

DAILY ONLINE ACTIVITIES SUMMARY

Date:	02/06/2020	Name:	Priya Nagari
Sem & Sec	Fourth SEM section B	USN:	4AL18CS063
Online Test Summary			
Subject	_____		
Max. Marks	_____	Score	_____
Certification Course Summary			
Course	The complete Android app development Masterclass: Build apps		
Certificate Provider	Udemy	Duration	29 hours
Coding Challenges 1. Write a Java program to find Perfect Sum Problem. 2. Given an array of positive integers. Write a C Program to find inversion count of array.			
Status:			
Uploaded the report in Github		YES	
If yes Repository name		Priya_Nagari link: https://github.com/alvas-education-foundation/Priya_Nagari	
Uploaded the report in slack		YES	

Online Test Details: 3rd test

NO TEST .

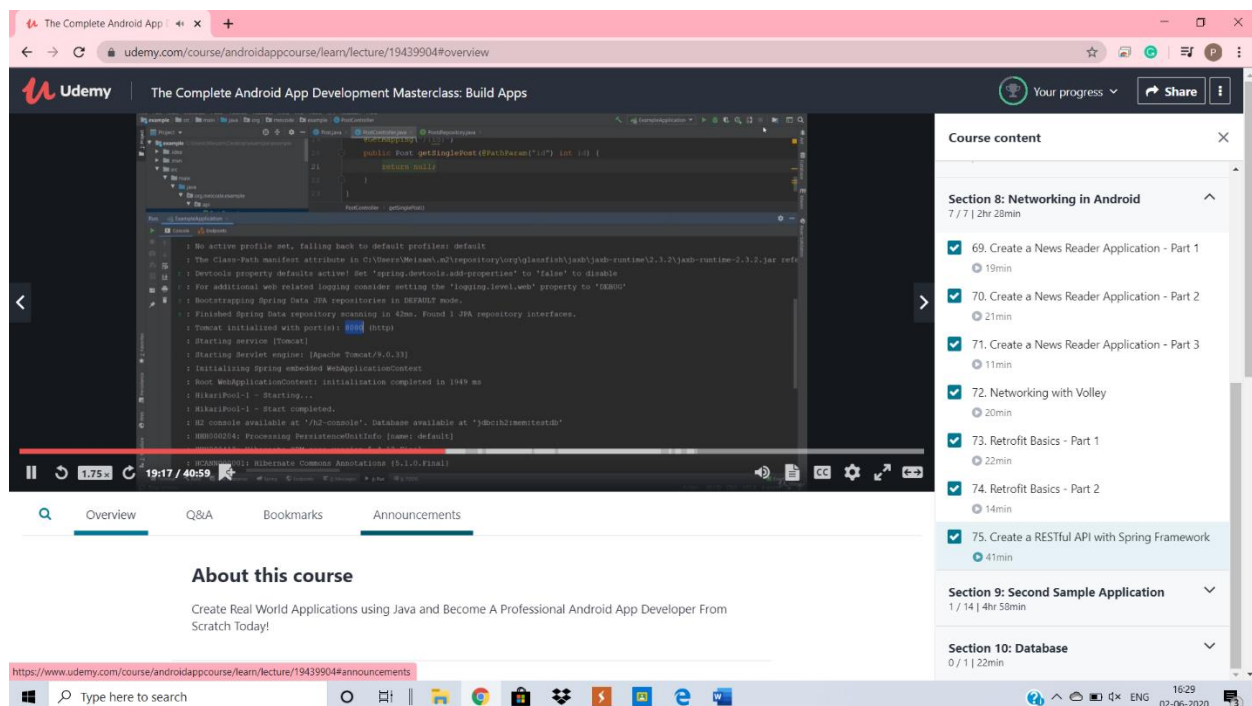
Certification Course Details:

Name of the course: The complete Android app development Masterclass: Build apps

Certificate Provider: Udemy

total duration is 29 hours.

Today I learnt some basic concepts of Retrofit and started with spring FrameWork.



Online Coding Details:

Problem Statement 1: Write a Java program to find Perfect Sum Problem.

The screenshot shows a web browser displaying a GitHub repository page for 'PerfectSum.java'. The browser's address bar shows the URL 'github.com/alvas-education-foundation/Priya_Nagari/blob/master/coding_solutions/JAVA_PROGRAMS/PerfectSum.java'. The page header includes a 'Sign up' button and a navigation bar with the repository path 'Branch: master Priya_Nagari / coding_solutions / JAVA_PROGRAMS / PerfectSum.java'. Below the navigation bar, the commit history shows a single commit by 'priya6426' on '02/06/2020'. The file details indicate it is 62 lines (52 sloc) and 1.71 KB. The code is displayed in a light blue editor with syntax highlighting. The code is a Java program to find the Perfect Sum Problem. The code is as follows:

```
1 //Write a Java program to find Perfect Sum Problem
2 import java.util.*;
3
4 class PerfectSum {
5
6     // Function to print the subsets whose
7     // sum is equal to the given target K
8     public static void sumSubsets(
9         int set[], int n, int target)
10    {
11        // Create the new array with size
12        // equal to array set[] to create
13        // binary array as per n(decimal number)
14        int x[] = new int[set.length];
15        int j = set.length - 1;
16
17        // Convert the array into binary array
18        while (n > 0) {
19            x[j] = n % 2;
20            n = n / 2;
21            j--;
22        }
23    }
```

Problem Statement 2: Given an array of positive integers. Write a C Program to find inversion count of array.

The screenshot shows a web browser displaying a GitHub repository page for 'InvCount.c'. The browser's address bar shows the URL 'github.com/alvas-education-foundation/Priya_Nagari/blob/master/coding_solutions/C_PROGRAMS/InvCount.c'. The page header includes a 'Sign up' button and a navigation bar with the repository path 'Branch: master Priya_Nagari / coding_solutions / C_PROGRAMS / InvCount.c'. Below the navigation bar, the commit history shows a single commit by 'priya6426' on '02/06/2020'. The file details indicate it is 21 lines (19 sloc) and 600 Bytes. The code is displayed in a light blue editor with syntax highlighting. The code is a C program to find the inversion count of an array. The code is as follows:

```
1 //Given an array of positive integers. Write a C Program to find inversion count of array.
2 #include <bits/stdc++.h>
3 int getInvCount(int arr[], int n)
4 {
5     int inv_count = 0;
6     for (int i = 0; i < n - 1; i++)
7         for (int j = i + 1; j < n; j++)
8             if (arr[i] > arr[j])
9                 inv_count++;
10
11     return inv_count;
12 }
13
14 /* Driver progra to test above functions */
15 int main(int argc, char** args)
16 {
17     int arr[] = { 1, 20, 6, 4, 5 };
18     int n = sizeof(arr) / sizeof(arr[0]);
19     printf("Number of inversions are %d\n", getInvCount(arr, n));
20     return 0;
21 }
```