DAILY ONLINE ACTIVITIES SUMMARY

Date:	**21/05/2019**	**Name:**	**Safnaaz**
Sem & Sec	**8th B**	**USN:**	**4AL16CS081**

Online Test Summary

Subject	**SMS 2**		
Max. Marks	**60**	**Score**	**58**

Certification Course Summary

Course	**Introduction To Hadoop**		
Certificate Provider	**Great Learning Academy**	**Duration**	**30 mins**

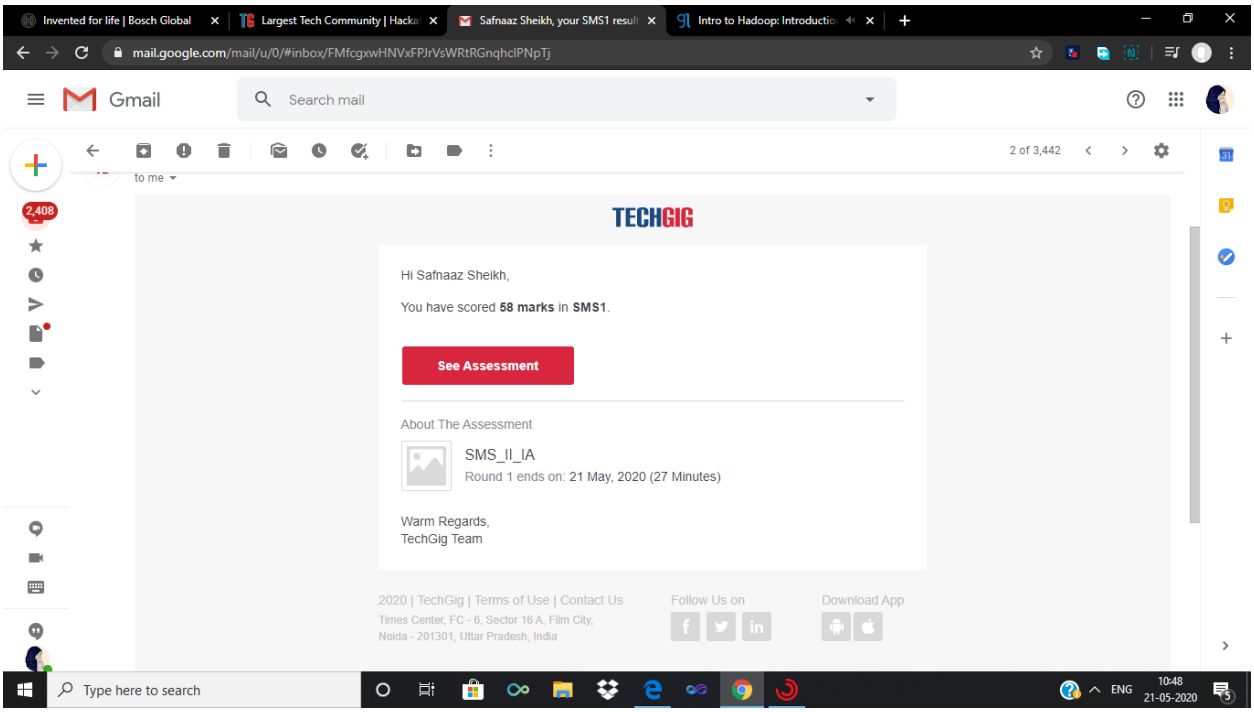
Coding Challenges

Problem Statement:1) To add some letters for a given word or letter then to find shortest palindrome possible.
2) To check whether the given linked list is palindrome or not.

Status: Solved

Uploaded the report in Github	**yes**
If yes Repository name	**Safnaazsheikh**
Uploaded the report in slack	**yes**

Online Test Details:



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.

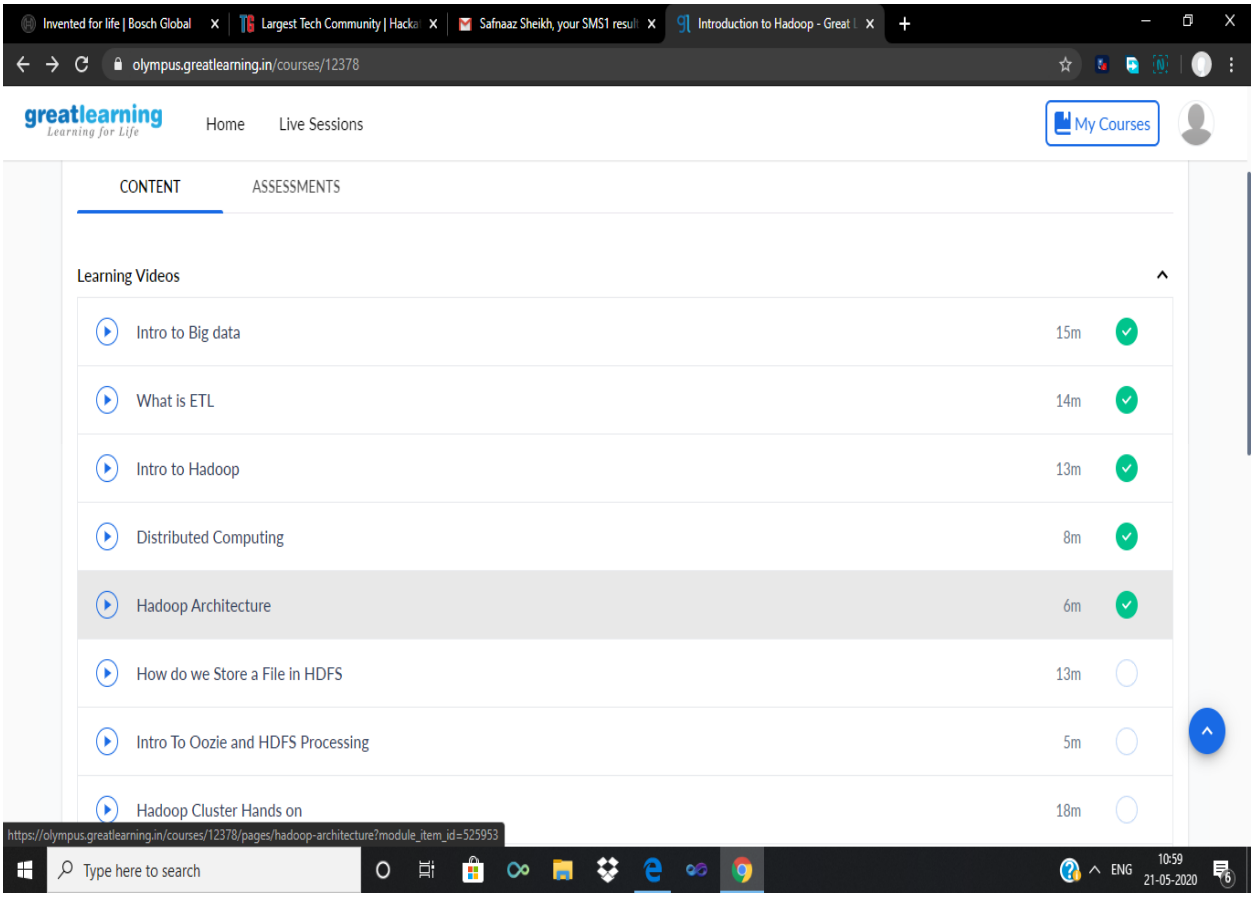
The Distributed Computing Environment (DCE) is a widely-used industry standard that supports this kind of distributed computing. On the Internet, third-party service providers now offer some generalized services that fit into this model. One of the first uses of grid computing was the breaking of a cryptographic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

program1:

```
#include<stdio.h>
#include<stdlib.h>
struct Node
{
int data;
struct Node* next;
};
pointer to the new head node. /
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));
```