DAILY ONLINE ACTIVITIES SUMMARY

Date:	**18/05/2020**	**Name:**	**KIRAN K**
Sem & Sec	**8th A**	**USN:**	**4AL16CS046**

Online Test Summary

Subject	**SMS**		
Max. Marks	**60**	**Score**	**43**

Certification Course Summary

Course	Introduction to Hadoop		
Certificate Provider	**Great learning**	**Duration**	**15 mins**

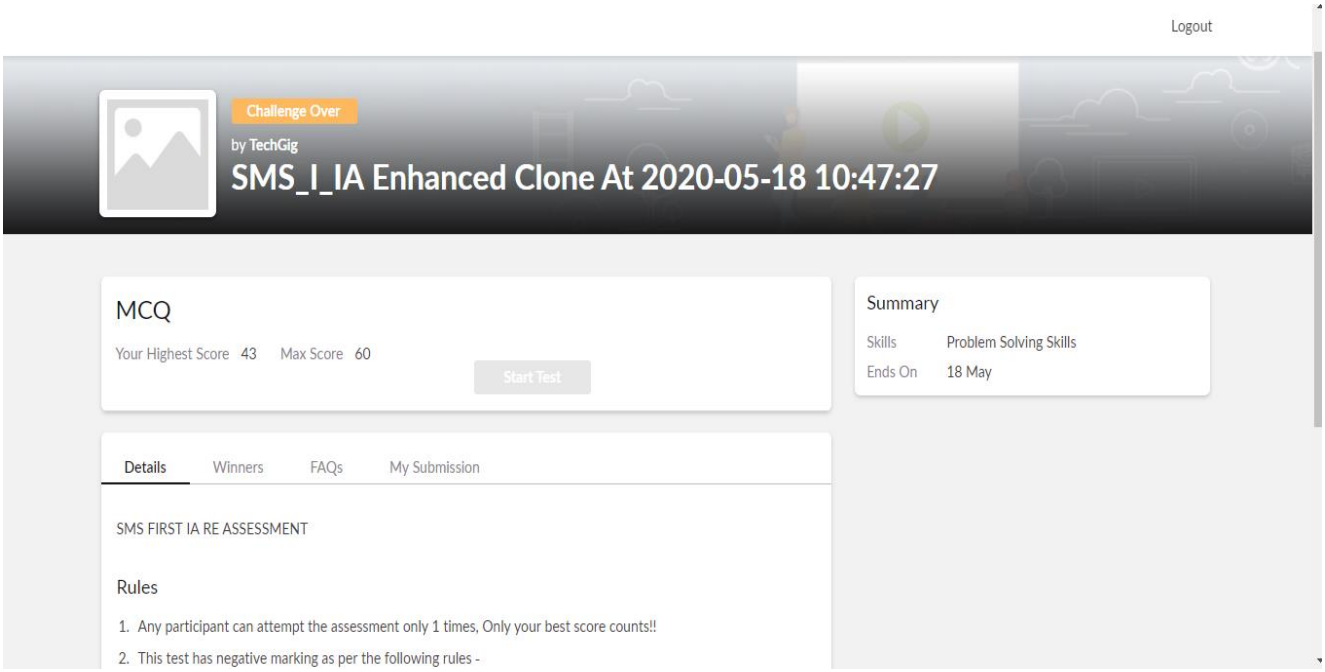
Coding Challenges

Problem Statement:	
Status: COMPLETED	
Uploaded the report in Github	**YES**
If yes Repository name	**KiranK27751**
Uploaded the report in slack	**YES**

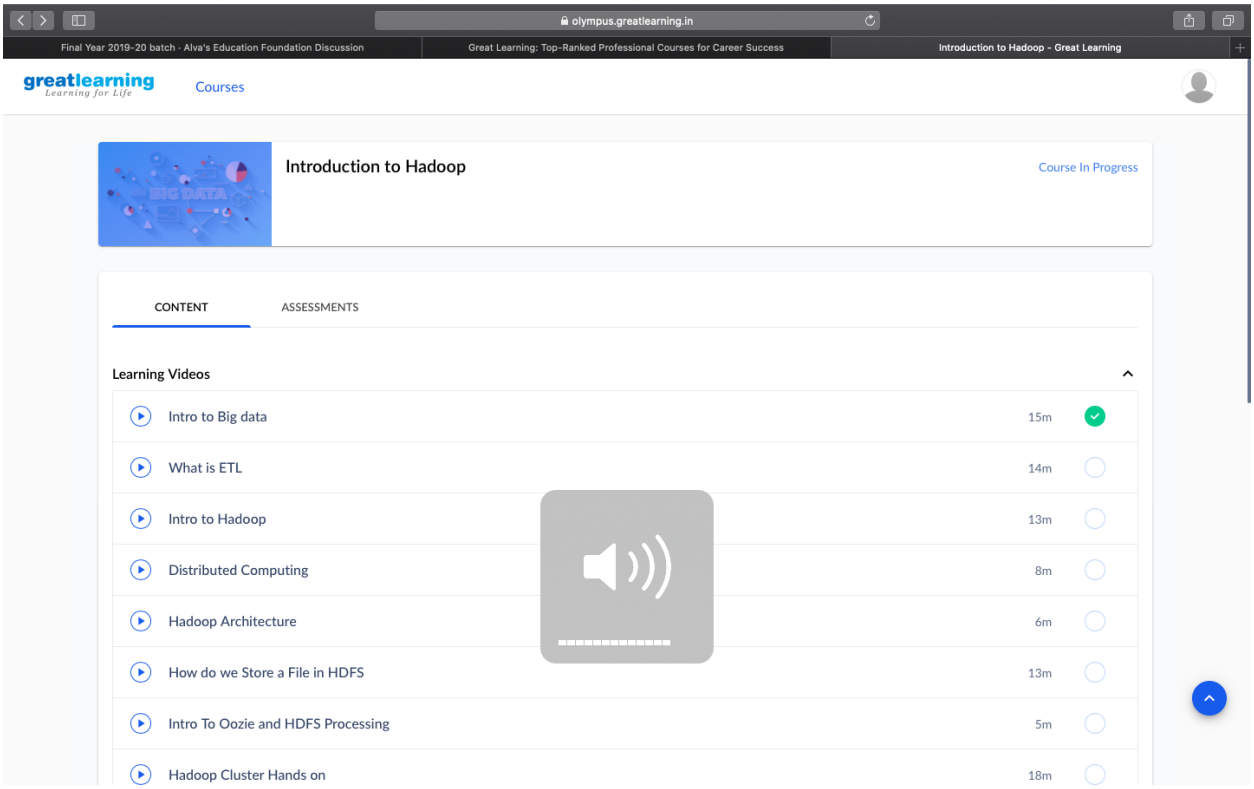
Online Test Details:

Test on module 3 (Random number generation)

Snapshot of test



Certification Course Details:



Introduction to BigData

Big Data is also **data** but with a **huge size**. Big Data is a term used to describe a collection of data that is huge in volume and yet growing exponentially with time. In short such data is so large and complex that none of the traditional data management tools are able to store it or process it efficiently.

Types Of Big Data

BigData' could be found in three forms:

1. **Structured**
2. **Unstructured**
3. **Semi-structured**

Coding Challenges Details

Program no:1

```
package pk;
import java.util.Scanner;
public class StringOperators
{
public static void main(String args[])
{
int i;
String str;

   int counter[] = new int[256];
   Scanner in = new Scanner(System.in);

   System.out.print("Enter a String : ");
   str=in.nextLine();

    for (i = 0; i < str.length(); i++) {
      counter[(int) str.charAt(i)]++;
   }
   // Print Frequency of characters
   for (i = 0; i < 256; i++) {
      if (counter[i] != 0) {
           System.out.println((char) i  + ":-" + counter[i] + " times");
      }
   }
}
}
```

Program no:2

```
public class PingPong extends Thread {
static StringBuilder object = new StringBuilder("");

public static void main(String[] args) throws InterruptedException {

Thread t1 = new PingPong();
```

```
Thread t2 = new PingPong();

t1.setName("\nping");
t2.setName(" pong");

t1.start();
t2.start();
}

@override
public void run() {
working();
}

void working() {
while (true) {
synchronized (object) {
try {
System.out.print(Thread.currentThread().getName());
object.notify();
object.wait();
} catch (InterruptedException e) {
e.printStackTrace();
}
}
}
}
}
```