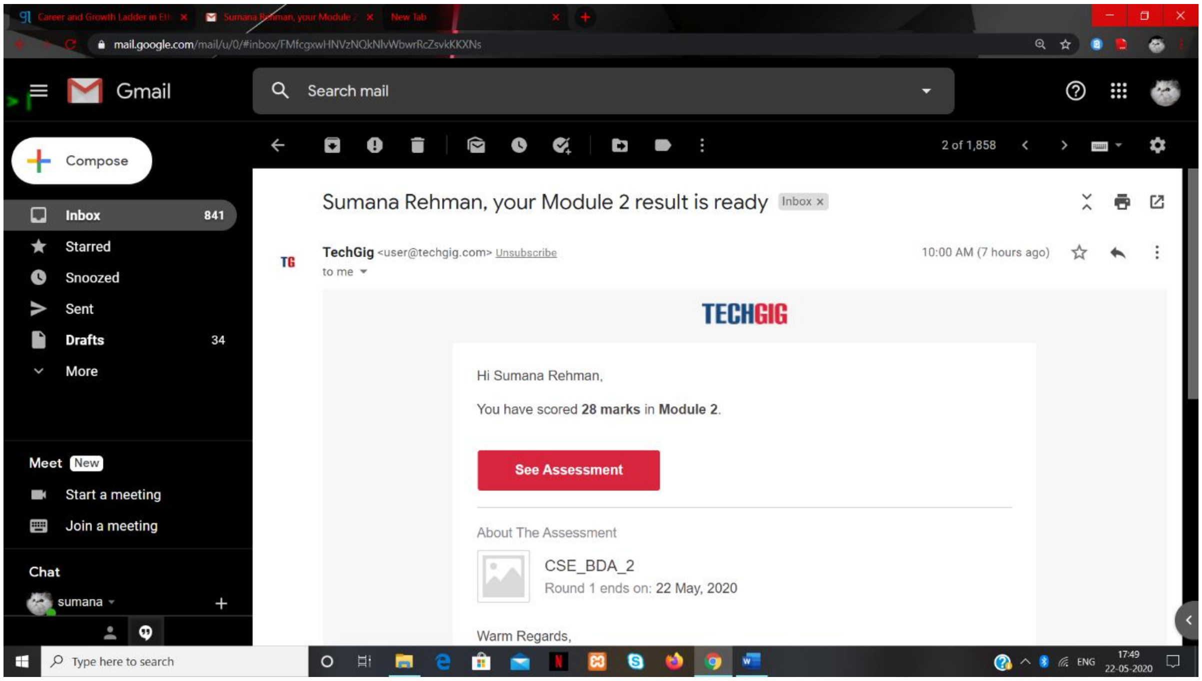
DAILY ONLINE ACTIVITIES SUMMARY

Date:	22/5/2020	Name:	Sumana
Sem & Sec	8th Sem B	USN:	4AL16CS107
Online Test Summary			
Subject	Big Data Analytics		
Max. Marks	40	Score	28
Certification Course Summary			
Course	Introduction to Ethical Hacking		
Certificate Provider	greatlearning.in	Duration	6 hrs
Coding Challenges			
Problem Statement: Write a C Program to implement various operations of Singly Linked List Stack.			
Status: Completed			
Uploaded the report in Github	Yes		
If yes Repository name	Alvas-education-foundation/Sumana		
Uploaded the report in slack	yes		

Online Test Details:



Sumana Rehman, your Module 2 result is ready Inbox ×

TechGig <user@techgig.com> Unsubscribe
to me
10:00 AM (7 hours ago)

TECHGIG

Hi Sumana Rehman,

You have scored **28 marks** in **Module 2**.

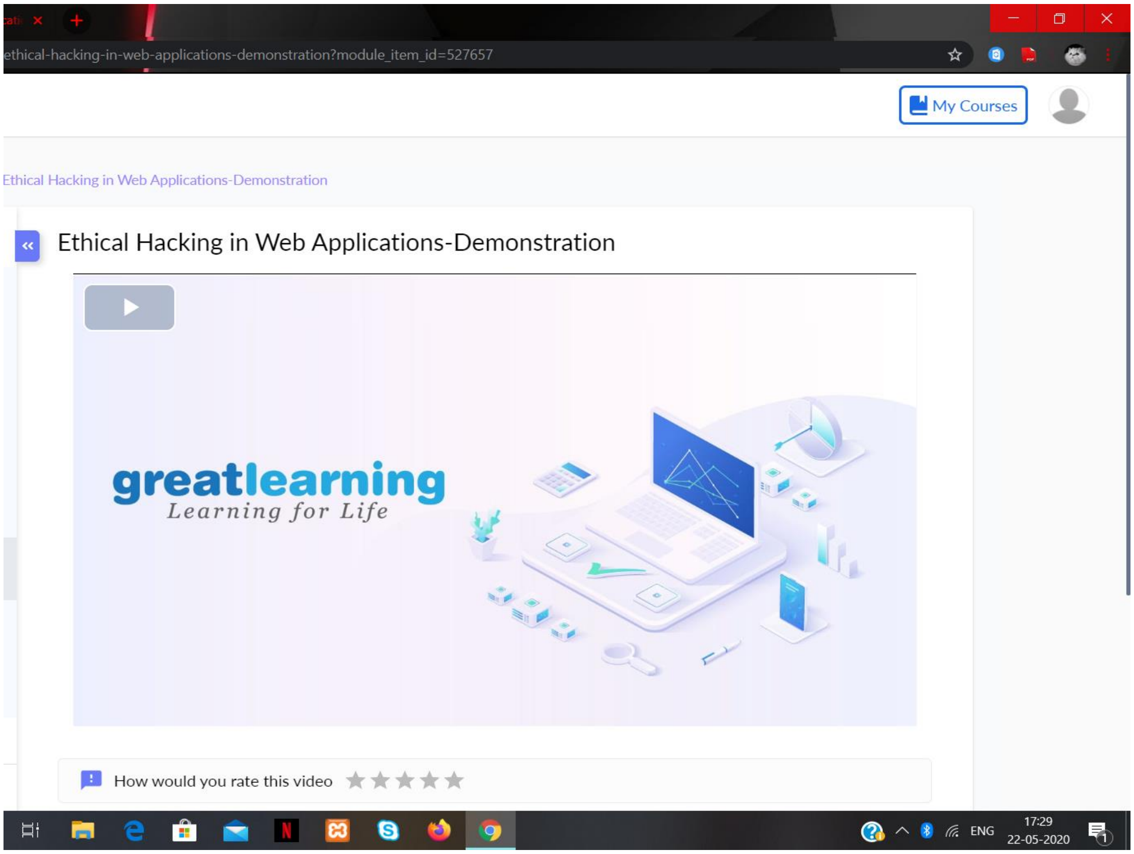
See Assessment

About The Assessment

CSE_BDA_2
Round 1 ends on: 22 May, 2020

Warm Regards,

Certification Course Details:


ethical-hacking-in-web-applications-demonstration?module_item_id=527657
My Courses
greatlearning
Learning for Life
ENG
17:29
22-05-2020

Ethical Hacking in Web Applications-Demonstration

Ethical Hacking in Web Applications-Demonstration

How would you rate this video ★★★★★

Agenda

- Why are Web Applications a target?
- bWAPP & OWASP
- Kali Linux and other suites
- Demonstration
- Ethical Hacking in Mobile Environment

Coding Challenges Details:

C program to implement various operations of SLL Stack :

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
typedef struct stack {
  int data;
  struct stack *next;
} node;
void Push(int, node **);
void Display(node **);
int Pop(node **);
int Sempty(node *);
void main() {
   node *top;
   int data, item, choice;
   char ans, ch;
  top = NULL;
   printf("\nStack Using Linked List : nn");
   do {
      printf("\n\n The main menu");
      printf("\n1.Push \n2.Pop \n3.Display \n4.Exit");
      printf("\n Enter Your Choice");
      scanf("%d", &choice);
      switch (choice) {
      case 1:
         printf("\nEnter the data");
         scanf("%d", &data);
         Push(data, &top);
         break;
      case 2:
         if (Sempty(top))
            printf("\nStack underflow!");
         else {
            item = Pop(&top);
            printf("\nThe popped node is%d", item);
         }
         break;
      case 3:
         Display(&top);
         break;
      case 4:
         printf("\nDo You want To Quit?(y/n)");
         ch = getch();
         if (ch == 'y')
            exit(0);
         else
            break;
      }
printf("\nDo you want to continue?");
      ans = getch();
      getch();
   } while (ans == 'Y' || ans == 'y');
   getch();
```

```
}
void Push(int Item, node **top) {
   node *New;
   node * get_node(int);
   New = get_node(Item);
   New->next = *top;
   *top = New;
}
node * get_node(int item) {
   node * temp;
   temp = (node *) malloc(sizeof(node));
   if (temp == NULL)
      printf("\nMemory Cannot be allocated");
   temp->data = item;
   temp->next = NULL;
   return (temp);
}
int Sempty(node *temp) {
   if (temp == NULL)
      return 1;
   else
```

```
      return 0;
}
int Pop(node **top) {
   int item;
   node *temp;
   item = (*top)->data;
   temp = *top;
   *top = (*top)->next;
   free(temp);
   return (item);
}
void Display(node **head) {
   node *temp;
   temp = *head;
   if (Sempty(temp))
      printf("\nThe stack is empty!");
   else {
      while (temp != NULL) {
         printf("%d\n", temp->data);
         temp = temp->next;
      }
   }  getch();

}
```