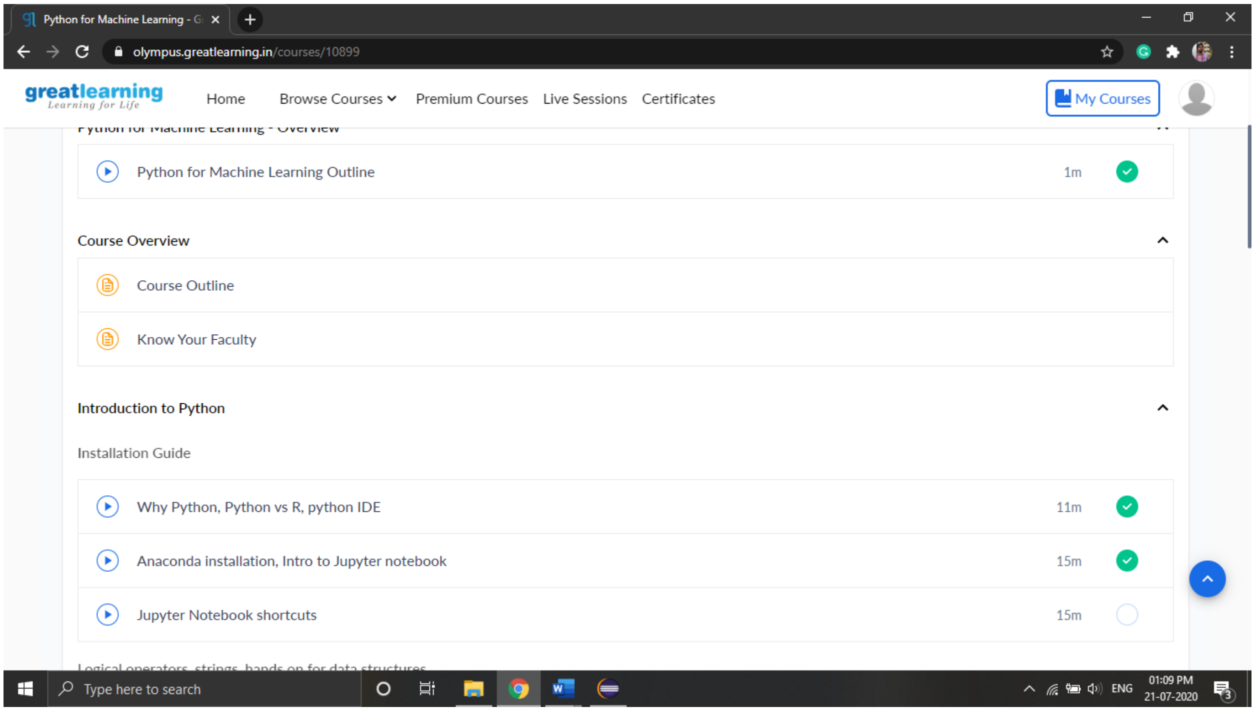
DAILY ONLINE ACTIVITIES SUMMARY

Date:	**20-07-2020**	**Name:**	**Nayan. P. Joshi**
Sem & Sec	**8th Sem A**	**USN:**	**4AL16CS058**
Online Test Summary			
Subject	----------		
Max. Marks	-----------	**Score**	---------------
Certification Course Summary			
Course	Python for Machine Learning		
Certificate Provider	**Great learning academy**	**Duration**	**2hrs**
Coding Challenges			
Problem Statement: Java Program to perform quick sort			
Status: Solved			
Uploaded the report in GitHub	**Yes**		
If yes Repository name	**nayan1998**		
Uploaded the report in slack	**Yes**		



Python for Machine Learning - Overview

Python for Machine Learning Outline 1m ✓

Course Overview

Course Outline

Know Your Faculty

Introduction to Python

Installation Guide

Why Python, Python vs R, python IDE 11m ✓

Anaconda installation, Intro to Jupyter notebook 15m ✓

Jupyter Notebook shortcuts 15m ○

Logical operators, strings, hands on for data structures

QUICK SORT JAVA PROGRAM

```
package pk;

public class Quick_Sort {
     public static void main(String[] args) {
        int i;
        int[] arr={90,23,101,45,65,23,67,89,34,23};
        quickSort(arr, 0, 9);
        System.out.println("\n The sorted array is: \n");
        for(i=0;i<10;i++)
        System.out.println(arr[i]);
    }
    public static int partition(int a[], int beg, int end)
    {

        int left, right, temp, loc, flag;
        loc = left = beg;
        right = end;
        flag = 0;
        while(flag != 1)
        {
            while((a[loc] <= a[right]) && (loc!=right))
            right--;
            if(loc==right)
            flag =1;
            else if(a[loc]>a[right])
            {
                temp = a[loc];
                a[loc] = a[right];
                a[right] = temp;
                loc = right;
            }
            if(flag!=1)
            {
                while((a[loc] >= a[left]) && (loc!=left))
                left++;
                if(loc==left)
                flag =1;
                else if(a[loc] <a[left])
                {
                    temp = a[loc];
                    a[loc] = a[left];
                    a[left] = temp;
                    loc = left;
                }
            }
        }
        return loc;
    }
    static void quickSort(int a[], int beg, int end)
    {

        int loc;
        if(beg<end)
        {
            loc = partition(a, beg, end);
            quickSort(a, beg, loc-1);
            quickSort(a, loc+1, end);
        }
    }
}
```