Name : Khushi Ingale
Assignment 35

1. Java Program to Check Palindrome (String/Number)
->

```
import java.util.Scanner;

public class Palindrome {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string or number: ");
        String input = scanner.nextLine();

        String reversed = new StringBuilder(input).reverse().toString();

        if (input.equals(reversed)) {
            System.out.println(input + " is a palindrome.");
        } else {
            System.out.println(input + " is not a palindrome.");
        }
    }
}
```

2. Java Program to Check Factorial of a Number
->

```
import java.util.Scanner;

public class Factorial {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        int result = 1;

        for (int i = 1; i <= num; i++) {
            result *= i;
        }

        System.out.println("The factorial of " + num + " is " + result);
    }
}
```

3. Java Program to Remove Duplicates from Array
->

```
import java.util.Arrays;
import java.util.HashSet;

public class RemoveDuplicates {
    public static void main(String[] args) {
        int[] arr = {1, 2, 2, 3, 4, 4, 5};
        HashSet<Integer> set = new HashSet<>();

        for (int num : arr) {
            set.add(num);
        }

        Integer[] uniqueArray = set.toArray(new Integer[0]);
        System.out.println("Array after removing duplicates: " +
Arrays.toString(uniqueArray));
    }
}
```

4. JavaScript Function to Return Only Even Numbers from an Array
->

```
function getEvenNumbers(arr) {
    return arr.filter(num => num % 2 === 0);
}

console.log(getEvenNumbers([1, 2, 3, 4, 5, 6])); // Output: [2, 4, 6]
```

5.JavaScript Function to Check if a Given Number is Prime

->

```
function isPrime(number) {
    if (number <= 1) return false;
    for (let i = 2; i < number; i++) {
        if (number % i === 0) {
            return false;
        }
    }
    return true;
}

console.log(isPrime(7));  // Output: true
console.log(isPrime(10)); // Output: false
```

6.HTML Pages for Various Functionalities

->

a.Create a registration form with validations

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Form</title>
</head>
<body>
    <form id="registrationForm" onsubmit="return validateForm()">
        <label for="name">Name:</label>
        <input type="text" id="name" required><br><br>

        <label for="email">Email:</label>
        <input type="email" id="email" required><br><br>

        <label for="password">Password:</label>
        <input type="password" id="password" required><br><br>

        <button type="submit">Register</button>
    </form>

    <script>
        function validateForm() {
            let name = document.getElementById('name').value;
            if (name === "") {
                alert("Name must be filled out");
                return false;
            }
            let email = document.getElementById('email').value;
            if (email === "") {
                alert("Email must be filled out");
                return false;
            }
            return true;
        }
    </script>
</body>
</html>
```

b.Create a responsive navigation bar

```
->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Navigation Bar</title>
    <style>
        nav {
            display: flex;
            justify-content: space-around;
            background-color: #333;
            padding: 10px;
        }
        nav a {
            color: white;
            text-decoration: none;
            padding: 10px;
        }
        nav a:hover {
            background-color: #ddd;
        }
        @media (max-width: 600px) {
            nav {
                flex-direction: column;
                align-items: center;
            }
        }
    </style>
</head>
<body>
    <nav>
        <a href="#">Home</a>
        <a href="#">About</a>
        <a href="#">Services</a>
        <a href="#">Contact</a>
    </nav>
</body>
</html>
```

c.Table with Rowspan and Colspan

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Table with Rowspan and Colspan</title>
</head>
<body>
    <table border="1">
        <tr>
            <th rowspan="2">Name</th>
            <th colspan="2">Marks</th>
        </tr>
        <tr>
            <td>Math</td>
            <td>Science</td>
        </tr>
        <tr>
            <td>John</td>
            <td>85</td>
            <td>90</td>
        </tr>
    </table>
```

```
</body>
</html>
```

d.Card Layout Using HTML and Minimal CSS

->

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Card Layout</title>
    <style>
        .card {
            width: 200px;
            padding: 20px;
            margin: 10px;
            border: 1px solid #ddd;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }
    </style>
</head>
<body>
    <div class="card">
        <h3>Card Title</h3>
        <p>This is a card description.</p>
    </div>
</body>
</html>
```

e.HTML Form with Input Types (Text, Date, Checkbox, etc.)

->

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form with Input Types</title>
</head>
<body>
    <form>
        <label for="name">Name:</label>
        <input type="text" id="name" required><br><br>

        <label for="dob">Date of Birth:</label>
        <input type="date" id="dob" required><br><br>

        <label for="subscribe">Subscribe to Newsletter:</label>
        <input type="checkbox" id="subscribe"><br><br>

        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

7. Create an API to Calculate Factorial of a Number

->

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class FactorialAPI {

    @GetMapping("/factorial")
    public String calculateFactorial(@RequestParam int number) {
        int result = 1;
        for (int i = 1; i <= number; i++) {
            result *= i;
        }
        return "The factorial of " + number + " is " + result;
    }
}
```

8. Implement a REST Endpoint to Check if a Number is Prime
->

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class PrimeNumberAPI {

    @GetMapping("/isPrime")
    public String isPrime(@RequestParam int number) {
        if (number <= 1) {
            return number + " is not a prime number.";
        }
        for (int i = 2; i < number; i++) {
            if (number % i == 0) {
                return number + " is not a prime number.";
            }
        }
        return number + " is a prime number.";
    }
}
```

9. Build a Simple Calculator Using React
->

```
import React, { useState } from 'react';

const Calculator = () => {
  const [input, setInput] = useState('');

  const handleClick = (value) => {
    setInput(input + value);
  };

  const handleEvaluate = () => {
    try {
      setInput(eval(input));
    } catch (e) {
      setInput('Error');
    }
  };

  const handleClear = () => {
    setInput('');
  };

  return (
    <div>
      <input type="text" value={input} disabled />
      <div>
        <button onClick={() => handleClick('1')}>1</button>
        <button onClick={() => handleClick('2')}>2</button>
        <button onClick={() => handleClick('3')}>3</button>
        <button onClick={() => handleClick('+')}>+</button>
        <button onClick={() => handleClick('4')}>4</button>
        <button onClick={() => handleClick('5')}>5</button>
        <button onClick={() => handleClick('6')}>6</button>
        <button onClick={() => handleClick('-')}>-</button>
        <button onClick={() => handleClick('7')}>7</button>
        <button onClick={() => handleClick('8')}>8</button>
        <button onClick={() => handleClick('9')}>9</button>
        <button onClick={() => handleClick('*')}>*</button>
        <button onClick={() => handleClick('0')}>0</button>
        <button onClick={handleClear}>C</button>
        <button onClick={handleEvaluate}>=</button>
        <button onClick={() => handleClick('/')}>/</button>
      </div>
    </div>
  );
}

export default Calculator;
```

10. Implement a Dropdown List Using useState in React

->

```
import React, { useState } from 'react';

const Dropdown = () => {
  const [selectedOption, setSelectedOption] = useState('');

  const handleSelect = (e) => {
    setSelectedOption(e.target.value);
  };

  return (
    <div>
      <select onChange={handleSelect}>
        <option value="">Select an option</option>
        <option value="Option 1">Option 1</option>
        <option value="Option 2">Option 2</option>
        <option value="Option 3">Option 3</option>
      </select>
      <p>You selected: {selectedOption}</p>
    </div>
  );
}

export default Dropdown;
```