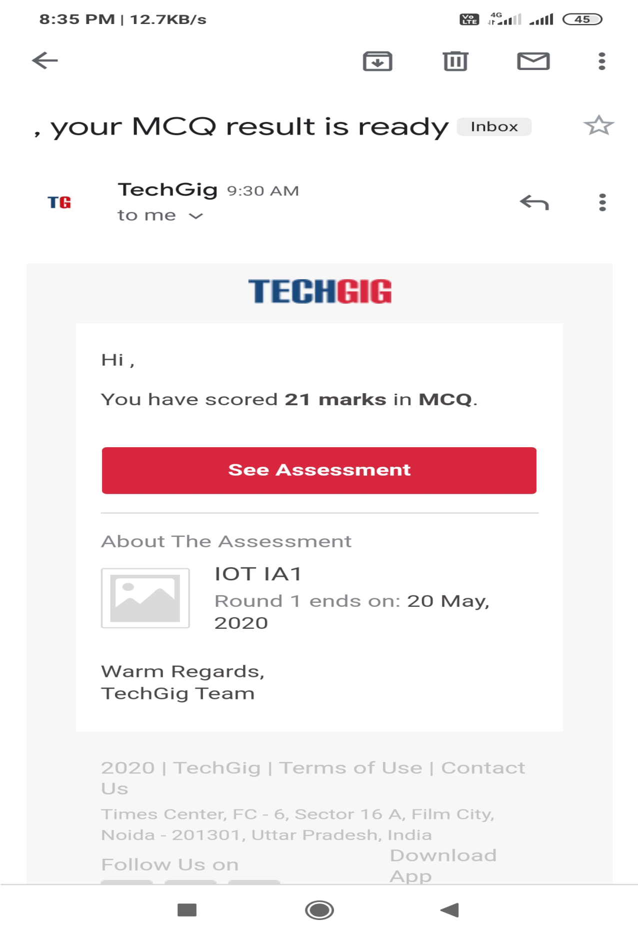
DAILY ONLINE ACTIVITIES SUMMARY

Date:	20/05/20	Name:	Anagha Iyengar S
Sem & Sec	VIII A	USN:	4AL16CS011
Online Test Summary			
Subject	Internet of Things		
Max. Marks	30	Score	21
Certification Course Summary			
Course	Introduction to Ethical Hacking		
Certificate Provider	Great Learner Acdemy	Duration	6 Hours
Coding Challenges			
Problem Statement: 1. Java program to count the occurance of letters in a given string 2. Python program to check whether the given number is an Amstrong number			
Status: Solved			
Uploaded the report in Github	Yes		
If yes Repository name	anaghaiyengar/online_certificate		
Uploaded the report in slack	Yes		



8:35 PM | 12.7KB/s

, your MCQ result is ready

TechGig 9:30 AM
to me

TECHGIG

Hi ,

You have scored **21 marks** in **MCQ**.

About The Assessment

IOT IA1
Round 1 ends on: 20 May, 2020

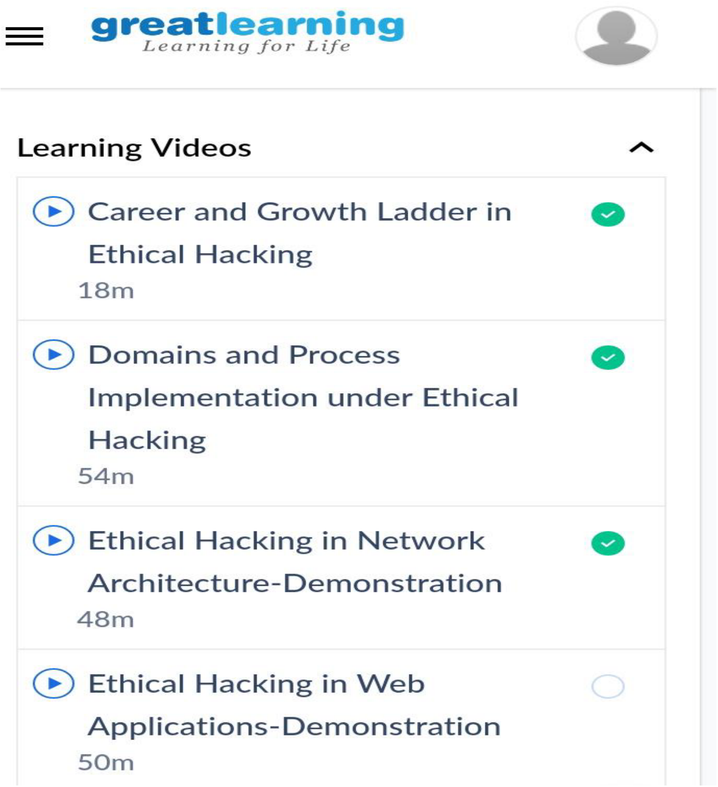
Warm Regards,
TechGig Team

2020 | TechGig | Terms of Use | Contact Us

Times Center, FC - 6, Sector 16 A, Film City, Noida - 201301, Uttar Pradesh, India

Follow Us on

Download App



Learning Videos

- Career and Growth Ladder in Ethical Hacking ✓
  18m
- Domains and Process Implementation under Ethical Hacking ✓
  54m
- Ethical Hacking in Network Architecture-Demonstration ✓
  48m
- Ethical Hacking in Web Applications-Demonstration ○
  50m

Program

```
import java.util.Stack;

// Data Structure to store a linked list node
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<Integer> 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.");
            }
        }
    }
```