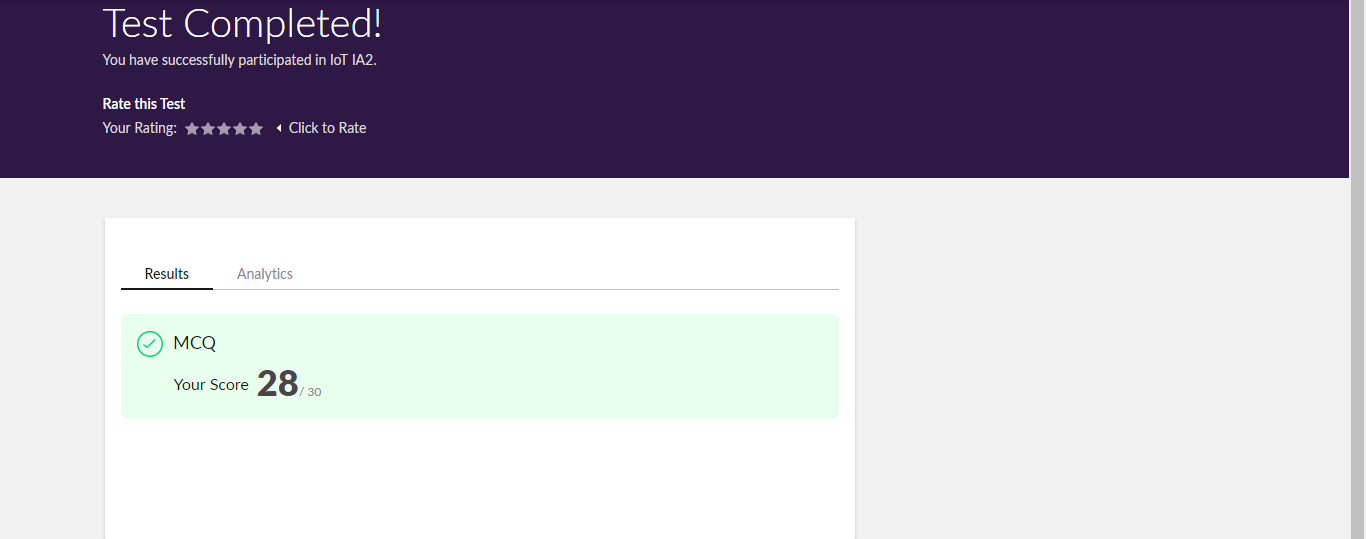
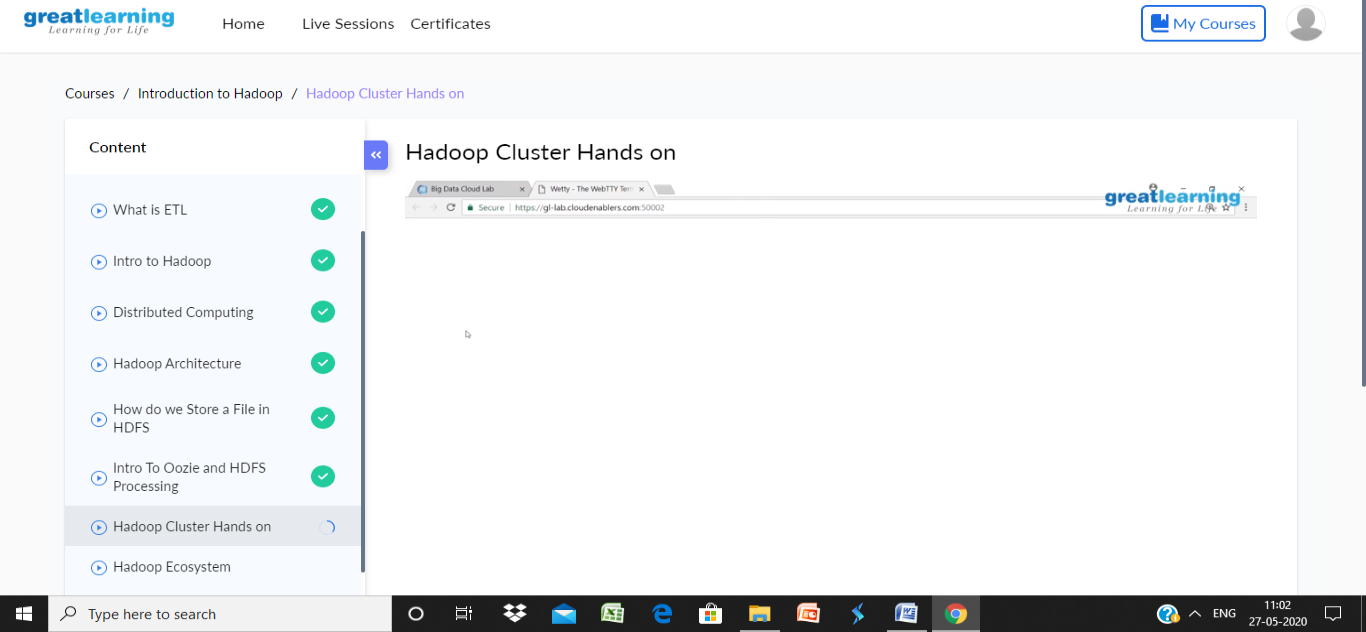
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **27-5-2020** | | | | | **Name:** | **poojashree** | |
| **Sem & Sec** | **8th sem A sec** | | | | | **USN:** | **4al16cs065** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **Introduction to things** | | | | | | |
| **Max. Marks** | | **30** | | **Score** | | | **28** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Introduction to hadoop** | | | | | | | |
| **Certificate Provider** | | | **Great learning academy** | | **Duration** | | | **3.0hr** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:** **1**. **find the largest element on the left side of each index which is smaller than the element present at that index.**  **2. In Bubble sort, each pass consists of comparison each element in the file with its successor (i.e. x[i] with x[i+1]) and interchanging two elements if they are not in the proper order** Top of Form | | | | | | | | |
| **Status:completed** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | **Poojashree** | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

**Online test**

****

**Certification course**



**Coding**

**Program 1**

1. **find the largest element on the left side of each index which is smaller than the element present at that index.**

**def findMaximumBefore(arr, n):**

**for i in range(n):**

**currAns = -1**

**for j in range(i-1,-1,-1):**

**if (arr[j] > currAns and**

**arr[j] < arr[i]):**

**currAns = arr[j]**

**print(currAns,end=" ")**

**if \_\_name\_\_ == '\_\_main\_\_':**

**arr=[4, 7, 6, 8, 5 ]**

**n = len(arr)**

**# Function Call**

**findMaximumBefore(arr, n)**

**coding 2**

1. **In Bubble sort, each pass consists of comparison each element in the file with its successor (i.e. x[i] with x[i+1]) and interchanging two elements if they are not in the proper order. The array may be sorted in any pass. If the array is sorted, then remaining passes should be skipped off. Write a C Program to sort an array of integers in ascending order and display the sorted array and Number of passes performed for sorting.**

|  |  |  |
| --- | --- | --- |
|  | BubbleSort (Arr, N) BubbleSort (Arr, N) | |
|  | For ( I:= 1 to (N-1) ) |
|  | { |
|  |  |
|  | noSwap = true; |
|  | For ( J:= 1 to (N-I) ) |
|  | { |
|  | If ( Arr [J] > Arr[J+1] ) |
|  | { |
|  | Swap( Arr[j], Arr[J+1] ); |
|  | noSwap = false; |
|  | } |
|  | } |
|  | If (noSwap) |
|  | break; |
|  | } |
|  | } |