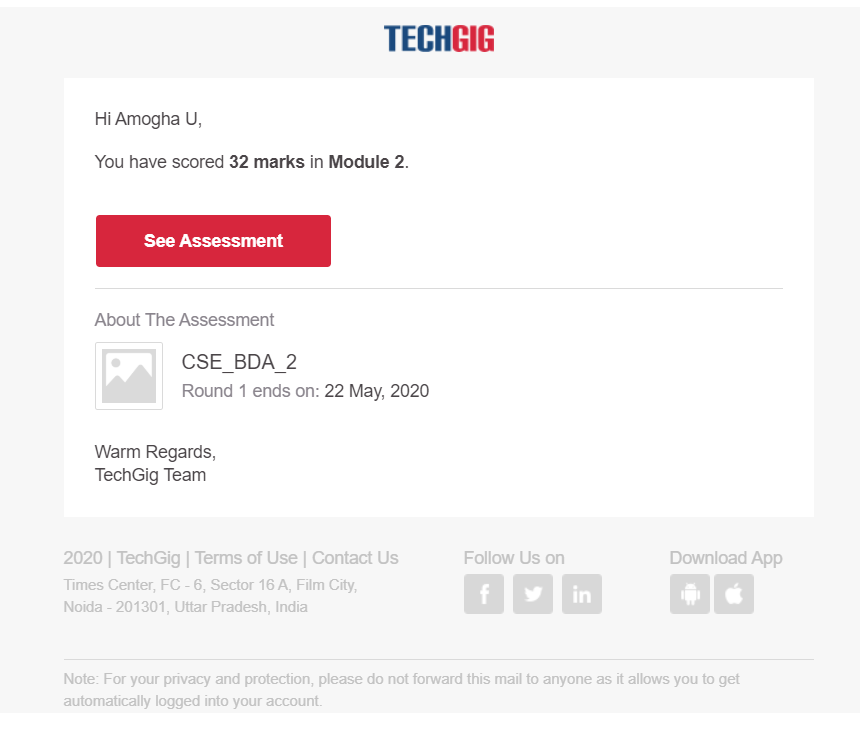
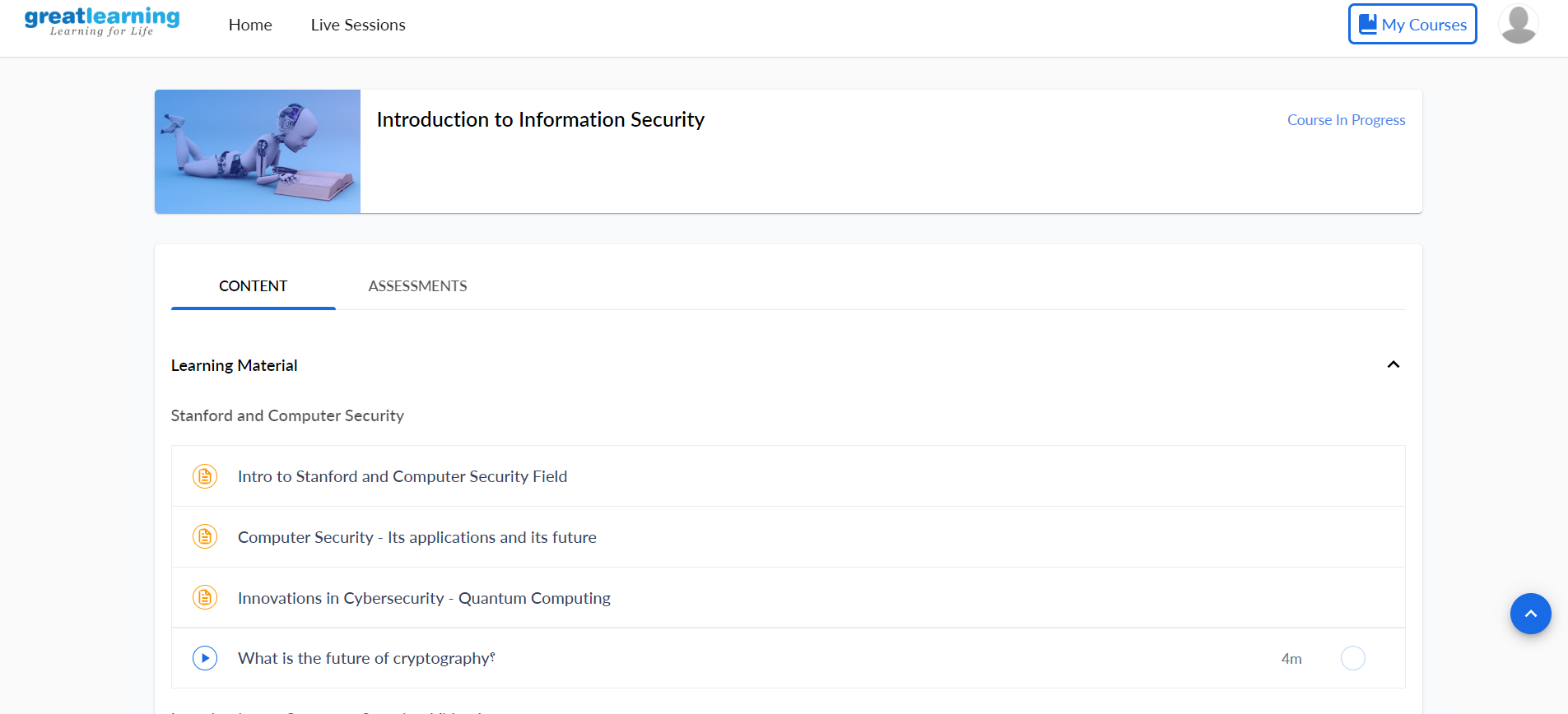
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **22/5/2020** | | | | | **Name:** | **Amogha U** | |
| **Sem & Sec** | **8th Sem** | | | | | **USN:** | **4AL16CS010** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **Big data Analytics** | | | | | | |
| **Max. Marks** | | **40** | | **Score** | | | **32** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Introduction to Information Security** | | | | | | | |
| **Certificate Provider** | | | **greatlearning** | | **Duration** | | | **5.5 hrs** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:**  1: 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** | | | | | amogha-u | | | |
| **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)

#include <stdio.h>

#include <stdlib.h>

struct node

{

int info;

struct node \*ptr;

}\*top,\*top1,\*temp;

int topelement();

void push(int data);

void pop();

void empty();

void display();

void destroy();

void stack\_count();

void create();

int count = 0;

void main()

{

int no, ch, e;

printf("\n 1 - Push");

printf("\n 2 - Pop");

printf("\n 3 - Top");

printf("\n 4 - Empty");

printf("\n 5 - Exit");

printf("\n 6 - Dipslay");

printf("\n 7 - Stack Count");

printf("\n 8 - Destroy stack");

create();

while (1)

{

printf("\n Enter choice : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

printf("Enter data : ");

scanf("%d", &no);

push(no);

break;

case 2:

pop();

break;

case 3:

if (top == NULL)

printf("No elements in stack");

else

{

e = topelement();

printf("\n Top element : %d", e);

}

break;

case 4:

empty();

break;

case 5:

exit(0);

case 6:

display();

break;

case 7:

stack\_count();

break;

case 8:

destroy();

break;

default :

printf(" Wrong choice, Please enter correct choice ");

break;

}

}