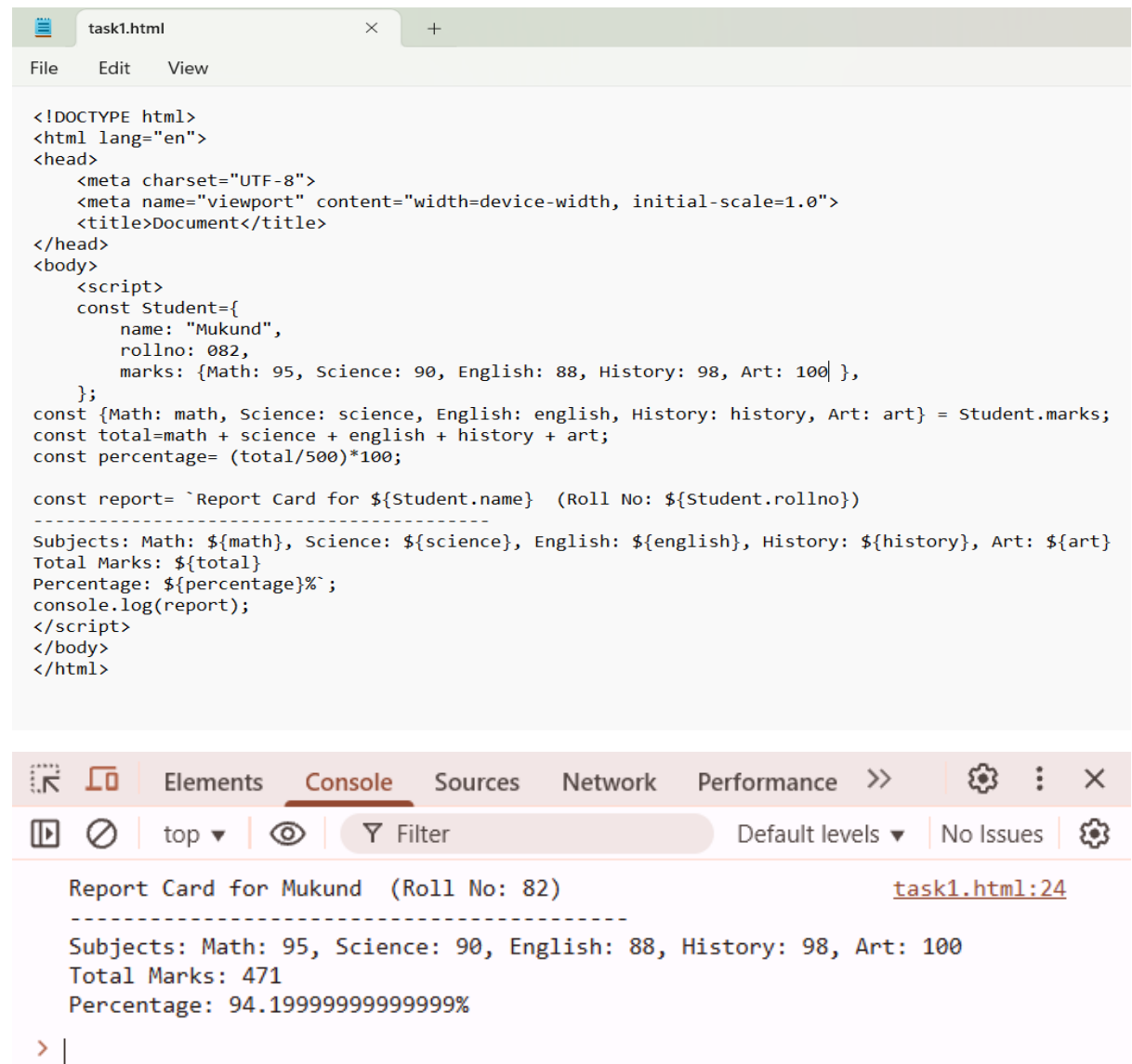


Name: Mukund Kuthe

3rd Year Section B (B1)

Practical 2 (JavaScript)

Task 1:



The screenshot shows a web browser window with a single tab titled 'task1.html'. The browser's developer tools are open, showing the 'Console' tab. The console displays the output of a JavaScript script that calculates a student's report card. The script defines a student object with the name 'Mukund', roll number '082', and marks for Math (95), Science (90), English (88), History (98), and Art (100). It then calculates the total marks (471) and the percentage (94.19999999999999%). The console output shows the report card details for Mukund (Roll No: 82), including the subjects, total marks, and percentage.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    const Student={
      name: "Mukund",
      rollno: 082,
      marks: {Math: 95, Science: 90, English: 88, History: 98, Art: 100| },
    };
    const {Math: math, Science: science, English: english, History: history, Art: art} = Student.marks;
    const total=math + science + english + history + art;
    const percentage= (total/500)*100;

    const report= `Report Card for ${Student.name} (Roll No: ${Student.rollno})
    -----
    Subjects: Math: ${math}, Science: ${science}, English: ${english}, History: ${history}, Art: ${art}
    Total Marks: ${total}
    Percentage: ${percentage}%`;
    console.log(report);
  </script>
</body>
</html>
```

Report Card for Mukund (Roll No: 82) task1.html:24

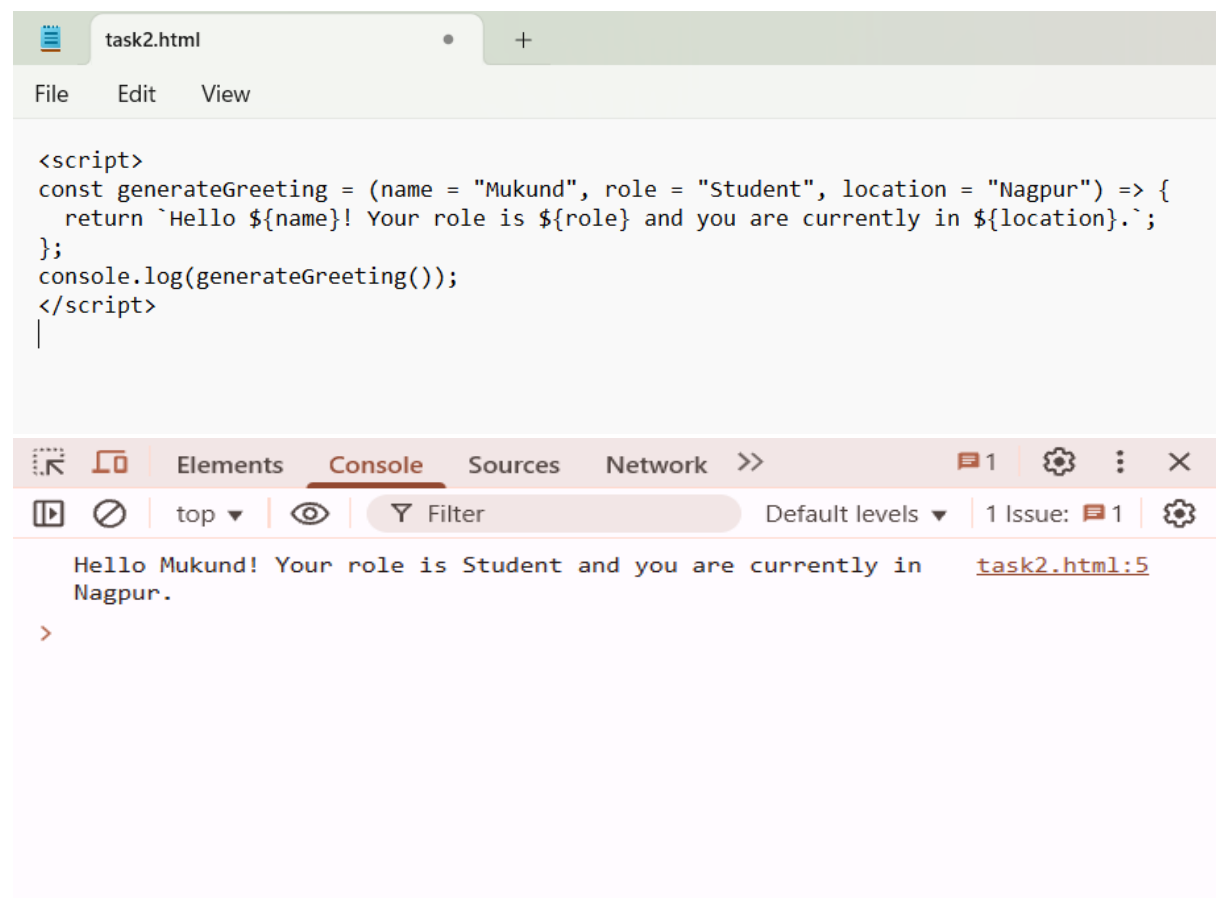
Subjects: Math: 95, Science: 90, English: 88, History: 98, Art: 100

Total Marks: 471

Percentage: 94.19999999999999%

> |

Task 2:



The screenshot shows a web browser with a single tab titled "task2.html". The browser's developer tools are open, with the "Console" tab selected. The console displays the output of a JavaScript function: "Hello Mukund! Your role is Student and you are currently in Nagpur." The message is followed by a red squiggly line indicating an error, and the source is listed as "task2.html:5".

```
<script>
const generateGreeting = (name = "Mukund", role = "Student", location = "Nagpur") => {
  return `Hello ${name}! Your role is ${role} and you are currently in ${location}.`;
};
console.log(generateGreeting());
</script>
```

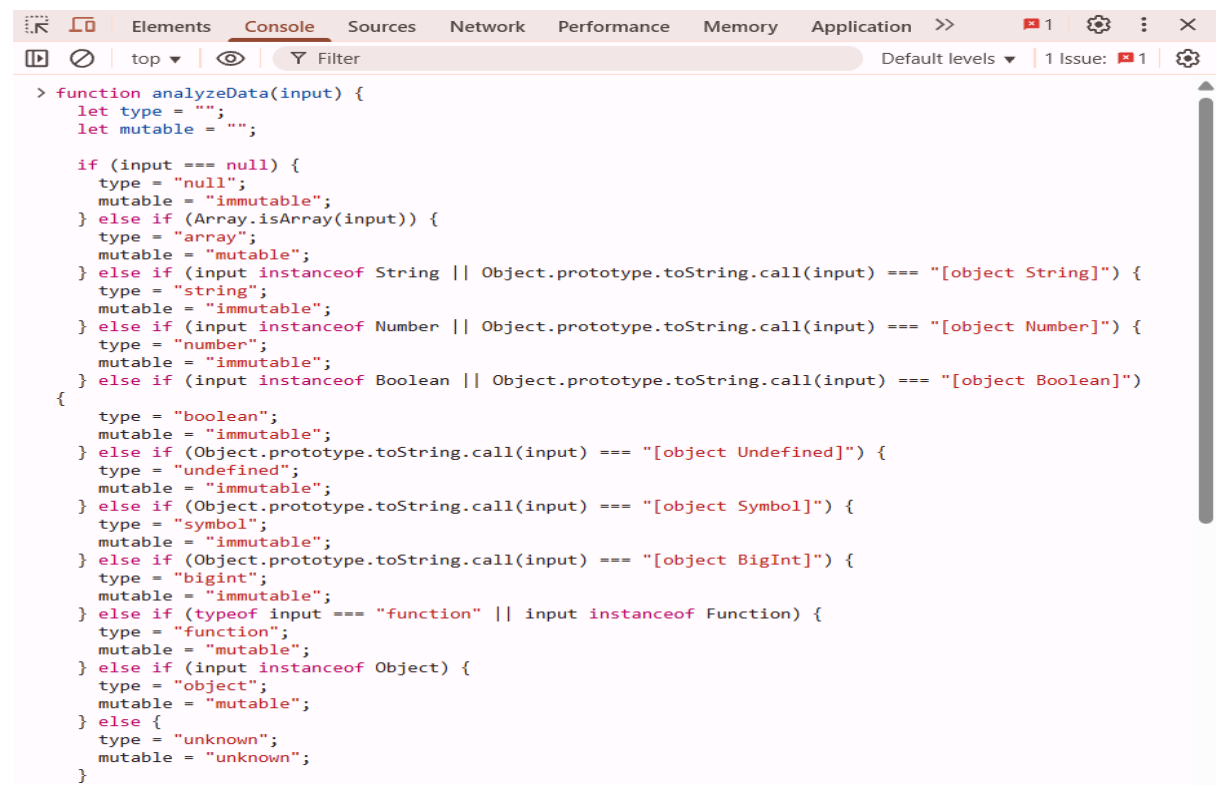
Elements Console Sources Network >> 1 Issue: 1

top Filter Default levels 1 Issue: 1

Hello Mukund! Your role is Student and you are currently in Nagpur.

>

Task 3:



The screenshot shows a web browser with a single tab titled "task3.html". The browser's developer tools are open, with the "Console" tab selected. The console displays the output of a JavaScript function: "Hello Mukund! Your role is Student and you are currently in Nagpur." The message is followed by a red squiggly line indicating an error, and the source is listed as "task3.html:5".

```
> function analyzeData(input) {
  let type = "";
  let mutable = "";

  if (input === null) {
    type = "null";
    mutable = "immutable";
  } else if (Array.isArray(input)) {
    type = "array";
    mutable = "mutable";
  } else if (input instanceof String || Object.prototype.toString.call(input) === "[object String]") {
    type = "string";
    mutable = "immutable";
  } else if (input instanceof Number || Object.prototype.toString.call(input) === "[object Number]") {
    type = "number";
    mutable = "immutable";
  } else if (input instanceof Boolean || Object.prototype.toString.call(input) === "[object Boolean]") {
    type = "boolean";
    mutable = "immutable";
  } else if (Object.prototype.toString.call(input) === "[object Undefined]") {
    type = "undefined";
    mutable = "immutable";
  } else if (Object.prototype.toString.call(input) === "[object Symbol]") {
    type = "symbol";
    mutable = "immutable";
  } else if (Object.prototype.toString.call(input) === "[object BigInt]") {
    type = "bigint";
    mutable = "immutable";
  } else if (typeof input === "function" || input instanceof Function) {
    type = "function";
    mutable = "mutable";
  } else if (input instanceof Object) {
    type = "object";
    mutable = "mutable";
  } else {
    type = "unknown";
    mutable = "unknown";
  }
}
```

Elements Console Sources Network Performance Memory Application >> 1 Issue: 1

top Filter Default levels 1 Issue: 1

>

```

    console.log(`Type: ${type}`);
    console.log(`Mutability: ${mutable}`);
    console.log(`Summary: The given input is of type '${type}' and it is ${mutable}.`);
  }

  analyzeData([1, 2, 3]);
  analyzeData("Hello");
  analyzeData(null);
  analyzeData({ key: "value" });

```

Type: array	VM70:40
Mutability: mutable	VM70:41
Summary: The given input is of type 'array' and it is mutable.	VM70:42
Type: string	VM70:40
Mutability: immutable	VM70:41
Summary: The given input is of type 'string' and it is immutable.	VM70:42
Type: null	VM70:40
Mutability: immutable	VM70:41
Summary: The given input is of type 'null' and it is immutable.	VM70:42
Type: object	VM70:40
Mutability: mutable	VM70:41
Summary: The given input is of type 'object' and it is mutable.	VM70:42

← undefined

> |

Task 4:

Elements
Console
Sources
Network
>>
1
⚙️
⋮
✕

top ▼
🔍 Filter
Default levels ▼
1 Issue: 🔴 1
⚙️

```

> const colors1 = ["red", "green"];
   const colors2 = ["blue", "yellow"];

   const palette = [...colors1, ...colors2];

   const [first, ...restColors] = palette;
   const last = restColors.pop();
   const others = restColors;

   console.log(`Main colors: ${capitalize(first)} and ${capitalize(last)}.
   Others in the palette: ${others.map(capitalize).join(", ")}.`);

   function capitalize(word) {
     return word.charAt(0).toUpperCase() + word.slice(1);
   }

```

Main colors: Red and Yellow. Others in the palette: Green, Blue.	VM64:10
--	-------------------------

← undefined

>