Logo, company name

Description automatically generated

Assignment # 02

Course Title : Mobile App Development

Submitted to : Muhammad kamran

Submitted by : **Nadeem Mumtaz**

Registration No. : **CIIT /FA20-BCS-060/ATK**

Date of Submission : 15 April, 2022

**Question 1 :String functions:**

1. **toUpperCase() –** Converts a string to uppercase letters.

Let str = ‘hello world’;

Let upperStr = str.toUpperCase();

Console.log(upperStr); // Output: HELLO WORLD

1. **toLowerCase() –** Converts a string to lowercase letters.

Let str = ‘HELLO WORLD’;

Let lowerStr = str.toLowerCase();

Console.log(lowerStr); // Output: hello world

1. **Concat() –** Joins two or more strings.

Let str1 = ‘Hello’;

Let str2 = ‘World’;

Let concatStr = str1.concat(‘ ‘, str2);

Console.log(concatStr); // Output: Hello World

1. **CharAt() –** Returns the character at a specified index in a string.

Let str = ‘Hello’;

Let char = str.charAt(1);

Console.log(char); // Output: e

1. **indexOf() –** Returns the position of the first occurrence of a specified value in a string.

Let str = ‘Hello World’;

Let index = str.indexOf(‘o’);

Console.log(index); // Output: 4

**Question 2: Array functions**

1.**map() -** Creates a new array with the results of calling a provided function on every element in the original array.

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

const newArr = arr.map(item => item \* 2);

console.log(newArr); // Output: [2, 4, 6, 8, 10]

2.**filter() -** Creates a new array with all elements that pass the test implemented by the provided function.

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

const newArr = arr.filter(item => item % 2 === 0);

console.log(newArr); // Output: [2, 4]

3.**reduce() -** Applies a function against an accumulator and each element in the array (from left to right) to reduce it to a single value.

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

const sum = arr.reduce((accumulator, currentValue) => accumulator + currentValue);

console.log(sum); // Output: 15

4. **forEach() -** Executes a provided function once for each array element.

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

arr.forEach(item => console.log(item)); // Output: 1, 2, 3, 4, 5

5.**slice() -** Returns a shallow copy of a portion of an array into a new array object selected from begin to end (end not included).

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

const newArr = arr.slice(1, 3);

console.log(newArr); // Output: [2, 3]

**Question 3 :Chess board design in react native**

**Code:**

Import React from ‘react’;

Import { View, StyleSheet } from ‘react-native’;

Const styles = StyleSheet.create({

Container: {

Flex: 1,

flexDirection: ‘column’,

alignItems: ‘center’,

justifyContent: ‘center’,

},

Row: {

Flex: 1,

flexDirection: ‘row’,

alignItems: ‘center’,

justifyContent: ‘center’,

},

Square: {

Width: 50,

Height: 50,

},

blackSquare: {

backgroundColor: ‘black’,

},

whiteSquare: {

backgroundColor: ‘white’,

},

});

Const ChessBoard = () => {

Const renderSquare = (I, j) => {

Const isBlack = (I + j) % 2 === 1;

Const squareColor = isBlack ? styles.blackSquare : styles.whiteSquare;

Return (

<View style={[styles.square, squareColor]} key={`${i}${j}`} />

);

};

Const renderRow = (i) => {

Const row = [];

For (let j = 0; j < 8; j++) {

Row.push(renderSquare(I, j));

}

Return (

<View style={styles.row} key={i}>

{row}

</View>

);

};

Const rows = [];

For (let I = 0; I < 8; i++) {

Rows.push(renderRow(i));

}

Return (

<View style={styles.container}>

{rows}

</View>

);

};

Export default ChessBoard;