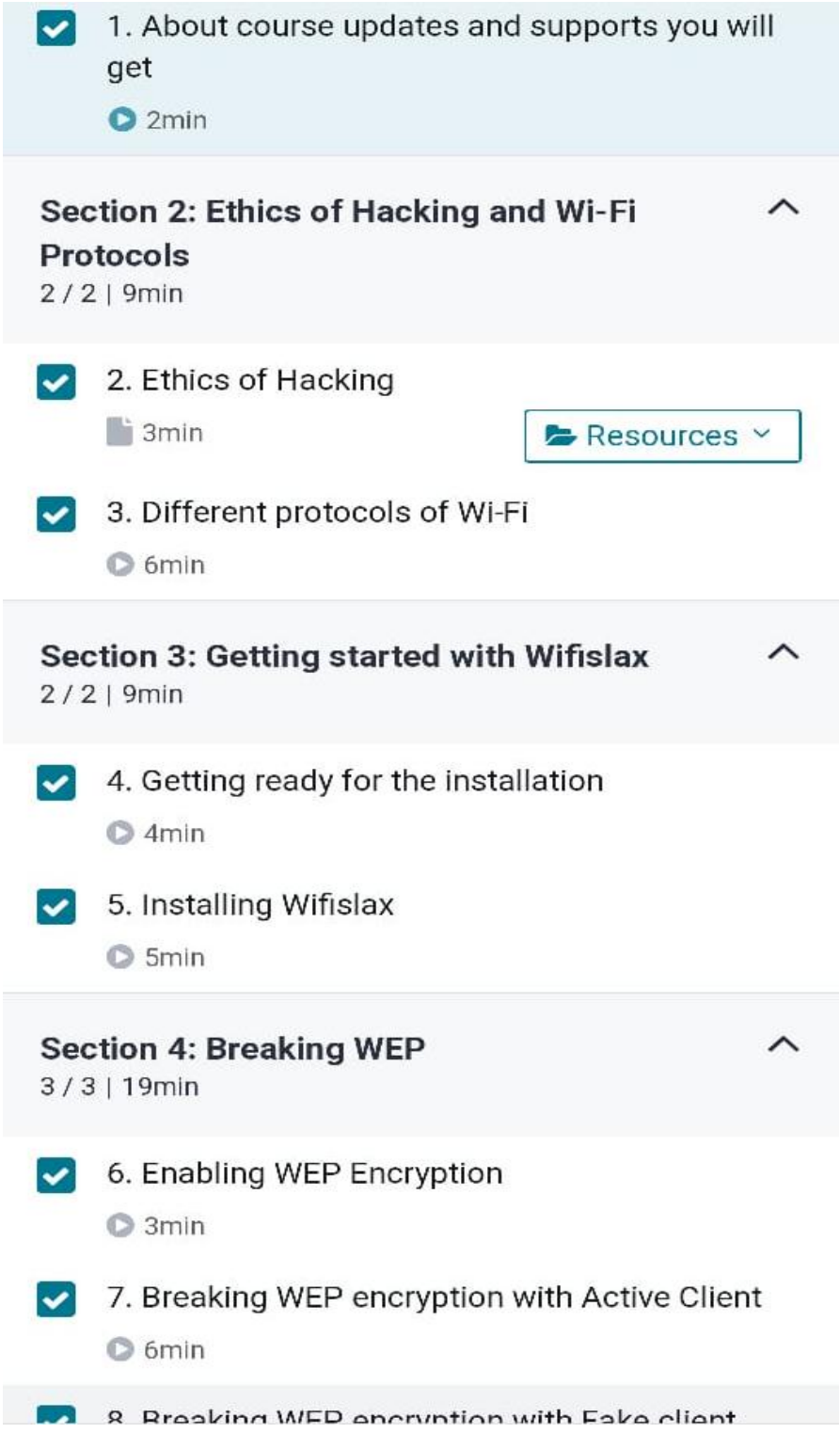
DAILY ONLINE ACTIVITIES SUMMARY

Date:	**25/06/2020**	**Name:**	**Deepa**
Sem & Sec	**8th Sem**	**USN:**	**4AL16CS029**
Online Test Summary			
Subject	**System Model-ling and Simulation**		
Max. Marks	**30**	**Score**	**- -**
Certification Course Summary			
Course	**The complete wi-fi hacking course: beginner to advance 2019**		
Certificate Provider	**udemy.com/**	**Duration**	**3 hrs**
Coding Challenges			
Problem Statement: 1) Write a C program to find if an array is a subset of another array			
Status: Completed			
Uploaded the report in Github	**Yes**		
If yes Repository name	**Daily_report**		
Uploaded the report in slack	**yes**		

Online Test Details:

- -

Certification Course Details:



- [x] 9. Configuring WPA/WPA2 and enabling WPS
  2min
- [x] 10. Methodology of cracking WPS
  3min
- [x] 11. Cracking WPS enabled WPA/WPA2 encryption
  7min

Section 6: Breaking WPA/WPA2 (WPS Disabled)
4 / 4 | 19min

- [x] 12. Deauthenticating client(s) and collecting WPA handshakes
  3min
- [x] 13. Creating effective password dictionary
  6min
- [x] 14. Cracking WPA/WPA2 encryption
  4min
- [x] 15. Cracking WPA/WPA2 encryption using Rainbow tables.
  6min

Section 7: Breaking WEP on Windows
1 / 2 | 9min

- [x] 16. Gathering necessary hacking tools
  3min

Coding Challenges Details:

Program 1:

```
#include<stdio.h>


int isSubset(int arr1[], int arr2[],
             int m, int n)
{
   int i = 0;
   int j = 0;
   for (i = 0; i < n; i++)
   {
      for (j = 0; j < m; j++)
      {
         if(arr2[i] == arr1[j])
            break;
      }


      if (j == m)
         return 0;
   }


   return 1;
```

```
}

int main()
{
   int m,n;
   printf("\nEnter the size of array 1 : ");
   scanf("%d", &m);
   printf("\nEnter the size of array 2 : ");
   scanf("%d", &n);
   int arr1[m],arr2[n];
   int i;
   printf("\nEnter the array 1 elements : ");
   for(i=0;i<m;i++)
   {
      scanf("%d",&arr1[i]);
   }
   printf("\nEnter the array 2 elements : ");
   for(i=0;i<n;i++)
   {
      scanf("%d",&arr2[i]);
   }

   if(isSubset(arr1, arr2, m, n))
```

```
    printf("\nArray2 is a subset of Array1\n ");

  else

    printf("\nArray2 is not a subset of Array1\n");
  getchar();

  return 0;

}
```