INTERNSHIP REPORT

WEEK 5 DAY 2

Submitted to:	**Ali Hyder**	Submission Date:	**22nd July, 2025**
Internship Domain:	**Front Development**	Internship Name:	**ProSensia**
Student Name:	**Yasal Qamar**	Roll No.	**S25031**

JavaScript Advanced

Topics: JavaScript Objects & Object Methods

Objective

To understand how to create and use objects in JavaScript, how properties and methods work, and how objects enhance data organization and functionality in JavaScript applications.

Topics Covered

1. JavaScript Objects

- Objects are key-value pairs used to group related data and functionality.

```
const person = {
  name: "Yasal",
  age: 21,
  isStudent: true
};
```

2. Accessing Object Properties

- Dot Notation: person.name
- Bracket Notation: person["age"]

```
<select onchange="alert(this.value)">
```

```
  <option value="HTML">HTML</option>
  <option value="CSS">CSS</option>
  <option value="JavaScript">JavaScript</option>
</select>
```

3. Modifying Object Properties

```
person.age = 22;
person["name"] = "Qamar";
```

4. Adding New Properties

```
person.city = "Haripur";
```

5. Deleting Properties

```
delete person.isStudent;
```

6. Object Methods

- Functions inside objects.

```
const car = {
  brand: "Toyota",
  start: function () {
    return "Car has started!";
  }
};
car.start(); // Output: "Car has started!"
```

7. this Keyword in Object Methods

- Refers to the object itself.

```
const user = {
  name: "Yasal",
  greet: function () {
    return `Hello, my name is ${this.name}`;
```

```
  }
};
```

8. Looping Through Objects

- Using for...in loop.

```
for (let key in person) {
  console.log(key + ": " + person[key]);
}
```

Learning Outcome

- Gained hands-on experience with core JavaScript events used in real-world web applications.
- Understood the difference between inline (onclick, onchange) and modern (addEventListener) event handling.
- Learned to build a user-friendly interface that reacts dynamically to input.

CODING

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Student Grade Calculator</title>
  <style>
    body {
      font-family: 'Segoe UI', sans-serif;
      background-color: #f4f9ff;
      padding: 40px;
    }

    h2 {
      text-align: center;
      color: #2c3e50;
    }
```

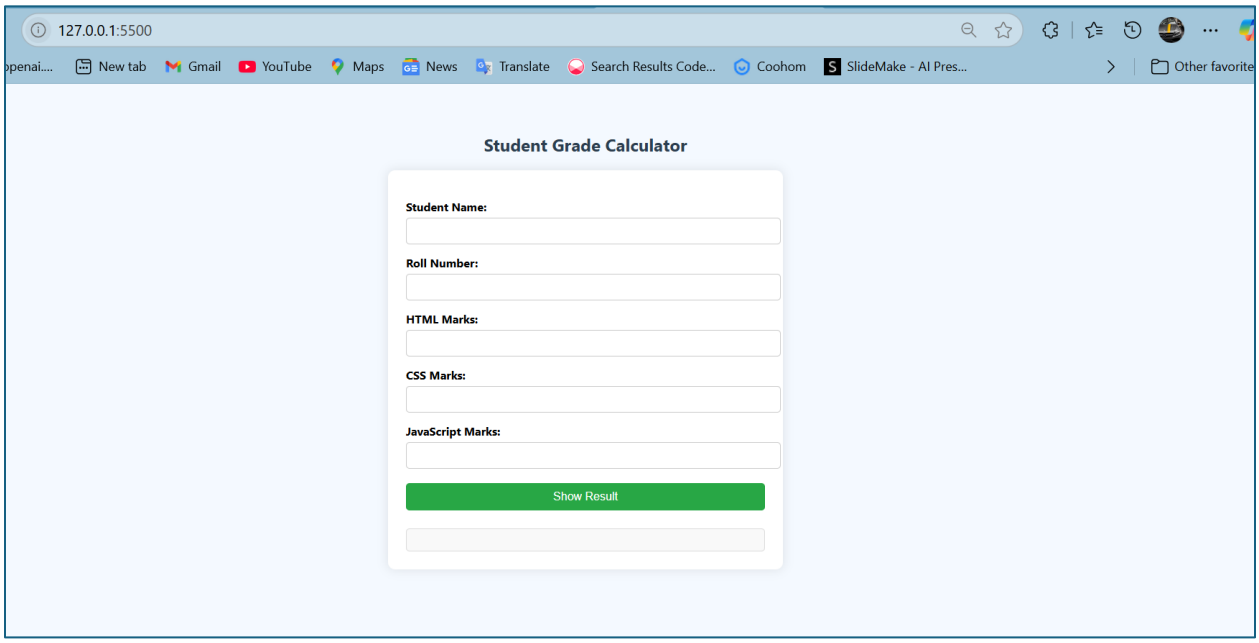
```css
.container {
  max-width: 500px;
  margin: auto;
  background: #ffffff;
  padding: 25px;
  border-radius: 10px;
  box-shadow: 0 0 15px rgba(0,0,0,0.1);
}

label {
  font-weight: bold;
  display: block;
  margin-top: 15px;
}

input {
  width: 100%;
  padding: 10px;
  margin-top: 5px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

button {
  margin-top: 20px;
  width: 100%;
  padding: 10px;
  font-size: 16px;
  background-color: #28a745;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #218838;
}

.output {
  margin-top: 25px;
  background-color: #f9f9f9;
  padding: 15px;
  border: 1px solid #ddd;
  border-radius: 5px;
```

```
      font-size: 16px;
    }

    .grade {
      font-weight: bold;
      padding: 6px 12px;
      border-radius: 6px;
      display: inline-block;
    }

    .Aplus { background: #2ecc71; color: white; }
    .A { background: #3498db; color: white; }
    .B { background: #f1c40f; color: black; }
    .C { background: #e67e22; color: white; }
    .Fail { background: #e74c3c; color: white; }
  </style>
</head>
<body>

  <h2>Student Grade Calculator</h2>
  <div class="container">
    <label for="name">Student Name:</label>
    <input type="text" id="name">

    <label for="roll">Roll Number:</label>
    <input type="text" id="roll">

    <label for="html">HTML Marks:</label>
    <input type="number" id="html">

    <label for="css">CSS Marks:</label>
    <input type="number" id="css">

    <label for="js">JavaScript Marks:</label>
    <input type="number" id="js">

    <button onclick="generateResult()">Show Result</button>

    <div class="output" id="resultArea"></div>
  </div>

  <script>
    function generateResult() {
      const name = document.getElementById('name').value;
      const roll = document.getElementById('roll').value;
```

```
      const html = parseFloat(document.getElementById('html').value);
      const css = parseFloat(document.getElementById('css').value);
      const js = parseFloat(document.getElementById('js').value);

      const student = {
        name: name,
        rollNo: roll,
        marks: {
          html: html,
          css: css,
          js: js
        },
        getTotal: function () {
          return this.marks.html + this.marks.css + this.marks.js;
        },
        getAverage: function () {
          return this.getTotal() / 3;
        },
        getGrade: function () {
          const avg = this.getAverage();
          if (avg >= 90) return { text: "A+", class: "Aplus" };
          else if (avg >= 80) return { text: "A", class: "A" };
          else if (avg >= 70) return { text: "B", class: "B" };
          else if (avg >= 60) return { text: "C", class: "C" };
          else return { text: "Fail", class: "Fail" };
        },
        display: function () {
          const gradeObj = this.getGrade();
          return `
            <strong>Name:</strong> ${this.name}<br>
            <strong>Roll Number:</strong> ${this.rollNo}<br>
            <strong>Total Marks:</strong> ${this.getTotal()} / 300<br>
            <strong>Average:</strong> ${this.getAverage().toFixed(2)}<br>
            <strong>Grade:</strong> <span class="grade
${gradeObj.class}">${gradeObj.text}</span>
          `;
        }
      };

      document.getElementById('resultArea').innerHTML = student.display();
    }
  </script>

</body>
</html>
```

Explanation:

- Objects
- Methods
- Dynamic input handling
- How output is shown and styled



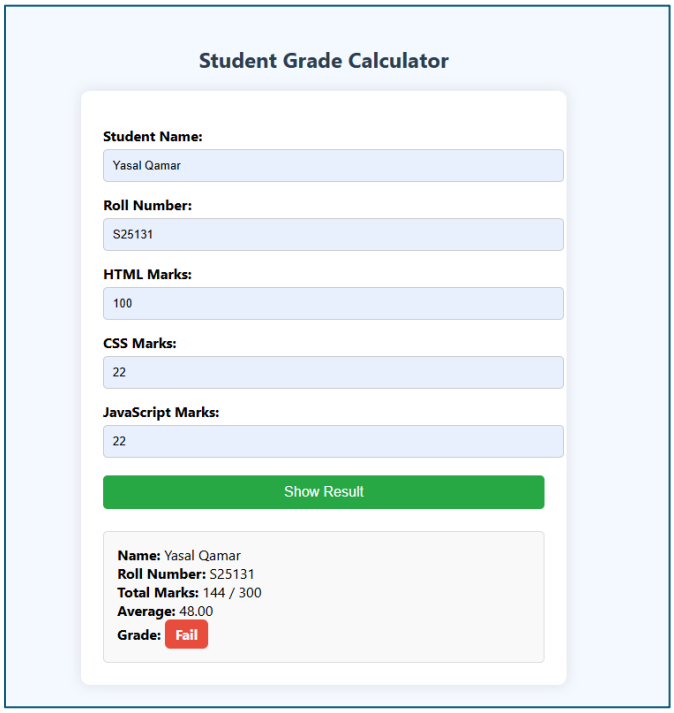
1. **Object Creation from User Input**

```
const student = {
  name: name,
  rollNo: roll,
  marks: {
    html: html,
    css: css,
    js: js
  },
  ...
}
```

Object Properties:

Property	Value
name	You enter it in input
rollNo	You enter it

marks	HTML, CSS, JS marks entered by you



2. Object Methods (Functions inside the object)

These methods act on the data inside the object.

Method 1: getTotal()

```
getTotal: function () {
  return this.marks.html + this.marks.css + this.marks.js;
}
```

Method 2: getAverage()

```
getAverage: function () {
  return this.getTotal() / 3;
}
```

Method 3: getGrade()

```
getGrade: function () {
  const avg = this.getAverage();
```

```
  if (avg >= 90) return { text: "A+", class: "Aplus" };
  else if (avg >= 80) return { text: "A", class: "A" };
  else if (avg >= 70) return { text: "B", class: "B" };
  else if (avg >= 60) return { text: "C", class: "C" };
  else return { text: "Fail", class: "Fail" };
}
```

Method 4: display()

```
display: function () {
  const gradeObj = this.getGrade();
  return `
    <strong>Name:</strong> ${this.name}<br>
    <strong>Roll No:<strong> ${this.rollNo}<br>
    <strong>Total Number:</strong> ${this.getTotal()} / 300<br>
    <strong>Average:</strong> ${this.getAverage().toFixed(2)}<br>
    <strong>Grade:</strong> <span class="grade ${gradeObj.class}">${gradeObj.text}</span>
  `;
}
```

3. Button Action: generateResult()

Step	What Happens
1	Takes your input values (name, roll, marks)
2	Creates the student object
3	Calls the display() method
4	Inserts the HTML inside #resultArea on the page

CONCUSLION:

Today's session helped me understand the core structure and behavior of JavaScript objects. I now know how to organize data efficiently and use methods to perform actions on that data. These concepts are essential for building dynamic and interactive JavaScript applications.