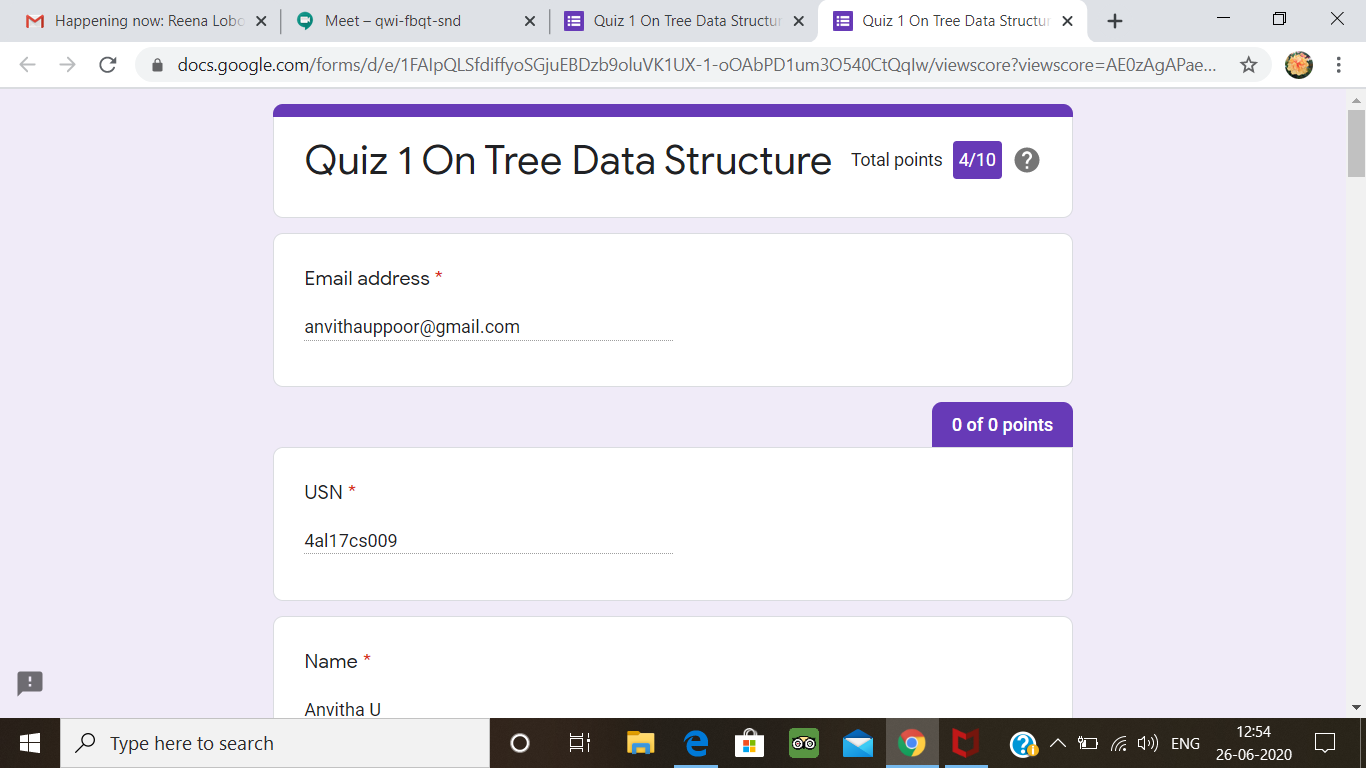
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **26-06-2020** | | | | | **Name:** | | **Anvitha U** | |
| **Sem & Sec** | **6th  Sem ‘A’ Sec** | | | | | **USN:** | | **4AL17CS009** | |
| **Online Test Summary** | | | | | | | | | |
| **Subject** | | **Cyber Security and Data structures in C .** | | | | | | | |
| **Max. Marks** | | **DS Quiz1=10.** | | **Score** | | | | **DS Quiz1=4.** | |
| **Pre-Placement Training Summary** | | | | | | | | | |
| **Course** | **Workshop of Cyber Security and Data structures in C .** | | | | | | | | |
| **Faculty** | | | **Manjunath Sir.**  **Venkatesh Sir.** | | | | **Duration** | | **4 hours** |
| **Coding Challenges** | | | | | | | | | |
| **Problem Statement:** 1. Write a program to print all permutations of a given string. | | | | | | | | | |
| **Status: done** | | | | | | | | | |
| **Uploaded the report in Github** | | | | | **Yes** | | | | |
| **If yes Repository name** | | | | | Daily Report =<https://github.com/anvithauppoor/online_coding_activity> | | | | |
| **Uploaded the report in slack** | | | | | **Yes** | | | | |

**Class and Quiz Snapshots:**

**Cyber Security and Data structures in C :**

****

**Coding Challenge:**

1. Write a program to print all permutations of a given string.

Description:  
Here is the steps to implement string permutations:

Take out the first char and keep it constant. And permute rest of the characters.  
User recursive method call to permute rest of the string except first character.  
While making recursive call, we accumulate each character being constant along with recursive call response.

public class Permutation

{

public static void main(String[] args)

{

String str = "ABC";

int n = str.length();

Permutation permutation = new Permutation();

permutation.permute(str, 0, n-1);

}

private void permute(String str, int l, int r)

{

if (l == r)

System.out.println(str);

else

{

for (int i = l; i <= r; i++)

{

str = swap(str,l,i);

permute(str, l+1, r);

str = swap(str,l,i);

}

}

}

public String swap(String a, int i, int j)

{

char temp;

char[] charArray = a.toCharArray();

temp = charArray[i] ;

charArray[i] = charArray[j];

charArray[j] = temp;

return String.valueOf(charArray);

}

}

**Output:**

