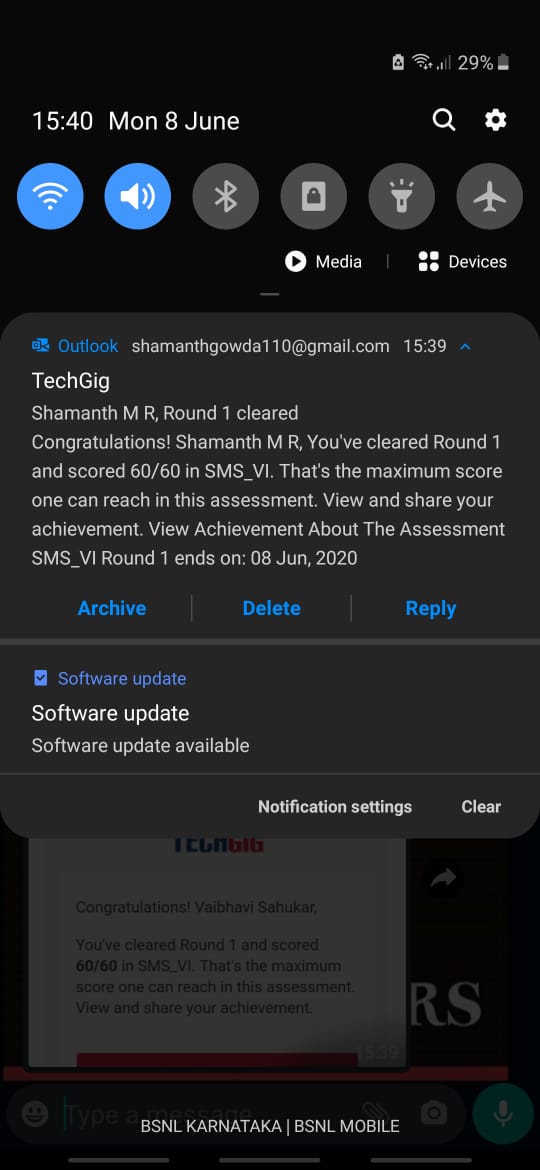
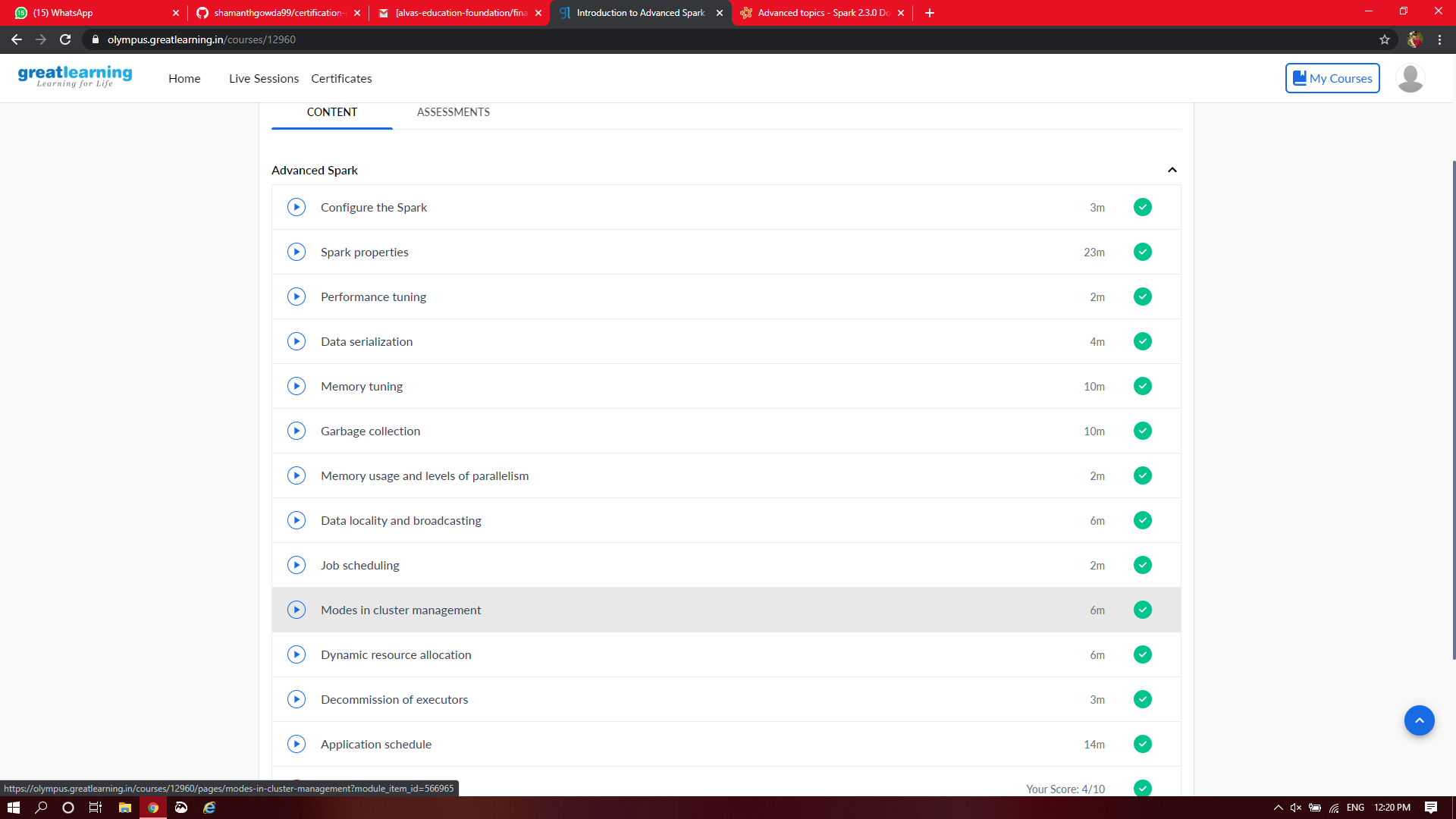
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **08-06-2020** | | | | | **Name:** | **Shamanth M R** | |
| **Sem & Sec** | **8th sem B sec** | | | | | **USN:** | **4AL16CS088** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **SMS** | | | | | | |
| **Max. Marks** | | **60** | | **Score** | | | **60** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Cloud foundations** | | | | | | | |
| **Certificate Provider** | | | **Great learning** | | **Duration** | | | **10-2.30** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement-** : **generate all unique partition of integer.** | | | | | | | | |
| **Status: completed** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | **Shamanth-Gowda** | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

Online Test Details: (Attach the snapshot and briefly write the report for the same)

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

Saw videos on machine learning

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

Coding was given and it was uploaded for github and slack

**generate all unique partition of integer**

Coding was given n it was uploaded for github and slack

**Generate all unique partitions of an integer**

**def printArray(p, n):**

**for i in range(0, n):**

**print(p[i], end = " ")**

**print()**

**def printAllUniqueParts(n):**

**p = [0] \* n # An array to store a partition**

**k = 0 # Index of last element in a partition**

**p[k] = n # Initialize first partition**

**# as number itself**

**while True:**

**printArray(p, k + 1)**

**rem\_val = 0**

**while k >= 0 and p[k] == 1:**

**rem\_val += p[k]**

**k -= 1**

**if k < 0:**

**print()**

**return**

**p[k] -= 1**

**rem\_val += 1**

**while rem\_val > p[k]:**

**p[k + 1] = p[k]**

**rem\_val = rem\_val - p[k]**

**k += 1**

**p[k + 1] = rem\_val**

**k += 1**

**print('All Unique Partitions of 2')**

**printAllUniqueParts(2)**

**print('All Unique Partitions of 3')**

**printAllUniqueParts(3)**

**print('All Unique Partitions of 4')**

**printAllUniqueParts(4)**