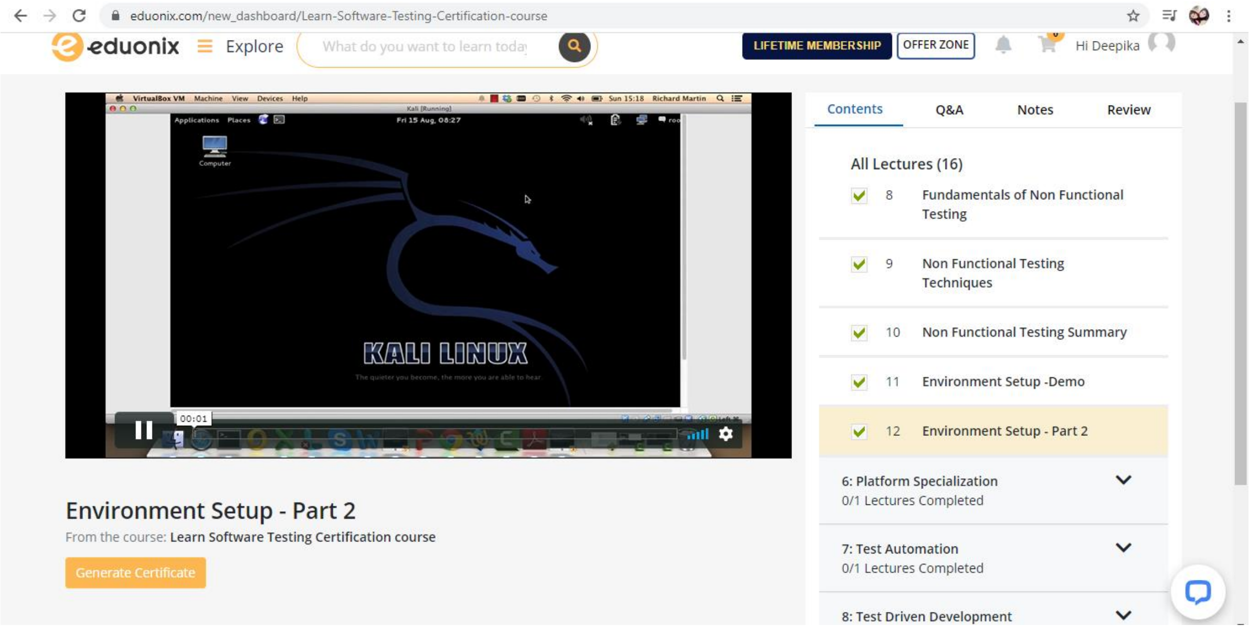
DAILY ONLINE ACTIVITIES SUMMARY

Date:	11-07-2020	**Name:**	**Deepika K V**
Sem & Sec	**8th sem 'A' sec**	**USN:**	**4AL16CS030**
Online Test Summary			
Subject	**-**		
Max. Marks	**-**	**Score**	-
Certification Course Summary			
Course	Learn Software Testing Certification Course.		
Certificate Provider	**eduonix**	**Duration**	**10.5 hrs**
Coding Challenges			
Problem Statement: C program to replace substring in a string.			
Status: SUBMITTED			
Uploaded the report in Github	**YES**		
If yes Repository name	**Codes**		
Uploaded the report in slack	**YES**		

Online test details:

Certification Course Details:


eduonix.com/new_dashboard/Learn-Software-Testing-Certification-course
eduonix
Explore
What do you want to learn today
LIFETIME MEMBERSHIP
OFFER ZONE
Hi Deepika
KALI LINUX
00:01
Contents
Q&A
Notes
Review

All Lectures (16)

- ✓ 8 Fundamentals of Non Functional Testing
- ✓ 9 Non Functional Testing Techniques
- ✓ 10 Non Functional Testing Summary
- ✓ 11 Environment Setup -Demo
- ✓ 12 Environment Setup - Part 2

6: Platform Specialization
0/1 Lectures Completed

7: Test Automation
0/1 Lectures Completed

8: Test Driven Development

Environment Setup - Part 2

From the course: Learn Software Testing Certification course

Generate Certificate

Coding Challenge:

```
#include<stdlib.h>
                     #include<stdio.h>
                     #define NO_OF_CHARS 256


                     int *get_char_count(char *str)
                     {
                             int *count = (int *)calloc(sizeof(int), NO_OF_CHARS);
                             int i;
                             for (i = 0; *(str+i); i++)
                                     count[*(str+i)]++;
                             return count;
                     }


                     int first_non_repeating_character(char *str)
                     {
                             int *count = get_char_count(str);
                             int index = -1, i;
                     for (i = 0; *(str+i); i++)
                             {
                                     if (count[*(str+i)] == 1)
                                             {
                                                     index = i;
                                                     break;
                                             }
                             }


                             free(count);
                             return index;
                     }


                     int main()
                     {
                             char str[NO_OF_CHARS];
                             printf("\nEnter the string : ");
                             scanf("%s",&str);
                             int index = first_non_repeating_character(str);
                             if (index == -1)
                                     printf("All the characters are repetitive");
                             else
                                     printf("First non-repeating character is %c", str[index]);
                             getchar();
                             return 0;
                     }
```