

**MOBILE APPLICATION DEVELOPMENT**

**ASSIGNMENT 02**

**NAME:**

**MUHAMMAD MUAAZ SHOAIB**

**REG. NO:**

**FA20-BCS-074**

**CLASS:**

**BCS-6B**

**SUBMITTED TO:**

**MR. MUHAMMAD KAMRAN**

**DATE:**

**19-MAR-2023**

# Q1: Array functions in JavaScript with examples.

## push

This method adds one or more elements to the end of an array and returns the new length of the array.

// FA20-BCS-074

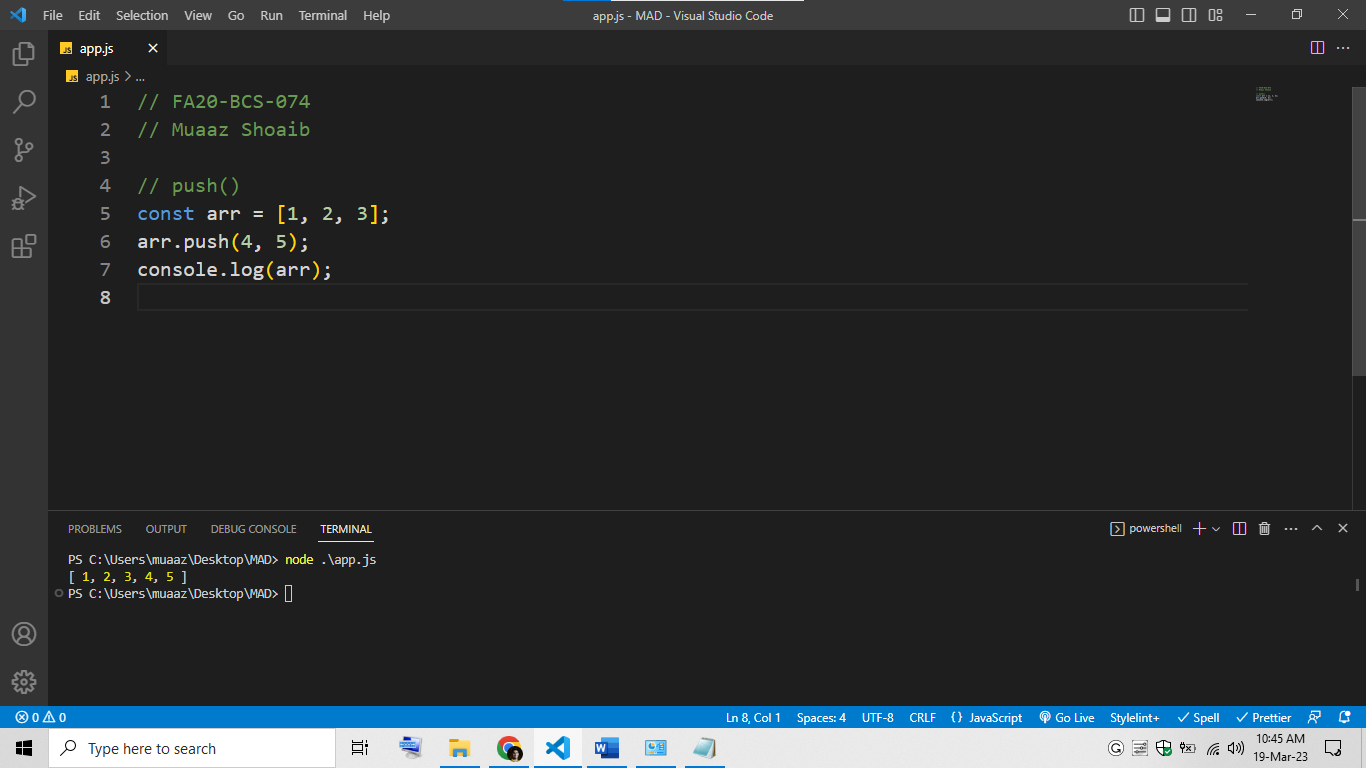
// Muaaz Shoaib

// push()

const arr = [1, 2, 3];

arr.push(4, 5);

console.log(arr);



## pop

This method removes the last element from an array and returns that element.

// FA20-BCS-074

// Muaaz Shoaib

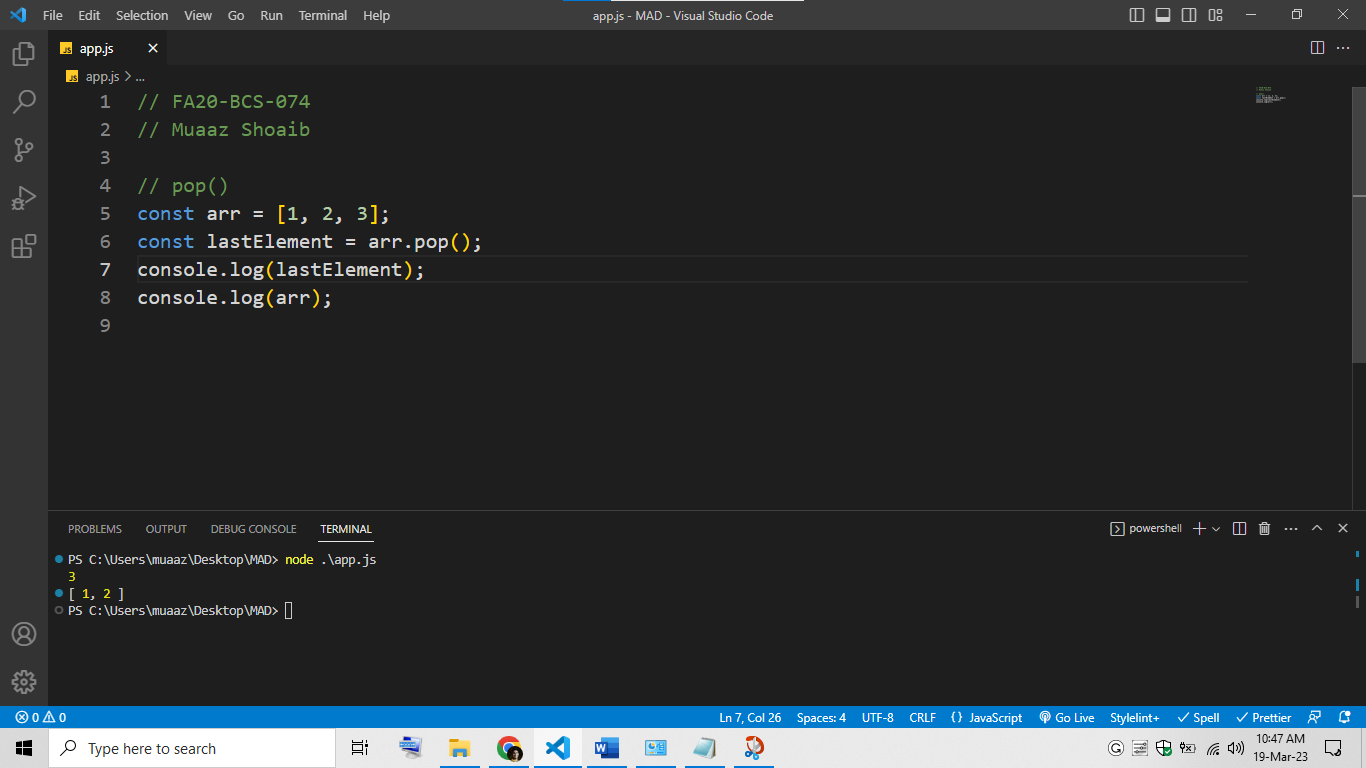
// pop()

const arr = [1, 2, 3];

const lastElement = arr.pop();

console.log(lastElement);

console.log(arr);



## shift

This method removes the first element from an array and returns that element.

// FA20-BCS-074

// Muaaz Shoaib

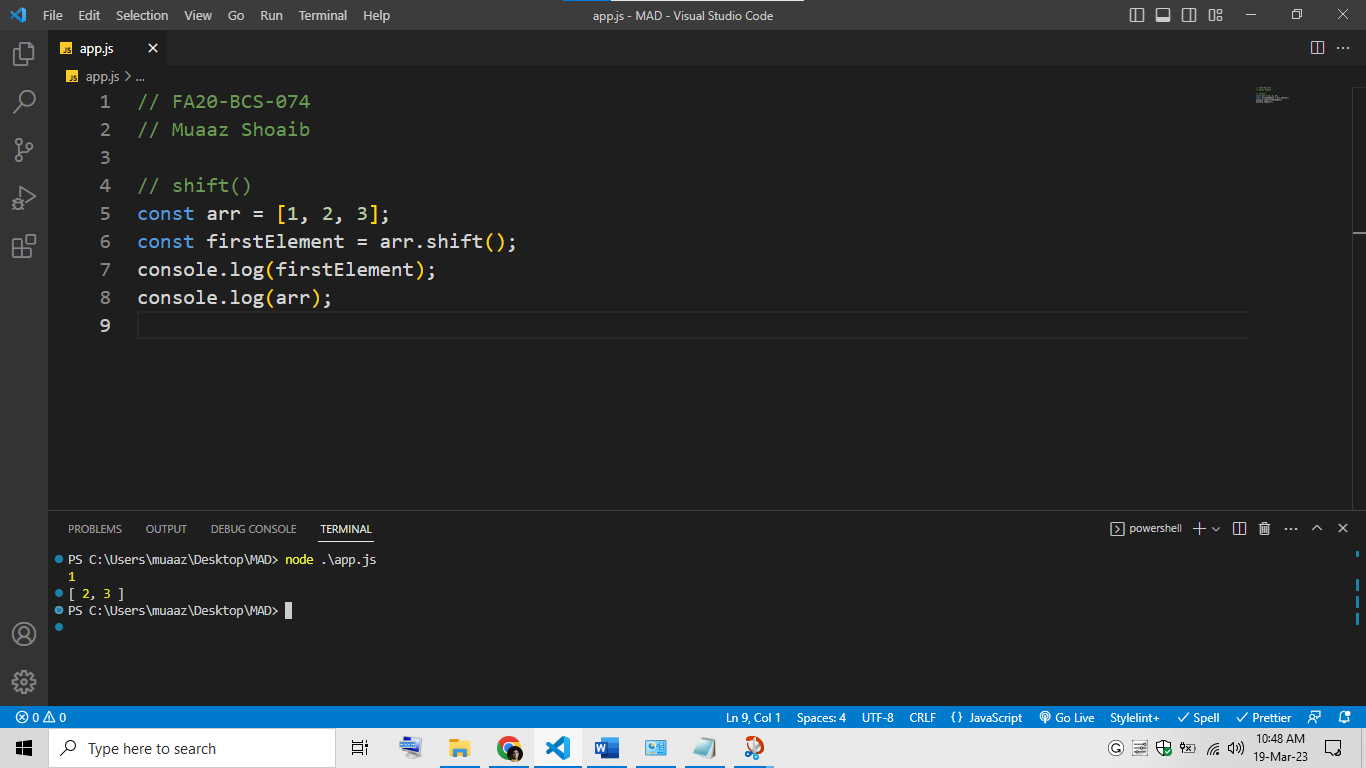
// shift()

const arr = [1, 2, 3];

const firstElement = arr.shift();

console.log(firstElement);

console.log(arr);



## unshift

This method adds one or more elements to the beginning of an array and returns the new length of the array.

// FA20-BCS-074

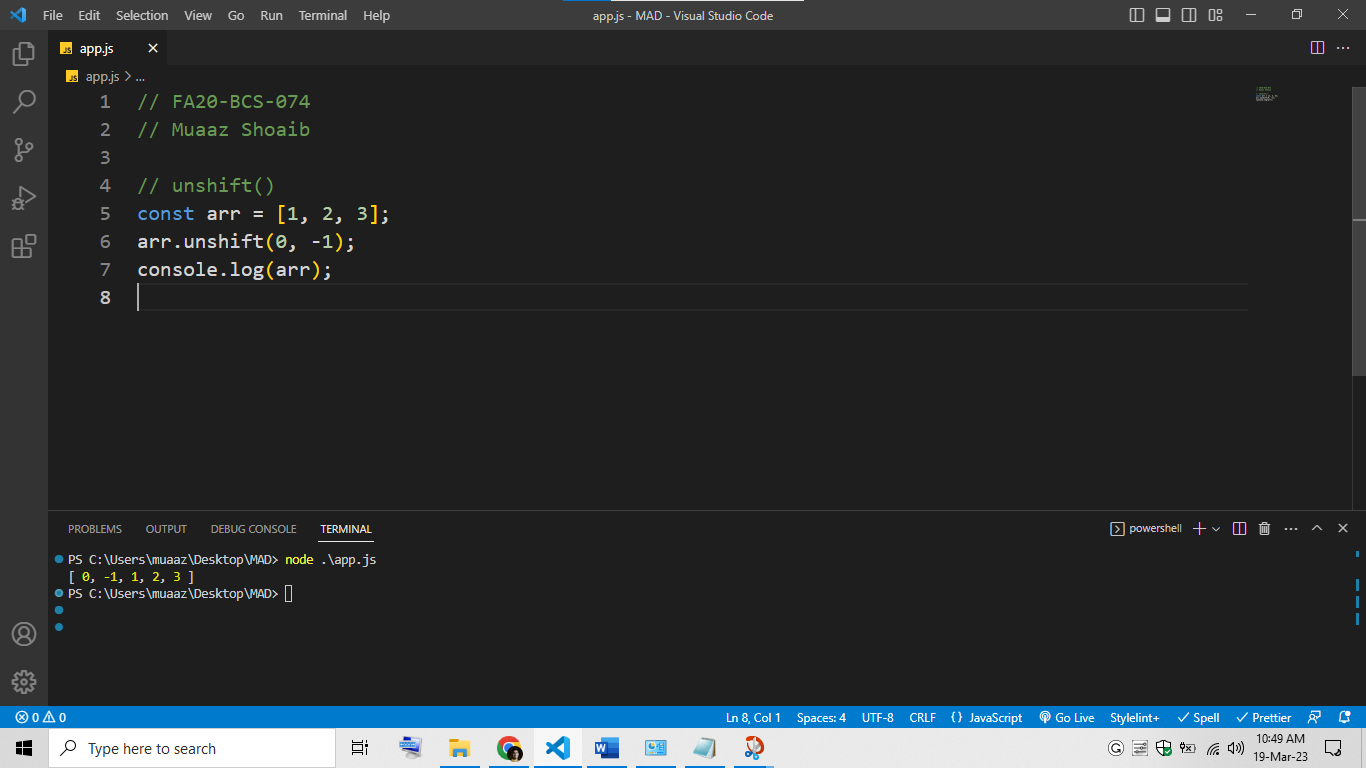
// Muaaz Shoaib

// unshift()

const arr = [1, 2, 3];

arr.unshift(0, -1);

console.log(arr);



## slice

This method returns a shallow copy of a portion of an array into a new array.

// FA20-BCS-074

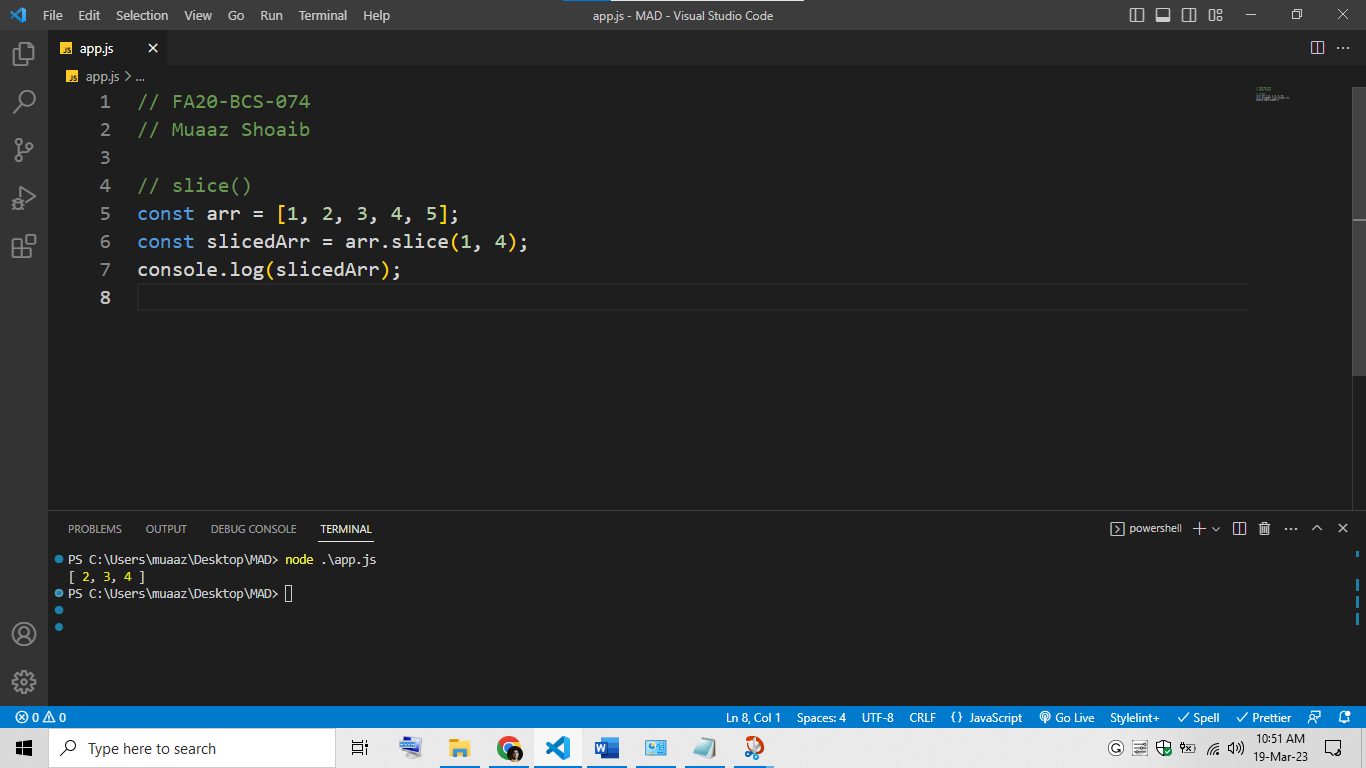
// Muaaz Shoaib

// slice()

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

const slicedArr = arr.slice(1, 4);

console.log(slicedArr);



## splice

This method changes the contents of an array by removing or replacing existing elements and/or adding new elements.

// FA20-BCS-074

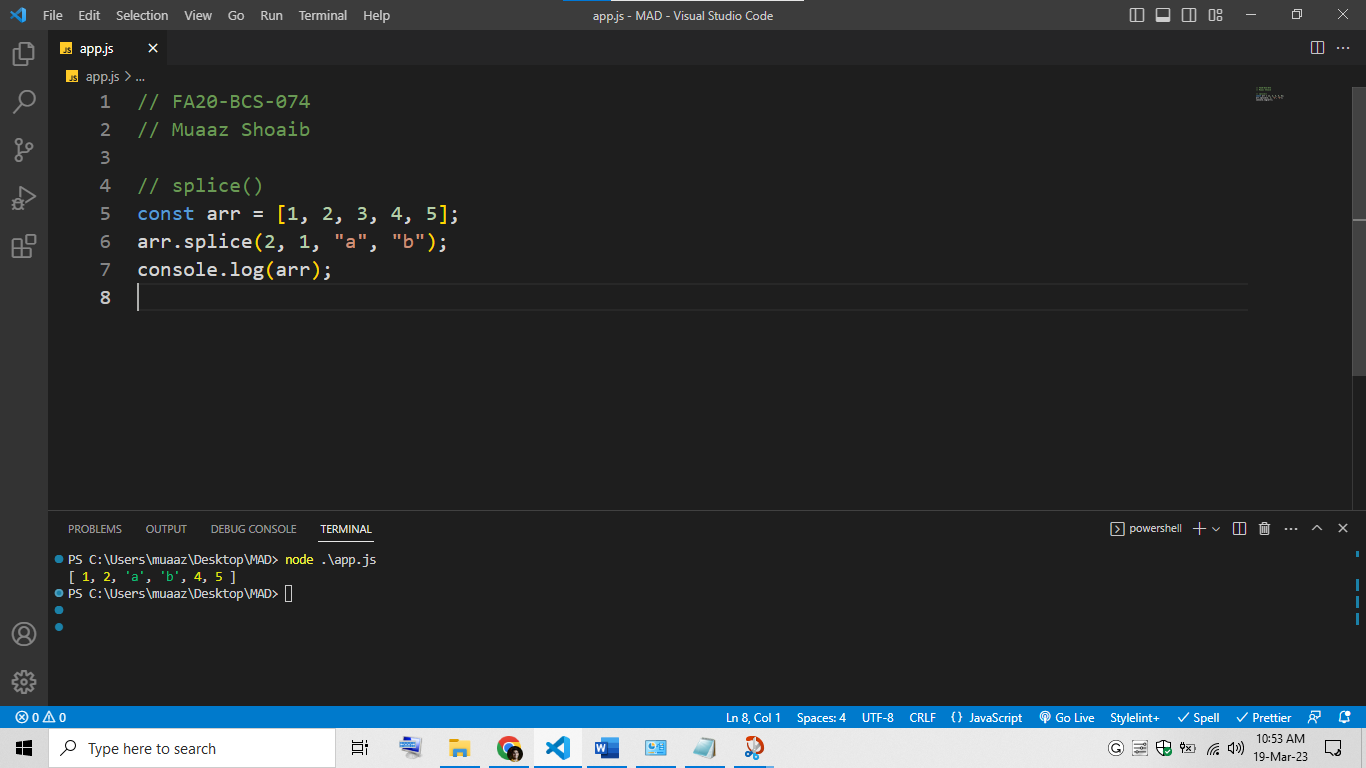
// Muaaz Shoaib

// splice()

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

arr.splice(2, 1, "a", "b");

console.log(arr);



## concat

This method merges two or more arrays into a new array.

// FA20-BCS-074

// Muaaz Shoaib

// concat()

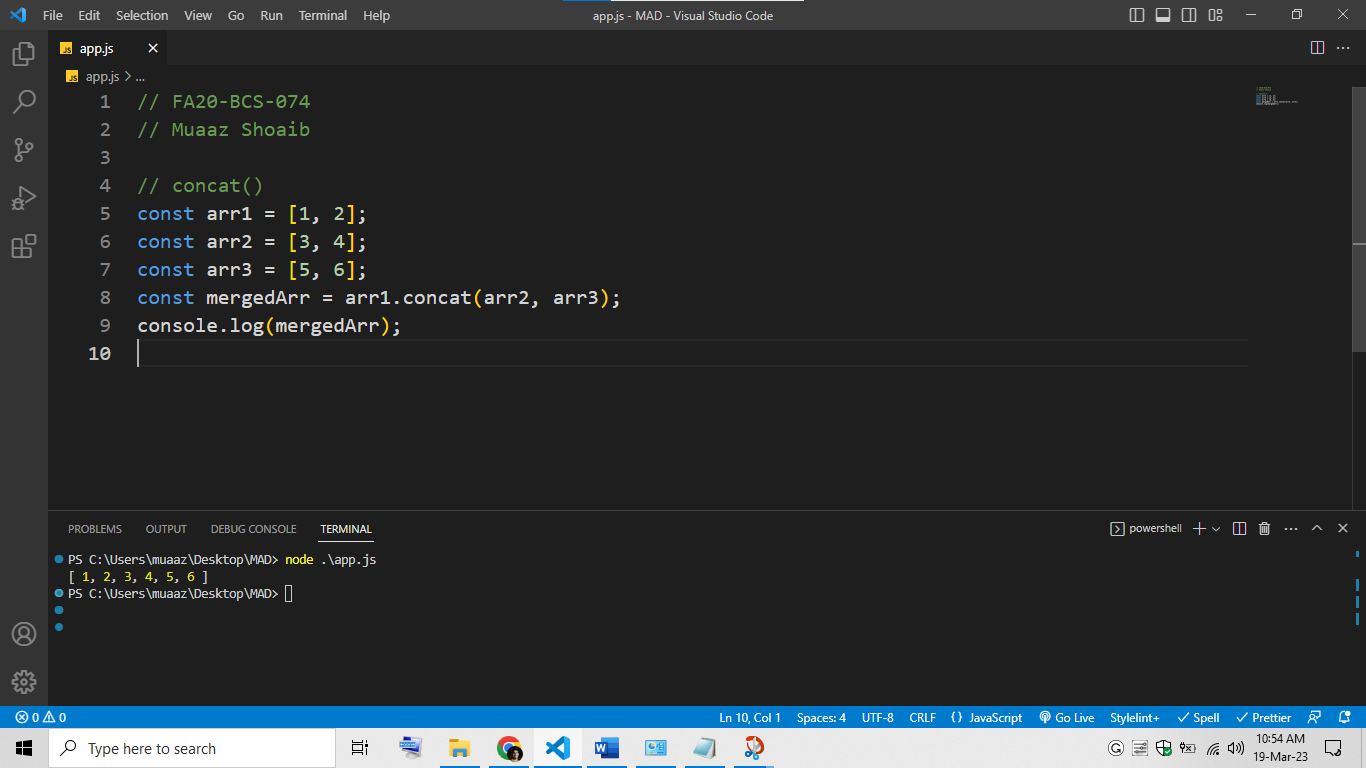
const arr1 = [1, 2];

const arr2 = [3, 4];

const arr3 = [5, 6];

const mergedArr = arr1.concat(arr2, arr3);

console.log(mergedArr);



## reverse

This method reverses the order of the elements in an array.

// FA20-BCS-074

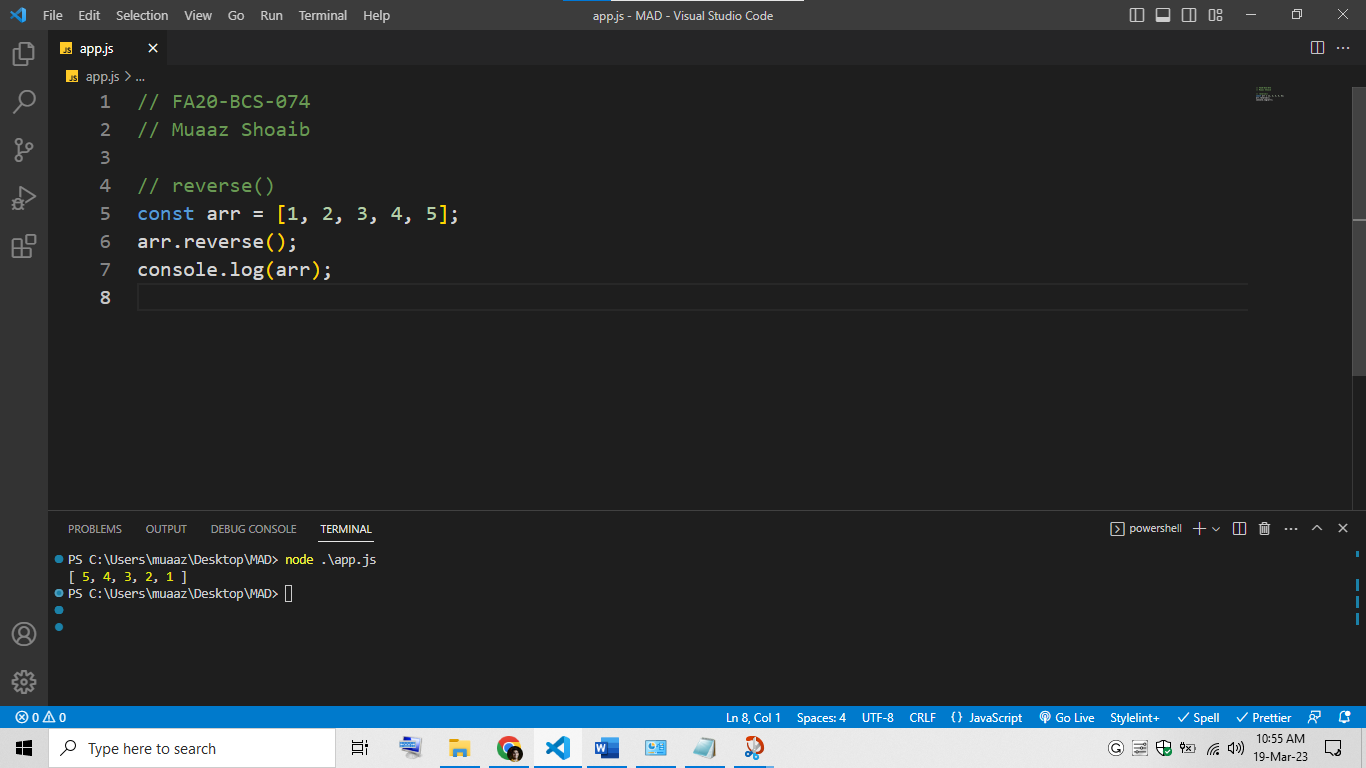
// Muaaz Shoaib

// reverse()

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

arr.reverse();

console.log(arr);



## join

This method joins all elements of an array into a string.

// FA20-BCS-074

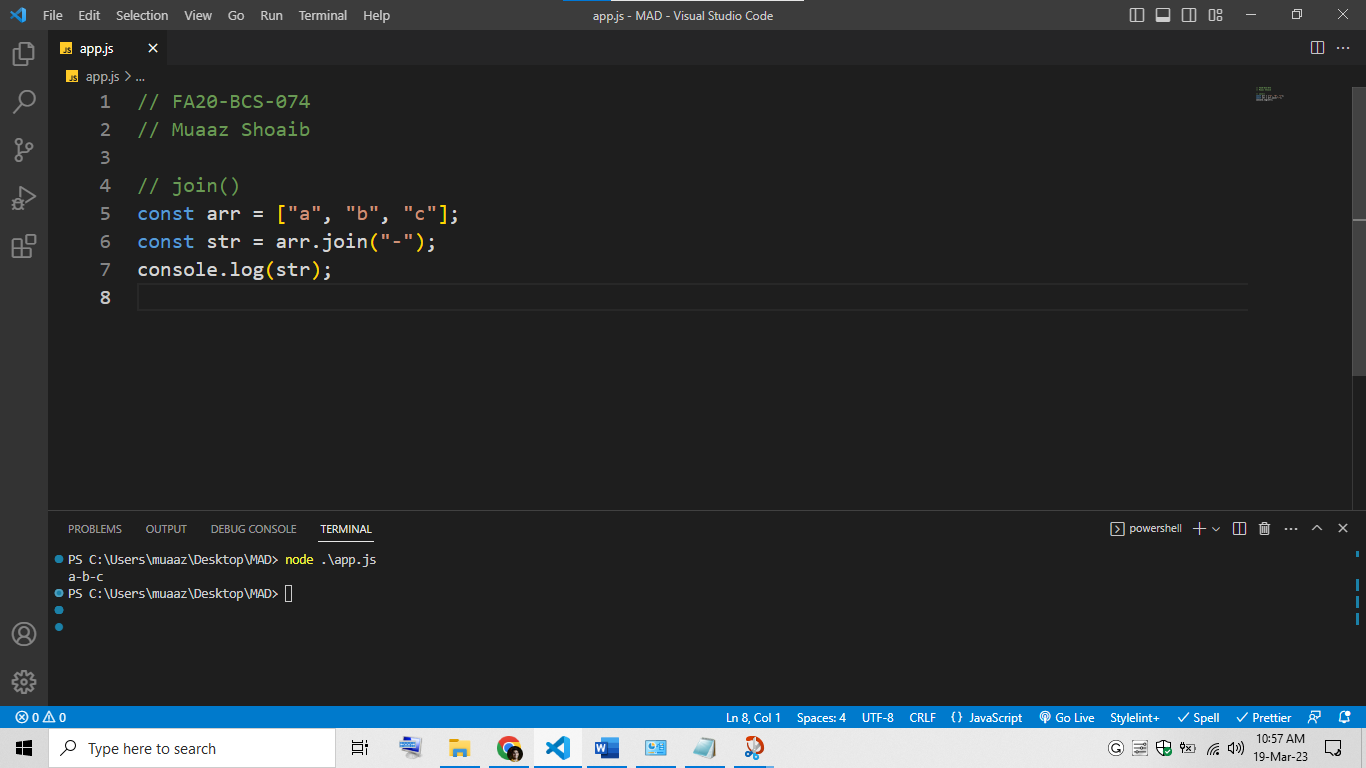
// Muaaz Shoaib

// join()

const arr = ["a", "b", "c"];

const str = arr.join("-");

console.log(str);



## indexOf

This method returns the first index at which a given element can be found in an array, or -1 if it is not present.

// FA20-BCS-074

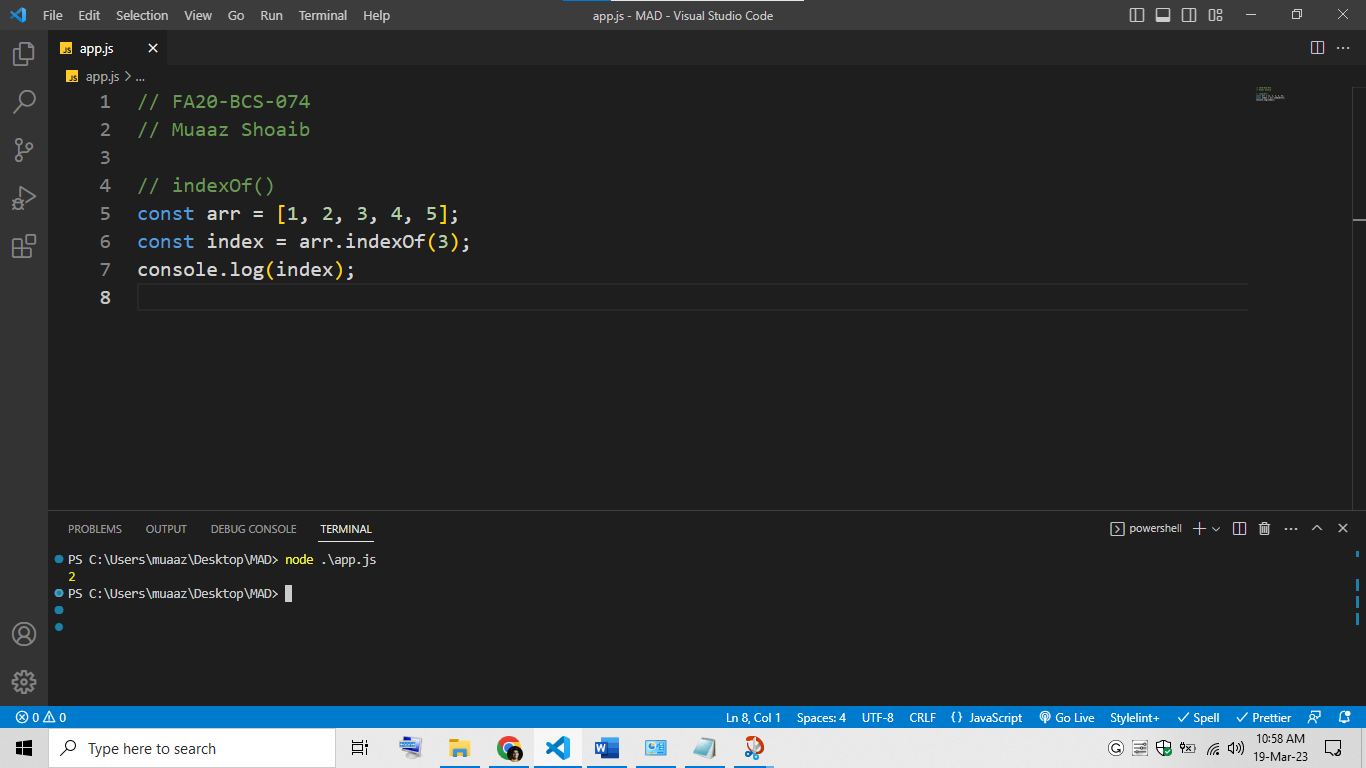
// Muaaz Shoaib

// indexOf()

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

const index = arr.indexOf(3);

console.log(index);



# Q2: String functions in JavaScript with example.

## length

This property returns the length of a string.

// FA20-BCS-074

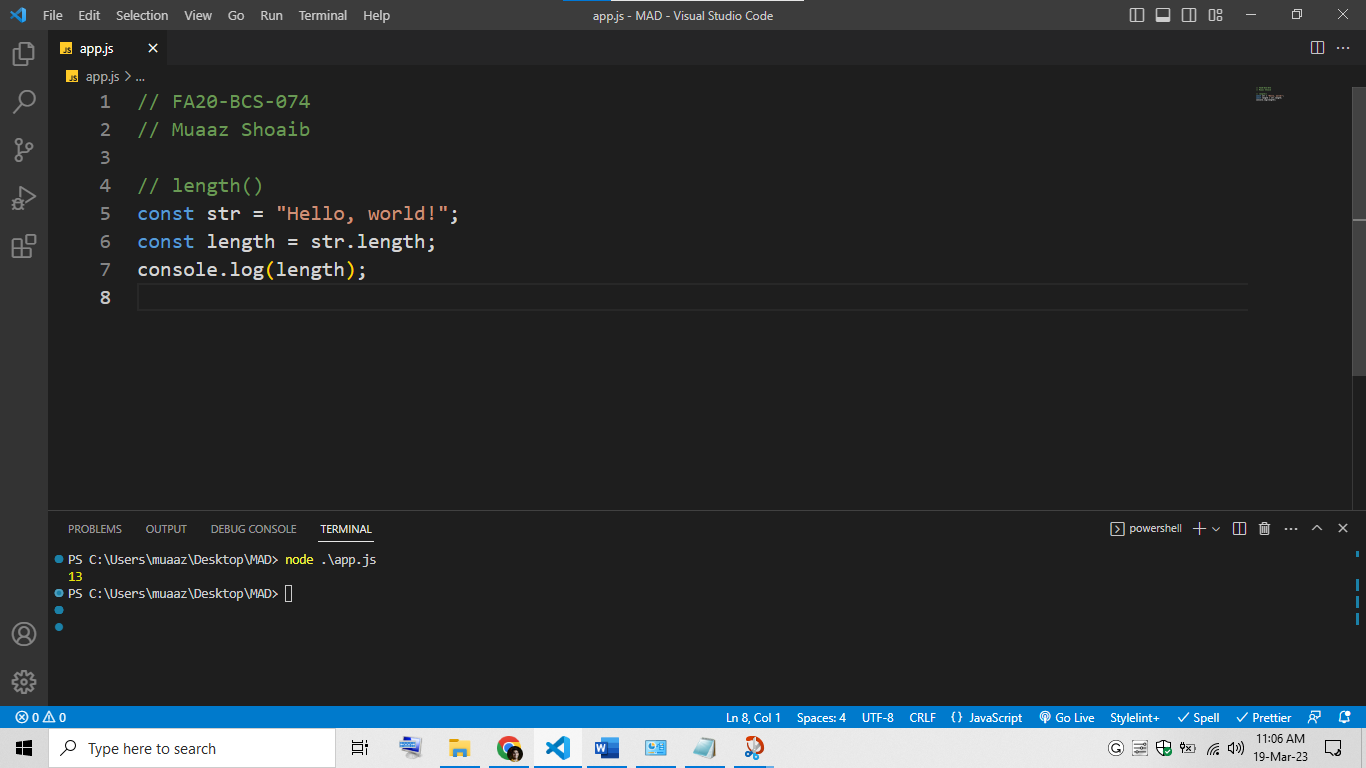
// Muaaz Shoaib

// length()

const str = "Hello, world!";

const length = str.length;

console.log(length);



## charAt

This method returns the character at a specified index in a string.

// FA20-BCS-074

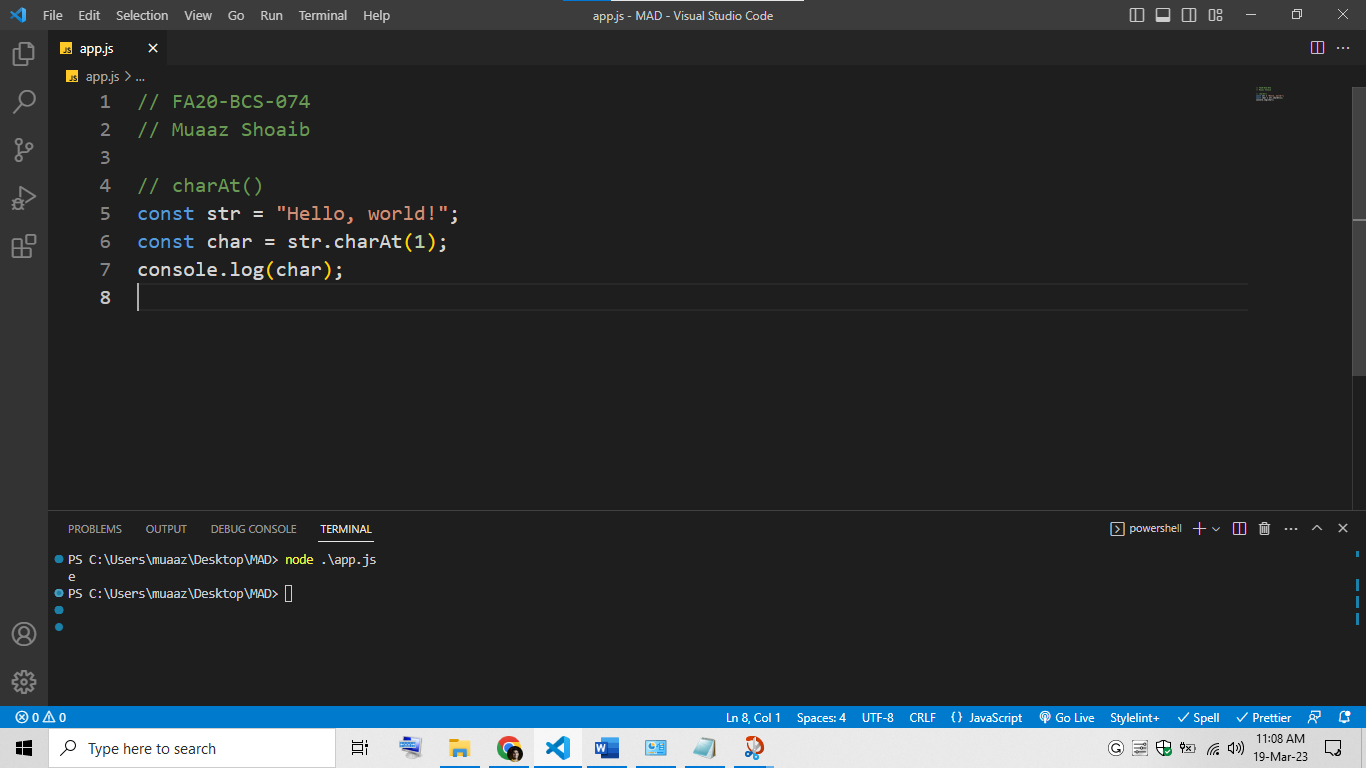
// Muaaz Shoaib

// charAt()

const str = "Hello, world!";

const char = str.charAt(1);

console.log(char);



## concat

This method concatenates two or more strings and returns the new string.

// FA20-BCS-074

// Muaaz Shoaib

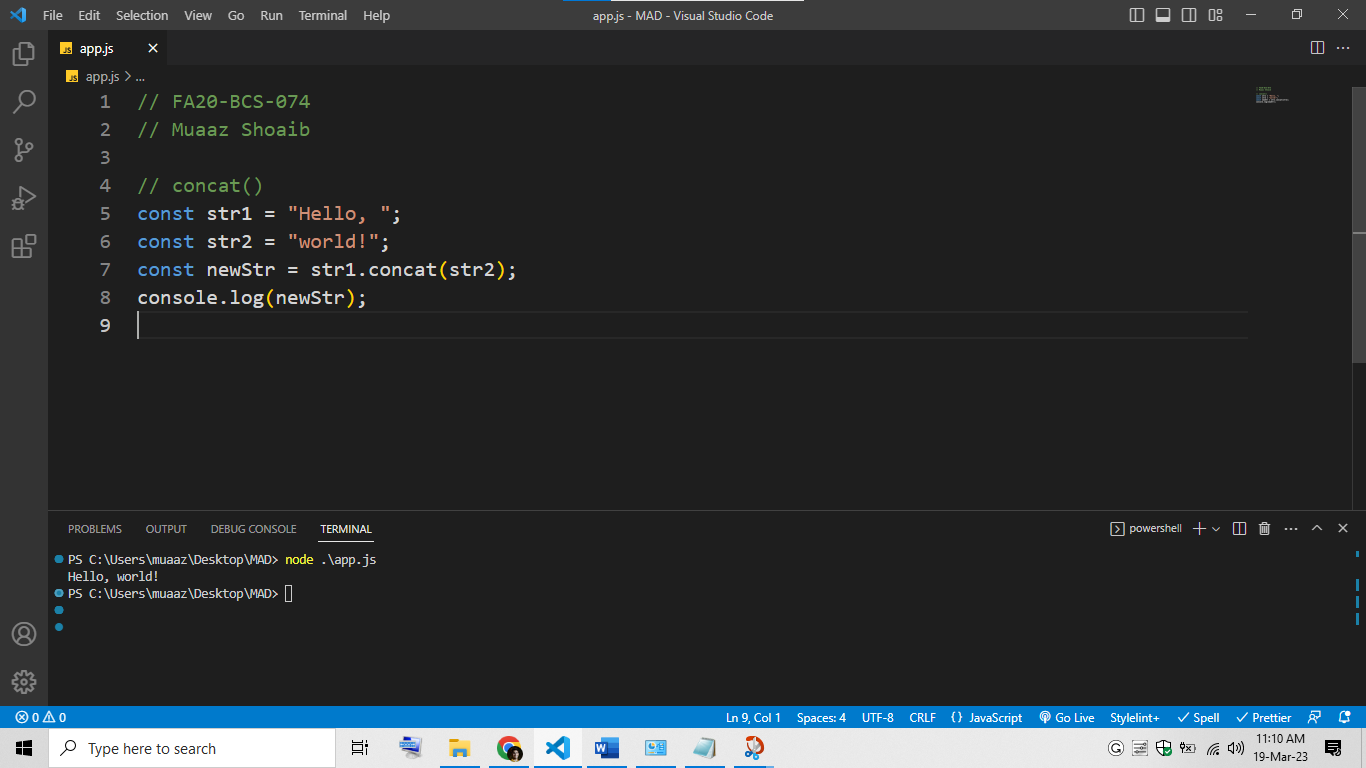
// concat()

const str1 = "Hello, ";

const str2 = "world!";

const newStr = str1.concat(str2);

console.log(newStr);



## slice

This method returns a portion of a string into a new string.

// FA20-BCS-074

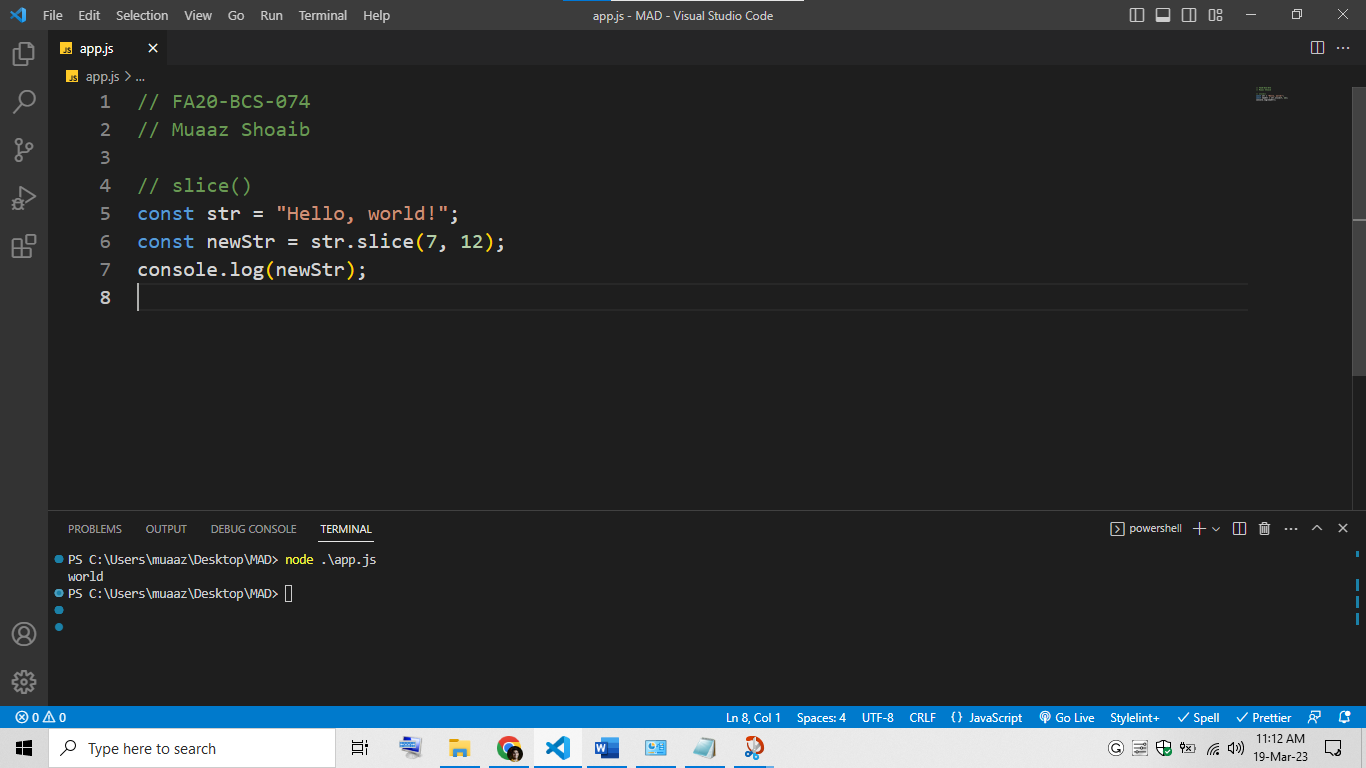
// Muaaz Shoaib

// slice()

const str = "Hello, world!";

const newStr = str.slice(7, 12);

console.log(newStr);



## toLowerCase

This method returns a new string with all characters in lowercase.

// FA20-BCS-074

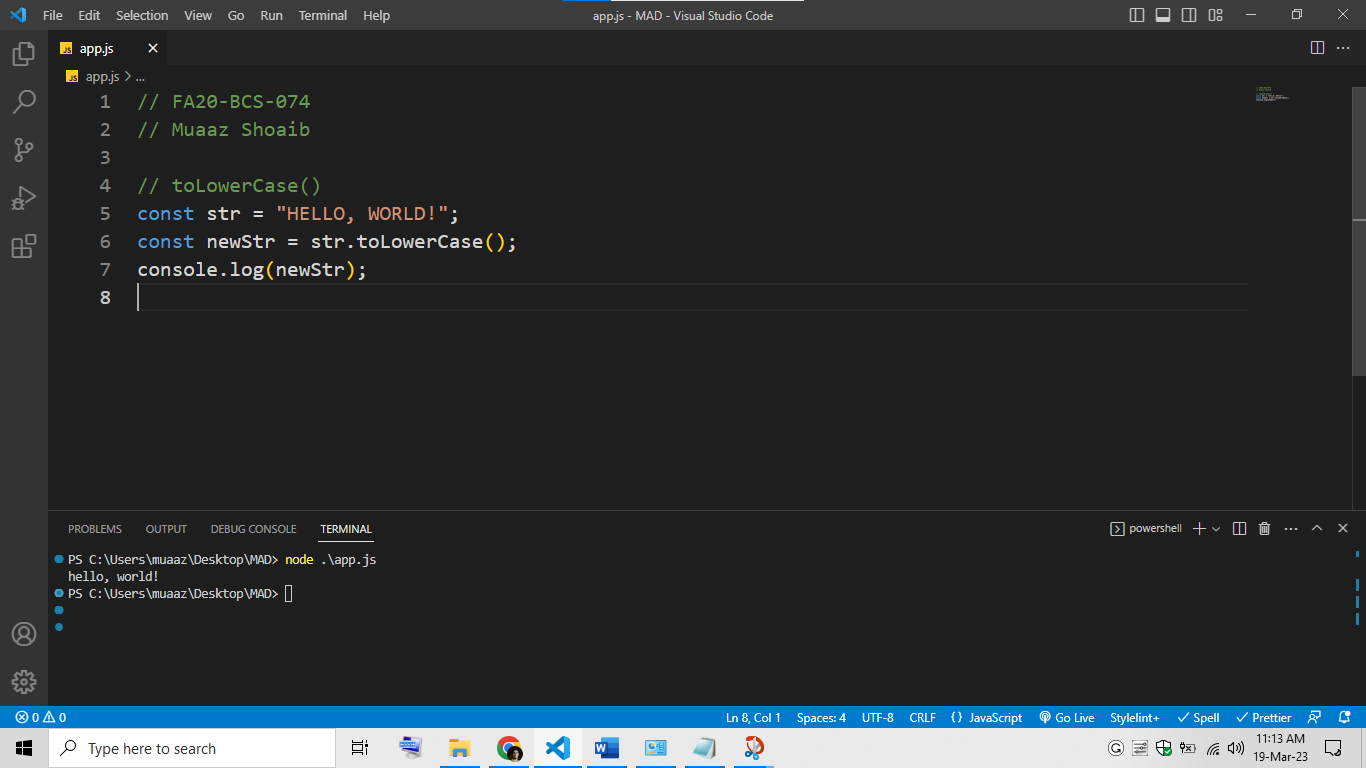
// Muaaz Shoaib

// toLowerCase()

const str = "HELLO, WORLD!";

const newStr = str.toLowerCase();

console.log(newStr);



## toUpperCase

This method returns a new string with all characters in uppercase.

// FA20-BCS-074

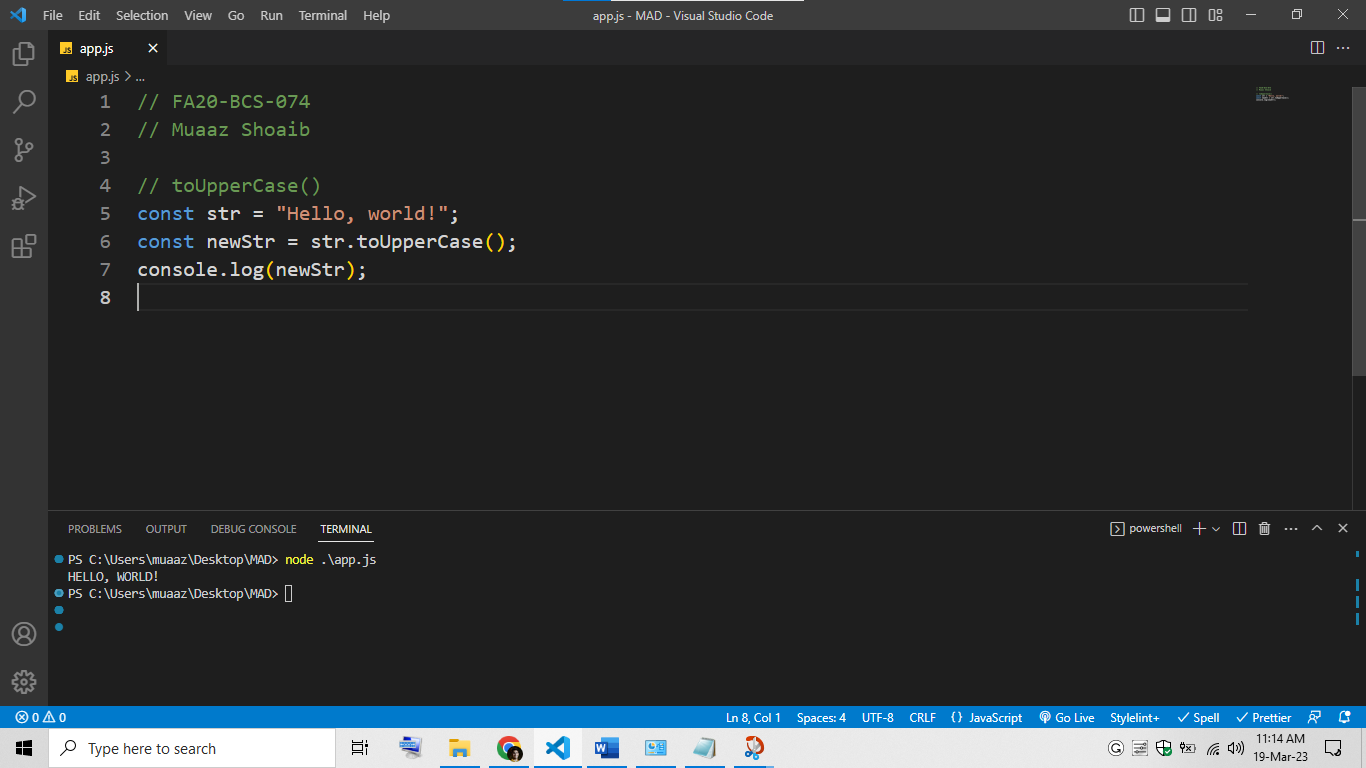
// Muaaz Shoaib

// toUpperCase()

const str = "Hello, world!";

const newStr = str.toUpperCase();

console.log(newStr);



## trim

This method removes whitespace from both ends of a string.

// FA20-BCS-074

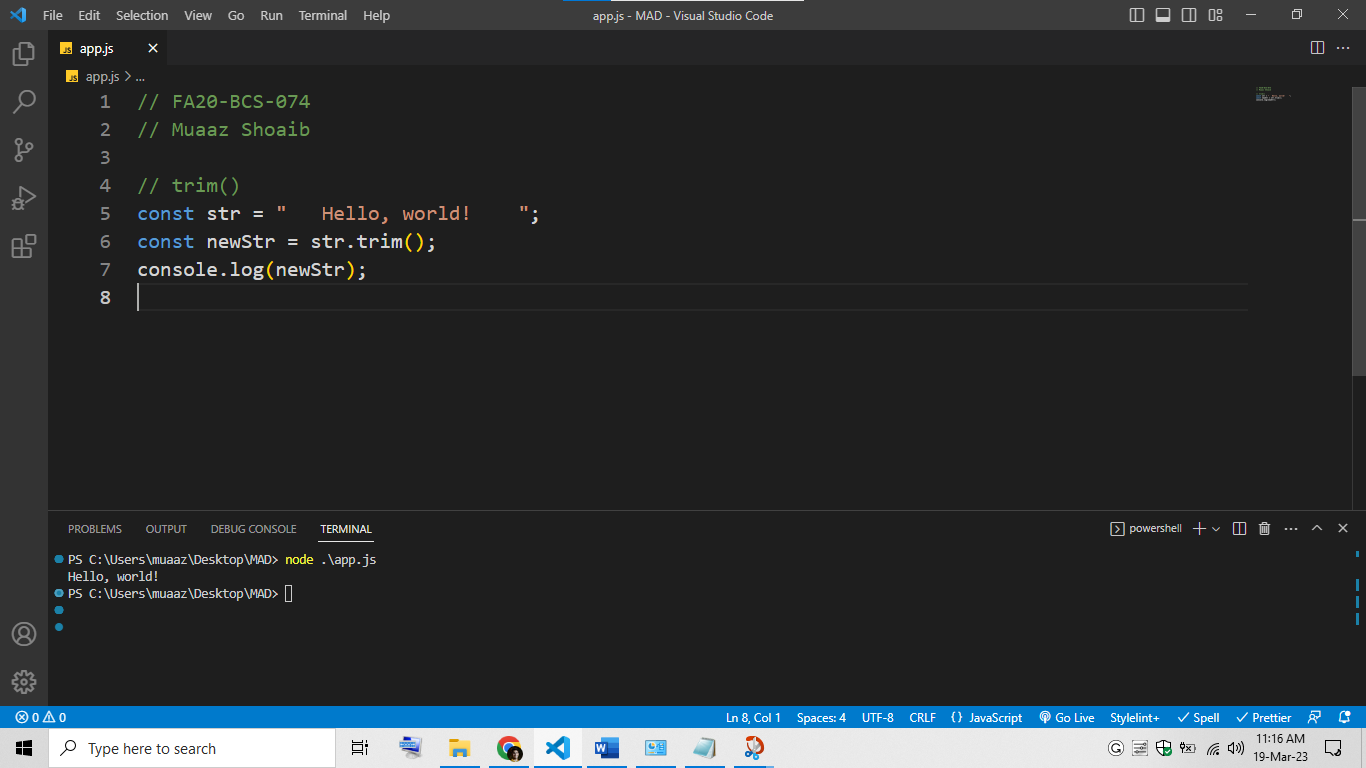
// Muaaz Shoaib

// trim()

const str = "   Hello, world!    ";

const newStr = str.trim();

console.log(newStr);



## indexOf

This method returns the index of the first occurrence of a specified value in a string, or -1 if it is not found.

// FA20-BCS-074

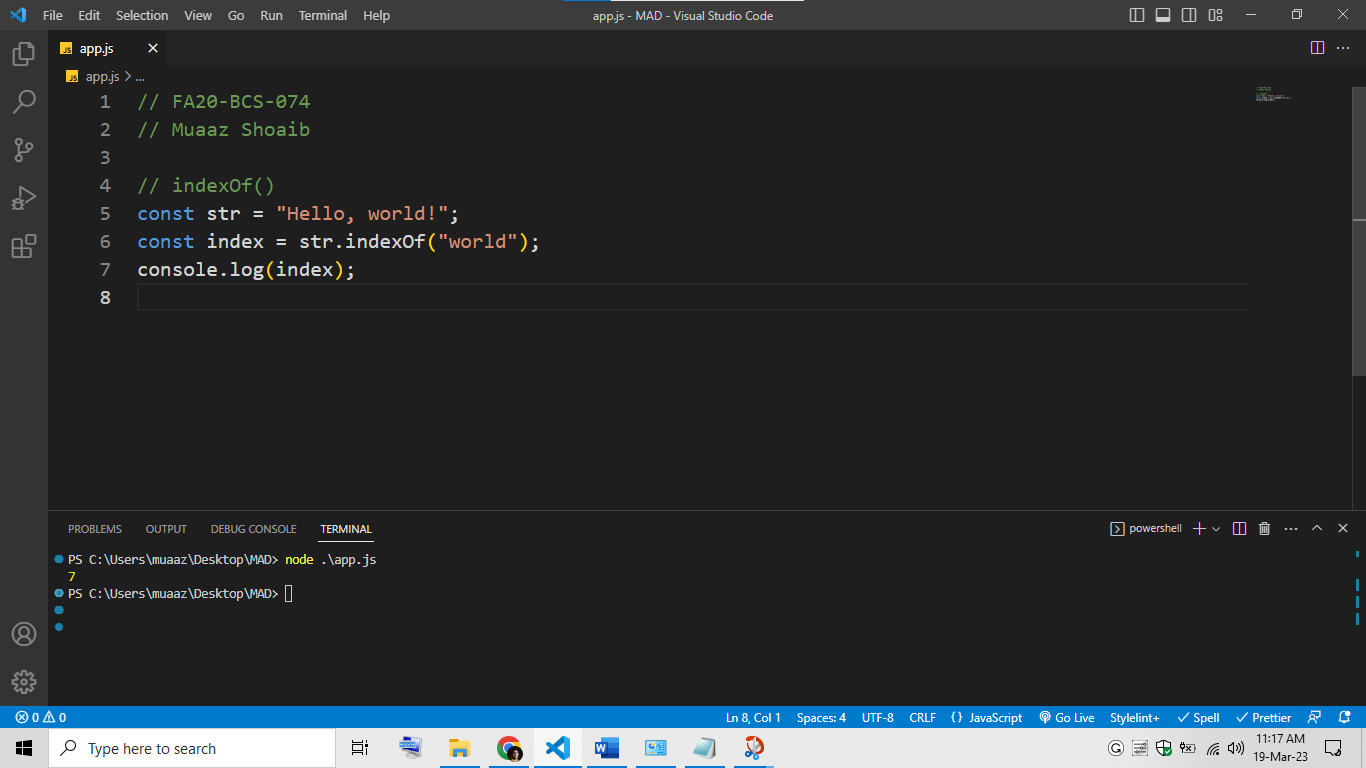
// Muaaz Shoaib

// indexOf()

const str = "Hello, world!";

const index = str.indexOf("world");

console.log(index);



## replace

This method replaces a specified value with another value in a string.

// FA20-BCS-074

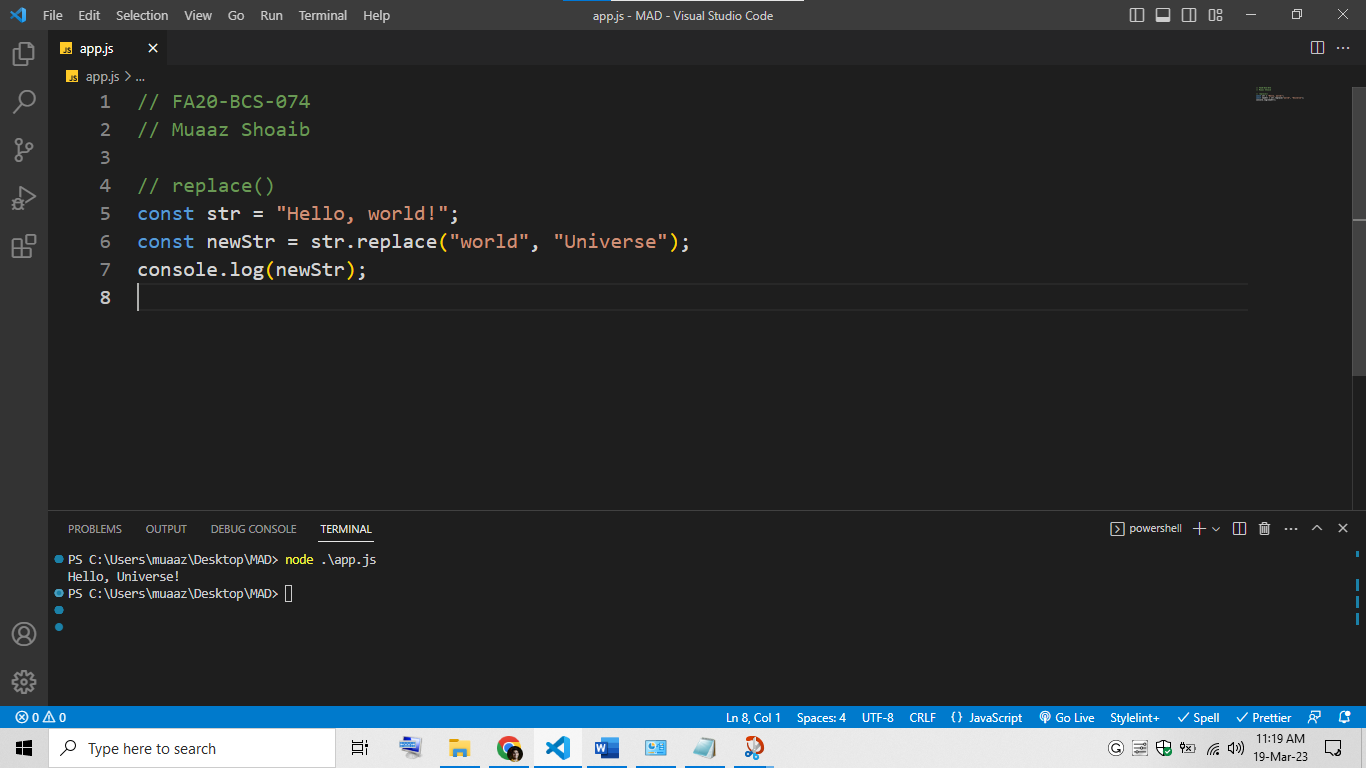
// Muaaz Shoaib

// replace()

const str = "Hello, world!";

const newStr = str.replace("world", "Universe");

console.log(newStr);



## split

This method splits a string into an array of substrings based on a specified separator.

// FA20-BCS-074

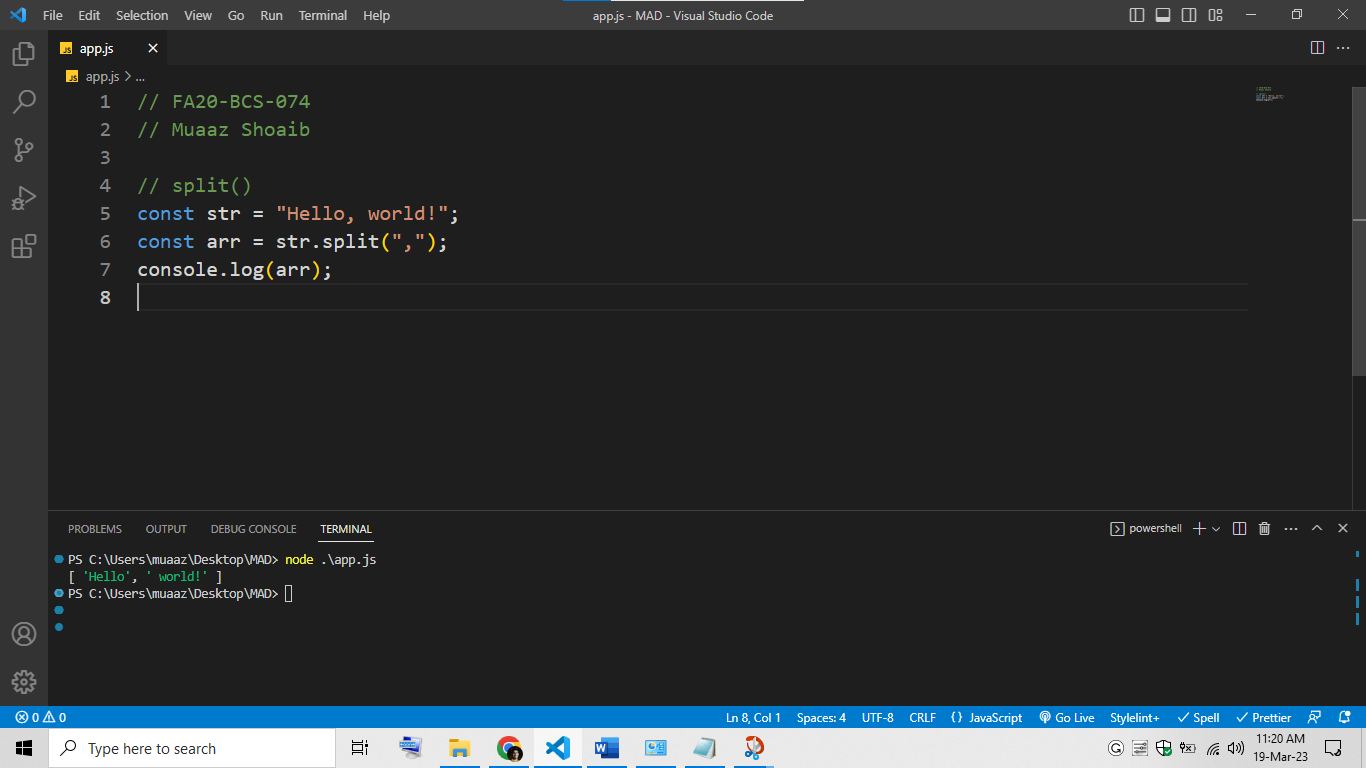
// Muaaz Shoaib

// split()

const str = "Hello, world!";

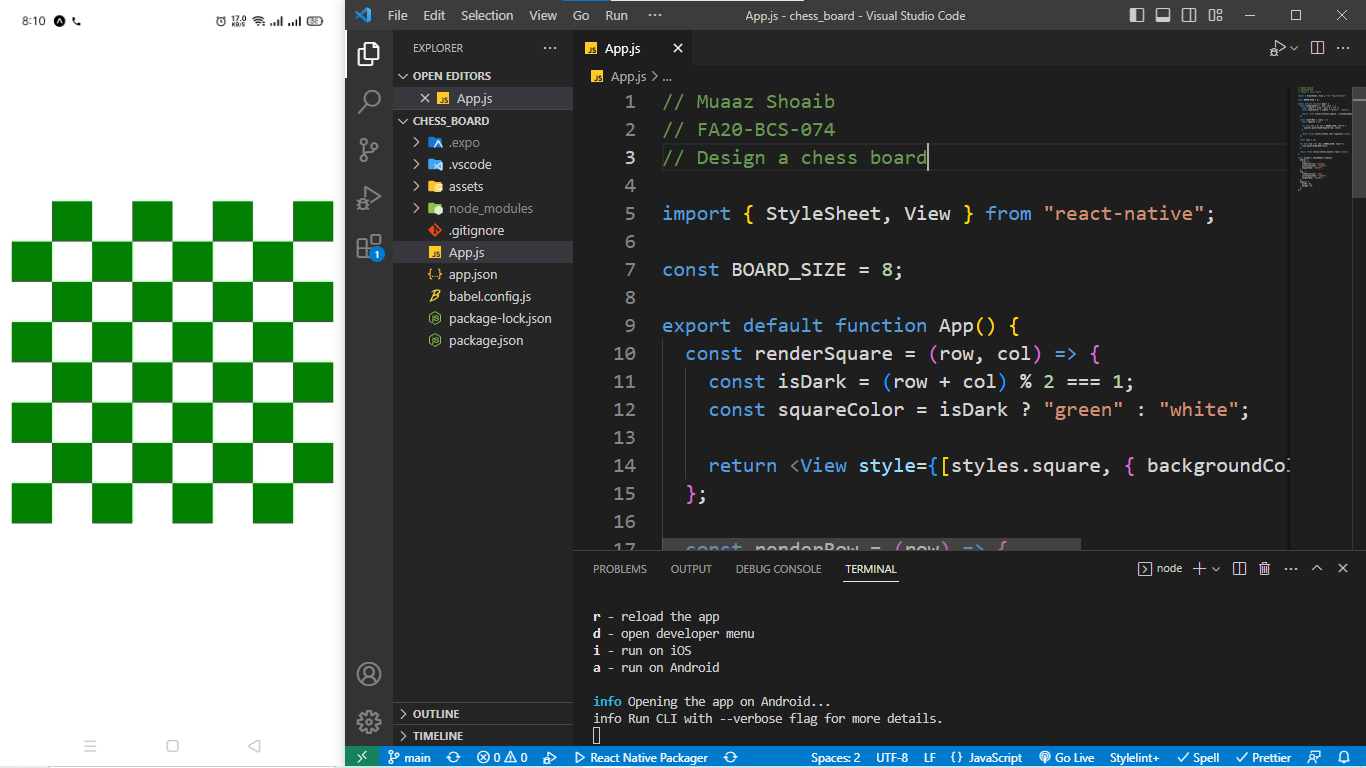
const arr = str.split(",");

console.log(arr);



# Q3: Design chess board UI in react native?

You will find a complete project by clicking on this link: <https://github.com/muaazshoaib/chess_board>



// Muaaz Shoaib

// FA20-BCS-074

// Design a chess board

import { StyleSheet, View } from "react-native";

const BOARD\_SIZE = 8;

export default function App() {

  const renderSquare = (row, col) => {

    const isDark = (row + col) % 2 === 1;

    const squareColor = isDark ? "green" : "white";

    return <View style={[styles.square, { backgroundColor: squareColor }]} />;

  };

  const renderRow = (row) => {

    const squares = [];

    for (let col = 0; col < BOARD\_SIZE; col++) {

      squares.push(renderSquare(row, col));

    }

    return <View style={styles.row}>{squares}</View>;

  };

  const rows = [];

  for (let row = 0; row < BOARD\_SIZE; row++) {

    rows.push(renderRow(row));

  }

  return <View style={styles.board}>{rows}</View>;

}

const styles = StyleSheet.create({

  board: {

    flex: 1,

    flexDirection: "column",

    justifyContent: "center",

    alignItems: "center",

  },

  row: {

    flexDirection: "row",

    justifyContent: "center",

    alignItems: "center",

  },

  square: {

    width: 42,

    height: 42,

  },

});