DAILY ONLINE ACTIVITIES SUMMARY

Date:	**19/05/2020**	**Name:**	**Shruthi**
Sem & Sec	**8B**	**USN:**	**4AL16CS100**

Online Test Summary

Subject	**BDA**		
Max. Marks	**30**	**Score**	**21**

Certification Course Summary(Internship)

Task	**Study-1.Learning the concepts of ETL and ETL tools.** **2.Creating an UI using an ETL tool.**		
company	**Gain-insights**	**Duration**	**9 hrs**

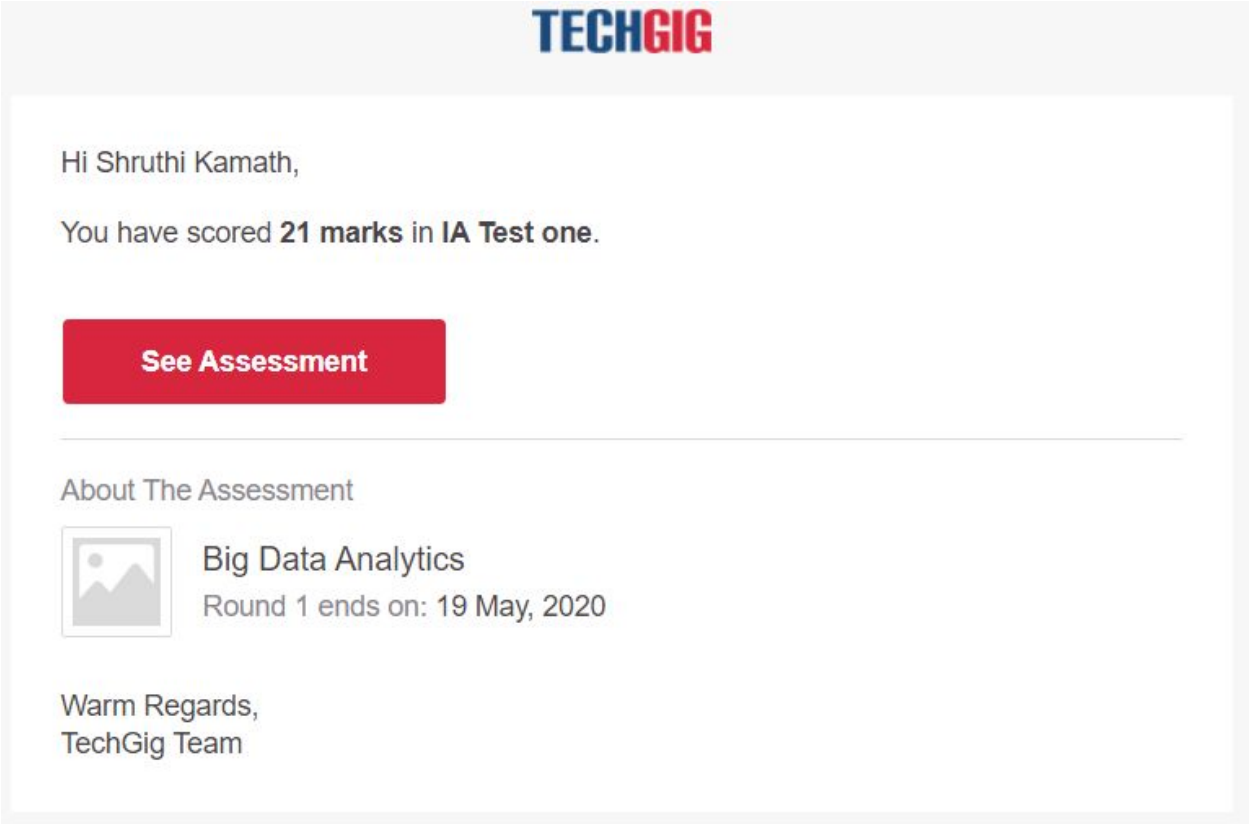
Coding Challenges

Problem Statement:1) Program to add some characters to the given string or character and find out what will be the shortest palindrome string by using simple java program.
2)Java program to identify given linked list is palindrome or not by using stack.

Status:completed

Uploaded the report in Github	**Yes**
If yes Repository name	shruthikamath
Uploaded the report in slack	**Yes**

ONLINE TEST:



CODING CHALLENGE:

PROGRAM 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 2 :

```
import java.util.Stack;
class Node {
int data;
Node next;
Node(int i)
{
this.data = i;
this.next = null;
}
};
class Main
{
public static boolean isPalindrome(Node head)
{

Stack s = new Stack<>();
Node node = head;
while (node != null) {
        s.push(node.data);
        node = node.next;
}

node = head;
while (node != null)
{
        int top = s.pop();
        if (top != node.data) {
                return false;
        }
        node = node.next;
}
```

```
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.");
}
}
```