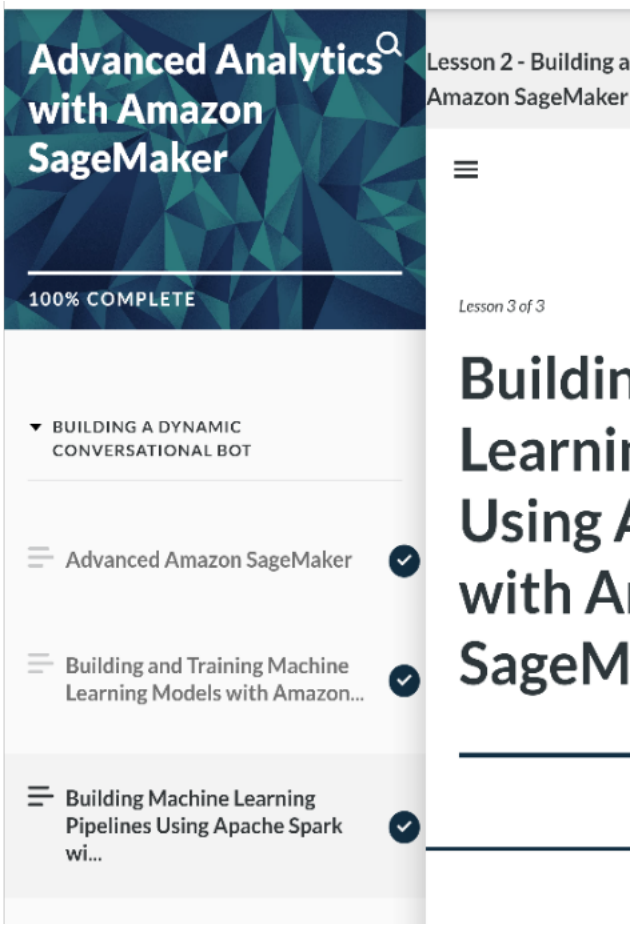
# DAILY ONLINE ACTIVITIES SUMMARY

| | | | |
|---|---|---|---|
| Date: | 08-06-2020 | Name: | Sinchana Kamath |
| Sem & Sec | 8th sem B sec | USN: | 4AL16CS102 |
| Online Test Summary | | | |
| Subject | SMS | | |
| Max. Marks | 60 | Score | 60 |
| Certification Course Summary | | | |
| Course | Machine learning | | |
| Certificate Provider | Aws | Duration | 3hr |
| Coding Challenges | | | |
| Problem Statement: generate all unique partition of integer | | | |
| Status: completed | | | |
| Uploaded the report in Github | | yes | |
| If yes Repository name | | sinchana Kamath | |
| Uploaded the report in slack | | yes | |

YOnline 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)


Advanced Analytics with Amazon SageMaker
100% COMPLETE
Lesson 2 - Building a Amazon SageMaker
Lesson 3 of 3
BUILDING A DYNAMIC CONVERSATIONAL BOT
Advanced Amazon SageMaker
Building and Training Machine Learning Models with Amazon...
Building Machine Learning Pipelines Using Apache Spark wi...


aws training and certification
Certificate of Completion
sinchana S Kamath
Has successfully completed
Advanced Analytics with Amazon SageMaker
Director, Training and Certification
25 minutes
Duration
8 June, 2020
Completion Date

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

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)
```