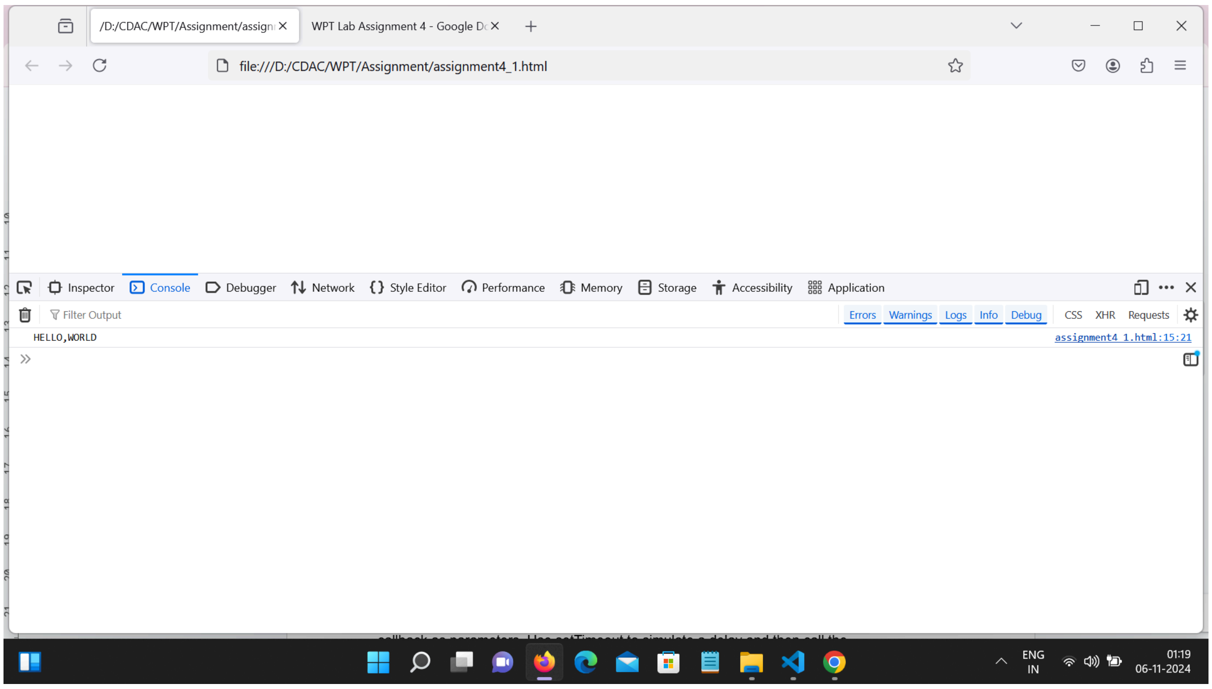

Module: WPT
Topic: Lab Assignment-4
Based on Callback Function

Exercise 1:
Create a function processData that takes two parameters: a string and a callback function. Your task is to write a callback that converts the string to uppercase and then call it within processData.
Requirements:
- Define a function toUpperCase that will serve as a callback.
- Pass a string and toUpperCase to processData and log the output.

```
<!DOCTYPE html>
<html lang="en">
    <body>
        <script>
            function processData (input,callback)
            {
               return callback(input);
            }

            function toUpperCase (str)
            {
                return str.toUpperCase();
            }

            console.log(processData("hello,world",toUpperCase));

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

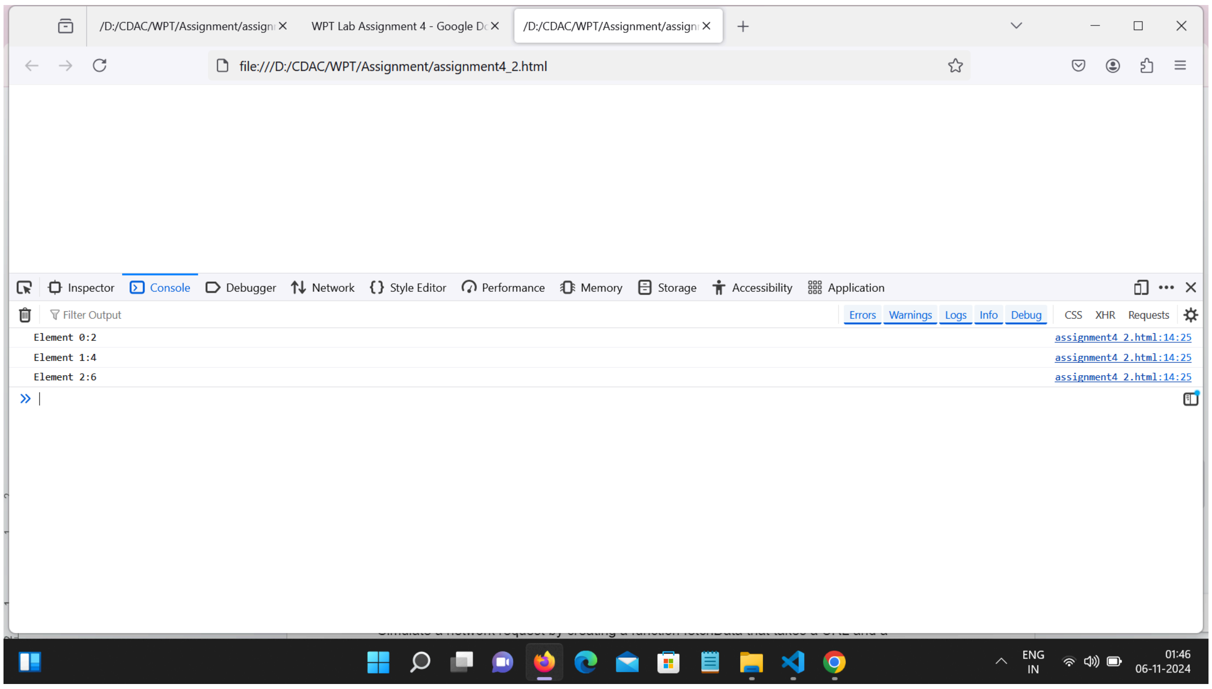


```
file:///D:/CDAC/WPT/Assignment/assignment4_1.html
```

```
HELLO,WORLD                                                    assignment4_1.html:15:21
```

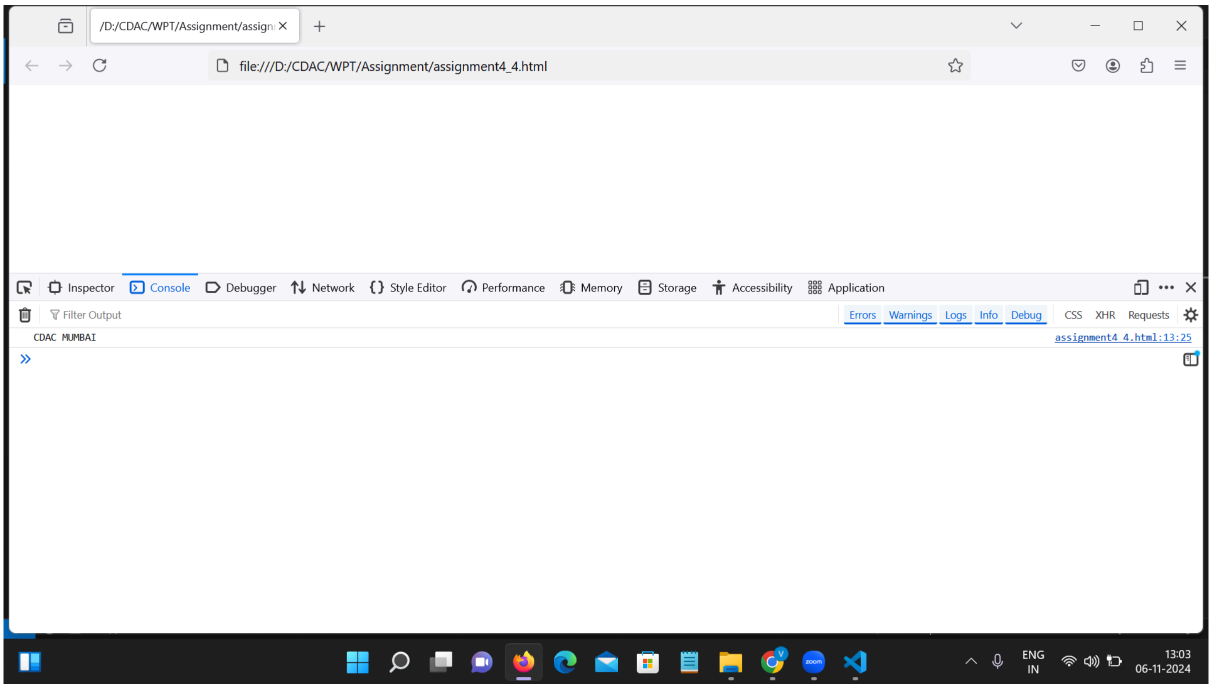
Exercise 2:
Write a function forEachElement that accepts an array and a callback. This function should apply the callback to each element of the array.
Requirements:
● Pass an anonymous function as the callback that multiplies each element by 2 and logs the result with the index.

```
<!DOCTYPE html>
<html>
    <body>
        <script>
            function forEachElement(arr,callback)
            {
                for(var i=0;i<arr.length;i++)
                {
                    callback(arr[i],i);
                }
            }

            forEachElement([1,2,3],(element,index)=>{
                console.log(`Element ${index}:${element*2}`);
            });

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



```
Element 0:2                                                assignment4_2.html:14:25
Element 1:4                                                assignment4_2.html:14:25
Element 2:6                                                assignment4_2.html:14:25
```

Exercise 3:
Simulate a network request by creating a function fetchData that takes a URL and a callback as parameters. Use setTimeout to simulate a delay and then call the callback with a string representing a response.
Requirements:
● After a delay, log the "response" to the console.
CDAC Mumbai



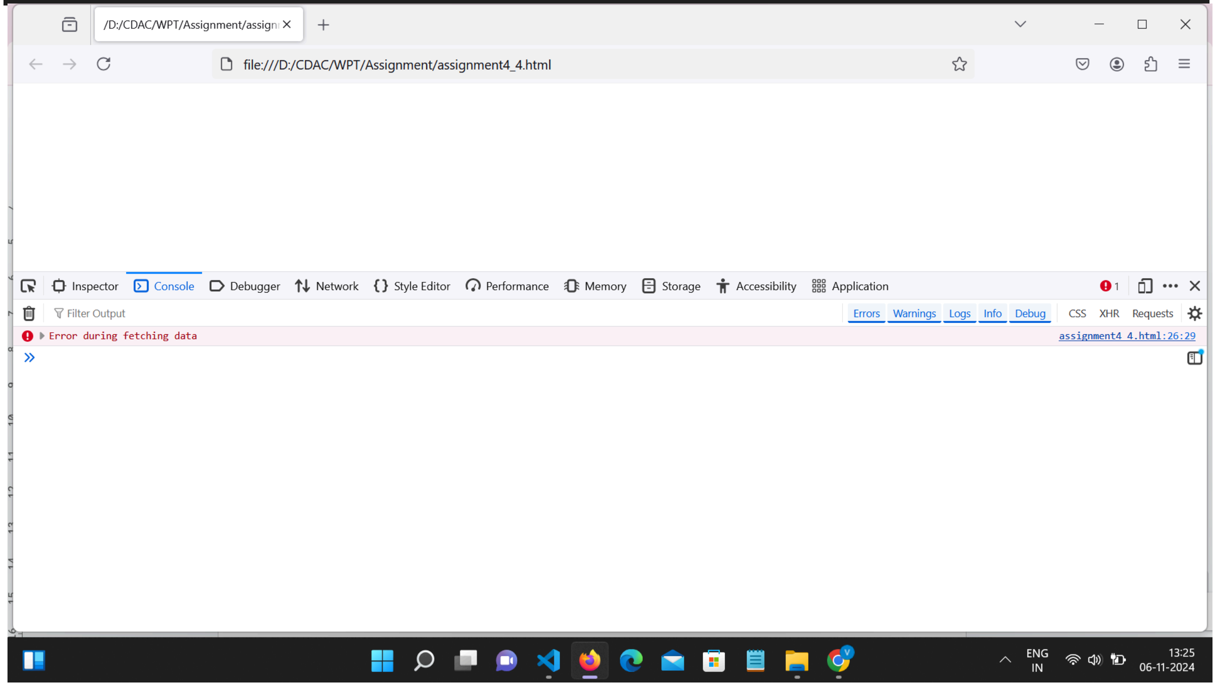
```
CDAC MUMBAI                                                assignment4_4.html:13:25
```

```
<!DOCTYPE html>
<html lang="en">
    <body>
```

```
        <script>
            function fetchData(url,callback)
            {
                setTimeout(()=>{
                    const data='CDAC MUMBAI';
                    callback(data);
                },2000)
            }
            fetchData("https://www.google.co.in/",(response)=>{
                console.log(response);
            })
        </script>
    </body>
</html>
```

Exercise 4:
Modify fetchData from Exercise 3 to include error handling.
Requirements:
● Call the callback with an error message if an error occurs; otherwise, pass the "response."
● Handle the error gracefully by logging it if it occurs.

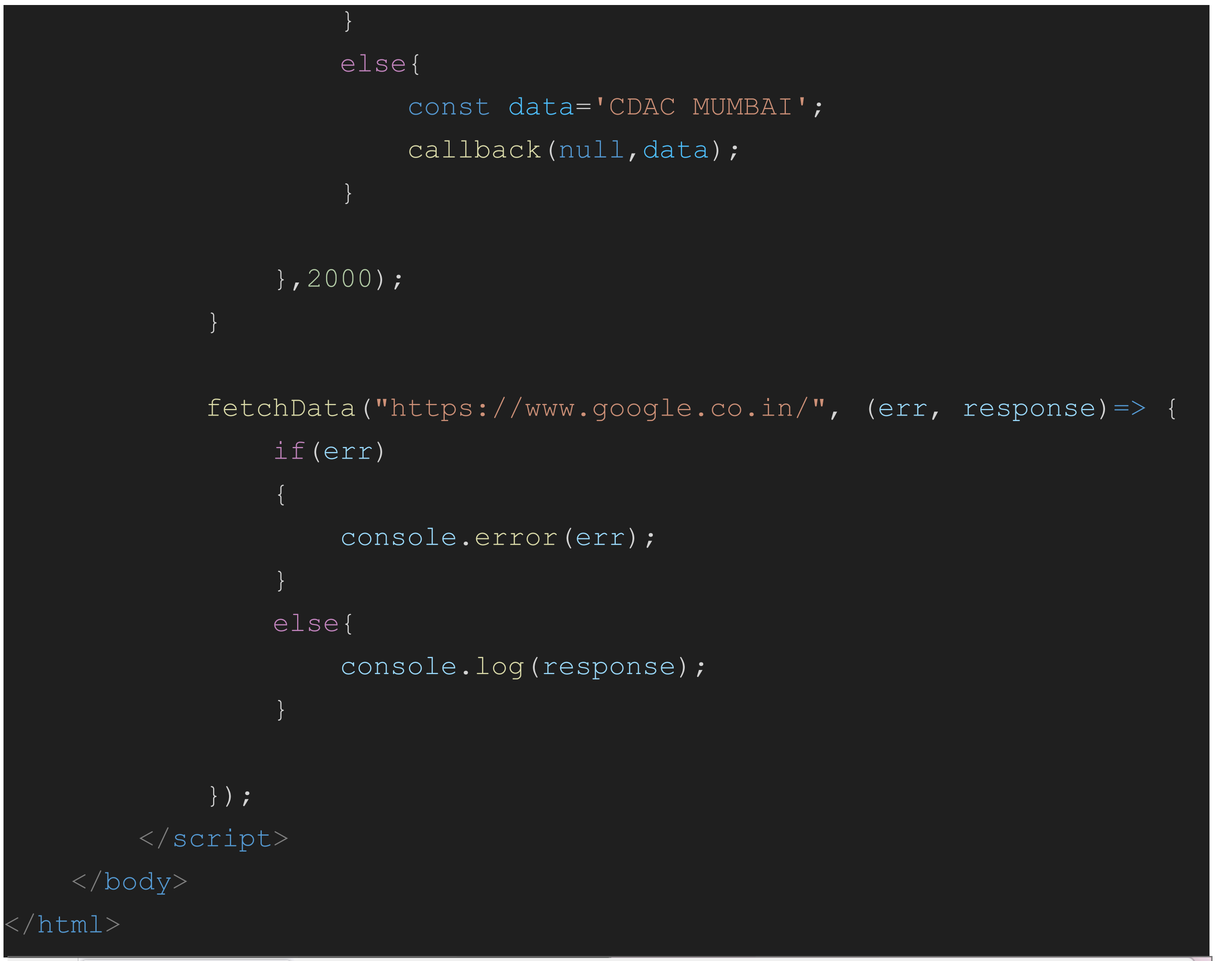
```
<!DOCTYPE html>
<html lang="en">
    <body>
        <script>
            function fetchData(url,callback)
            {

                setTimeout(()=>{
                const error = 5/0;
                    const data='CDAC MUMBAI';
                    if(error)
                    {
                        callback('Error during fetching data',null);

                    }
                    else{
                        callback(null,data);
                    }

                },2000)
            }

            fetchData("https://www.google.co.in/",(err,response)=>{
                if(err)
                {
                    console.error(err);
                }
                else{
                    console.log(response);
                }


            });
        </script>
    </body>
</html>
```



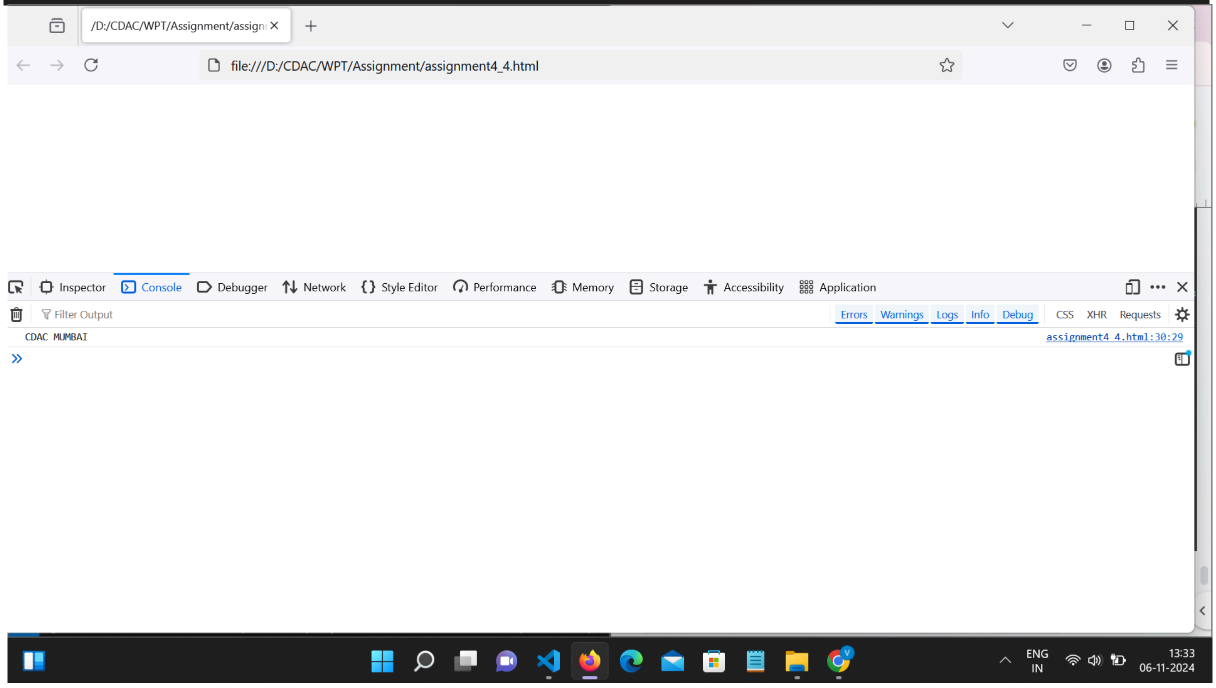
```
file:///D:/CDAC/WPT/Assignment/assignment4_4.html
```

```
Error during fetching data                                    assignment4_4.html:26:29
```

```
<!DOCTYPE html>
<html lang="en">
    <body>
        <script>
            function fetchData(url,callback)
            {

                setTimeout(()=>{
                const error = false;

                    if(error)
                    {
                        callback('Error during fetching data',null);
```



```
                }
                else{
                    const data='CDAC MUMBAI';
                    callback(null,data);
                }

            },2000);
        }

        fetchData("https://www.google.co.in/", (err, response)=> {
            if(err)
            {
                console.error(err);
            }
            else{
                console.log(response);
            }

        });
    </script>
  </body>
</html>
```

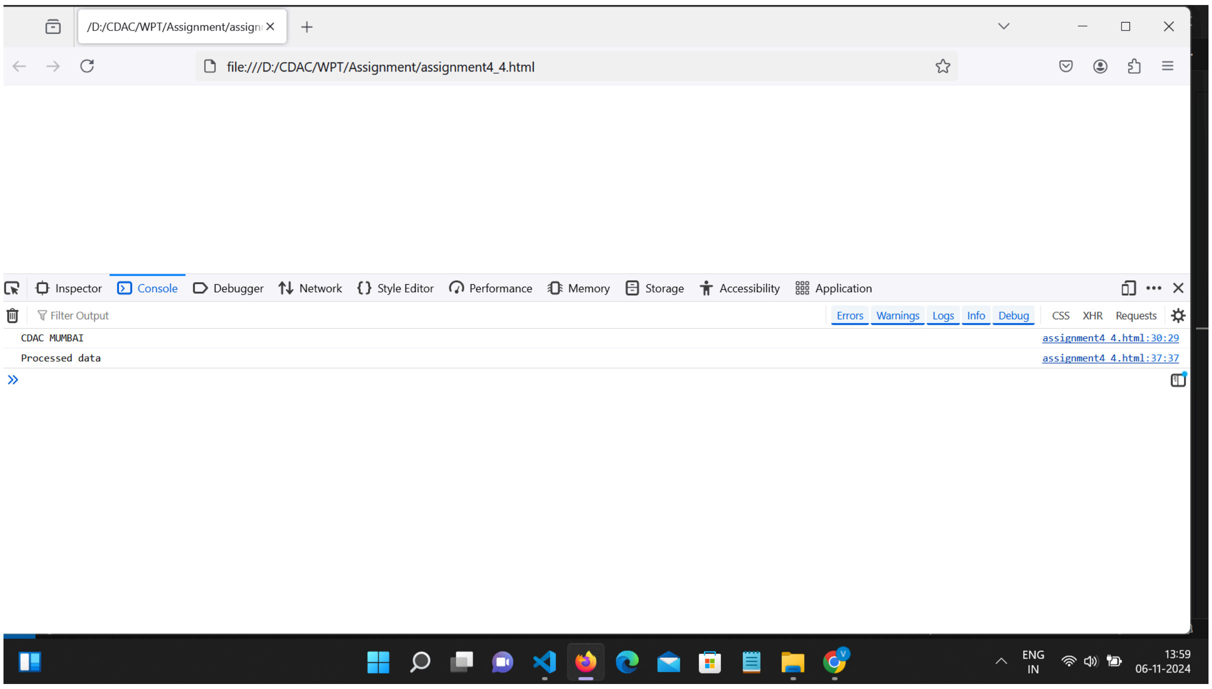


```
CDAC MUMBAI                                                        assignment4_4.html:30:29
```

Exercise 5:
Using fetchData from Exercise 4, create another function processData that simulates processing the fetched data. Chain these functions together using nested callbacks.
Requirements:

● First, call fetchData. Once the response is received, pass it to processData.
● processData should modify the data and log the processed result.

```
<!DOCTYPE html>
<html lang="en">
    <body>
        <script>
            function fetchData(url,callback)
            {

                setTimeout(()=>{

                        const data='CDAC MUMBAI';
                        callback(null,data);

                },1000);
            }

            function processData(data,callback){
                setTimeout(() => {
                    const processedData = 'Processed data';
                    callback(null,processedData);
                },1000);

            }

            fetchData("https://www.google.co.in/", (err, data)=> {
                if(err)
                {
                    console.error(err);
                }
                else{
                    console.log(data);

                    processData(data, (err,processedData) => {
                        if(err){
                            console.error(err);
                        }
                        else{
                            console.log(processedData);
                        }
                    });
                }
            });
        </script>
    </body>
</html>
```



file:///D:/CDAC/WPT/Assignment/assignment4_4.html

```
CDAC MUMBAI                                         assignment4_4.html:30:29
Processed data                                      assignment4_4.html:37:37
```