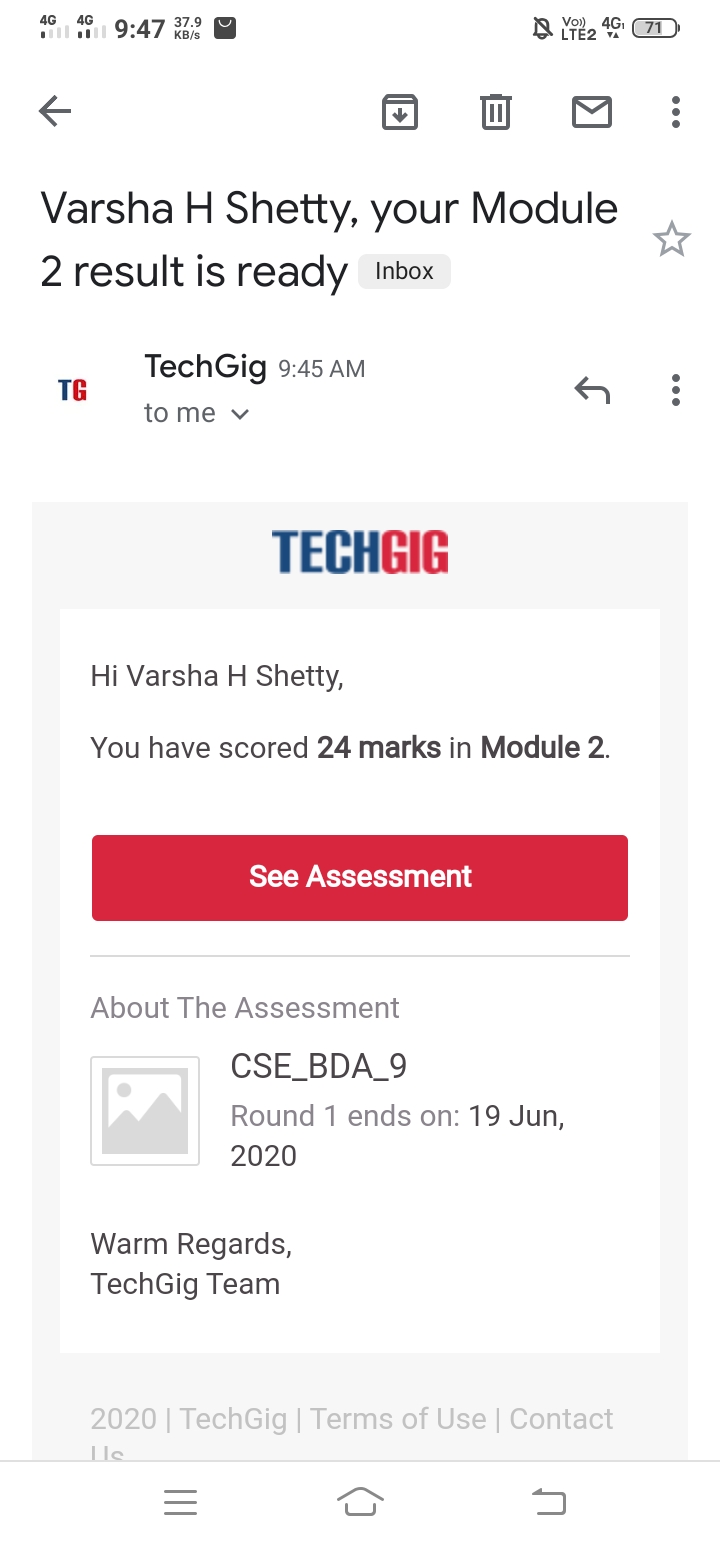
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **19-06-2020** | | | | **Name:** | **Varsha H Shetty** | |
| **Sem & Sec** | **8th sem B sec** | | | | **USN:** | **4AL16CS117** | |
| **Online Test Summary** | | | | | | | |
| **Subject** | | **BDA** | | | | | |
| **Max. Marks** | | **30** | | **Score** | | **24** | |
| **Certification Course Summary** | | | | | | | |
| **Course** | **Authentication and authorisation with aws** | | | | | | |
| **Certificate Provider** | | | **Awd** | **Duration** | | | **15 min** |
| **Coding Challenges** | | | | | | | |
| **Problem Statement-** :  Write a Java program to find the row, column position of a specified number (row, column position) in a given 2-dimensional array | | | | | | | |
| **Status: completed** | | | | | | | |
| **Uploaded the report in Github** | | | | **yes** | | | |
| **If yes Repository name** | | | | **Varsha-Shetty** | | | |
| **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)

Coding was given and it was uploaded for github and slack

import java.util.\*;

public class abc {

  public static void main(String[] args) {

   int nums[][] = {{12, 20, 30, 40},

                  {15, 25, 35, 45},

                  {24, 29, 39, 51},

                  {35, 30, 39, 50},

                  {50, 60, 75, 72}};

int rows = 5;

int search\_element = 39;

     int ans[] = Saddleback(nums, rows - 1, 0, search\_element);

        System.out.println("Position of "+search\_element+" in the matrix is ("+ans[0] + "," + ans[1]+")");

    }

    private static int[] Saddleback(int nums[][], int row, int col, int search\_element) {

        //numsay to store the row and column of the searched element

        int element\_pos[] = {-1, -1};

        if (row < 0 || col >= nums[row].length) {

            return element\_pos;

        }

        if (nums[row][col] == search\_element) {

            element\_pos[0] = row;

            element\_pos[1] = col;

            return element\_pos;

        }

        else if (nums[row][col] > search\_element) {

            return Saddleback(nums, row - 1, col, search\_element);

        }

        return Saddleback(nums, row, col + 1, search\_element);

    }

}

