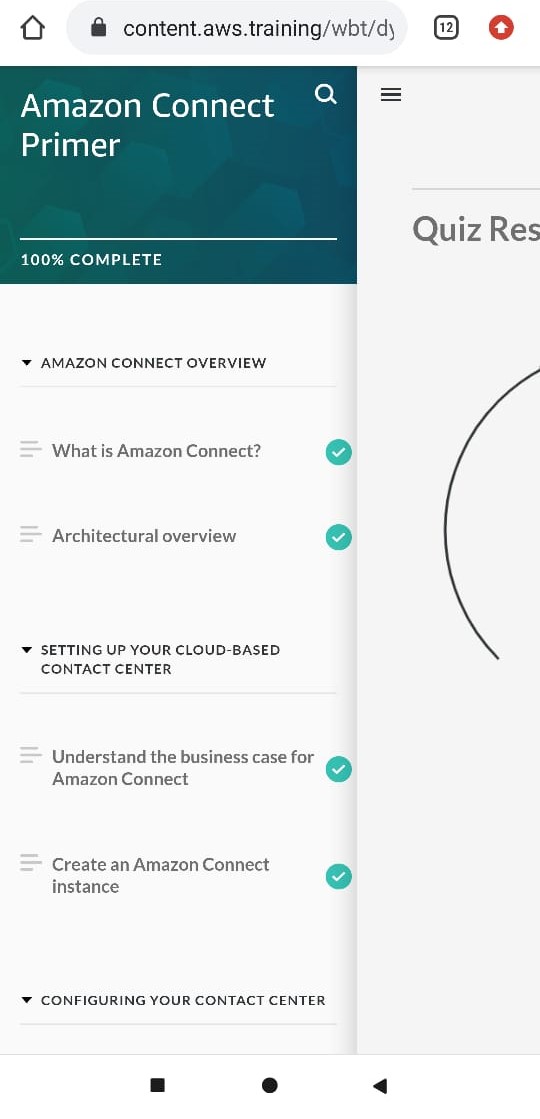
DAILY ONLINE ACTIVITIES SUMMARY

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Date: | 08-07-2020 | | | | Name: | Veekshith Shetty | |
| Sem & Sec | 8th B | | | | USN: | 4AL16CS097 | |
| Online Test Summary | | | | | | | |
| Subject | | ----- | | | | | |
| Max. Marks | | ----- | | Score | | ----- | |
| Certification Course Summary | | | | | | | |
| Course | Amazon connect primer | | | | | | |
| Certificate Provider | | | AWS | Duration | | | 4 hr |
| Coding Challenges | | | | | | | |
| Problem Statement- : generate all unique partition of integer. | | | | | | | |
| Status: completed | | | | | | | |
| Uploaded the report in Github | | | | yes | | | |
| If yes Repository name | | | | Veekshith-Shetty | | | |
| Uploaded the report in slack | | | | yes | | | |

Online Test Details:

Certification Course Details



Coding Challenges Details

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)