DAILY ONLINE ACTIVITIES SUMMARY

Date:	**21-06-2020**	**Name:**	**Shaima Abdul Kader**
Sem & Sec	**VIII Semester & B Section**	**USN:**	**4AL16CS087**

Online Test Summary

Subject	**-**		
Max. Marks	**-**	**Score**	**-**

Certification Course Summary

Course	**Introduction to Data Science (completed)**		
Certificate Provider	**IBM**	**Duration**	**3 Hrs**

Coding Challenges

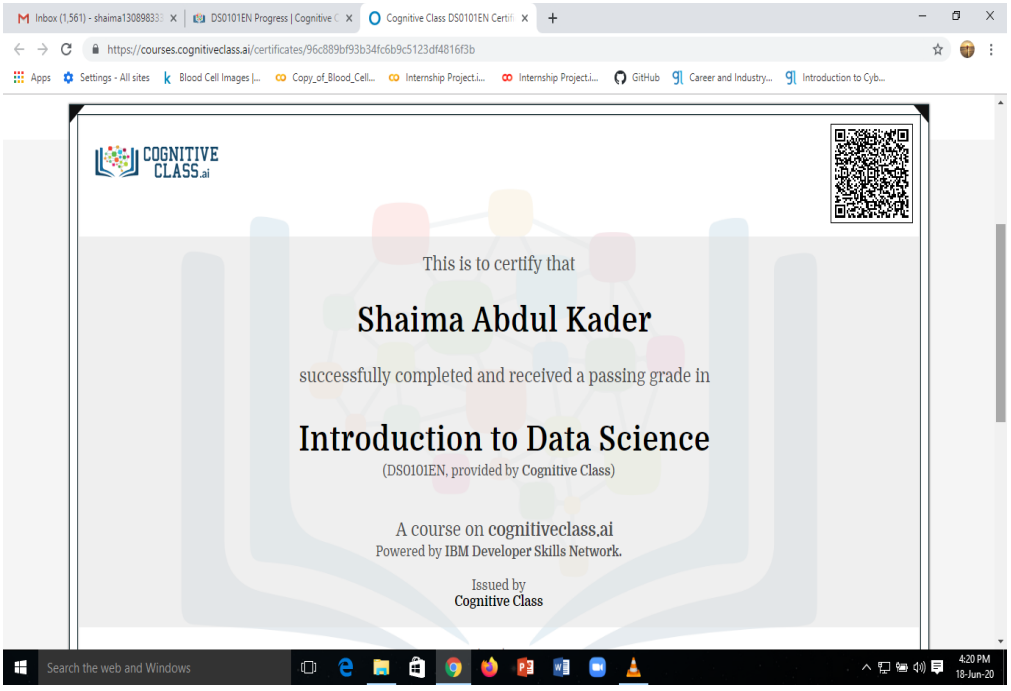
Problem Statement: C Program to Find Transpose of a Matrix.

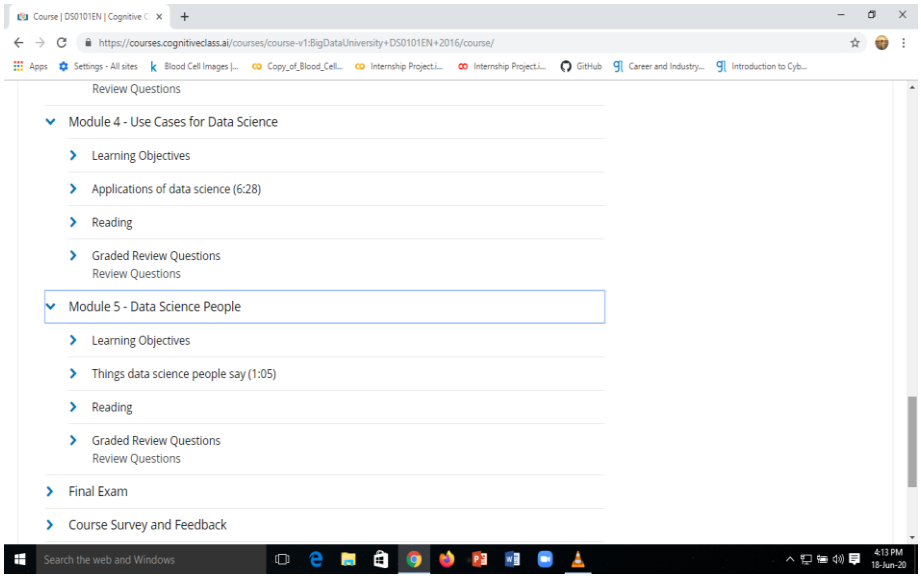
Status: COMPLETED

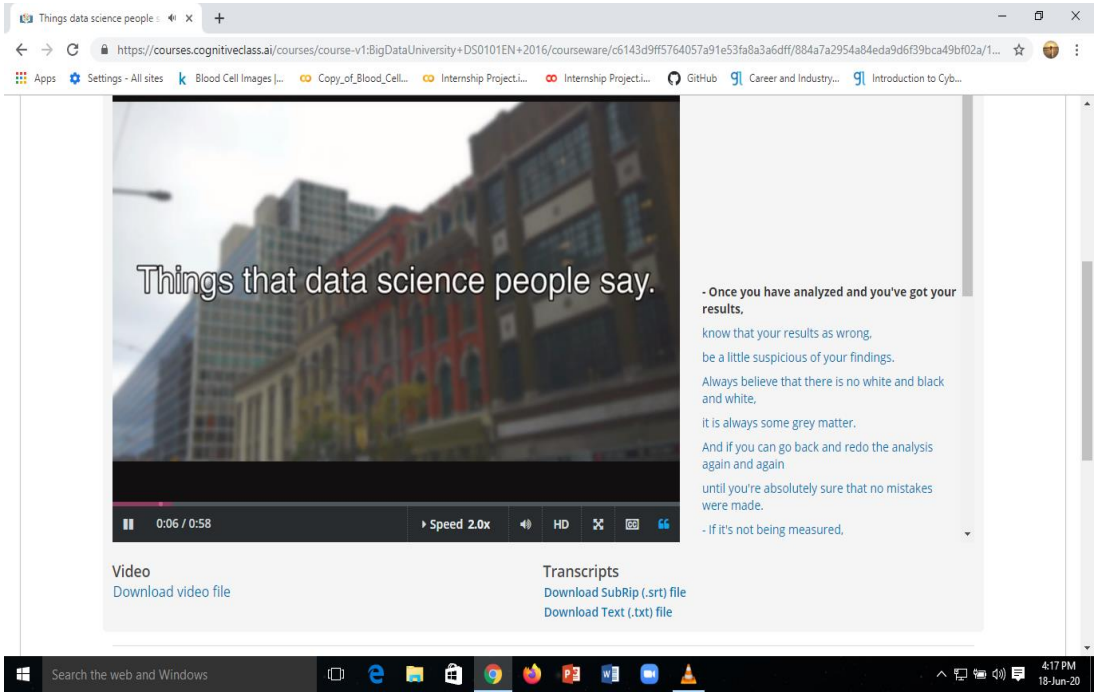
Uploaded the report in Github	**YES**
If yes Repository name	**shaima**
Uploaded the report in slack	**YES**

Online Test Details:

Certification Course Details:




Review Questions
Module 4 - Use Cases for Data Science
Learning Objectives
Applications of data science (6:28)
Reading
Graded Review Questions
Review Questions
Module 5 - Data Science People
Learning Objectives
Things data science people say (1:05)
Reading
Graded Review Questions
Review Questions
Final Exam
Course Survey and Feedback


Things that data science people say.
0:06 / 0:58
Speed 2.0x
HD
- Once you have analyzed and you've got your results,
know that your results as wrong,
be a little suspicious of your findings.
Always believe that there is no white and black and white,
it is always some grey matter.
And if you can go back and redo the analysis again and again
until you're absolutely sure that no mistakes were made.
- If it's not being measured,
Video
Download video file
Transcripts
Download SubRip (.srt) file
Download Text (.txt) file

Coding challenges online details

C Program to Find Transpose of a Matrix

```
#include <stdio.h>
int main() {
   int a[10][10], transpose[10][10], r, c, i, j;
   printf("Enter rows and columns: ");
   scanf("%d %d", &r, &c);

   // Assigning elements to the matrix
   printf("\nEnter matrix elements:\n");
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         printf("Enter element a%d%d: ", i + 1, j + 1);
         scanf("%d", &a[i][j]);
```

```
    }

// Displaying the matrix a[][]
printf("\nEntered matrix: \n");
for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        printf("%d  ", a[i][j]);
        if (j == c - 1)
            printf("\n");
    }

// Finding the transpose of matrix a
for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        transpose[j][i] = a[i][j];
    }

// Displaying the transpose of matrix a
printf("\nTranspose of the matrix:\n");
for (i = 0; i < c; ++i)
    for (j = 0; j < r; ++j) {
        printf("%d  ", transpose[i][j]);
```

```
            if (j == r - 1)
                printf("\n");
        }
    return 0;
}
```