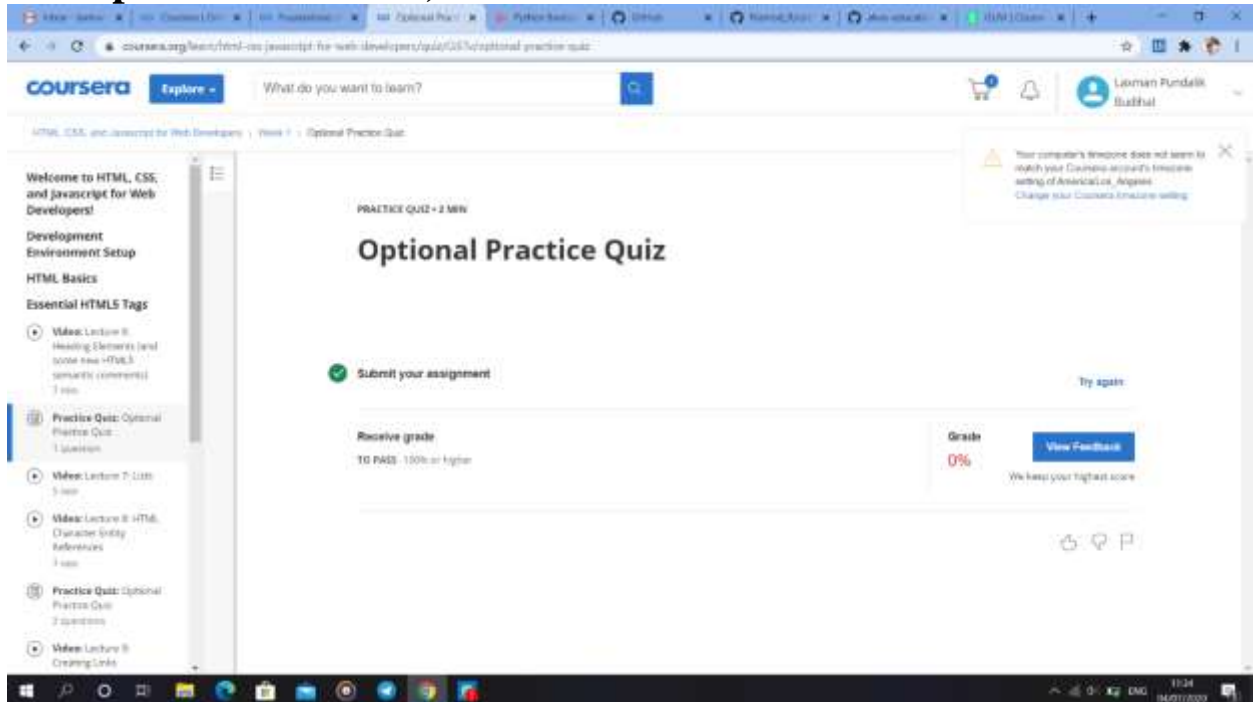


DAILY ONLINE ACTIVITIES SUMMARY

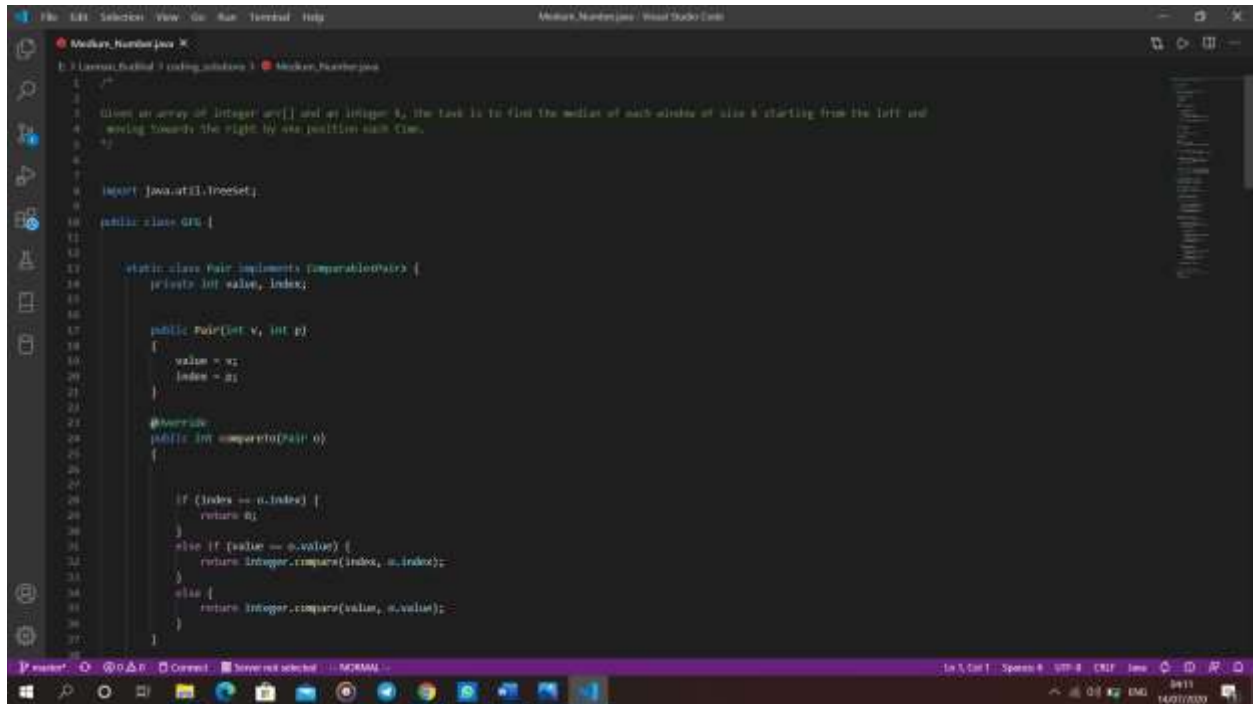
Date:	04/07/2020	Name:	Laxman Pundalik Budihal
Sem & Sec	4 rd sem (A sec)	USN:	4AL18CS043
Online Test Summary			
Subject	-		
Max. Marks	-	Score	-
Certification Course Summary			
Course	HTML, CSS, and Javascript for Web Developers		
Certificate Provider	Coursera	Duration	15 hours
Coding Challenges			
Problem Statement: Given an array of integer arr[] and an integer k, the task is to find the median of each window of size k starting from the left and moving			
Status: Completed			
Uploaded the report in GitHub		YES	
If yes Repository name		https://github.com/alvas-education-foundation/Laxman_Budihal	
Uploaded the report in slack		YES	

Certification Course Details: (Attach the snapshot and briefly write the report for the same)



The today's topic is about HTML, CSS, and Javascript for Web Developers

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)



```
1 // Median.java
2
3 // Given an array of integer arr[] and an integer k, the task is to find the median of each window of size k starting from the left and
4 // moving towards the right by one position each time.
5
6
7
8 import java.util.TreeSet;
9
10 public class Median {
11
12     static class Pair implements Comparable<Pair> {
13         private int value, index;
14
15         public Pair(int v, int p) {
16             value = v;
17             index = p;
18         }
19
20         @Override
21         public int compareTo(Pair o) {
22             if (index == o.index) {
23                 return 0;
24             }
25             else if (value == o.value) {
26                 return Integer.compare(index, o.index);
27             }
28             else {
29                 return Integer.compare(value, o.value);
30             }
31         }
32     }
33
34     static int findMedian(int arr[], int k) {
35         TreeSet<Pair> ts = new TreeSet<>();
36         for (int i = 0; i < arr.length; i++) {
37             ts.add(new Pair(arr[i], i));
38             if (i < k - 1) continue;
39             if (i >= k) ts.remove(new Pair(arr[i - k], i - k));
40             int median = ts.get(k / 2).value;
41             System.out.print(median + " ");
42         }
43     }
44 }
```

The question I took to code is: Given an array of integer `arr[]` and an integer `k`, the task is to find the median of each window of size `k` starting from the left and moving towards the right by one position each time.