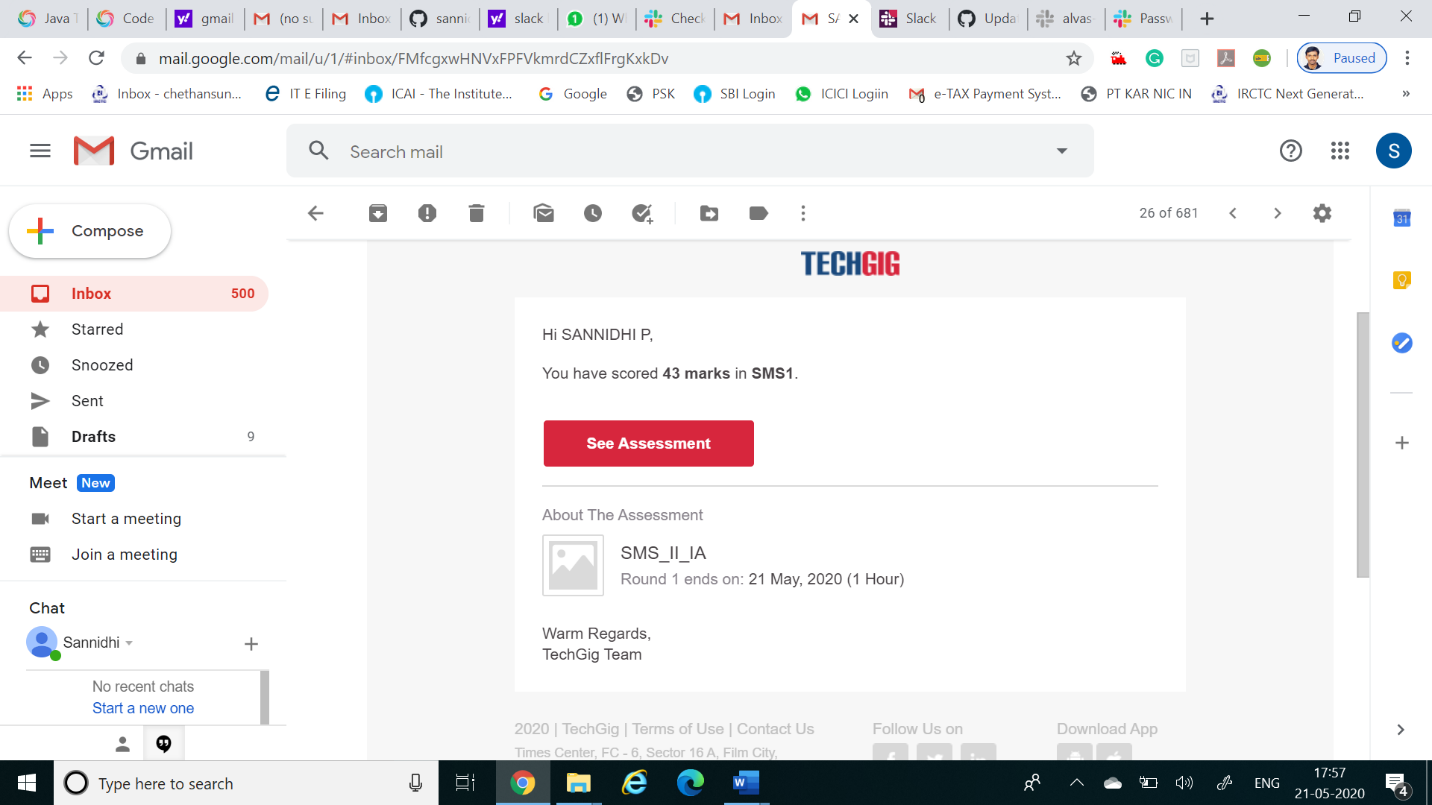
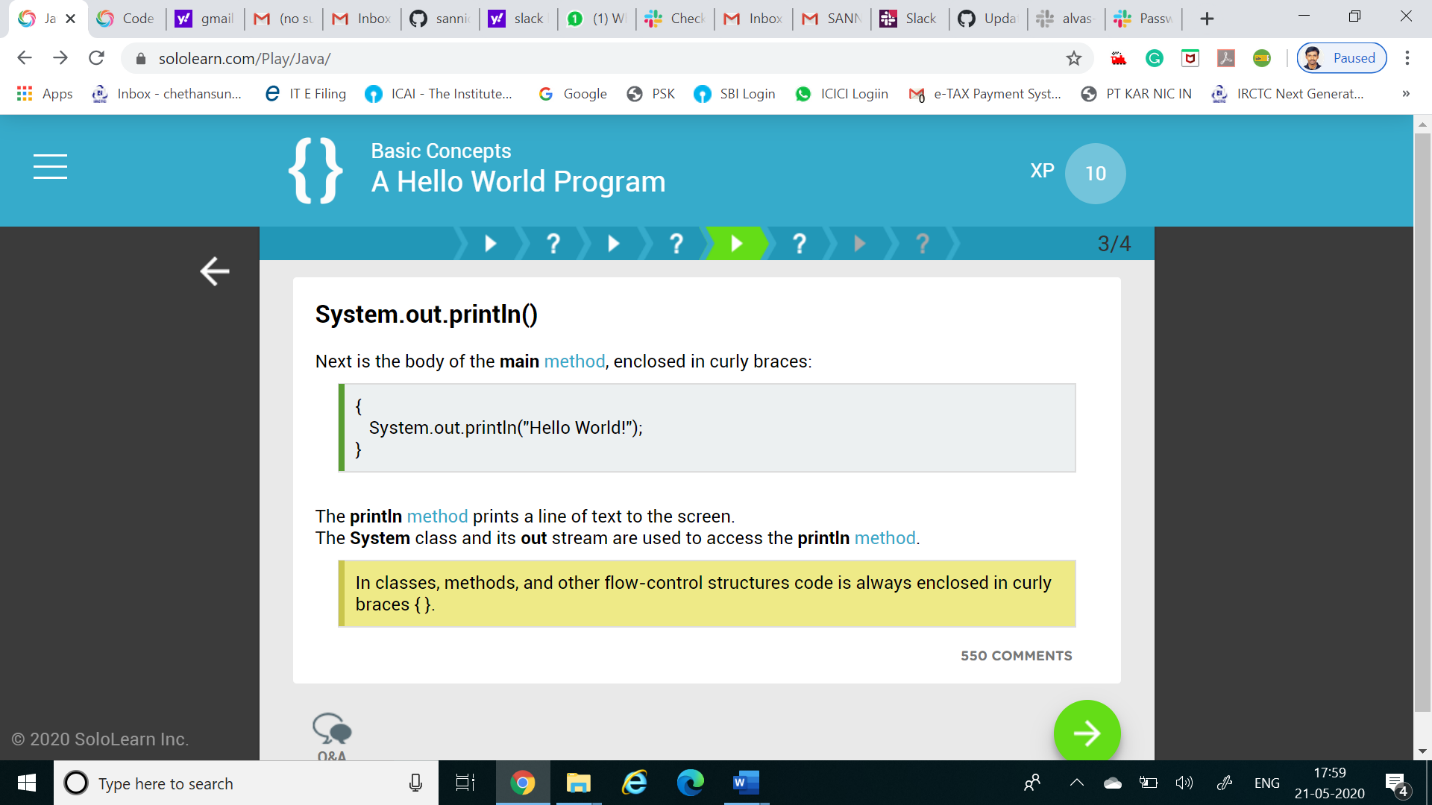
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **21-05-2020** | | | | | **Name:** | **Sannidhi P** | |
| **Sem & Sec** | **8th sem B sec** | | | | | **USN:** | **4AL16CS084** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **SMS** | | | | | | |
| **Max. Marks** | | **60** | | **Score** | | | **43** | |
| **Certification Course Summary(Internship)** | | | | | | | | |
| **Course** | **JAVASCRIPT TUTORIAL** | | | | | | | |
| **Certificate Provider** | | | **SOLOLEARN** | | **Duration** | | | **30min** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:** C Program to Reverse a Linked List in groups of given size. | | | | | | | | |
| **Status: completed** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | **Sannidhi1** | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

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

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



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

Coding was given n it was uploaded for github and slack

PROGRAM1

#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; ko

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;

else

prev->next=p;

prev=p;

}

return 0;

}