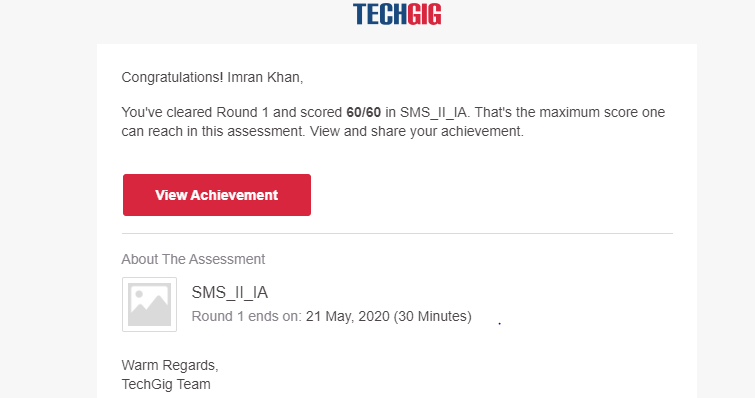
**DAILY ONLINE ACTIVITIES SUMMARY**

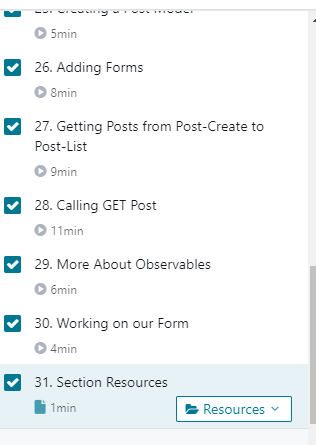
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **21/05/2020** | | | | **Name:** | **Imran Khan** | |
| **Sem & Sec** | **8th A** | | | | **USN:** | **4AL16CS040** | |
| **Online Test Summary** | | | | | | | |
| **Subject** | | **SMS** | | | | | |
| **Max. Marks** | | **60** | | **Score** | | **60** | |
| **Certification Course Summary** | | | | | | | |
| **Course** | **Angular and node js –The mean stack** | | | | | | |
| **Certificate Provider** | | | **Udemy** | **Duration** | | | **20hrs** |
| **Coding Challenges** | | | | | | | |
| Problem Statement: Write C Program to create Singly Liked List with n elements and reverse the elements using C **.** | | | | | | | |
| **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**:

Learn how to connect your Angular Frontend to a NodeJS & Express & MongoDB Backend by building a real Application



**Coding Challenges Details**:

**program1:**

|  |
| --- |
|  |
|  | #include <stdlib.h> |
|  |  |
|  | struct node |
|  | { |
|  | int num; |
|  | struct node \*nextptr; |
|  | }\*stnode; |
|  |  |
|  | void createNodeList(int n); |
|  | void reverseDispList(); |
|  | void displayList(); |
|  |  |
|  | int main() |
|  | { |
|  | int n; |
|  | printf("\n\n Linked List : Create a singly linked list and print it in reverse order :\n"); |
|  | printf("----------------------------------------------------------------------------\n"); |
|  |  |
|  | printf(" Input the number of nodes : "); |
|  | scanf("%d", &n); |
|  | createNodeList(n); |
|  | printf("\n Data entered in the list are : \n"); |
|  | displayList(); |
|  | reverseDispList(); |
|  | printf("\n The list in reverse are : \n"); |
|  | displayList(); |
|  | return 0; |
|  | } |
|  |  |
|  | void createNodeList(int n) |
|  | { |
|  | struct node \*fnNode, \*tmp; |
|  | int num, i; |
|  | stnode = (struct node \*)malloc(sizeof(struct node)); |
|  | if(stnode == NULL) |
|  | { |
|  | printf(" Memory can not be allocated."); |
|  | } |
|  | else |
|  | { |
|  |  |
|  | printf(" Input data for node 1 : "); |
|  | scanf("%d", &num); |
|  | stnode-> num = num; |
|  | stnode-> nextptr = NULL; |
|  | tmp = stnode; |
|  |  |
|  | for(i=2; i<=n; i++) |
|  | { |
|  | fnNode = (struct node \*)malloc(sizeof(struct node)); |
|  | if(fnNode == NULL) |
|  | { |
|  | printf(" Memory can not be allocated."); |
|  | break; |
|  | } |
|  | else |
|  | { |
|  | printf(" Input data for node %d : ", i); |
|  | scanf(" %d", &num); |
|  | fnNode->num = num; |
|  | fnNode->nextptr = NULL; |
|  | tmp->nextptr = fnNode; |
|  | tmp = tmp->nextptr; |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | void reverseDispList() |
|  | { |
|  | struct node \*prevNode, \*curNode; |
|  |  |
|  | if(stnode != NULL) |
|  | { |
|  | prevNode = stnode; |
|  | curNode = stnode->nextptr; |
|  | stnode = stnode->nextptr; |
|  |  |
|  | prevNode->nextptr = NULL; |
|  |  |
|  | while(stnode != NULL) |
|  | { |
|  | stnode = stnode->nextptr; |
|  | curNode->nextptr = prevNode; |
|  |  |
|  | prevNode = curNode; |
|  | curNode = stnode; |
|  | } |
|  | stnode = prevNode; |
|  | } |
|  | } |
|  |  |
|  | void displayList() |
|  | { |
|  | struct node \*tmp; |
|  | if(stnode == NULL) |
|  | { |
|  | printf(" No data found in the list."); |
|  | } |
|  | else |
|  | { |
|  | tmp = stnode; |
|  | while(tmp != NULL) |
|  | { |
|  | printf(" Data = %d\n", tmp->num); |
|  | tmp = tmp->nextptr; |
|  | } |
|  | } |
|  | } |