JAVASCRIPT-DAY-4-Hands-on-Problems-srinu

Problem 1. Responsive_page

1.Problem Statement

Assessment Goal: Ensure learners understand responsiveness and screen adaptability.

Hands-on Tasks:

1. Add viewport meta tag to the HTML page
2. Use media queries to: Change background color on mobile screen Adjust font size for smaller screens
3. Convert navigation into vertical layout on mobile
4. Test the page using browser responsive mode

Expected Outcome:
A webpage that looks different and readable on mobile and desktop screens.

2. Code

Index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <!--Viewport meta tag -->

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Responsive Website Demo</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <!-- Navigation -->

  <nav class="navbar">

    <h2 class="logo">MyWebsite</h2>

    <ul class="nav-links">

      <li><a href="#">Home</a></li>
```

```
        <li><a href="#">Courses</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav>
    <!-- Main Content -->
    <main class="container">
      <h1>Welcome to Responsive Webpage</h1>
      <p>
        This page changes layout, font size, navigation style,
        and background color based on screen size in Mobile view.
      </p>
    </main>
</body>
</html>
```

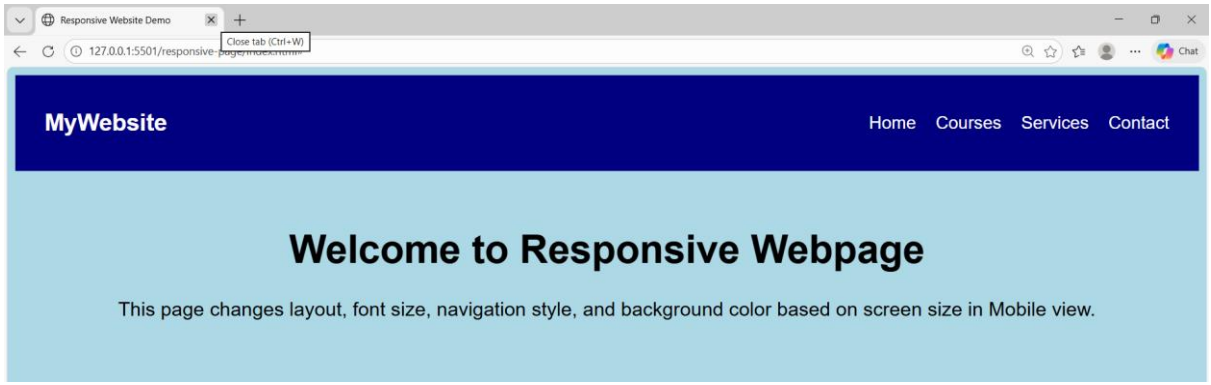
Style.css

```
body {
    font-family: Arial, sans-serif;
    background-color: lightblue;
}
/* Navigation */
.navbar {
    display: flex;
    justify-content: space-between;
    align-items: center;
    background-color: navy;
    color: white;;
    padding: 15px 30px;
}
```

```
.nav-links {
    list-style: none;
    display: flex;
    gap: 20px;
}
.nav-links a {
    color: white;
    text-decoration: none;
    font-size: 18px;
}
/* Main Content */
.container {
    padding: 30px;
    text-align: center;
}
h1 {
    font-size: 40px;
}
p {
    font-size: 20px;
}
/*  Media Query for Mobile Screen   */
@media (max-width: 768px) {
    /*Change background color */
    body {
        background-color: lightcoral;
    }
    /*Adjust font size */
    h1 {
```

```
        font-size: 24px;
    }
    p {
        font-size: 14px;
    }

    /*Convert navigation to vertical layout */
    .navbar {
        flex-direction: column;
        align-items: center;
    }
    .nav-links {
        flex-direction: column;
        gap: 10px;
        margin-top: 10px;
        text-align: center;
    }
}
```

3. Output Screenshot



4.Code Explanation

<!DOCTYPE html> → Declares the document as HTML5.

<html lang="en"> → Root element of the webpage and sets the language to English.

<meta charset="UTF-8"> → Defines character encoding as UTF-8 to support all characters.

<meta name="viewport" content="width=device-width, initial-scale=1.0"> → Makes the webpage responsive by adjusting layout based on device screen width.

<title>Responsive Website Demo</title> → Sets the title shown in the browser tab.

<link rel="stylesheet" href="style.css"> → Links the external CSS file to apply styling.

<nav class="navbar"> → Creates the navigation bar and applies navbar styles.

<h2 class="logo">MyWebsite</h2> → Displays the website name/logo.

<ul class="nav-links"> → Creates an unordered list for navigation links.

 → Represents each navigation item.

<a href="#"> → Creates clickable hyperlinks.

<main class="container"> → Defines the main content section with container styling.

<h1> → Main heading of the webpage.

<p> → Paragraph describing responsive behavior.

body → Styles the entire webpage including font and background color.

font-family: Arial, sans-serif; → Sets the font style for the whole page.

background-color: lightblue; → Sets default background color.

.navbar → Styles the navigation bar.

display: flex; → Enables flexbox layout.

justify-content: space-between; → Places logo on left and links on right.

align-items: center; → Vertically centers items inside navbar.

background-color: navy; → Sets navbar background color.

color: white; → Sets text color inside navbar.

padding: 15px 30px; → Adds spacing inside navbar.

.nav-links → Styles the navigation list.

list-style: none; → Removes bullet points.

display: flex; → Aligns links horizontally.

gap: 20px; → Adds space between links.

.nav-links a → Styles anchor tags inside navbar.

color: white; → Sets link color.

text-decoration: none; → Removes underline from links.

font-size: 18px; → Sets link text size.

.container → Styles the main content section.

padding: 30px; → Adds spacing inside the container.

text-align: center; → Centers text horizontally.

h1 → Styles all h1 headings.

font-size: 40px; → Sets large heading size.

p → Styles all paragraph elements.

font-size: 20px; → Sets paragraph text size.

@media (max-width: 768px) → Applies styles only when screen width is 768px or less (mobile view).

body { background-color: lightcoral; } → Changes background color in mobile view.

h1 { font-size: 24px; } → Reduces heading size for smaller screens.

p { font-size: 14px; } → Reduces paragraph font size for mobile.

.navbar { flex-direction: column; } → Changes navbar layout from horizontal to vertical in mobile view.

align-items: center; → Centers navbar items vertically.

.nav-links { flex-direction: column; } → Displays navigation links vertically in mobile.

gap: 10px; → Reduces spacing between links in mobile.

margin-top: 10px; → Adds space above links in mobile.

text-align: center; → Centers navigation text in mobile view.

Problem 2: Student Grade Evaluator

1.Problem Statement

A school wants a simple JavaScript program to evaluate a student's performance based on marks obtained in a subject.

Requirements Accept the student's marks as a variable Use if–else statements to assign grades: Marks ≥ 75 → Grade A Marks ≥ 60 → Grade B Marks ≥ 40 → Grade C Marks < 40 → Fail

Display the grade on the web page or console

Technical Constraints Use JavaScript variables (let or const) Use numeric data types Use comparison and logical operators No functions or arrays allowed Output using console.log() or document.write()

Learning Outcome

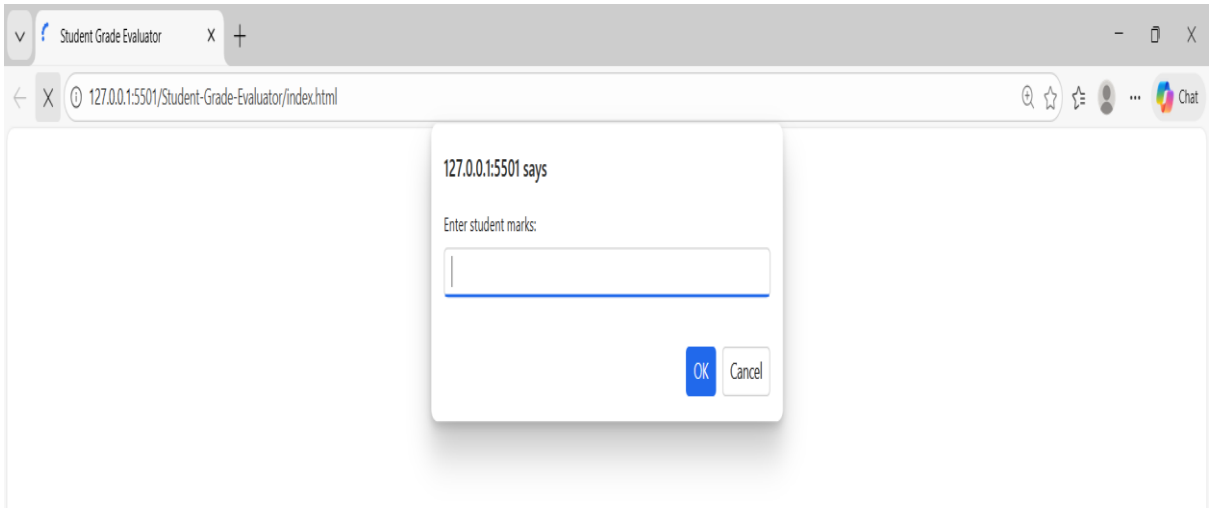
You should be able to: Declare and use variables Apply comparison operators Implement conditional logic using if–else Understand decision-making in JavaScript

2. Code

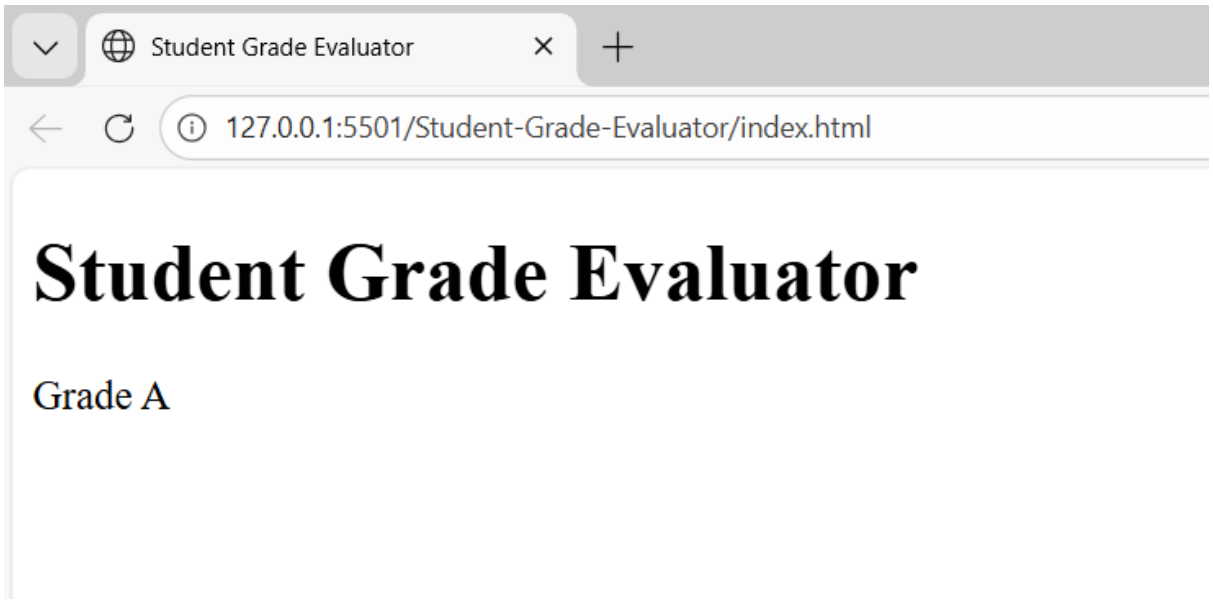
```
<!DOCTYPE html>

<html lang="en">


<head>

  <meta charset="UTF-8">

  <title>Student Grade Evaluator</title>

</head>


<body>

  <h1>Student Grade Evaluator</h1>


  <script>


    // Store raw input first

    let input = prompt("Enter student marks:");

    let marks = Number(input);
```

```
        let grade;
        if (input === "") {
            grade = "Marks cannot be empty";
        }
        else if (isNaN(marks)) {
            grade = "Invalid input";
        }
        else if (marks > 100 || marks < 0) {
            grade = "Invalid marks";
        }
        else if (marks >= 75) {
            grade = "Grade A";
        }
        else if (marks >= 60) {
            grade = "Grade B";
        }
        else if (marks >= 40) {
            grade = "Grade C";
        }
        else {
            grade = "Fail";
        }

        document.write(grade);

    </script>

</body>
</html>
```

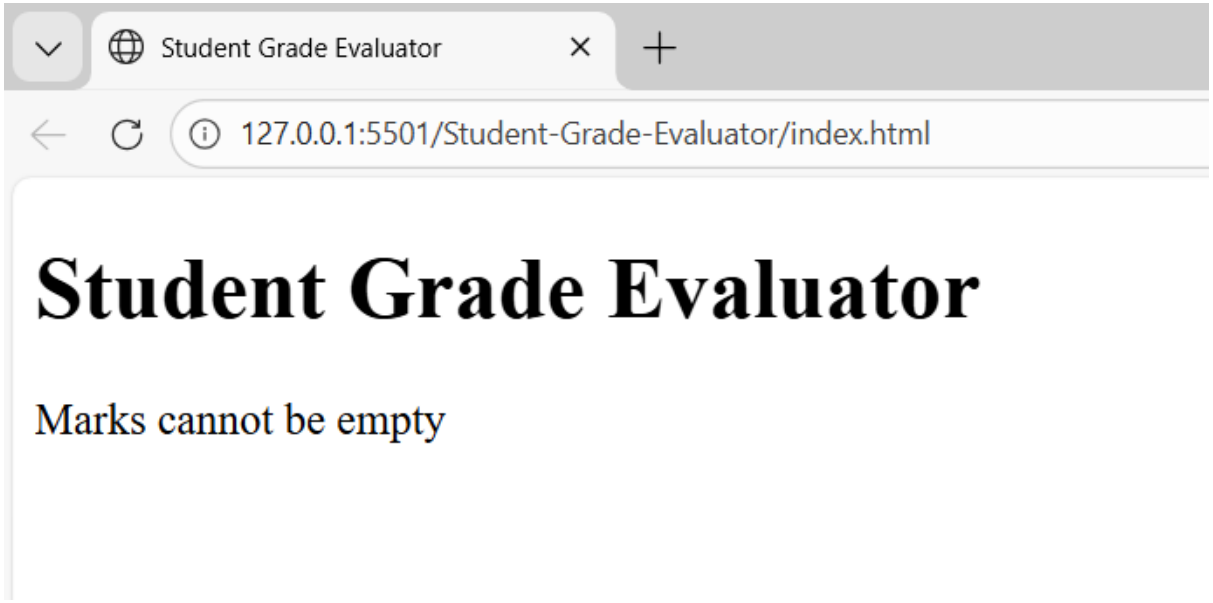
3. Code Output



I Entered 85 marks



Northing entered



4.Code Explanation

<!DOCTYPE html> → Declares the document as HTML5.

<html lang="en"> → Root element of the webpage and sets language to English.

<meta charset="UTF-8"> → Defines character encoding as UTF-8.

<title>Student Grade Evaluator</title> → Sets the title shown in the browser tab.

<h1>Student Grade Evaluator</h1> → Displays the main heading on the webpage.

<script> → Starts JavaScript code section.

let input = prompt("Enter student marks:"); → Displays a popup box to take marks input from the user as a string.

let marks = Number(input); → Converts the input string into a number.

let grade; → Declares a variable to store the final grade result.

if (input === "") → Checks if the user entered an empty value.

grade = "Marks cannot be empty"; → Assigns message if no input is given.

else if (isNaN(marks)) → Checks whether the entered value is not a number.

grade = "Invalid input"; → Assigns message for non-numeric input.

else if (marks > 100 || marks < 0) → Checks if marks are outside valid range (0–100).

grade = "Invalid marks"; → Assigns message if marks are out of range.

else if (marks >= 75) → Checks if marks are 75 or above.

grade = "Grade A"; → Assigns Grade A.

else if (marks >= 60) → Checks if marks are between 60 and 74.

grade = "Grade B"; → Assigns Grade B.

else if (marks >= 40) → Checks if marks are between 40 and 59.

grade = "Grade C"; → Assigns Grade C.

else → Executes if none of the above conditions are true.

grade = "Fail"; → Assigns Fail if marks are below 40.

document.write(grade); → Displays the final result (grade message) on the webpage.

</script> → Ends JavaScript section.

</body> → Ends the body of the webpage.

</html> → Ends the HTML document.

Problem 3: Simple Discount Calculator (Level-1)

1.Problem Statement

An online store wants to apply a discount based on the total purchase amount.

Requirements Store purchase amount in a variable Apply discount rules:

Amount ≥ 5000 → 20% discount

Amount ≥ 3000 → 10% discount

Amount < 3000 → No discount

Calculate and display: Discount amount Final payable amount

Technical Constraints Use arithmetic operators Use if–else statements Use only primitive data types

No user input (hardcoded values allowed)

Learning Outcome

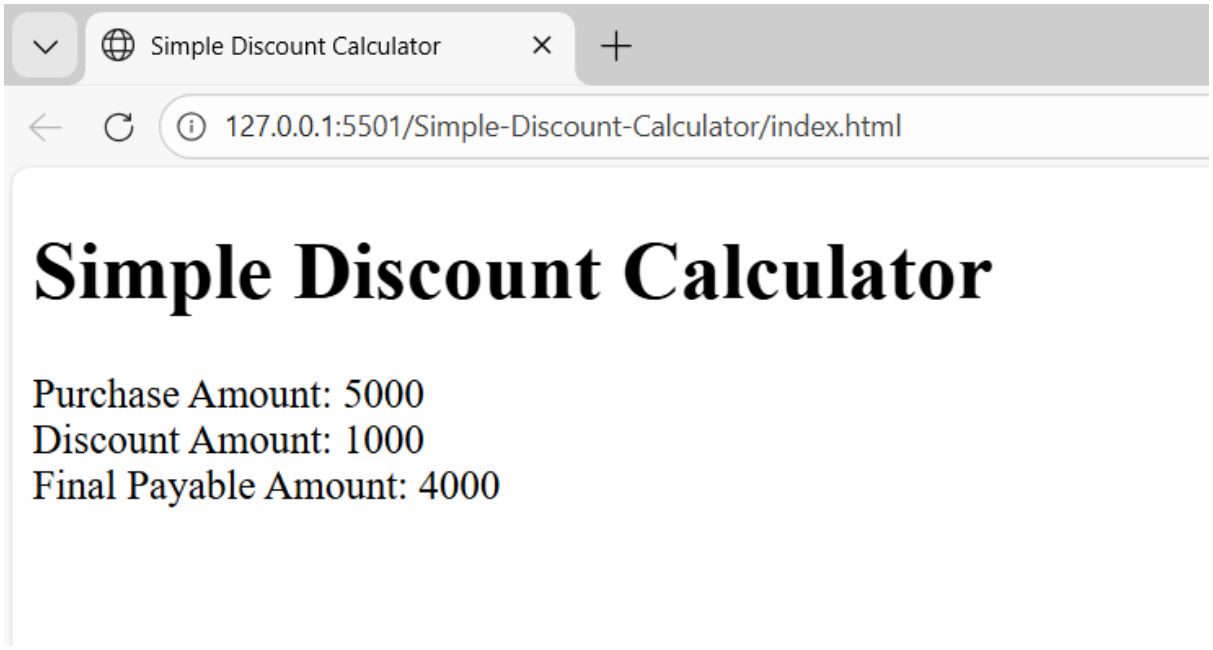
You will be able to: Perform calculations using operators Work with expressions Apply conditional statements Build real-world logic using JavaScript basics

2. Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Simple Discount Calculator</title>
</head>
<body>
<h1>Simple Discount Calculator</h1>
<script>
  let purchaseAmount = 5000;
  let discount = 0;
  let finalAmount = 0;
  if (purchaseAmount >= 5000) {
    discount = purchaseAmount * 0.20;
```

```
    }
    else if (purchaseAmount >= 3000) {
        discount = purchaseAmount * 0.10;
    }
    else {
        discount = 0;
    }
    finalAmount = purchaseAmount - discount;
    document.write("Purchase Amount: " + purchaseAmount + "<br>");
    document.write("Discount Amount: " + discount + "<br>");
    document.write("Final Payable Amount: " + finalAmount + "<br>");
</script>
</body>
</html>
```

3.Output Screenshot



4.Code Explanation

<!DOCTYPE html> → Declares the document as HTML5.

<html lang="en"> → Root element of the webpage and sets language to English.

<meta charset="UTF-8"> → Defines character encoding as UTF-8.

<title>Simple Discount Calculator</title> → Sets the title shown in the browser tab.

<h1>Simple Discount Calculator</h1> → Displays the main heading on the webpage.

<script> → Starts the JavaScript section.

let purchaseAmount = 5000; → Declares a variable and assigns the total purchase amount as 5000.

let discount = 0; → Declares a variable to store the discount amount and initializes it to 0.

let finalAmount = 0; → Declares a variable to store the final payable amount.

if (purchaseAmount >= 5000) → Checks if the purchase amount is 5000 or more.

discount = purchaseAmount * 0.20; → Calculates 20% discount for purchases 5000 or above.

else if (purchaseAmount >= 3000) → Checks if the purchase amount is between 3000 and 4999.

discount = purchaseAmount * 0.10; → Calculates 10% discount for purchases 3000 or above.

else → Executes if purchase amount is less than 3000.

discount = 0; → No discount is applied.

finalAmount = purchaseAmount - discount; → Calculates final payable amount after subtracting discount.

document.write("Purchase Amount: " + purchaseAmount + "
"); → Displays the purchase amount on the webpage.

document.write("Discount Amount: " + discount + "
"); → Displays the calculated discount.

document.write("Final Payable Amount: " + finalAmount + "
"); → Displays the final amount to be paid.

</script> → Ends the JavaScript section.

</body> → Ends the body section.

</html> → Ends the HTML document.

Problem 4: Traffic Signal Simulator (Level-2)

1.Problem Statement

A traffic control system needs a JavaScript program that displays instructions based on traffic signal color.

Requirements Store signal color in a variable ("red", "yellow", "green") Use a **switch statement** to display:

Red → Stop

Yellow → Get Ready

Green → Go

Handle invalid signal input gracefully

Technical Constraints Must use switch–case Use string data types Use console.log() for output No if–else allowed

Learning Outcome

Learners should be able to: Use switch statements effectively Compare string values

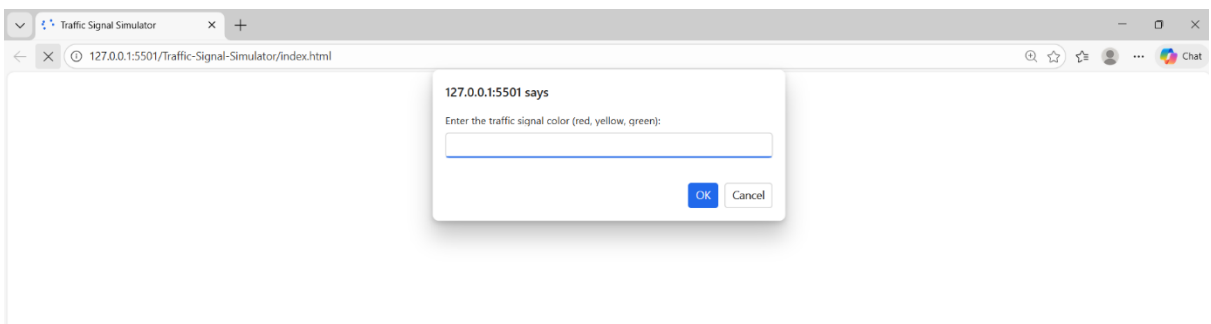
Handle multiple conditions cleanly Understand control flow alternatives

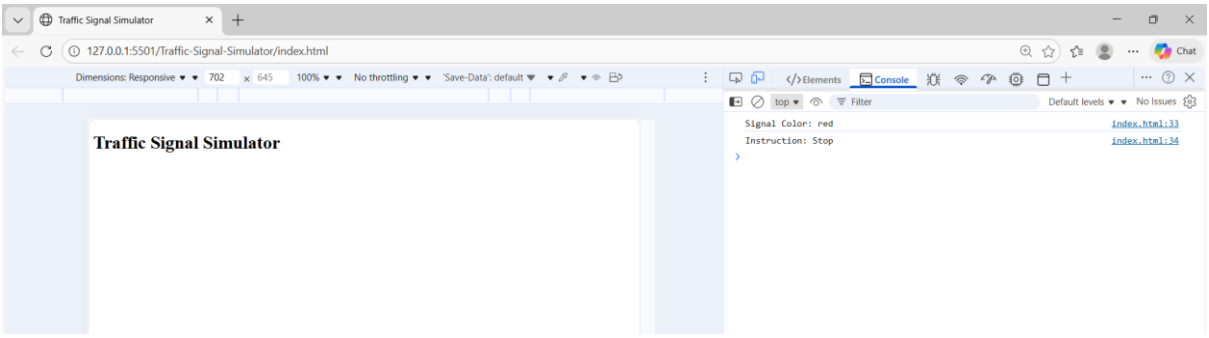
2. Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Traffic Signal Simulator</title>
</head>
<body>
<h1>Traffic Signal Simulator</h1>
<script>
  let signalColor = prompt("Enter the traffic signal color (red, yellow, green):");
  let instruction;
  switch (signalColor) {

    case "red":
```

```
            //Red signal instruction
            instruction = "Stop";
            break;

        case "yellow":
            //Yellow signal instruction
            instruction = "Get Ready";
            break;

        case "green":
            //Green signal instruction
            instruction = "Go";
            break;

        default:
            //Handle invalid signal input gracefully
            instruction = "Invalid signal color";
    }
    console.log("Signal Color:", signalColor);
    console.log("Instruction:", instruction);
</script>
</body>
</html>
```

3.Output Screenshot





4. Code Explanation

<h1>Traffic Signal Simulator</h1> → Displays the main heading on the webpage.

<script> → Starts the JavaScript section.

let signalColor = prompt("Enter the traffic signal color (red, yellow, green):"); → Displays a popup box to take traffic signal color input from the user.

let instruction; → Declares a variable to store the instruction based on signal color.

switch (signalColor) → Starts a switch statement to check different possible values of signalColor.

case "red": → Checks if the entered color is red.

instruction = "Stop"; → Assigns "Stop" instruction for red signal.

break; → Stops further checking once a match is found.

case "yellow": → Checks if the entered color is yellow.

instruction = "Get Ready"; → Assigns "Get Ready" instruction for yellow signal.

break; → Exits the switch block after execution.

case "green": → Checks if the entered color is green.

instruction = "Go"; → Assigns "Go" instruction for green signal.

break; → Stops execution after matching case.

default: → Executes if none of the cases match.

instruction = "Invalid signal color"; → Handles incorrect input gracefully.

console.log("Signal Color:", signalColor); → Displays the entered signal color in the browser console.

console.log("Instruction:", instruction); → Displays the corresponding instruction in the console.

</script> → Ends the JavaScript section.

Problem 5: Number Analysis Tool (Level-2)

1.Problem Statement

A utility program is required to analyze numbers and provide insights such as positivity, parity, and range.

Requirements Store a number in a variable Use **conditional (ternary) operator** to check: Positive or Negative Use **if–else** to check: Even or Odd Use a **loop** to print all numbers from 1 to the given number

Technical Constraints Store a number in a variable Use conditional (ternary) operator to check: Positive or Negative Use if–else to check Even or Odd Use a loop to print all numbers from 1 to the given number

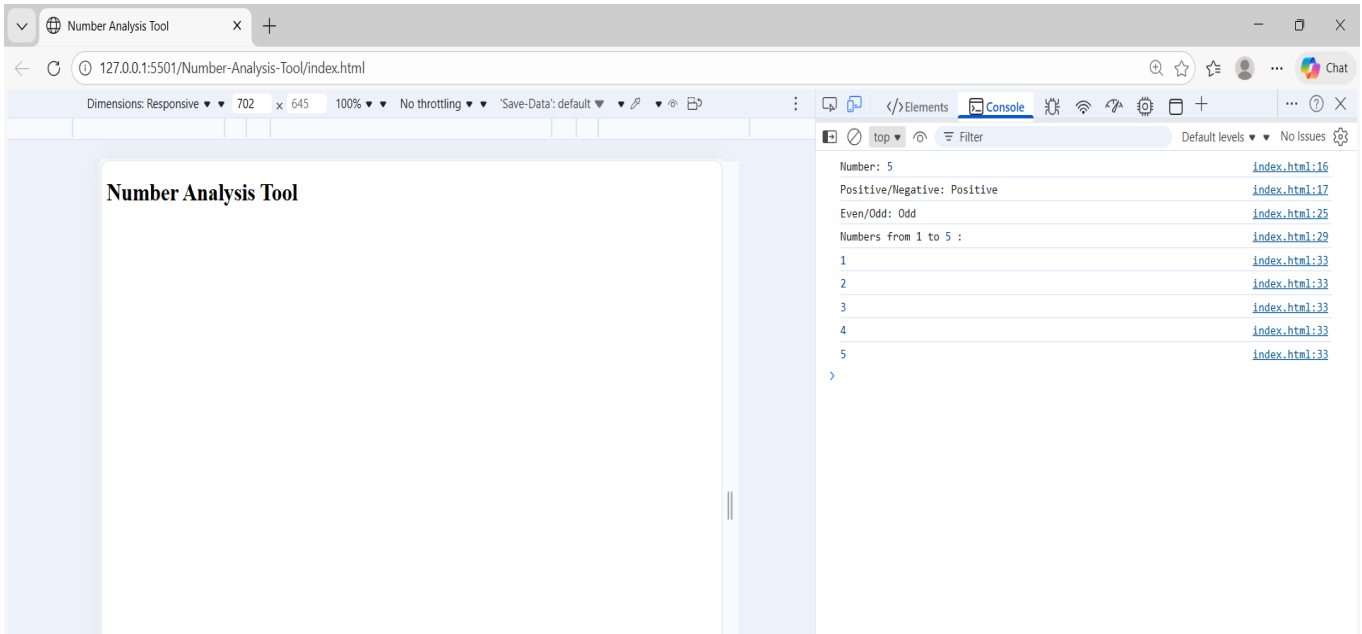
Learning Outcome You will be able to: Combine multiple control flow techniques Use loops for iteration Apply conditional operators Build multi-step logical programs

2. Code

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Number Analysis Tool</title>

</head>

<body>


<h1>Number Analysis Tool</h1>


<script>

  let number = 5;


  // (ternary) operator to check Positive or Negative

  let result = (number >= 0) ? "Positive" : "Negative";

  console.log("Number:", number);

  console.log("Positive/Negative:", result);
```

```
    // Use if–else to check Even or Odd
    if (number % 2 === 0) {
        console.log("Even/Odd: Even");
    }
    else {
        console.log("Even/Odd: Odd");
    }

    //loop to print numbers from 1 to the given number
    console.log("Numbers from 1 to", number, ":");
    // Using for loop for iteration
    for (let i = 1; i <= number; i++) {

        console.log(i);
    }
</script>
</body>
</html>
```

3.Output Screenshot



4.Code Explanation

<!DOCTYPE html> → Declares the document as HTML5.

<html lang="en"> → Root element of the webpage and sets language to English.

<meta charset="UTF-8"> → Defines character encoding as UTF-8.

<title>Number Analysis Tool</title> → Sets the title shown in the browser tab.

<h1>Number Analysis Tool</h1> → Displays the main heading on the webpage.

<script> → Starts the JavaScript section.

let number = 5; → Declares a variable named number and assigns value 5.

let result = (number >= 0) ? "Positive" : "Negative"; → Uses ternary operator to check if number is positive or negative.

(number >= 0) → Condition to check whether the number is greater than or equal to 0.

? "Positive" → If condition is true, assigns "Positive".

: "Negative" → If condition is false, assigns "Negative".

console.log("Number:", number); → Displays the number in the browser console.

console.log("Positive/Negative:", result); → Displays whether the number is positive or negative.

if (number % 2 === 0) → Checks if the number is divisible by 2.

number % 2 → Gives remainder after dividing by 2.

=== 0 → Checks if remainder is exactly 0.

console.log("Even/Odd: Even"); → Prints "Even" if condition is true.

else → Executes if number is not divisible by 2.

console.log("Even/Odd: Odd"); → Prints "Odd" if condition is false.

console.log("Numbers from 1 to", number, ":"); → Displays heading message in console.

for (let i = 1; i <= number; i++) → Runs a loop starting from 1 up to the given number.

let i = 1 → Initializes loop variable.

i <= number → Loop continues while i is less than or equal to number.

i++ → Increments i by 1 after each iteration.

console.log(i); → Prints each value of i inside the loop.

</script> → Ends the JavaScript section.

</body> → Ends the body section.