DAILY ONLINE ACTIVITIES SUMMARY

Date:	**21/05/2020**	**Name:**	**AMEEN AHMED**
Sem & Sec	**8 A**	**USN:**	**4AL16CS009**

Online Test Summary

Subject	**SMS**		
Max. Marks	**60**	**Score**	**58**

Certification Course Summary

Course	**INTRODUCTION TO HADOOP**		
Certificate Provider	**GREAT LEARNING**	**Duration**	**30 MINS**

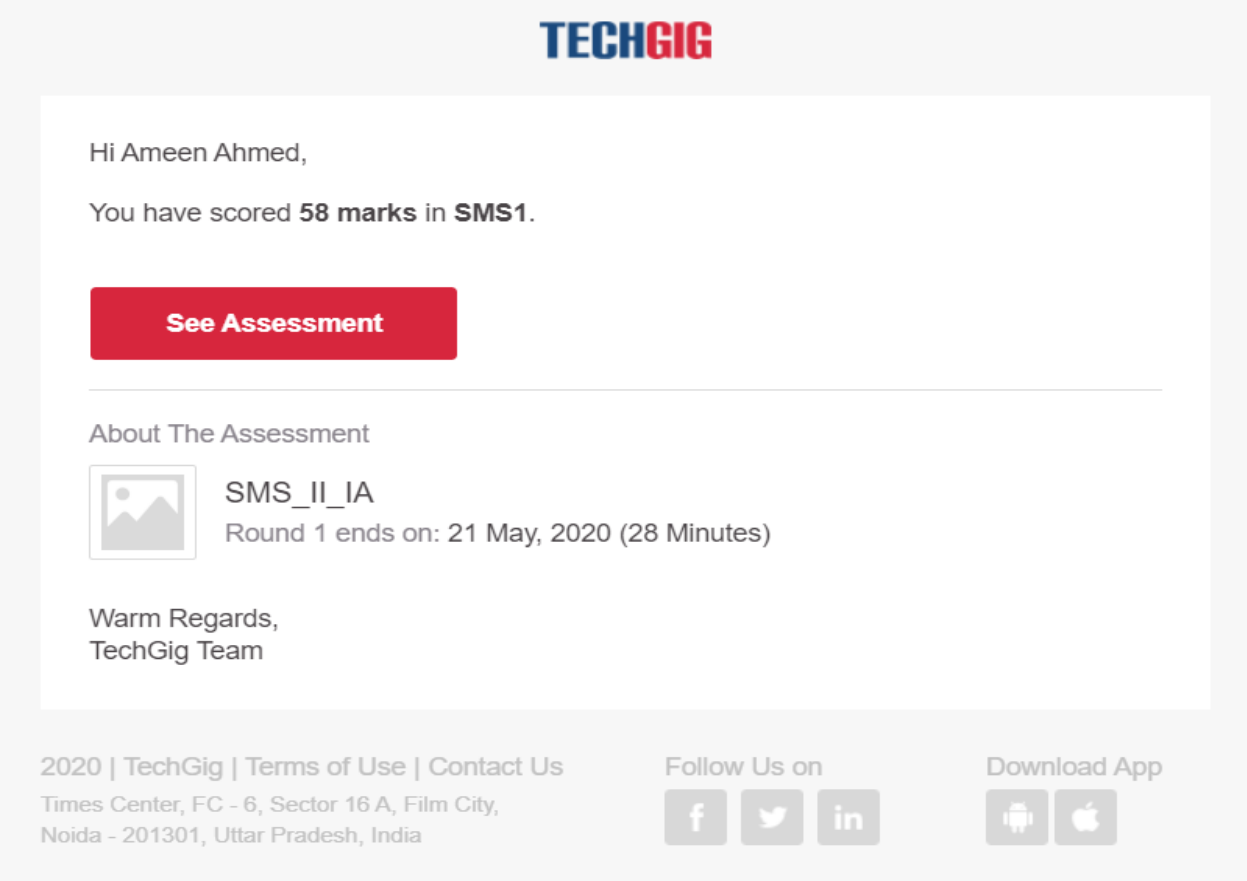
Coding Challenges

Problem Statement: C Program to Reverse a Linked List in groups of given size
Status: COMPLETED

Uploaded the report in Github	**YES**
If yes Repository name	**Ameen**
Uploaded the report in slack	**YES**

Online Test Details:

SMS – Test 2



Certification Course Details:

What is distribution computing?

Distributed computing is a model in which components of a software system are shared among multiple computers to improve efficiency and performance.

The Distributed Computing Environment (DCE) is a widely-used industry standard that supports this kind of distributed computing. On the Internet, third-party service providers now offer some generalized services that fit into this model. One of the first uses of grid computing was the breaking of a cryptographic code by a group that is now known as distributed.net. That group also describes its model as distributed computing.

Hadoop Architecture

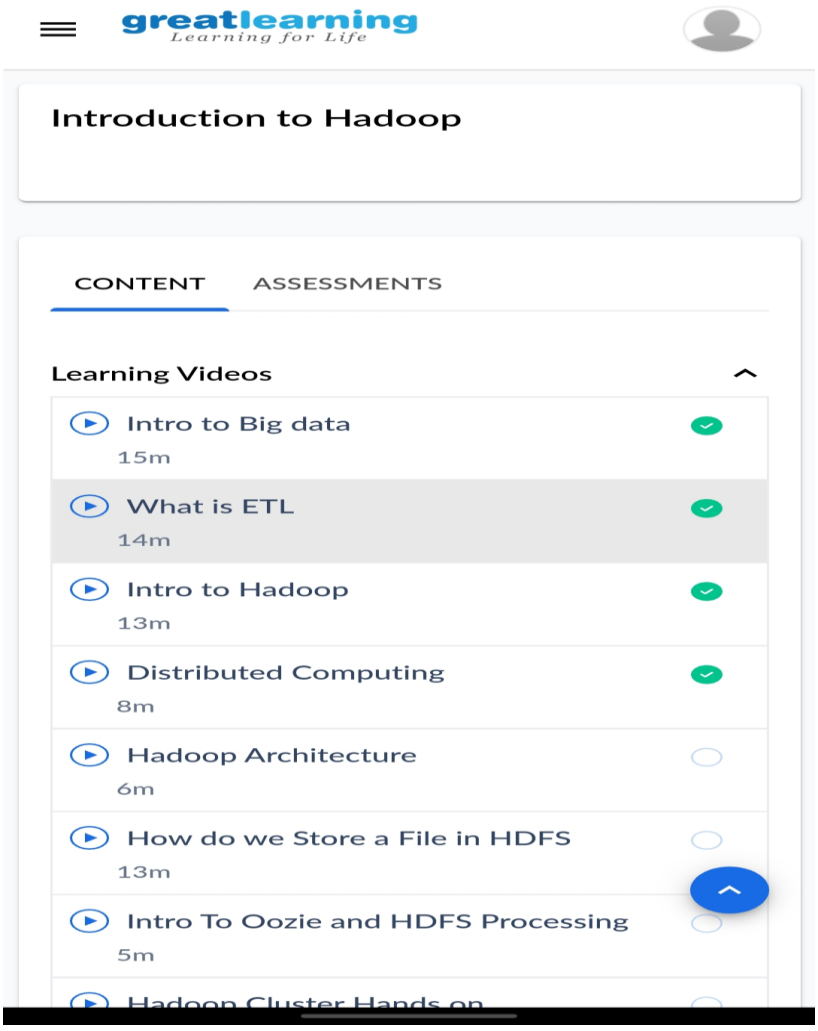
Apache Hadoop offers a scalable, flexible and reliable distributed computing big data framework for a cluster of systems with storage capacity and local computing power by leveraging commodity hardware. Hadoop follows a Master Slave architecture for the transformation and analysis of large datasets using Hadoop MapReduce paradigm.

The 3 important hadoop components that play a vital role in the Hadoop architecture are

Hadoop Distributed File System (HDFS) – Patterned after the UNIX file system

Hadoop MapReduce

Yet Another Resource Negotiator (YARN)



Coding Challenges Details:

Write a C Program to Reverse a Linked List in groups of given size.

```
#include<stdio.h>
#include<stdlib.h>
struct Node
{
   int data;
   struct Node* next;
   };
struct Node reverse(struct Node head,int k)
{
  struct Node current= head;
  struct Node next= Null;
  struct Node prev= Null;
   int count = 0;
          while(current!=Null && count<k)
          {
              next= current->next;
              current->next = prev;
              prev= current;
              current= next;
              count++;
          }
          if ( next!=Null)
          head->next= reverse( next,k);
          return prev;
         }
        void push( struct Node ==head_ref,int new_data)
         {
           struct Node= new_node= (struct Node*) malloc(sizeof(struct Node));


 }

}
int main()
{
    Struct node *prev,*head,*p;
    int n,i;
    printf ("number of elements:");
    scanf("%d",&n);
    head=NULL;
    for(i=0;i<n;i++)
    {
```

```
        p=malloc(sizeof(struct node));
        scanf("%d",&p->data);
        p->next=NULL;
        if(head==NULL)
            head=p;
        else
            prev->next=p;
        prev=p;
    }
    return 0;
}
```