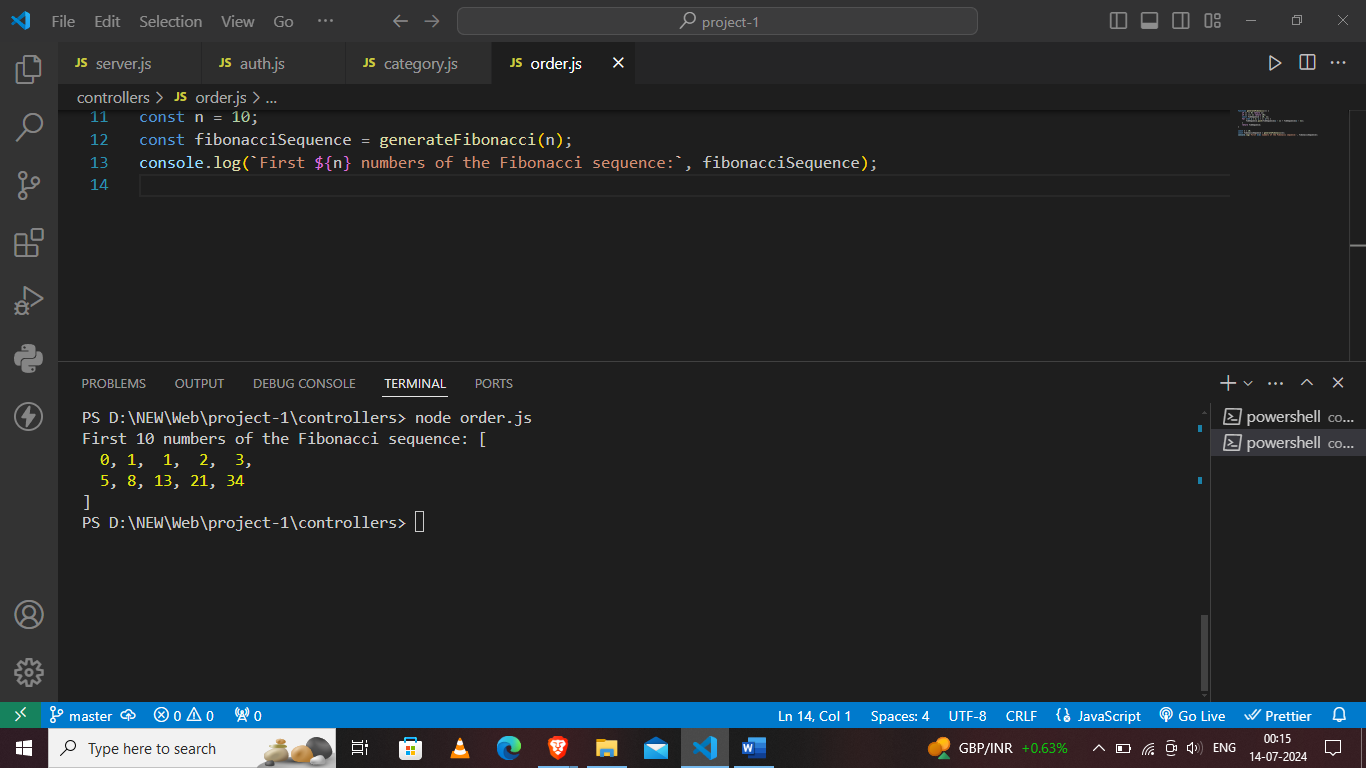
}

const n = 10;

const fibonacciSequence = generateFibonacci(n);

console.log(`First ${n} numbers of the Fibonacci sequence:`, fibonacciSequence);

**Output:**

****

**8.** Implement a simple HashMap class with put, get, and remove methods.

**Sol:-**

class HashMap {

    constructor() {

        this.map = {};

    }

    put(key, value) {

        this.map[key] = value;

    }

    get(key) {

        return this.map.hasOwnProperty(key) ? this.map[key] : undefined;

    }

    remove(key) {

        if (this.map.hasOwnProperty(key)) {

            delete this.map[key];

        }

    }

}

const hashMap = new HashMap();

hashMap.put("name", "John Doe");

hashMap.put("age", 30);

console.log(hashMap.get("name"));

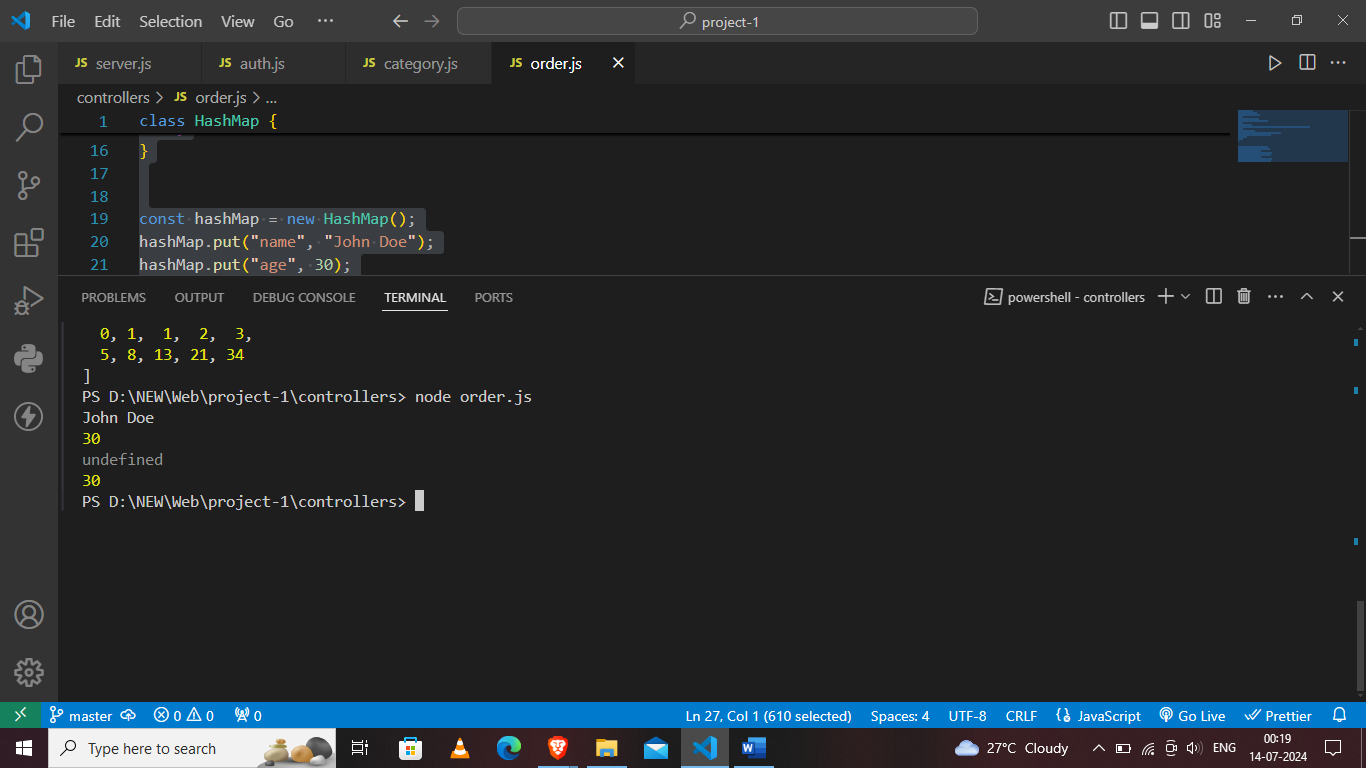
console.log(hashMap.get("age"));

hashMap.remove("name");

console.log(hashMap.get("name"));

console.log(hashMap.get("age"));

**Output:**

****

**9.** Write a function that filters out even numbers from an array.

**Sol:-**

function filterEvenNumbers(arr) {

    let oddNumbers = [];

    for (let i = 0; i < arr.length; i++) {

        let num = arr[i];

        if (num % 2 !== 0) {

            oddNumbers.push(num);

        }

    }

    return oddNumbers;

}

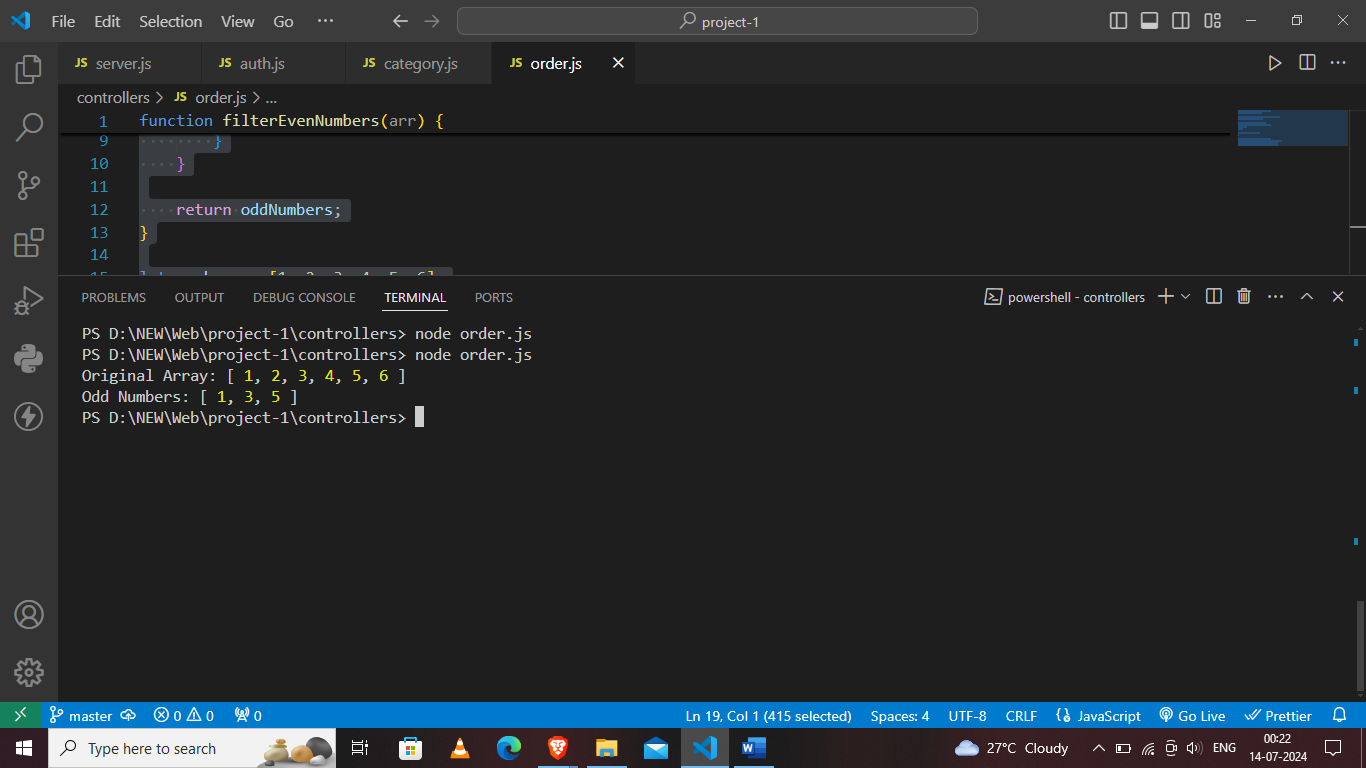
let numbers = [1, 2, 3, 4, 5, 6];

let oddNumbers = filterEvenNumbers(numbers);

console.log("Original Array:", numbers);

console.log("Odd Numbers:", oddNumbers);

**Output:**

****

**10.** Write a function that converts a given string to title case (capitalizing the first letter of each word).

function toTitleCase(str) {

    let words = str.toLowerCase().split(' ');

    for (let i = 0; i < words.length; i++) {

        words[i] = words[i].charAt(0).toUpperCase() + words[i].substring(1);

    }

    return words.join(' ');

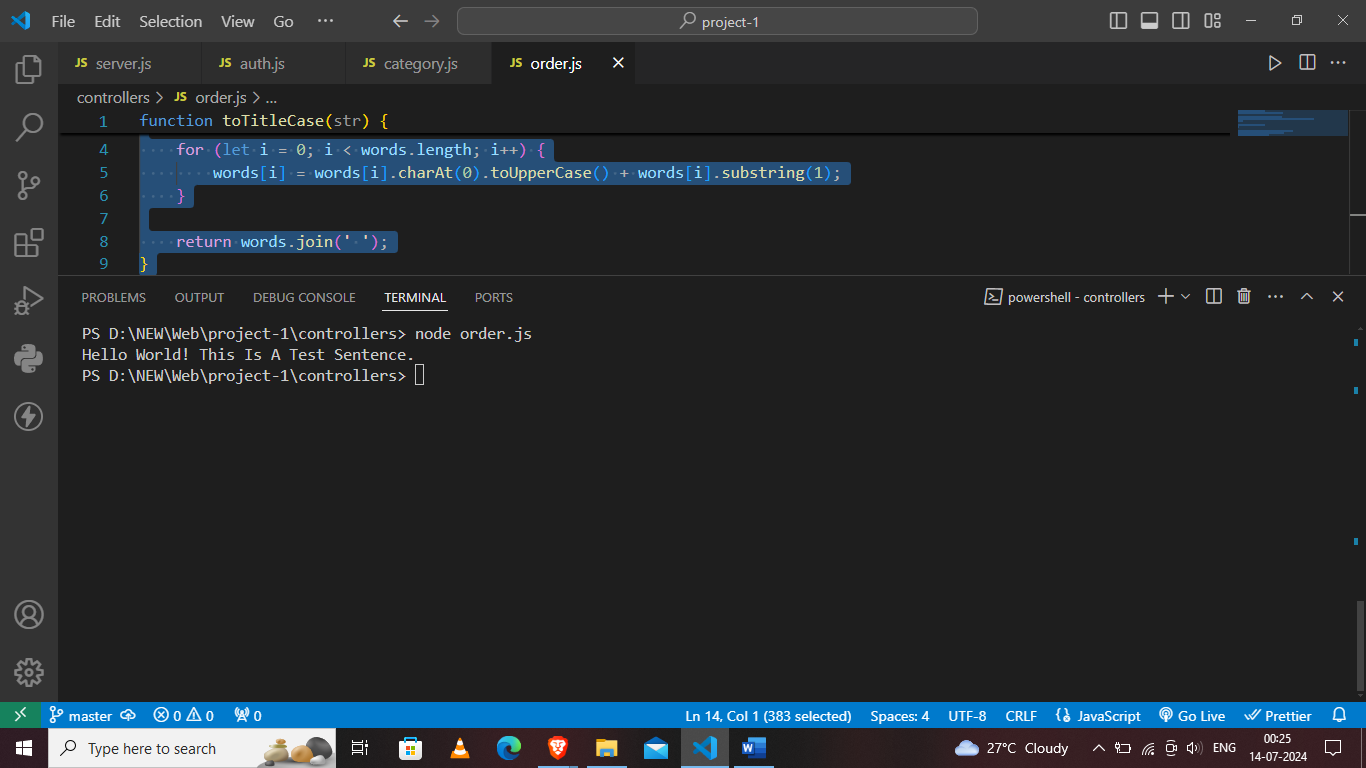
}

let sentence = "hello world! this is a test sentence.";

let titleCaseSentence = toTitleCase(sentence);

console.log(titleCaseSentence);

**Ouput:**

****