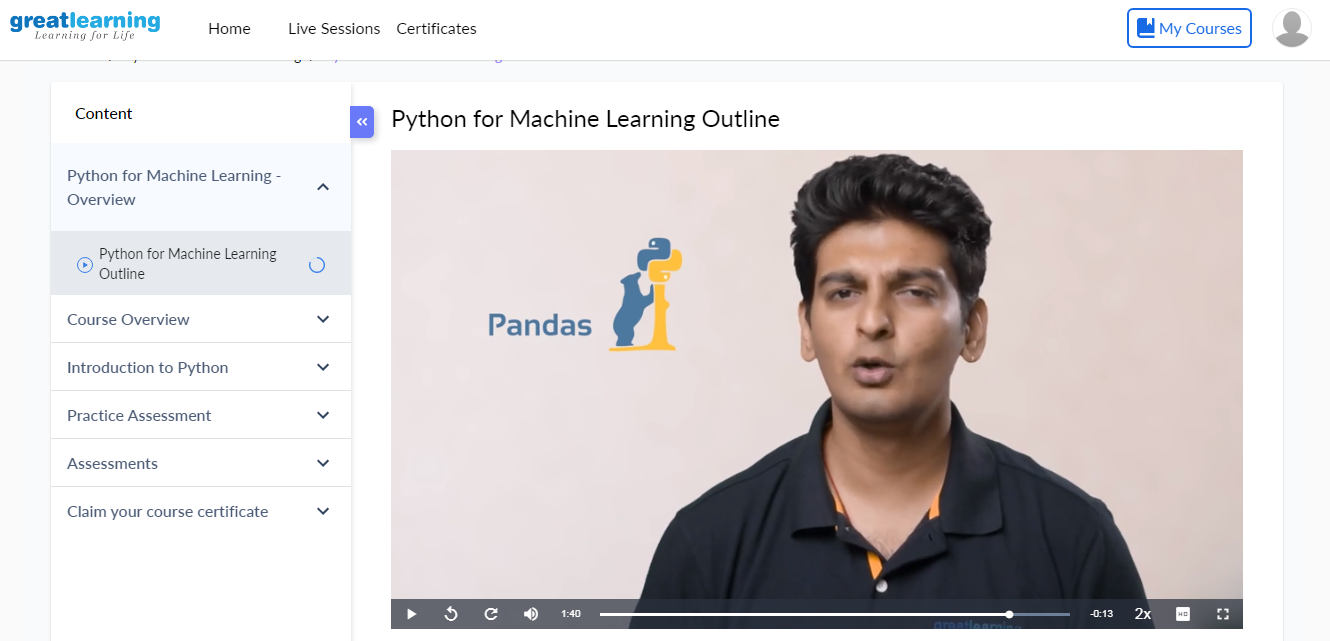
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **19/06/20** | | | | | **Name:** | **SARANG VK** | |
| **Sem & Sec** | **8th B** | | | | | **USN:** | **4AL16CS085** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **BDA** | | | | | | |
| **Max. Marks** | | **30** | | **Score** | | | **10** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Python for Machine Learning** | | | | | | | |
| **Certificate Provider** | | | **GREATLEARNING** | | **Duration** | | | **28 MINUTES** |
| **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** | | | | | **alvas-education-foundation/sarang\_vk** | | | |
| **Uploaded the report in slack** | | | | | **YES** | | | |

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 n it was uploaded for github and slack

PROGRAM1

importjava.util.\*;

publicclassabc{

  publicstaticvoidmain(String[]args){

   intnums[][]={{12,20,30,40},

                  {15,25,35,45},

                  {24,29,39,51},

                  {35,30,39,50},

                  {50,60,75,72}};

int rows =5;

intsearch\_element=39;

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

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

    }

    privatestaticint[]Saddleback(intnums[][],int row,int col,intsearch\_element){

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

        intelement\_pos[]={-1,-1};

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

            returnelement\_pos;

        }

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

            element\_pos[0]= row;

            element\_pos[1]= col;

            returnelement\_pos;

        }

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

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

        }

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

    }

}

