DAILY ONLINE ACTIVITIES SUMMARY

Date:	**04-06-2020**	**Name:**	**Deepika K V**
Sem & Sec	**8th sem 'A' sec**	**USN:**	**4AL16CS030**

Online Test Summary

Subject	**SMS**		
Max. Marks	**60**	**Score**	**60**

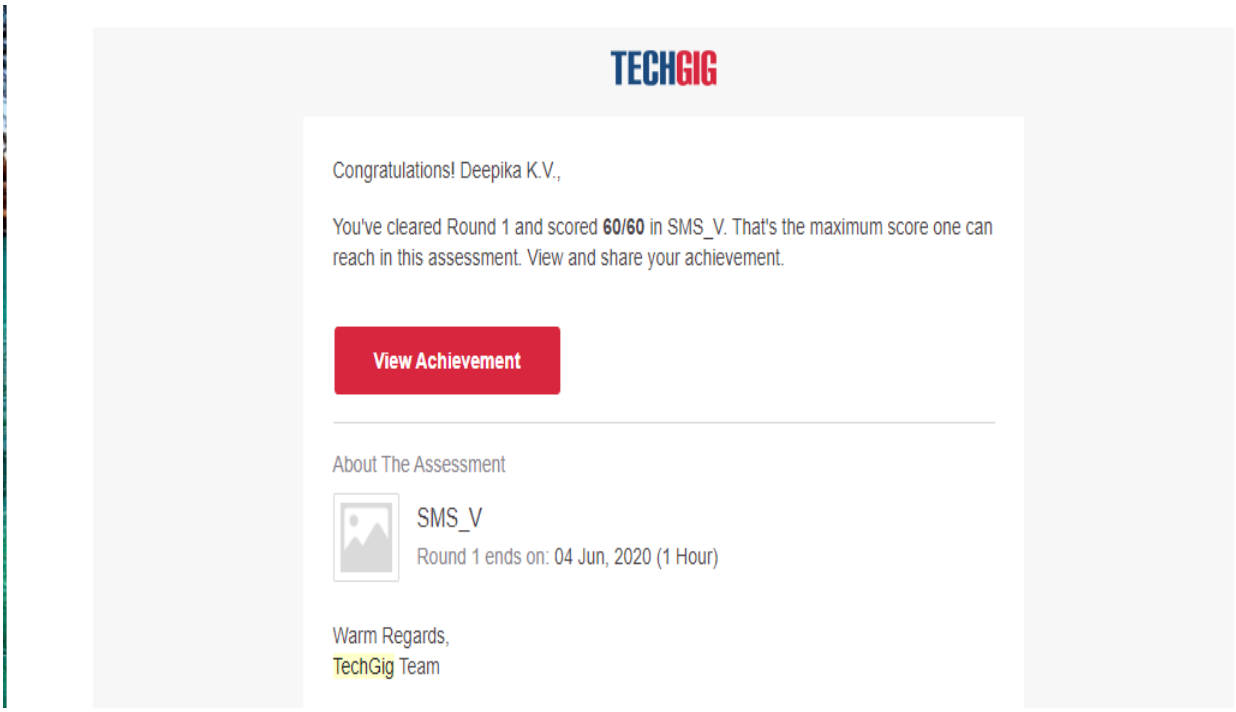
Certification Course Summary

Course	**Computer Networking**		
Certificate Provider	**Alison**	**Duration**	**2.5 hrs**

Coding Challenges

Problem Statement: Write a C program to check if the expression has balanced parenthesis.	
Status: SUBMITTED	
Uploaded the report in Github	**YES**
If yes Repository name	**Codes**
Uploaded the report in slack	**YES**

Online test details:



TECHGIG

Congratulations! Deepika K.V.,

You've cleared Round 1 and scored **60/60** in SMS_V. That's the maximum score one can reach in this assessment. View and share your achievement.

View Achievement

About The Assessment

SMS_V

Round 1 ends on: 04 Jun, 2020 (1 Hour)

Warm Regards,
TechGig Team

Certification Course Details:

Alison

Learner Achievement Verification

This is to certify that the management of Alison has decided to award Deepika K V living in India the certificate of completion in Computer Networking - Digital Network Security - Revised.

Learner Details

Name: Deepika K V

E-mail: deepikakv225@gmail.com

Country: India

Course and Result

Score
84%

Study Time
1:21:38

Computer Networking - Digital Network Security - Revised

The course begins by defining network infrastructure and network security. It then continues by describing the features and functions of the VPN protocol and the Point-to-Point Tunneling Protocol (PPTP). It will teach you how PPTP captures Point-To-Point (PPP) frames into IP datagrams for transmission over an IP-based network. You will also learn about L2TP and how it relies on IPSec in Transport Mode for encryption services.

Modules Studied

Module 1: Defining Network Infrastructure and Network Security

Module 2: Course assessment

Coding Challenge:

```
#include<stdio.h>
                    #include <stdlib.h>
                    #include <string.h>


                    int top = -1;
                    char stack[100];


                    void push(char);
                    void pop();
                    void find_top();


                    void main()
                    {
                           int i;
                           char a[100];
                           printf("enter expression\n");
                           scanf("%s", &a);
                           for (i = 0; a[i] != '\0';i++)
                           {
                                  if (a[i] == '(')
                                  {
                                         push(a[i]);
                                  }
                                  else if (a[i] == ')')
                                  {
                                         pop();
                                  }
                           }
                           find_top();
                    }


                    void push(char a)
                    {
                           stack[top] = a;
                           top++;
                    }
```

```
void pop()
{
	if (top == -1)
	{
		printf("expression is invalid\n");
		exit(0);
	}
	else
	{
		top--;
	}
}


void find_top()
{
	if (top == -1)
		printf("\nexpression is valid\n");
	else
		printf("\nexpression is invalid\n");
}
```