DAILY ONLINE ACTIVITIES SUMMARY

Date:	**21-05-2020**	**Name:**	**Ainab**
Sem & Sec	**VIII Semester & A Section**	**USN:**	**4AL16CS004**

Online Test Summary

Subject	**System Modeling & Simulation**		
Max. Marks	**60**	**Score**	**60**

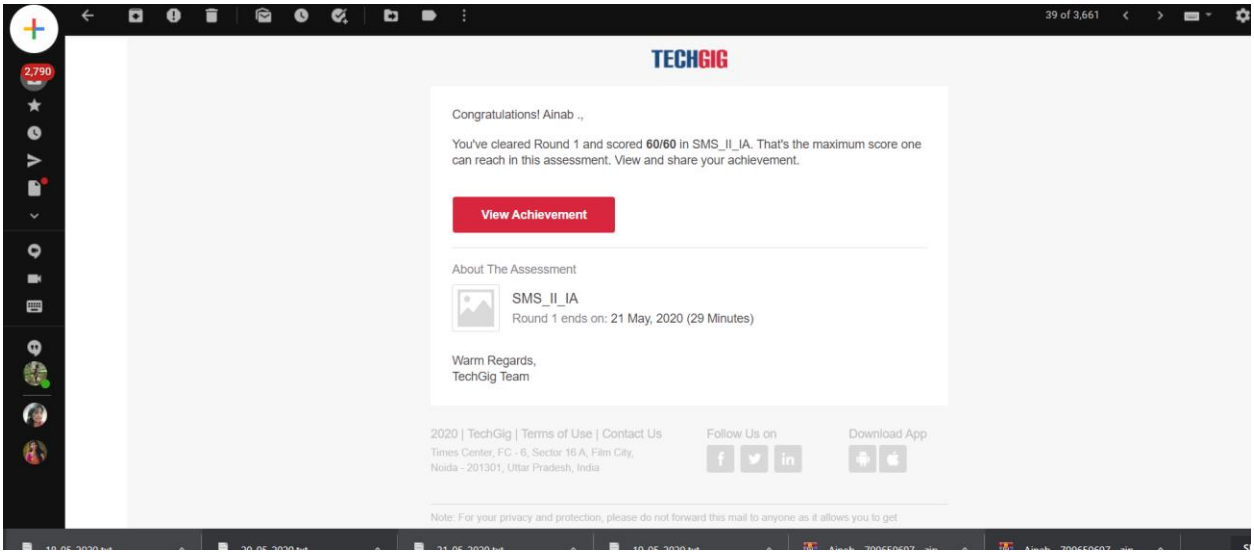
Certification Course Summary

Course	**Introduction to Hadoop**		
Certificate Provider	**Great Learning**	**Duration**	**8+6=14mins**

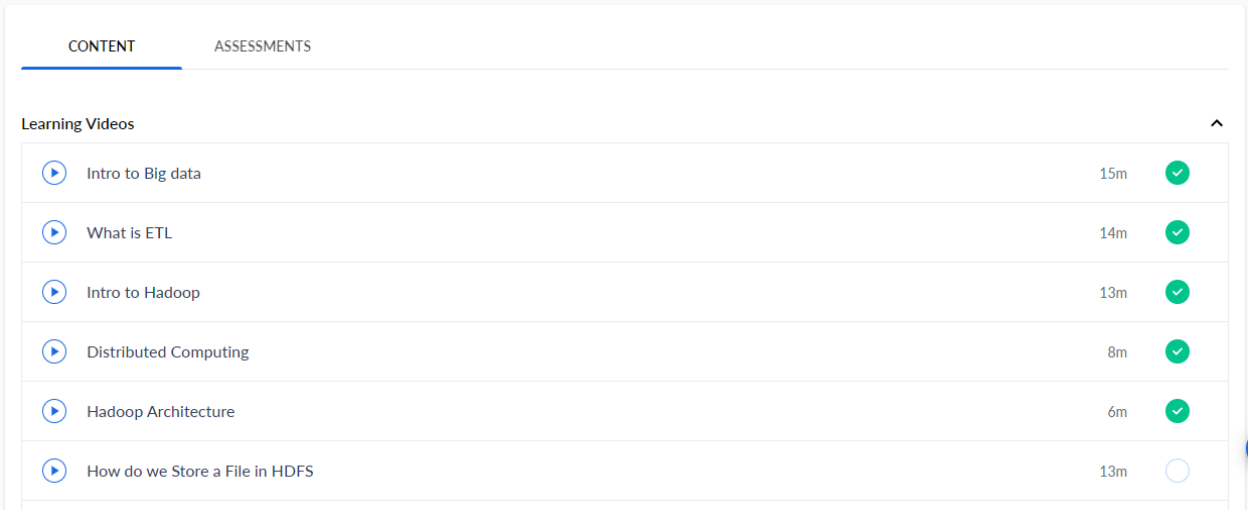
Coding Challenges

Problem Statement: Creating SLL and reversing the link into SSL until head becomes null.	
Status: COMPLETED	
Uploaded the report in Github	**YES**
If yes Repository name	**Ainab004**
Uploaded the report in slack	**YES**

Online Test Details:



Certification Course Details:



Distributed Computing:

The Hadoop take a bunch of machines, while installing Hadoop will ask which is master and which is slave. Hadoop is a framework in which we can expand to any number of machine and due to Hadoop the user does not have to worry about the storage because it can resize without downtime. We can add machine and also remove it only if there is no program running in the machine. This whole process is called as Hadoop

Clustering. Clustering means group of machines. The Hadoop clusters can also be built in desktops.

Hadoop Architecture:

Hadoop has three major components:

1. **HDFS**
2. **MapReduce**
3. **YARN**

In Hadoop Distributed File System there are two nodes namenode and datanode. Here namenode acts like a master and datanode acts like a slave we can have any number of slaves. In HDFS we can delete any file but small editing is little difficult but is possible by using other platform.HDFS takes care of storage.

Coding Challenges Details:

Create SLL, and then reverse the link in SLL until head becomes NULL. Each time reversing the link head must be moved to next immediate node.

Solution:

```
#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;
```