Task-1

Aim: Variables and Data Types

Description: Describe a variable using var ,let and const. Assign different data types to each variable and print their values.

Source Code:

```
var a;
var b;
const c=3;
a=1;
b=2;
console.log(a);
console.log(b);
console.log(c);
```

Output:

```
node /tmp/ikgrd2pkpz.js
1
2
3
```

Learning Outcome:
Understand various technologies and trends impacting single page web applications.

Task-2

Aim : Operators and Expressions.

Description : Write a function that takes two numbers as arguments and returns their sum, difference,product, and quotient using arithmetic operators.

Source Code :

```
var a=prompt("Enter 1st Number : ")
console.log(a);
var aa=parseInt(a);
var b=prompt("Enter 2nd Number");
console.log(b);
var bb=parseInt(b);
```

```
var add=add(aa,bb);
var sub=sub(aa,bb);
var mul=mul(aa,bb);
var div=div(aa,bb);
function add(n1,n2)
   {
      return (n1+n2);
   }
function sub(n1,n2)
   {
     return (n1-n2);
   }
function mul(n1,n2)
   {
      return (n1*n2);
   }
function div(n1,n2)
   {
      return (n1/n2);
   }
console.log("Addition of "+a+" and "+b+" is "+add);
console.log("Subtraction of "+a+" and "+b+" is "+sub);
console.log("Multiplication of "+a+" and "+b+" is "+mul);
console.log("Division of "+a+" and "+b+" is "+div);
```

Output :

```
node /tmp/uVAQkj1Vb5.js
Enter 1st Number : 5
5
Enter 2nd Number6
6
Addition of 5 and 6 is 11
Subtraction of 5 and 6 is -1
Multiplication of 5 and 6 is 30
Division of 5 and 6 is 0.8333333333333334
```

Learning Outcomes : Understand various technologies and trends impacting single page web applications.

Task-3

Aim : Control Flow

Description : Write a program that prompts the user to enter their age. Based on their age, display different messages:

○ If the age is less than 18, display "You are a minor."

○ If the age is between 18 and 65, display "You are an adult."

○ If the age is 65 or older, display "You are a senior citizen."

Source Code :

```
var a=parseInt(prompt("Enter Your Age : "));
console.log(a);
if(a<18)
{
   console.log("You are Minor");
}
else if(a>18 && a<65)
{
   console.log("You are Adult");
}
else{
   console.log("You are senior citizen");
}
```

Output :

```
node /tmp/uVAQkj1Vb5.js
Enter Your Age : 20
20
You are Adult
```

Learning Outcomes : Understand various technologies and trends impacting single page web applications.

Task-4

Aim : Functions

Description : Write a function that takes an array of salary as an argument and returns the min/max salary in the array**.**

Source Code :

```
var userInput = prompt("Enter salaries, separated by commas:");
var array = userInput.split(',');

array = array.map(function(element) {
  return element.trim();
});

console.log(array);
function findMinMax() {

    let minValue = Math.min(...array);
    let maxValue = Math.max(...array);

    console.log("Minimum element is : " + minValue);
    console.log("Maximum Element is : " + maxValue);
}

findMinMax()
```

Output :

```
node /tmp/uVAQkj1Vb5.js
Enter salaries, separated by commas:1000,2403,32985,1243,4743,546
[ '1000', '2403', '32985', '1243', '4743', '546' ]
Minimum element is : 546
Maximum Element is : 32985
```

Learning Outcomes : Understand various technologies and trends impacting single page web applications.

Task-5

Aim : Arrays and Objects.

Description : Create an array of your favorite books. Write a function that takes the array as an argument and displays each book title on a separate line.

Source Code :

```
var userInput = prompt("Enter Book Names, separated by commas:");
var array = userInput.split(',');

array.forEach((element) => {
  console.log(element.trim());
})
```

Output :

```
node /tmp/uVAQkj1Vb5.js
Enter Book Names, separated by commas:Abc,def,ghi,jkl,mno
Abc
def
ghi
jkl
mno
```

Learning Outcomes : Understand various technologies and trends impacting single page web applications.

Task-6

Aim : Scope and Hoisting

Description : Declare a variable inside a function and try to access it outside the function. Observe the scope behavior and explain the results. [var vs let vs const]

Source Code :

1 : using var

```
function myFunction() {
  var x = 10;
}

myFunction();
console.log(x); // Output: 10
```

2 : using let

```
function myFunction() {
  let x = 10;
  console.log(x); // Output: 10
}

myFunction();
console.log(x); // ReferenceError: x is not defined
```

3 : using const

```
function myFunction() {
  const x = 10;
  console.log(x); // Output: 10
}

myFunction();
console.log(x); // ReferenceError: x is not defined
```

Learning Outcome : Understand various technologies and trends impacting single page web applications.

Task-7

Aim : DOM Manipulation.

Description : Create an HTML page with a button. Write JavaScript code that adds an event listener to the button and changes its text when clicked.

Source Code :

```
<!DOCTYPE html>
<html>
<head>
  <title>Button Event Listener Example</title>
</head>
<body>
  <button id="myButton">Click Me</button>

  <script>
       var button = document.getElementById("myButton");

      button.addEventListener("click", function() {
         button.textContent = "Clicked!";
    });
  </script>
</body>
</html>
```

Learning Outcome : Understand various technologies and trends impacting single page web applications.

Task-8

Aim : Error Handling

Description : Write a function that takes a number as an argument and throws an error if the number is negative. Handle the error and display a custom error message.

Source Code :

```
function validateNumber(number) {
  if (number < 0) {
    throw new Error("Number cannot be negative");
  }

  // Continue with the rest of the function if the number is valid
  // ...
}
```

Learning Outcome : Understand various technologies and trends impacting single page web applications.

Task-9

Aim : Asynchronous JavaScript

Description : Write a function that uses setTimeout to simulate an asynchronous operation. Use a callback function to handle the result.

Source Code :

```
function simulateAsyncOperation(callback) {
  setTimeout(function() {
    // Simulate some asynchronous task
    var result = 42; // This can be any result or data you want to pass to the
callback

    // Call the callback function with the result
    callback(result);
  }, 2000); // Simulate a delay of 2000 milliseconds (2 seconds)
}
```

Learning Outcomes : Understand various technologies and trends impacting single page web applications.