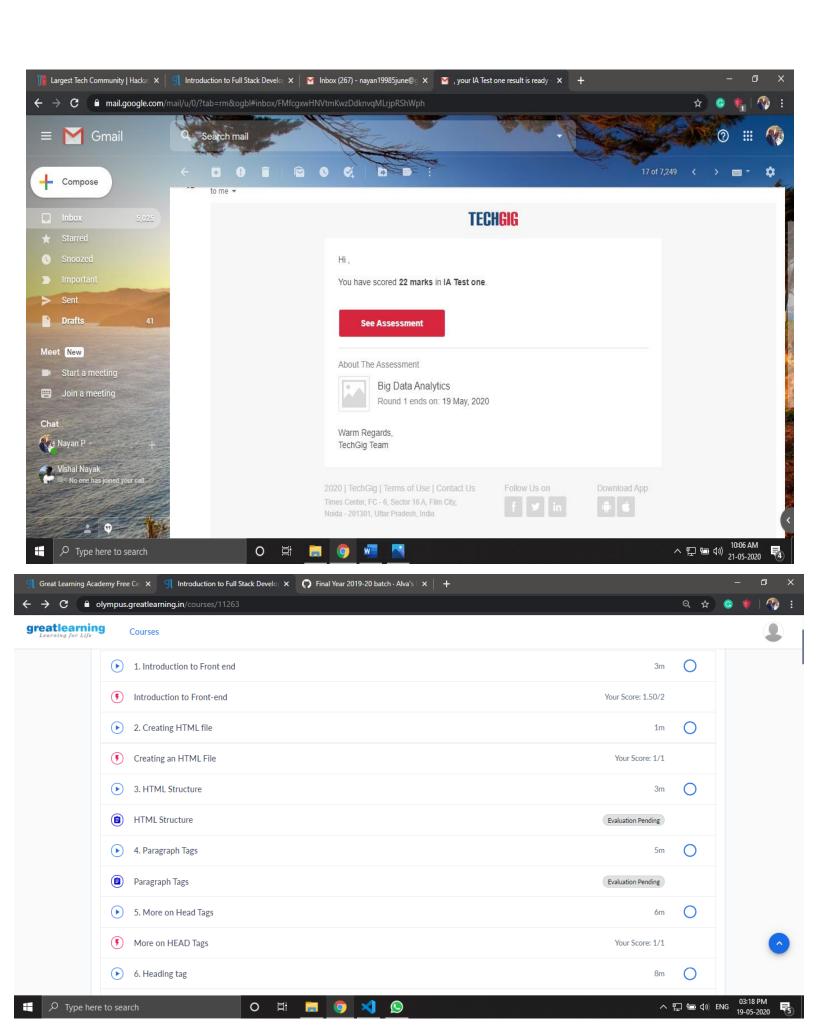
DAILY ONLINE ACTIVITIES SUMMARY

Date:	19-05-2020		Name:	Nayan. P. Joshi	
Sem & Sec	8 th Sem A		USN:	4AL16CS058	
Online Test Summary					
Subject BIG DATA ANALYTICS					
Max. Marks	30		Score	22	
Certification Course Summary					
Course	Introduction to Full Stack Development				
Certificate Provider		Great learning academy	Duration		0hrs
Coding Challenges					
Problem Statement: a prg add some letters to it and need to find out shortest palindrome Write a simple code to identify given linked list is palindrome or not by using stack.					
Status: Solved					
Uploaded the report in GitHub			yes		
If yes Repository name			nayan1998		
Uploaded the report in slack			yes		



Program1(ShortestPalindrome)

```
package pk;
import java.util.Scanner;
public class ShortestPalindromeDemo {
      public static String shortestPalindrome(String str) {
             int y=str.length()-1;
             while(y \ge 0){
             if(str.charAt(x)==str.charAt(y)){
             x++;
             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 <u>in</u> = 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));
             }
}
Program2(Write a simple code to identify given linked list is palindrome or not by using
stack.)
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)
{
      int top = s.pop();
      // compare the popped element with current node's data
      // return false if mismatch happens
      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.");
}
```