DAILY ONLINE ACTIVITIES SUMMARY

Date:	**20-05-2020**	**Name:**	**Rakesh M Kotian**
Sem & Sec	**8 th sec-b**	**USN:**	**R4al16cs072**

Online Test Summary

Subject	**sms**		
Max. Marks	**30**	**Score**	**43**

Certification Course Summary

Course	**Python for machine learning**		
Certificate Provider	**Great learning**	**Duration**	**6 hours**

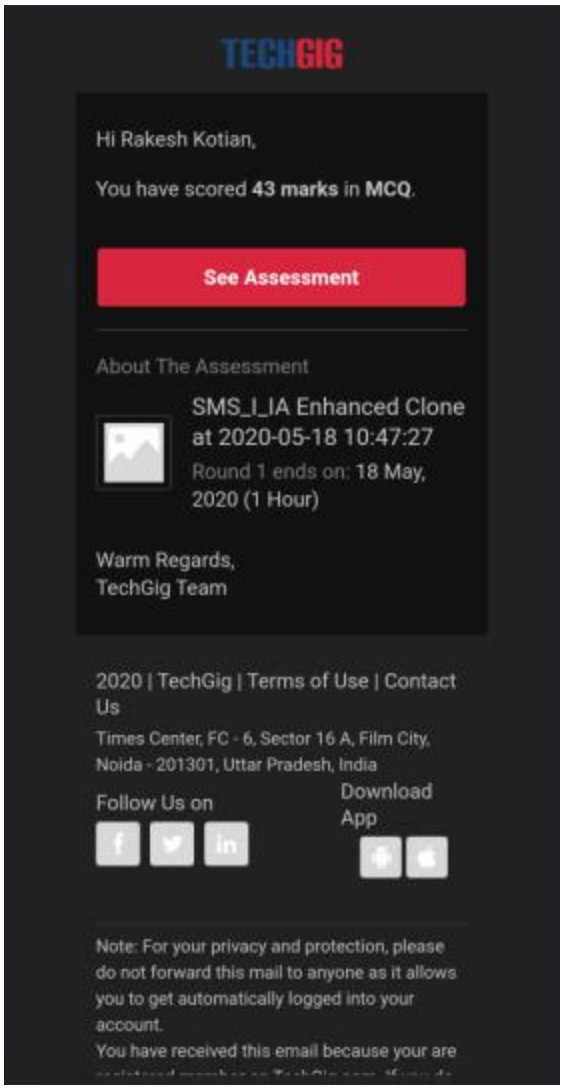
Coding Challenges

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

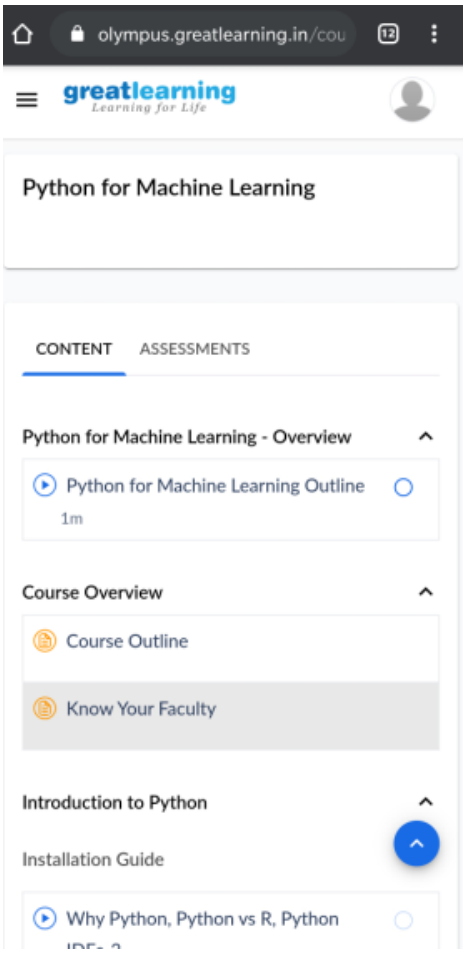
Status:solved

Uploaded the report in Github	**yes**
If yes Repository name	**Rakeshkotian08**
Uploaded the report in slack	**yes**

Online Test Details: (Attach the snapshot and briefly write the report for the same)



Certification Course Details: (Attach the snapshot and briefly write the report for the same)



Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

Prog1:

```
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.c
harAt(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));
}
```

Prog 2:

```
import java.util.Stack;
class Node {
int data;
Node next;
Node(int i)
{t
his.data = i;
this.next = null;
} };
class Main
{
public static boolean isPalindrome(Node head)
{
Stack s = new Stack<>();
Node node = head; // push
while (node != null)
{ s.push(node.data);
node = node.next;
}
// traverse node =
head; while (node
!= null)
{ int top =
s.pop(); //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.");
}}}
```