<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Assignment4</title>

</head>

<body>

    <script>

        // Define the toUpperCase callback function

function toUpperCase(str) {

    return str.toUpperCase();

}

// Define the processData function

function processData(inputString, callback) {

    // Call the callback with the inputString and log the result

    const result = callback(inputString);

    console.log(result);

}

// Example usage

const myString = "hello, world!";

processData(myString, toUpperCase);

    </script>

</body>

</html>

index.html:19 HELLO, WORLD!

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Que2</title>

</head>

<body>

    <script>

        function forEachElement(array, callback) {

    for (let i = 0; i < array.length; i++) {

        callback(array[i], i); // Apply the callback to each element with its index

    }

}

// Example usage

const numbers = [1, 2, 3, 4, 5];

forEachElement(numbers, function(element, index) {

    const result = element \* 2; // Multiply each element by 2

    console.log(`Index: ${index}, Result: ${result}`); // Log the index and result

});

    </script>

</body>

</html>

Index: 0, Result: 2

index1.html:21 Index: 1, Result: 4

index1.html:21 Index: 2, Result: 6

index1.html:21 Index: 3, Result: 8

index1.html:21 Index: 4, Result: 10

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Que3</title>

</head>

<body>

    <script>

        function fetchData(url, callback) {

    console.log(`Fetching data from ${url}...`);

    // Simulating a network delay

    setTimeout(() => {

        const response = `Response from ${url}`; // Simulated response

        callback(response); // Call the callback with the response

    }, 1000); // Delay of 1 second

}

// Example usage

fetchData("https://example.com/api", (response) => {

    console.log("Response:", response); // Log the response to the console

});

    </script>

</body>

</html>

Fetching data from https://example.com/api...

index2.html:22 Response: Response from <https://example.com/api>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Que4</title>

</head>

<body>

    <script>

        function fetchData(callback) {

    // Simulating an asynchronous data fetch

    setTimeout(() => {

        // Simulate success or error

        const success = Math.random() > 0.2; // 80% chance to succeed

        if (success) {

            const response = { data: "Fetched data" };

            callback(null, response); // Call with null error and response

        } else {

            const errorMessage = "An error occurred while fetching data.";

            console.error(errorMessage); // Log the error

            callback(errorMessage, null); // Call with error message

        }

    }, 1000);

}

    </script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Que 5</title>

</head>

<body>

    <script>

        function processData(response, callback) {

    // Simulating data processing

    const processedData = response.data.toUpperCase(); // Modify the data

    console.log("Processed Data:", processedData); // Log the processed result

    callback(null, processedData); // Call the callback with null error

}

// Chaining fetchData and processData with nested callbacks

fetchData((error, response) => {

    if (error) {

        console.error("Error:", error); // Handle error from fetchData

    } else {

        processData(response, (err, processedResult) => {

            if (err) {

                console.error("Error processing data:", err);

            } else {

                console.log("Final Result:", processedResult);

            }

        });

    }

});

    </script>

</body>

</html>