DAILY ONLINE ACTIVITIES SUMMARY

Date:	**19/05/2020**	**Name:**	**Safnaaz**
Sem & Sec	**8th B**	**USN:**	**4AL16CS081**

Online Test Summary

Subject	**Big data analytic (BDA)**		
Max. Marks	**30**	**Score**	**14**

Certification Course Summary

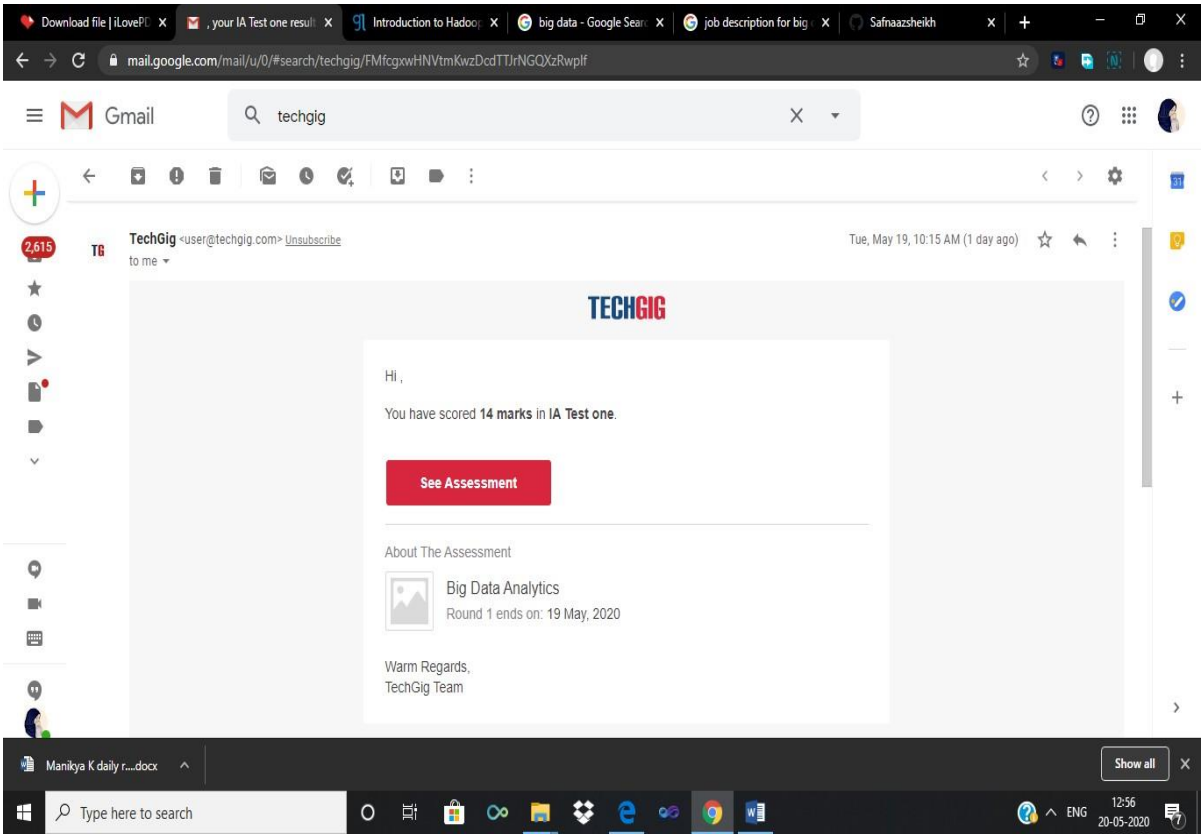
Course	**Introduction to Hadoop**		
Certificate Provider	**Great learning**	**Duration**	**30 mins**

Coding Challenges

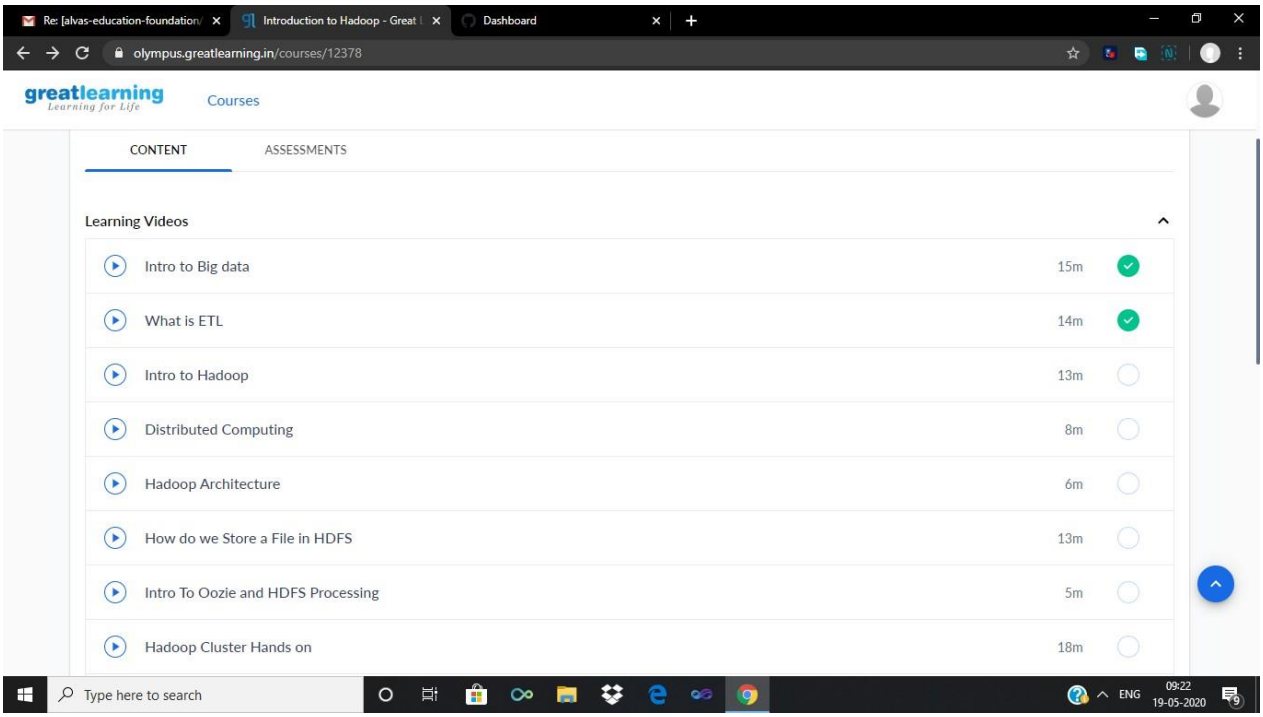
Problem Statement:1) finding frequency of each character in a string and to print even and odd for series. 2) java program	
Status: COMPLETED	
Uploaded the report in Github	**YES**
If yes Repository name	**Safnaazsheikh**
Uploaded the report in slack	**YES**

Online Test Details:

Snapshot of test



Certification Course Details:



What is ETL?

ETL is short for extract, transform, load, three database functions that are combined into one tool to pull data out of one database and place it into another database. Extract is the process of reading data from a database. Transformation occurs by using rules or lookup tables or by combining the data with other data.

Why ETL important

Businesses have relied on the ETL process for many years to get a consolidated view of the data that drives better business decisions. Today, this method of integrating data from multiple systems and sources is still a core component of an organization's data integration toolbox.

- When used with an enterprise data warehouse (data at rest), ETL provides deep historical context for the business.
- By providing a consolidated view, ETL makes it easier for business users to analyze and report on data relevant to their initiatives.

- ETL can improve data professionals' productivity because it codifies and reuses processes that move data without requiring technical skills to write code or scripts.
- ETL has evolved over time to support emerging integration requirements for things like streaming data.
- Organizations need both ETL and ELT to bring data together, maintain accuracy and provide the auditing typically required for data warehousing, reporting and analytics.

Coding Challenges Details

Program no:1

```
package shortestpalindromeexample.java;
import java.util.Scanner;
public class ShortestPalindromeDemo {

public static String shortestPalindrome(String str) {

int x=0;
int y=str.length()-1;

while(y>=0){
if(str.charAt(x)==str.charAt(y)){
x++;
}
y--;
}

if(x==str.length())
return str;

String suffix = str.substring(x);
String prefix = new StringBuilder(suffix).reverse().toString();
String mid = shortestPalindrome(str.substring(0, x));

return prefix+mid+suffix;
}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("Enter a String to find out shortest palindrome");

String str=in.nextLine();
```

```
  System.out.println("Shortest palindrome of "+str+" is "+shortestPalindrome(str));

  }

  Program no:2

 import java.util.Stack;

 class Node {
 int data;

 Node next;

 Node(int i)
{
 this.data = i;
 this.next = null;
 }
 };

 class Main
 {
 // Function to determine if a given linked list is palindrome or not
 public static boolean isPalindrome(Node head)
 {
 // construct an empty stack
 Stack s = new Stack<>();

 // push all elements of the linked list into the stack
 Node node = head;
 while (node != null) {
 s.push(node.data);
 node = node.next;
 }

  // traverse the linked list again
 node = head;
 while (node != null)
 {
 // pop the top element from the stack
 int top = s.pop();

 // compare the popped element with current node's data
// return false if mismatch happens
if (top != node.data) {
return false;
}
```

```
 // advance to the next node
 node = node.next;
 }

 // we reach here only when the linked list is palindrome
 return true;
 }

public static void main(String[] args)
{
Node head = new Node(1);
head.next = new Node(2);
head.next.next = new Node(3);
head.next.next.next = new Node(2);
head.next.next.next.next = new Node(1);

if (isPalindrome(head)) {
System.out.print("Linked List is a palindrome.");
} else {
System.out.print("Linked List is not a palindrome.");
}
}
}

}
}
}
```