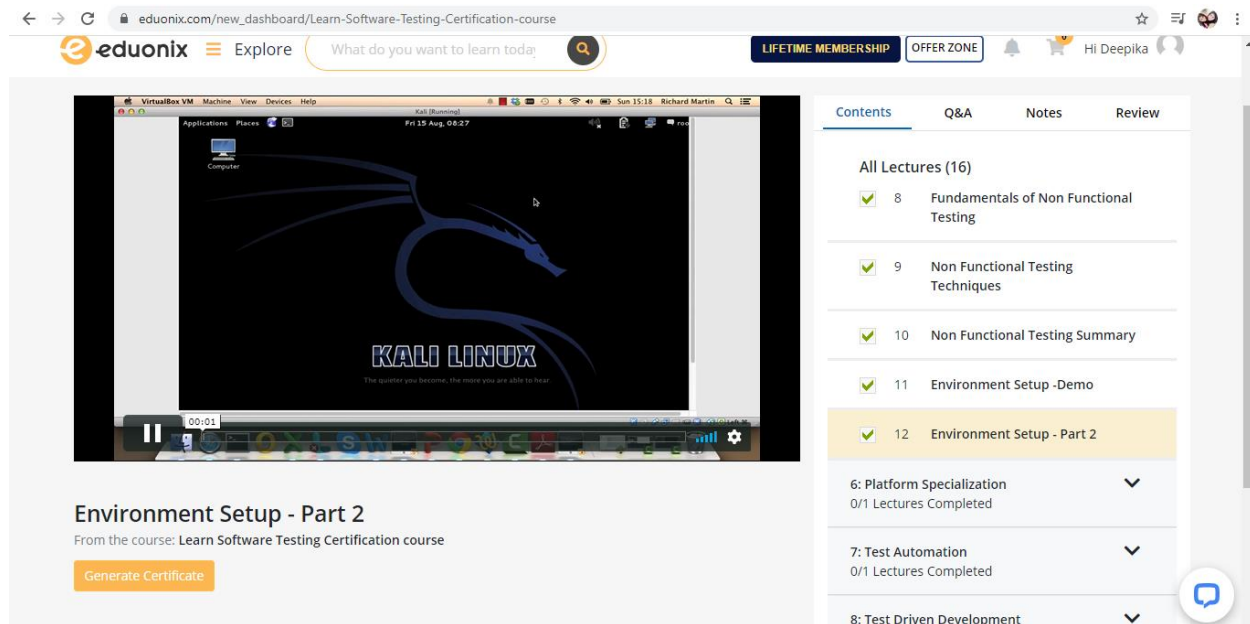


## DAILY ONLINE ACTIVITIES SUMMARY

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

## Online test details:

## Certification Course Details:



The screenshot shows the Eduonix website interface. At the top, there's a navigation bar with the Eduonix logo, an 'Explore' button, a search bar, and links for 'LIFETIME MEMBERSHIP' and 'OFFER ZONE'. The main content area features a video player displaying the Kali Linux desktop environment. Below the video player, the title 'Environment Setup - Part 2' is shown, along with the text 'From the course: Learn Software Testing Certification course' and a 'Generate Certificate' button. To the right of the video player is a sidebar with a 'Contents' tab selected. The sidebar lists 16 lectures, with the first 5 lectures (8-12) expanded. Lecture 12, 'Environment Setup - Part 2', is highlighted. Below the list, there are sections for '6: Platform Specialization' (0/1 Lectures Completed), '7: Test Automation' (0/1 Lectures Completed), and '8: Test Driven Development'.

Environment Setup - Part 2

From the course: Learn Software Testing Certification course

Generate Certificate

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

## 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;
}

```