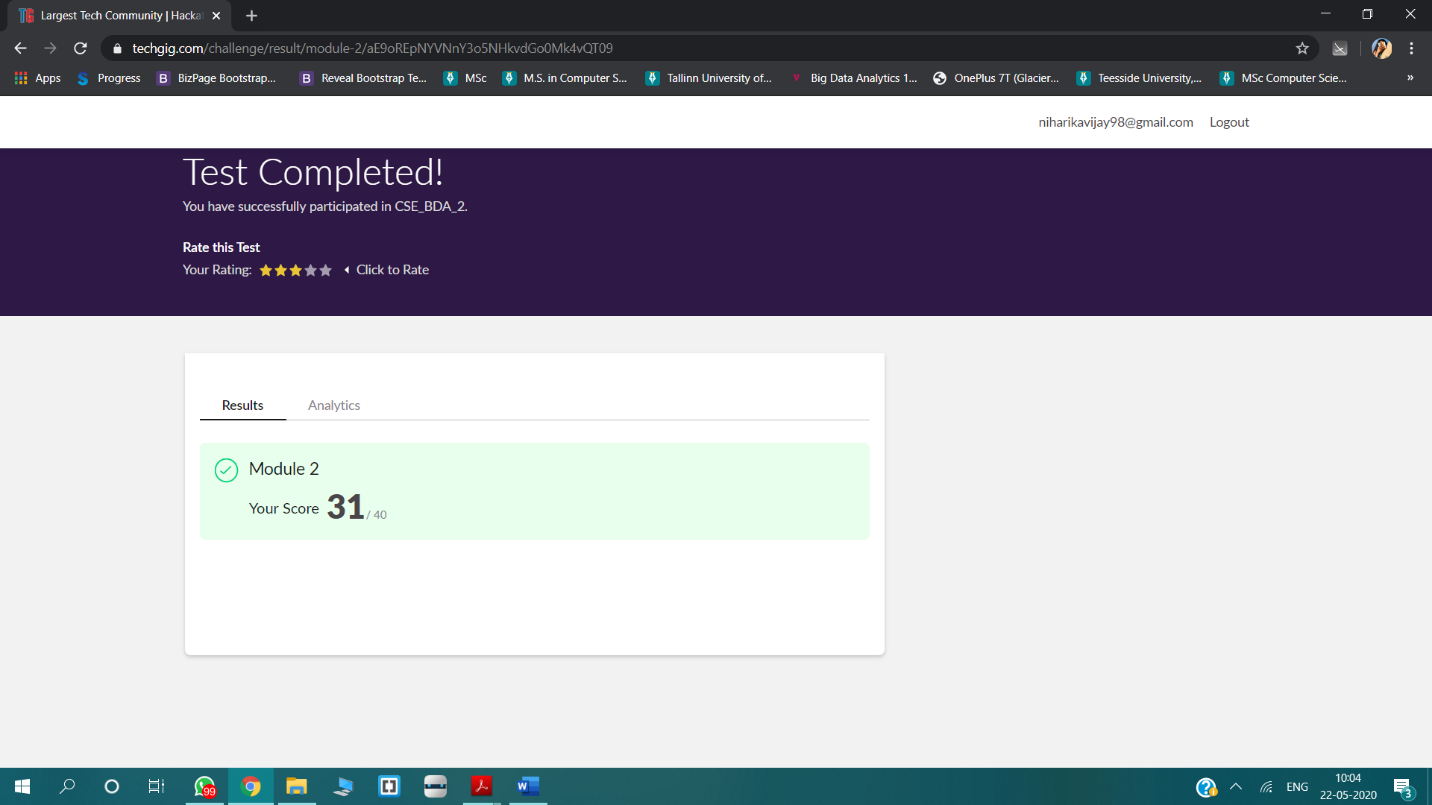
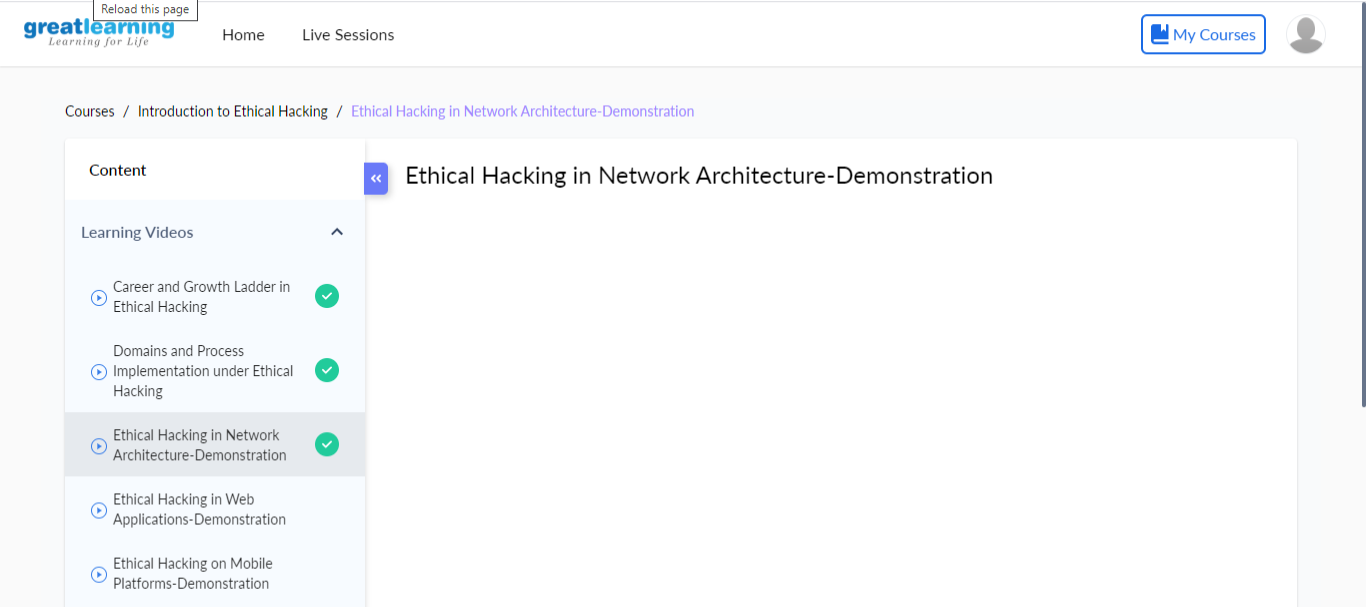
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **20-05-2020** | | | | | **Name:** | **Niharika G V** | |
| **Sem & Sec** | **8 A** | | | | | **USN:** | **4AL16CS059** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **IOT** | | | | | | |
| **Max. Marks** | | **40** | | **Score** | | | **31** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Introduction to Ethical Hacking** | | | | | | | |
| **Certificate Provider** | | | **Great Learning** | | **Certificate Provider** | | | **Bright Learning** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement: C Program to implement various operations of Singly Linked List Stack** | | | | | | | | |
| **Status: Solved** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **Yes** | | | |
| **If yes Repository name** | | | | | **Daily Progress Reports** | | | |
| **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)

**Program 1**Top of Form

Bottom of Form

**C Program to implement various operations of Singly Linked List Stack**

#include <stdio.h>

#include <stdlib.h>

struct node

{

int info;

struct node \*ptr;

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

void push(int data);

void pop();

void display();

void create();

int count = 0;

void main()

{

int no, ch, e;

printf("\n 1 - Push");

printf("\n 2 - Pop");

printf("\n 3 - Display");

printf("\n 4 - Destroy");

printf("\n 5 - Exit");

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:

display();

break;

case 4:

destroy();

break;

case 5:

exit(0);

default :

printf("Invalid Input");

break;

}

}

}

void create()

{

top = NULL;

}

void push(int data)

{

if (top == NULL)

{

top =(struct node \*)malloc(1\*sizeof(struct node));

top->ptr = NULL;

top->info = data;

}

else

{

temp =(struct node \*)malloc(1\*sizeof(struct node));

temp->ptr = top;

temp->info = data;

top = temp;

}

count++;

}

void display()

{

top1 = top;

if (top1 == NULL)

{

printf("Stack is empty");

return;

}

while (top1 != NULL)

{

printf("%d \n", top1->info);

top1 = top1->ptr;

}

}

void pop()

{

top1 = top;

if (top1 == NULL)

{

printf("\n Error : Not Able to pop from empty stack");

return;

}

else

top1 = top1->ptr;

printf("\n Popped value : %d", top->info);

free(top);

top = top1;

count--;

}

int topelement()

{

return(top->info);

}

void empty()

{

if (top == NULL)

printf("\n Stack is empty");

else

printf("\n Stack is not empty with %d elements", count);

}

void destroy()

{

top1 = top;

while (top1 != NULL)

{

top1 = top->ptr;

free(top);

top = top1;

top1 = top1->ptr;

}

free(top1);

top = NULL;

printf("\n All stack elements destroyed");

count = 0;

}

Top of Form

Bottom of Form