```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Validator Web Modules</title>
    <link rel="stylesheet" href="style.css">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&family=Roboto
:wght@400;500&display=swap" rel="stylesheet">
</head>
<body>
    <div class="grid-container">
        <header class="main-header">
            <h1>Activity 7</h1>
        </header>

        <div class="main-content">
            <div class="left-container">
                <h2>Prime Number Checker</h2>
                <p>Check if a number is a prime number or not</p>

                <div class ="prime-checker">
                    <input type="number" id="numInput" placeholder="Enter a
number " required>
                    <button id="checkPrime" onclick="checkPrime()">Check
Number</button>
                    <!--Output box -->
                </div>
                <div id="primeResult" class="output-box"></div>
            </div>

            <div class="mid-container">
                <h2>Square Root Calculator</h2>
                <p>Check the square root of a number</p>

                <div class ="prime-checker">
                    <input type="number" id="sqrtInput" placeholder="Enter a
number " required>
                    <button id="checkSqrt" onclick="checkSqrt()">Check
Number</button>
                </div>
                <!--Output box -->
                <div id="sqrtResult" class="output-box"></div>
            </div>
```

```
            <div class="right-container">
                <h2>Phone Number Format Checker</h2>
                <p>Check for Philippines or Singaporean Number</p>

                <div class="prime-checker">
                    <input
                        type="text"
                        id="phoneInput"
                        placeholder="Enter +639XXXXXXXXX, 09XXXXXXXXX, or
+65XXXXXXXX"
                        pattern="^(\+639\d{9}|09\d{9}|\+65\d{8})$"
                        title="Examples: +639171234567, 09171234567, +6591234567"
                        required
                    >
                    <button id="checkPhone" onclick="checkPhone()">Check
Number</button>
                </div>

                <!-- Output box -->
                <div id="phoneResult" class="output-box"></div>
            </div>
        </div>

        <footer class="main-footer">
            <p>&copy; 2025 John Rich Nicolas. All rights reserved.</p>
        </footer>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

CSS

```
body {
    font-family: 'Roboto', sans-serif;
    background-color: #121212;
    color: #E0E0E0;
    padding: 20px;
}

.grid-container {
    display: grid;
    grid-template-areas:
        "header header header"
```

```
        "main main main"
        "footer footer footer";
    grid-template-columns: repeat(3, 1fr);
    gap: 20px;
    max-width: 1200px;
    margin: auto;
    padding: 20px;
}

.main-footer {
    grid-area: footer;
    text-align: center;
}

/* Header Section */
.main-header {
    grid-area: header;
    display: flex;
    justify-content: space-between;
    align-items: center;
    flex-wrap: wrap;
    background-color: #00C896;
    color: #121212;
    padding: 30px;
    border-radius: 1.5rem;
    border-bottom: 5px solid #FF4081;
    box-shadow: 0 6px 16px rgba(0, 0, 0, 0.5);
    margin: 0;
}

.main-header h1 {
    font-family: 'Poppins', sans-serif;
    font-size: 2.5rem;
    letter-spacing: 1px;
    text-transform: uppercase;
}

/* Containters */
.main-content {
    grid-area: main;
    display: flex;
    flex-direction: row;
    gap: 20px;
    align-items: stretch;
}
```

```
.left-container,
.mid-container,
.right-container {
    display: flex;
    flex-direction: column;
    gap: 20px;
    text-align: center;
    flex: 1;
    padding: 20px;
    border-radius: 1rem;
    background-color: #1e1e2f;
    color: #E0E0E0;
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.3);
}

/*Input*/
.prime-checker input{
    padding: 12px 16px;
    border: none;
    border-radius: 8px;
    width: 100%;
    max-width: 250px;
    font-size: 1rem;
    box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
    margin:20px;
}
/*BUTTONS*/
button {
    background-color: #00C896;
    color: #121212;
    padding: 10px 20px;
    border: none;
    border-radius: 10px;
    font-weight: 500;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

button:hover {
    background-color: #FF4081;
    color: #E0E0E0;
}
/*Output*/
.output-box {
```

```
    margin-top: 10px;
    padding: 15px;
    background-color: #2e2e4f;
    border-radius: 10px;
    font-weight: bold;
    font-size: 1.1rem;
    min-height: 50px;
}
```

SCRIPT

```
function checkPrime() {
    const input = document.getElementById('numInput').value.trim();
    const resultBox = document.getElementById('primeResult');

    // Validate input
    if (input === "" || isNaN(input)) {
        resultBox.textContent = "Please enter a valid number.";
        resultBox.style.color = "#FF4081";
        return;
    }

    const num = parseInt(input);

    if (num <= 1) {
        resultBox.textContent = "Number must be greater than 1.";
        resultBox.style.color = "#FF4081";
        return;
    }

    if (num === 2) {
        resultBox.textContent = "Number is PRIME!";
        resultBox.style.color = "#00C896";
        return;
    }

    if (num % 2 === 0) {
        resultBox.textContent = "Number is NOT PRIME!";
        resultBox.style.color = "#FF4081";
        return;
    }

    // Check divisibility up to square root of num
    //chechck gang sqrt lang ng number, pag may divisible/divisor or factor na
pwede mag divid di na siya prime agad
```

```
    //skip din mga even numbers since even sila and di sila prime. except 2
    const sqrt = Math.sqrt(num);
    for (let i = 3; i <= sqrt; i += 2) {
        if (num % i === 0) {
            resultBox.textContent = "Number is NOT PRIME!";
            resultBox.style.color = "#FF4081";
            return;
        }
    }

    // Prime if no factors found
    resultBox.textContent = "Number is PRIME!";
    resultBox.style.color = "#00C896";
}

function checkSqrt() {
    const input = document.getElementById("sqrtInput").value;
    const resultBox = document.getElementById("sqrtResult");

    if (input === "" || isNaN(input)) {
        resultBox.textContent = "Please enter a valid number.";
        resultBox.style.color = "#FF4081";
    } else if (input < 0) {
        resultBox.textContent = "Cannot calculate square root of a negative
number.";
        resultBox.style.color = "#FF4081";
    } else {
        const sqrt = Math.sqrt(parseFloat(input)).toFixed(2);
        resultBox.textContent = `√${input} = ${sqrt}`;
        resultBox.style.color = "#00C896";
    }
}

function checkPhone() {
    const input = document.getElementById("phoneInput").value.trim();
    const resultBox = document.getElementById("phoneResult");

    function isValidPHLocal(num) {
        return num.startsWith("09") && num.length === 11;
    }

    function isValidPHIntl(num) {
        return num.startsWith("+639") && num.length === 13;
    }
```

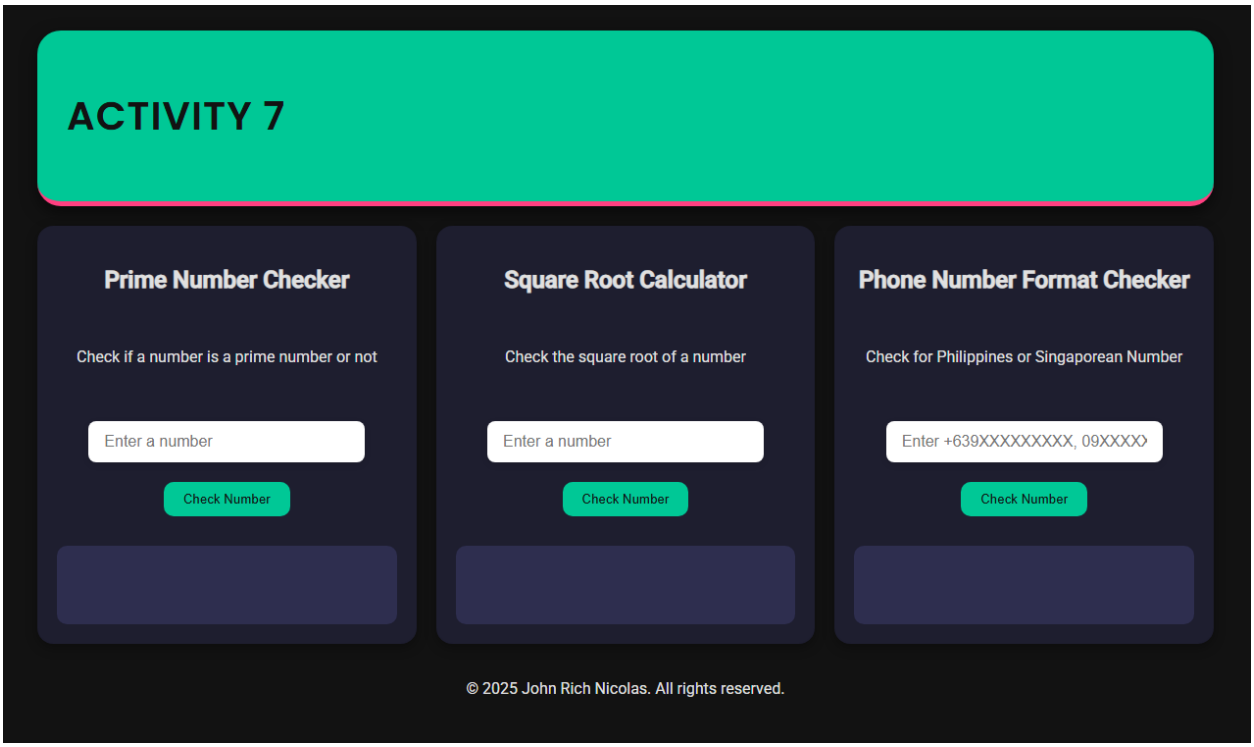
```
    function isValidSG(num) {
        return num.startsWith("+65") && num.length === 11;
    }

    if (isValidPHLocal(input) || isValidPHIntl(input)) {
        resultBox.textContent = "Valid Philippine number!";
        resultBox.style.color = "#00C896";
    } else if (isValidSG(input)) {
        resultBox.textContent = "Valid Singaporean number!";
        resultBox.style.color = "#00C896";
    } else {
        resultBox.textContent = "Invalid phone number format.";
        resultBox.style.color = "#FF4081";
    }
}
```

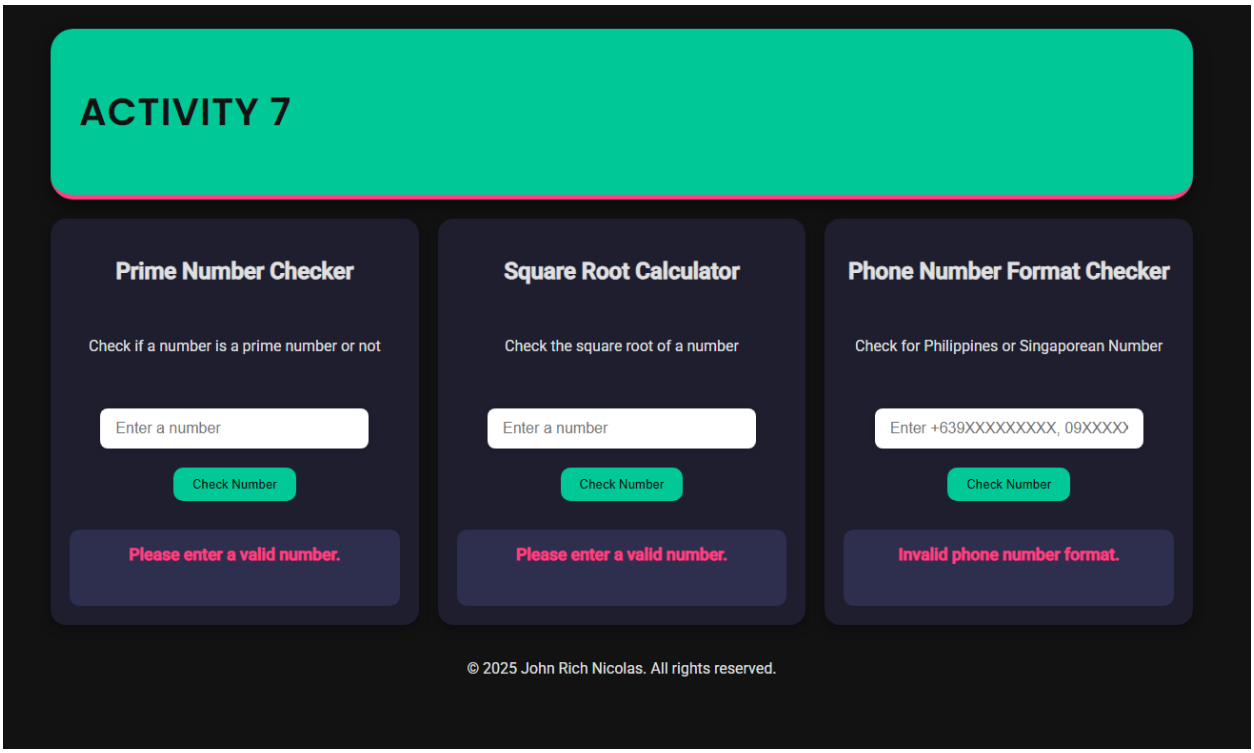
John Rich Nicolas
DAY 7

FRONTEND

OUTPUT



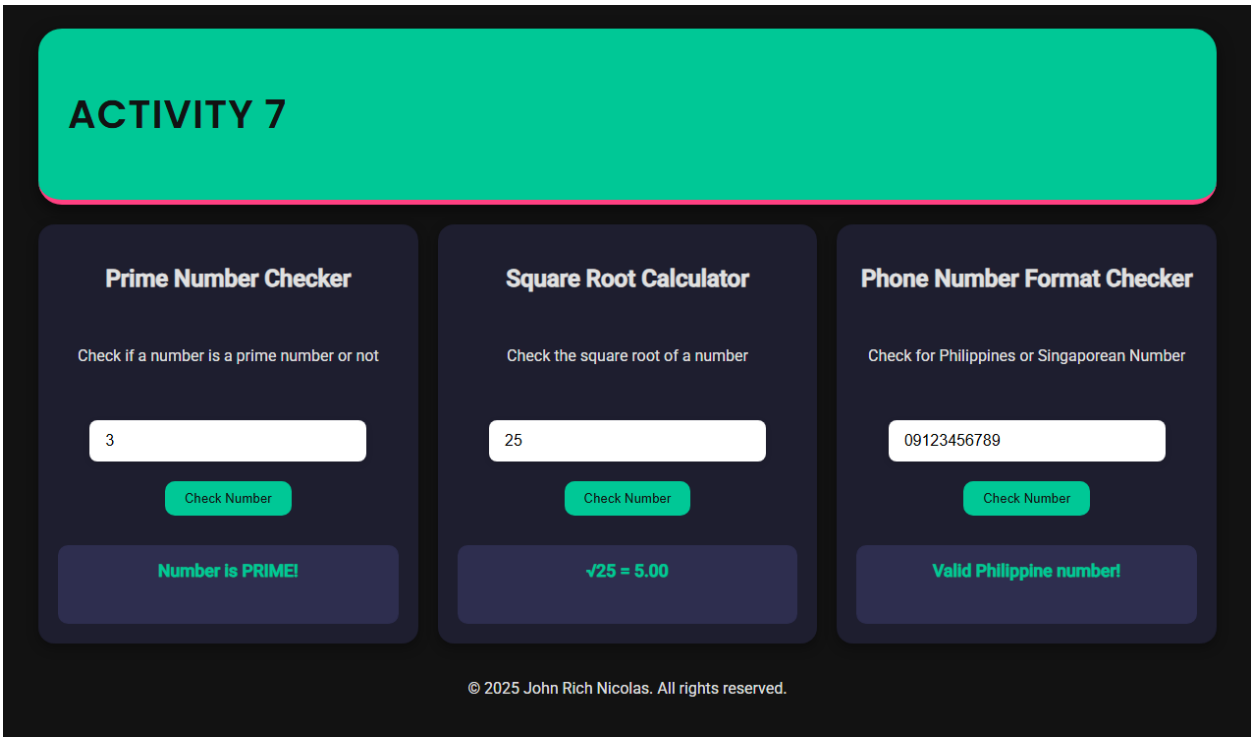
NO INPUTS

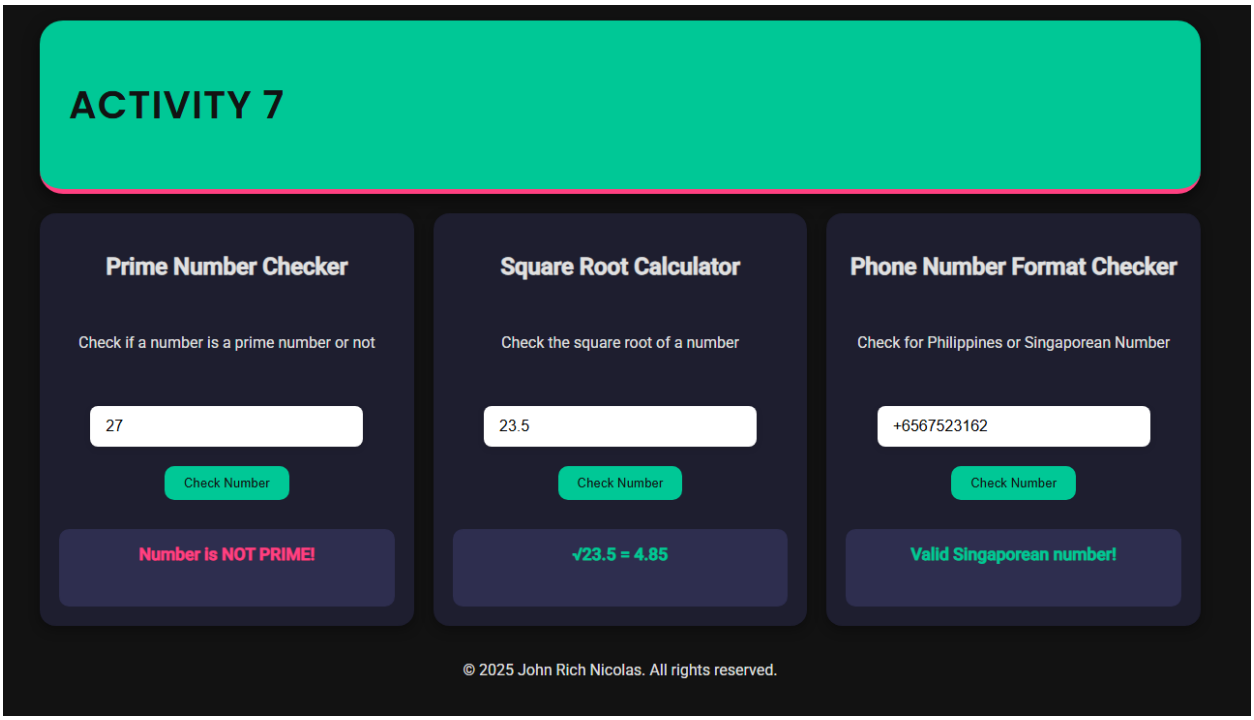


John Rich Nicolas
DAY 7

FRONTEND

VALID INPUT

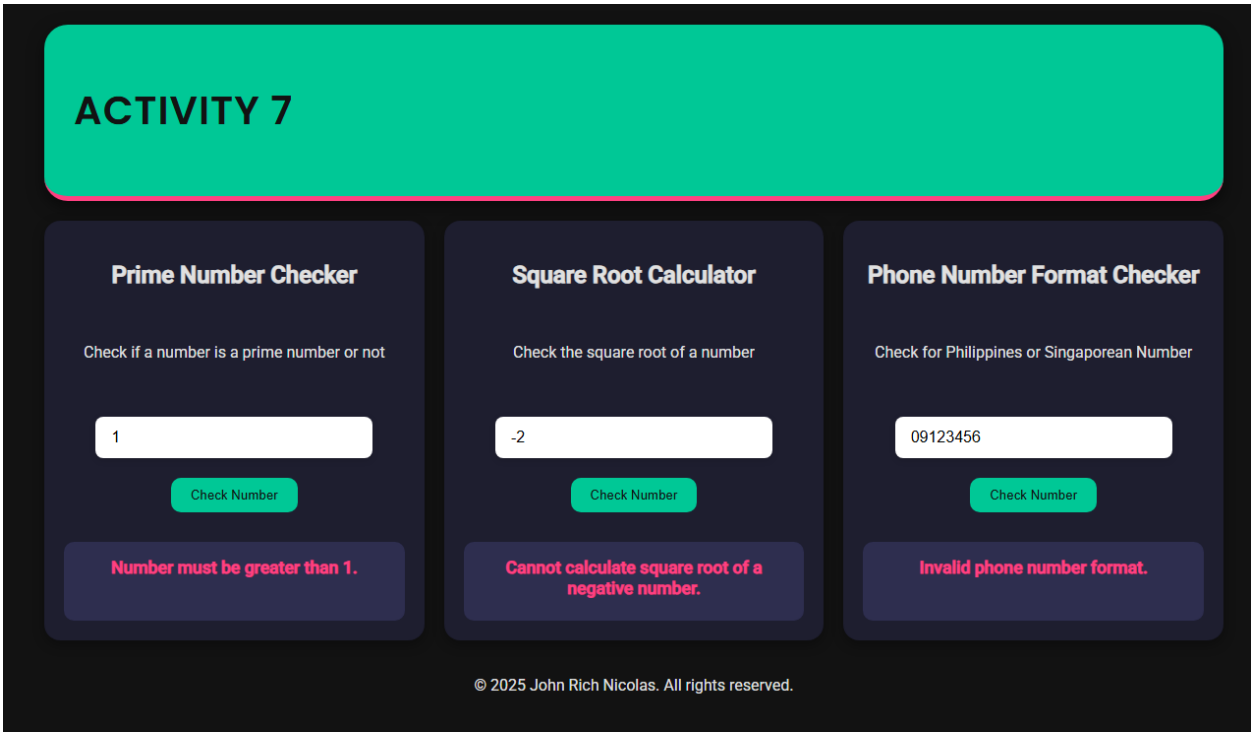


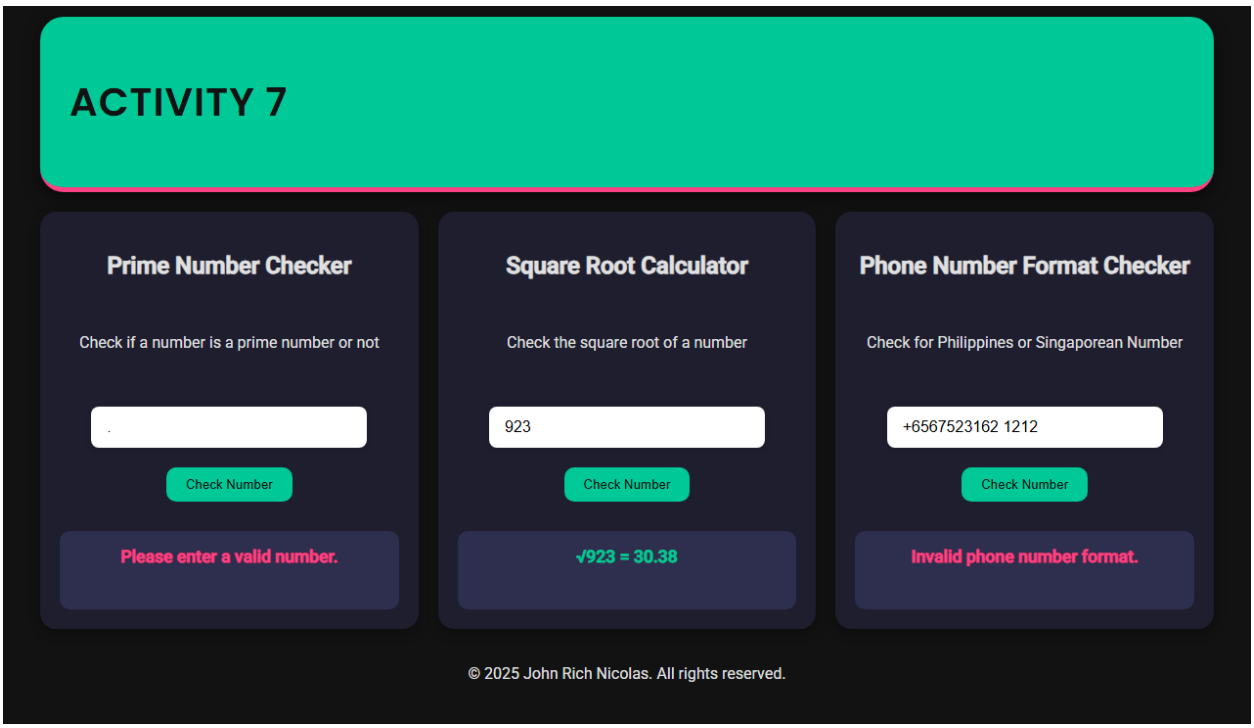


John Rich Nicolas
DAY 7

FRONTEND

INVALID INPUT





John Rich Nicolas FRONTEND
DAY 7

REFLECTION

In this activity, I structured my JavaScript code into three separate and reusable functions — one for each module: the prime number checker, the square root calculator, and the phone number format validator. Each function was written to handle a specific task, making the code easier to read, test, and debug. This approach follows modular programming, which allowed me to organize the logic in a way where each module is independent and manageable. If changes were needed for one function, I could make them without affecting the others. It also made the code scalable in case I wanted to add more features later.

To handle different user inputs, I used input validation and control flow statements such as if, else if, and else. For example, in the prime number checker, I checked if the input was a positive whole number greater than 1 before proceeding with the calculation. The square root module also included validation to reject negative numbers or empty fields. In the phone number validator, I used string manipulation and regular expressions to determine if the input started with valid Philippine prefixes (+63 or 09) or a Singaporean prefix (+65). This logic helped ensure the number followed a common pattern before accepting it as valid.

I also implemented mathematical functions, such as Math.sqrt() for the square root calculator and modulo checks to test for prime numbers. These were combined with logical conditions to compute and decide the correct results. For DOM interaction, I used document.getElementById() to get input values and used .textContent to display the output dynamically within specific result containers on the webpage. I actually found using .textContent really fun because it made the result pop out in a box, which added a nice visual touch to the program and made it feel more interactive. I discovered it while browsing online and thought it would be great for giving feedback to users clearly.

Throughout the activity, I learned more about how to use functions properly and how to validate different types of user input. One thing I initially struggled with was handling incorrect formats for phone numbers, but in the end testing various cases helped me fix it. If I were to improve the project, I would make the layout responsive on smaller screens. Overall, this activity helped me strengthen my understanding of JavaScript, especially in applying logic, structuring code, and interacting with the DOM effectively.