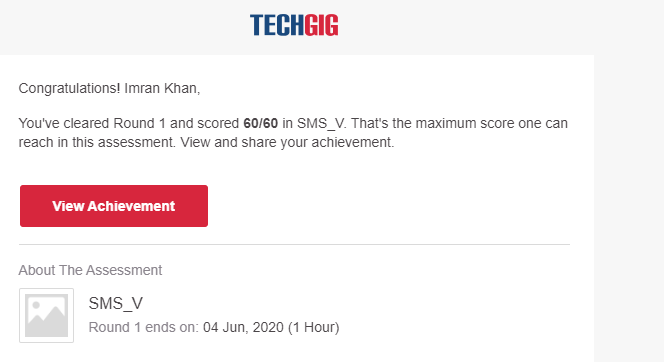
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **4/06/2020** | | | | **Name:** | **Imran Khan** | |
| **Sem & Sec** | **8th A** | | | | **USN:** | **4AL16CS040** | |
| **Online Test Summary** | | | | | | | |
| **Subject** | | **SMS** | | | | | |
| **Max. Marks** | | **60** | | **Score** | | **60** | |
| **Certification Course Summary** | | | | | | | |
| **Course** | **MATLAB FOR MACHINE LEARNING** | | | | | | |
| **Certificate Provider** | | | **IBM** | **Duration** | | | **3 HOURS** |
| **Coding Challenges** | | | | | | | |
| Problem Statement:  Write a C Program to balanced parenthesis.. | | | | | | | |
| **Status: Solved** | | | | | | | |
| **Uploaded the report in Github** | | | | **yes** | | | |
| **If yes Repository name** | | | | **Imran040** | | | |
| **Uploaded the report in slack** | | | | **yes** | | | |

Online test details:



**Certification Course Details**:



**Coding Challenges Details**:

**program1:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | | |
|  | |  | | #include <stdlib.h> | |
|  | |  | | #include <string.h> | |
|  | |  | |  | |
|  | |  | | int top = -1; | |
|  | |  | | char stack[100]; | |
|  | |  | |  | |
|  | |  | | void push(char); | |
|  | |  | | void pop(); | |
|  | |  | | void find\_top(); | |
|  | |  | |  | |
|  | |  | | void main() | |
|  | |  | | { | |
|  | |  | | int i; | |
|  | |  | | char a[100]; | |
|  | |  | | printf("enter expression\n"); | |
|  | |  | | scanf("%s", &a); | |
|  | |  | | for (i = 0; a[i] != '\0';i++) | |
|  | |  | | { | |
|  | |  | | if (a[i] == '(') | |
|  | |  | | { | |
|  | |  | | push(a[i]); | |
|  | |  | | } | |
|  | |  | | else if (a[i] == ')') | |
|  | |  | | { | |
|  | |  | | pop(); | |
|  | |  | | } | |
|  | |  | | } | |
|  | |  | | find\_top(); | |
|  | |  | | } | |
|  | |  | |  | |
|  | |  | | void push(char a) | |
|  | |  | | { | |
|  | |  | | stack[top] = a; | |
|  | |  | | top++; | |
|  | |  | | } | |
|  | |  | |  | |
|  | |  | | void pop() | |
|  | |  | | { | |
|  | |  | | if (top == -1) | |
|  | |  | | { | |
|  | |  | | printf("expression is invalid\n"); | |
|  | |  | | exit(0); | |
|  | |  | | } | |
|  | |  | | else | |
|  | |  | | { | |
|  | |  | | top--; | |
|  | |  | | } | |
|  | |  | | } | |
|  | |  | |  | |
|  | |  | | void find\_top() | |
|  | |  | | { | |
|  | |  | | if (top == -1) | |
|  | |  | | printf("\nexpression is valid\n"); | |
|  | |  | | else | |
|  | |  | | printf("\nexpression is invalid\n"); | |
|  | |  | | } | |