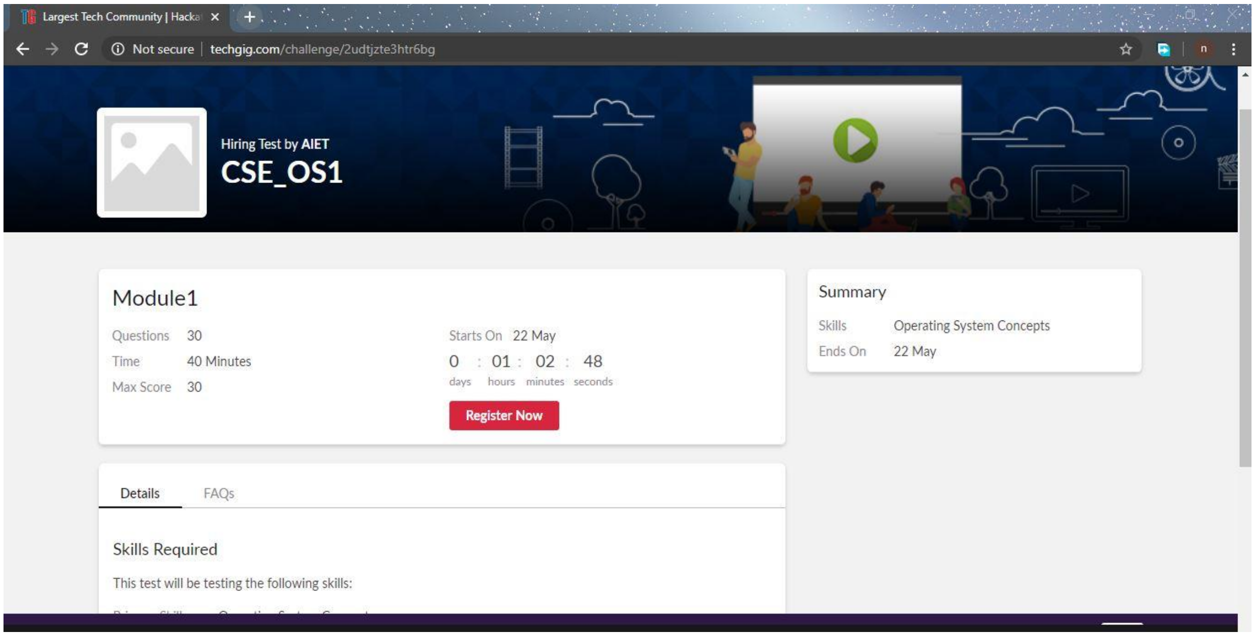
DAILY ONLINE ACTIVITIES SUMMARY

Date:	**22/05/2020**	**Name:**	**NIKHIL KUMAR**
Sem & Sec	**4th SEM & ‘B’ SEC.**	**USN:**	**4AL19CS400**
Online Test Summary			
Subject	**OPERATING SYSTEMS**		
Max. Marks	**30**	**Score**	**19**
Certification Course Summary			
Course	**Python for Machine Learning**		
Certificate Provider	**Greatlearning academy**	**Duration**	**5 Hrs.**
Coding Challenges			
Problem statement : Write a C Program to implement various operations on Singly Linked List Stack.			
Status: Executed			
Uploaded the report in Github	**Yes**		
If yes Repository name	**c-coding**		
Uploaded the report in slack	**Yes**		

Online Test Summary: Operating systems test was scheduled from 09:15Am to 09:55Am .The portion for the IA was 1st module there were 30 questions and the time assigned was 40 minutes the questions were MCQ type and to predict the output of the given program.

The snapshot of the test is uploaded which shows score allotted at the end of the test.


Largest Tech Community | Hacka
Not secure | techgig.com/challenge/2udtjzte3htr6bg

Hiring Test by AIET

CSE_OS1

Module1

Questions	30	Starts On	22 May
Time	40 Minutes	0 days : 01 hours : 02 minutes : 48 seconds	
Max Score	30		

Summary

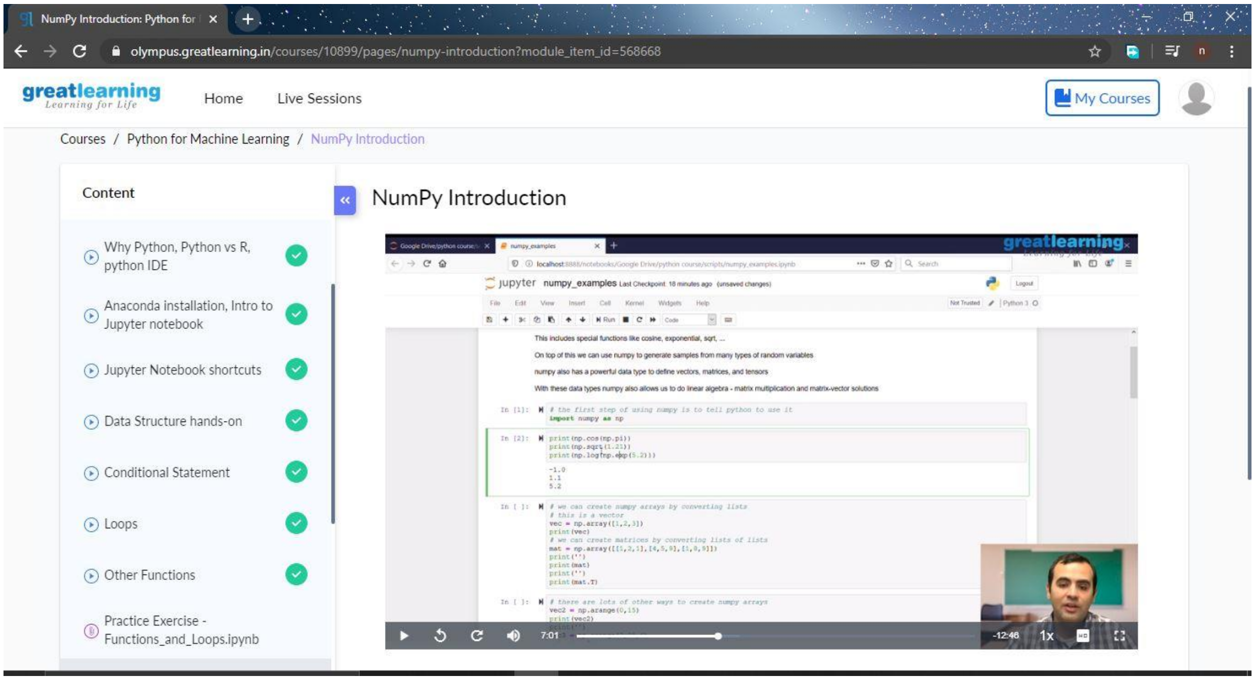
Skills	Operating System Concepts
Ends On	22 May

Skills Required

This test will be testing the following skills:

<u>Online Certification Course Summary</u>: Today I have learn about numpy introduction.

Snapshot of today’s cource is given below :


NumPy Introduction: Python for
olympus.greatlearning.in/courses/10899/pages/numpy-introduction?module_item_id=568668
greatlearning Learning for Life
Home
Live Sessions
My Courses
Courses / Python for Machine Learning / NumPy Introduction
Content
Why Python, Python vs R, python IDE
Anaconda installation, Intro to Jupyter notebook
Jupyter Notebook shortcuts
Data Structure hands-on
Conditional Statement
Loops
Other Functions
Practice Exercise - Functions_and_Loops.ipynb

NumPy Introduction

numpy_examples
localhost:8888/notebooks/Google Drive/python course/scripts/numpy_examples.ipynb
jupyter numpy_examples Last Checkpoint 18 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
7:01
-12:46
1x

This includes special functions like cosine, exponential, sqrt, ...

On top of this we can use numpy to generate samples from many types of random variables

numpy also has a powerful data type to define vectors, matrices, and tensors

With these data types numpy also allows us to do linear algebra - matrix multiplication and matrix-vector solutions

```
In [1]:  # the first step of using numpy is to tell python to use it
         import numpy as np

In [2]:  print(np.cos(np.pi))
         print(np.sqrt(1.21))
         print(np.log(np.exp(5.2)))

         -1.0
         1.1
         5.2

In [ ]:  # we can create numpy arrays by converting lists
         # this is a vector
         vec = np.array([1,2,3])
         print(vec)
         # we can create matrices by converting lists of lists
         mat = np.array([[1,2,1],[4,5,9],[1,8,9]])
         print('')
         print(mat)
         print('')
         print(mat.T)

In [ ]:  # there are lots of other ways to create numpy arrays
         vec2 = np.arange(0,15)
         print(vec2)
```

Online Coding Summary:.

: Everyday we are given with new question of coding related to the language of java and c. it seems interesting how we imbibe ourselves in depth to understand the logic break it and then code for it.

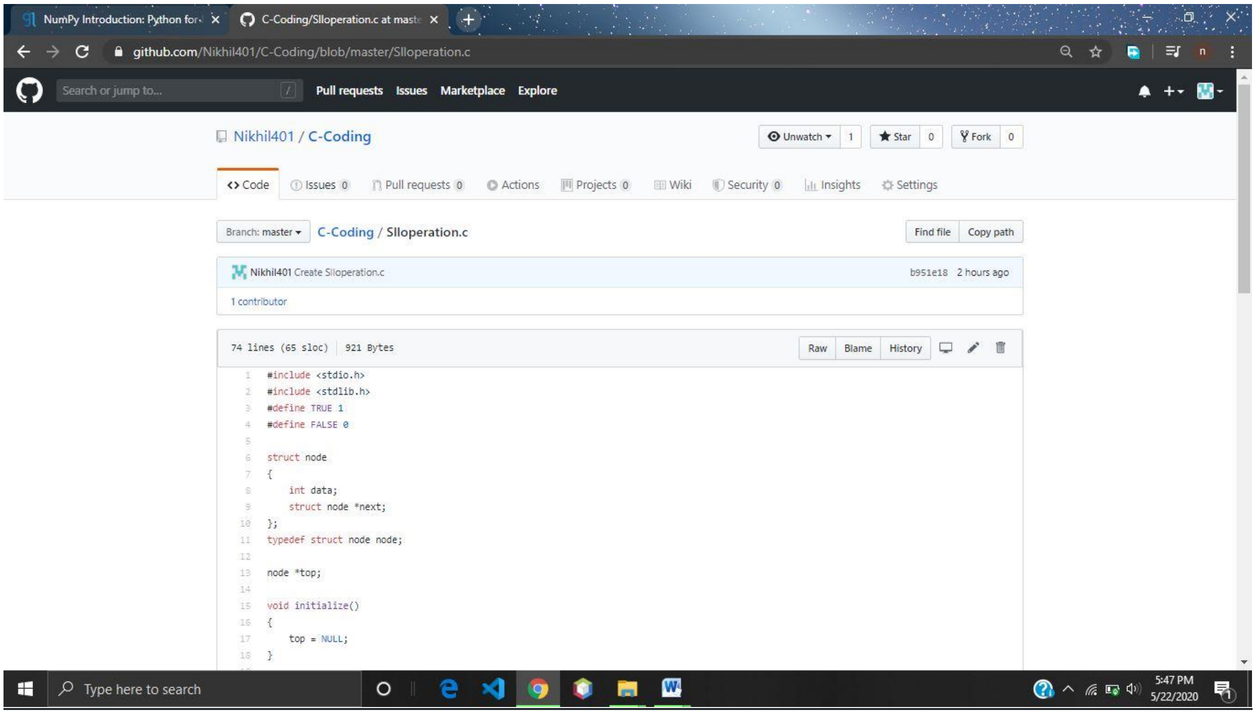
Today's question was :

Write a C Program to implement various operations on Singly Linked List Stack.

Hint: First Create a Singly Linked List Stack with the node corresponding to First Element is the base of the stack; and its link field must be always Null.
When you push First Element, It is the First and it is Base of the stack. Its Link must be Null. top pointer pointing to First. (top = First)
When you push any element, (No need of checking Stack full case because SLL is dynamic) Create a new node called temp using malloc function and insert the a number into Data field, and Link field must be pointing to top; and move the pointer top to point to temp.
When you pop, First check for stack Empty. if First == NULL, then Stack Empty. If it is not empty, The pointer temp must be pointing to top. Move the pointer top to top->link. delete temp.
When you display the stack element, First Check for Stack Empty as in pop operation. If it is not empty, Display all the elements of current stack starting from top to First.

Using the above given hints I have made a c code for the question

Snapshot:



```c
#include <stdio.h>
#include <stdlib.h>
#define TRUE 1
#define FALSE 0

struct node
{
    int data;
    struct node *next;
};
typedef struct node node;

node *top;

void initialize()
{
    top = NULL;
}
```

Link : https://github.com/Nikhil401/C-Coding/blob/master/Slloperation.c