DAILY ONLINE ACTIVITIES SUMMARY

Date:	**21/05/2020**	**Name:**	**Vignesha M. Shetty**
Sem & Sec	**8th,B**	**USN:**	**4AAL16CS124**

Online Test Summary

Subject	**System Modeling and Simulation**		
Max. Marks	**60**	**Score**	**45**

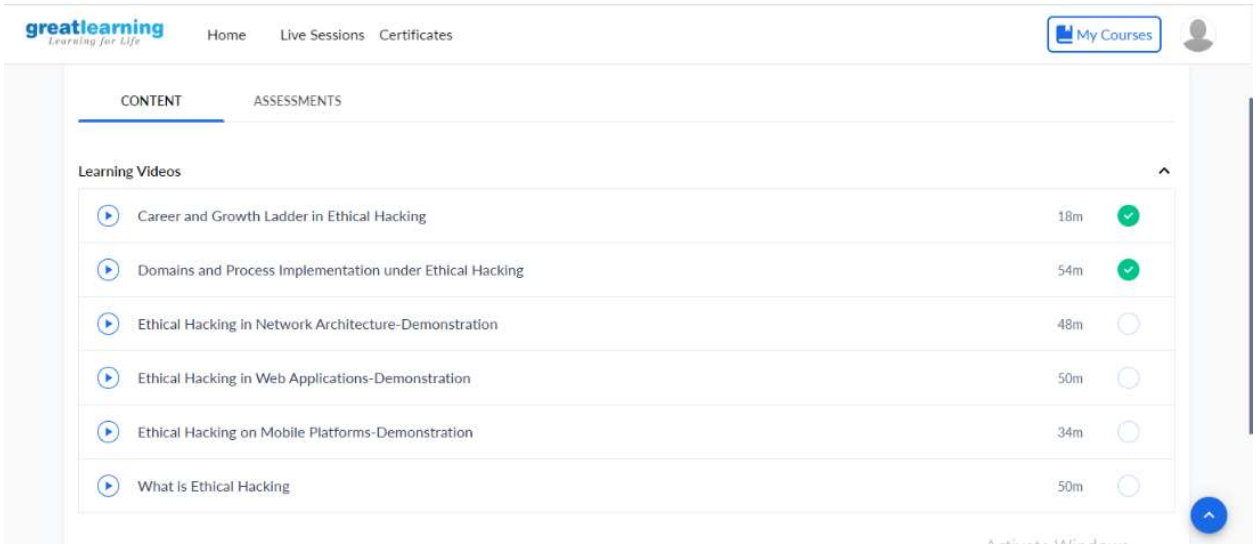
Certification Course Summary

Course	**Introduction to Ethical Hacking**		
Certificate Provider	**Great Learning**	**Duration**	**6 hours**

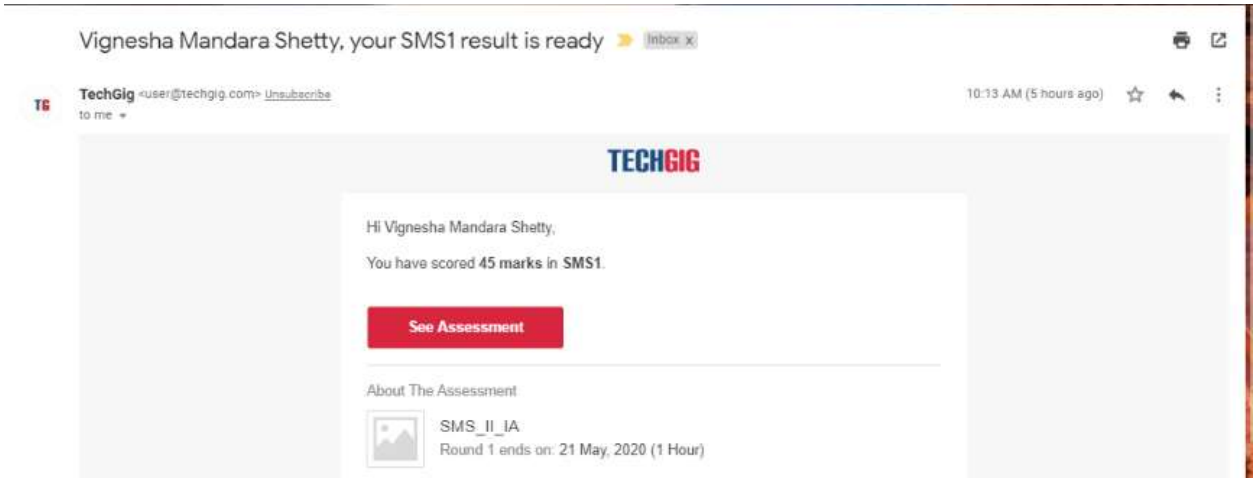
Coding Challenges

Problem Statement: Create singly linked list and reverse it until the head becomes NULL and each time head should be moved to next.	
Status: Solved	
Uploaded the report in Github	**yes**
If yes Repository name	**vigneshshetty/vignesh124** **vigneshshetty/Online_Certifications** **vigneshshetty/online_coding** **vigneshshetty/Daily_progress_report**
Uploaded the report in slack	**yes**

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



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



IA2:Random Variate Generators

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

52 lines (51 sloc) | 932 Bytes Raw Blame History

```
Hint: Crete the SLL, and then Reverse the Link in SLL until Head becomes NULL. Each time Reversing the Link, Head must be moved to next nod

#include <stdio.h>
#include <stdlib.h>
struct node
{
int data;
struct node next;
};
struct Node reverse(struct Node head,int k)
{
struct Node current= head;
struct Node next= NULL;
struct Node prev= NULL;
int count = 0;
while(current!=NULL && count<k)
{
next= current->next;
current->next = prev;
prev= current;
current= next;
count++;
}
if ( next!=NULL)
head->next= reverse( next,k);
return prev;
}
void push( struct Node **head_ref,int new_data)
{
struct Node* new_node= (struct Node) malloc(sizeof(struct Node));
}
}
int main()
{
Struct node *prev,*head,*p;
int n,i;
printf ("number of elements:");
scanf("%d",&n);
head=NULL;
for(i=0;i<n;i++)
{
p=malloc(sizeof(struct node));
scanf("%d",&p->data);
p->next=NULL;
if(head==NULL)
head=p;
```