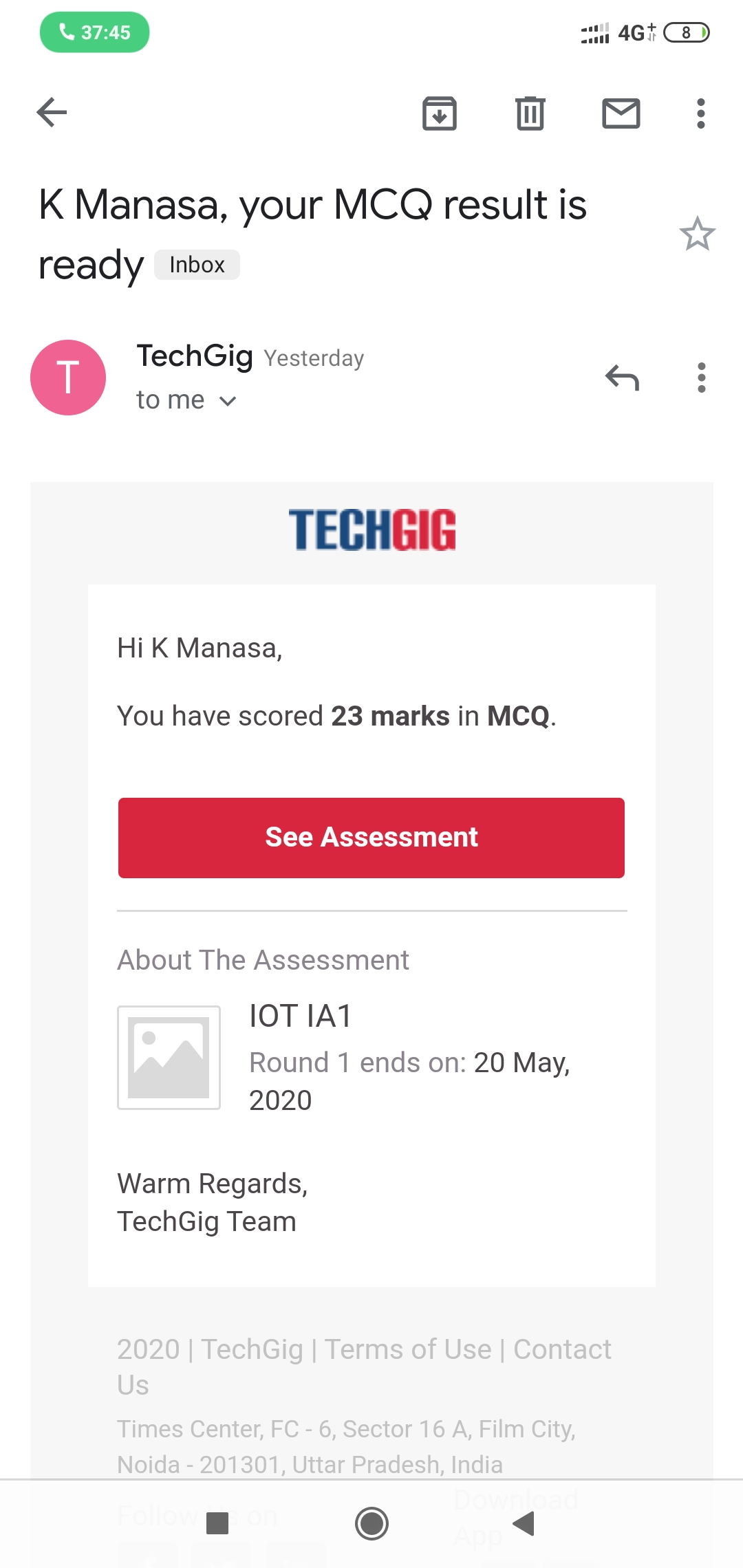
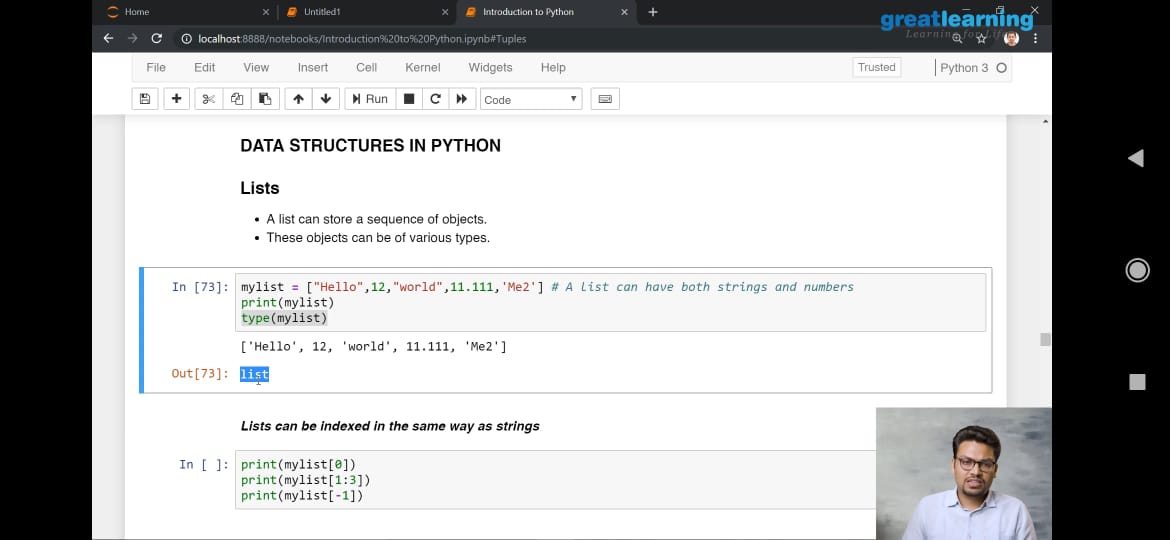
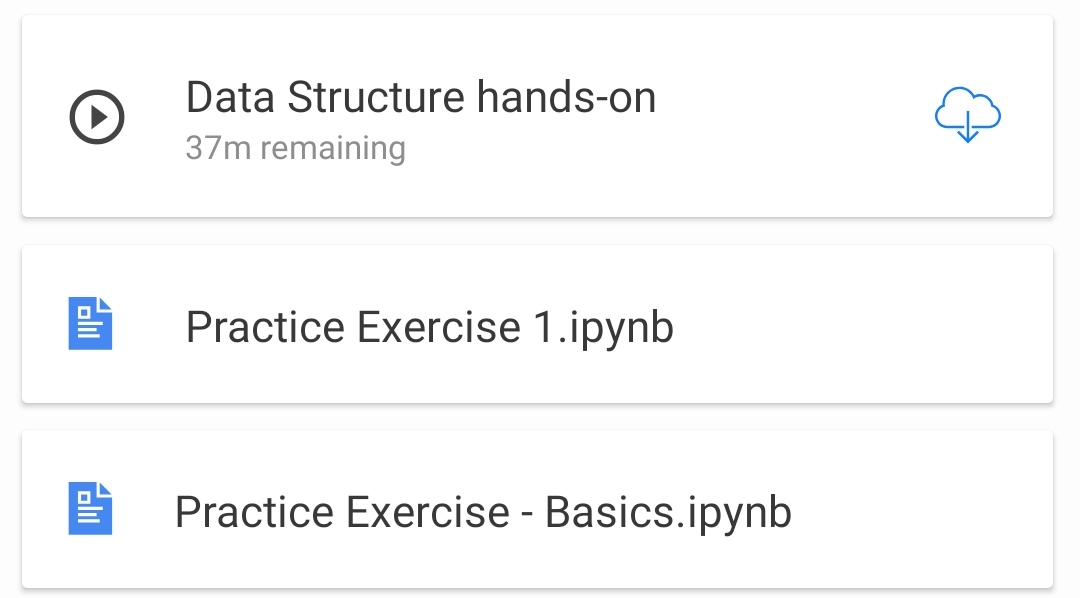
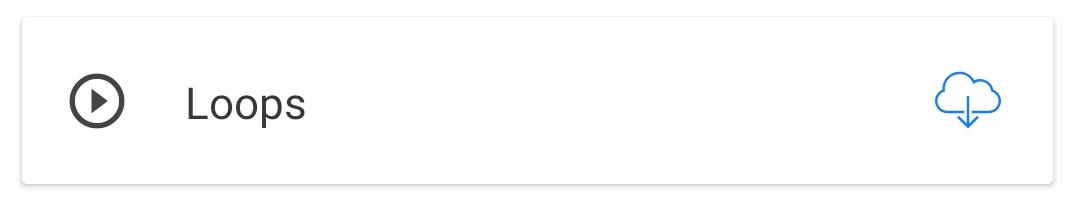
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **20-05-2020** | | | | **Name:** | **K Manasa** | |
| **Sem & Sec** | **8th 'A'** | | | | **USN:** | **4AL16CS043** | |
| **Online Test Summary** | | | | | | | |
| **Subject** | | **Internet of things** | | | | | |
| **Max. Marks** | | **60** | | **Score** | | **23** | |
| **Certification Course Summary** | | | | | | | |
| **Course** | **Python for Machine learning** | | | | | | |
| **Certificate Provider** | | | **Great learning** | **Duration** | | | **1hrs** |
| **Coding Challenges** | | | | | | | |
| **Problem Statement:Finding a frequency of each character in a string and to point the even and odd for series** | | | | | | | |
| **Status:Solved** | | | | | | | |
| **Uploaded the report in Github** | | | | **Yes** | | | |
| **If yes Repository name** | | | | **Manasa** | | | |
| **Uploaded the report in slack** | | | | **Yes** | | | |

Online Test Details: (Attach the snapshot and briefly write the report for the same)



Certification Course Details: (Attach the snapshot and briefly write the report for the same)



Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

PROGRAM1

struct Node

{

int data;

struct Node\* next;

};

pointer to the new head node. /

struct Node reverse (struct Node head, int k)

{

struct Node current = head;

struct Node next = NULL;

struct Node prev = NULL;

int count = 0;

while (current != NULL && count < k)

{

next = current->next;

current->next = prev;

prev = current;

current = next;

count++;

}

if (next != NULL)

head->next = reverse(next, k);

return prev;

}

void push(struct Node\*\* head\_ref, int new\_data)

{

struct Node\* new\_node =

(struct Node\*) malloc(sizeof(struct Node));

new\_node->data = new\_data;

new\_node->next = (\*head\_ref);

(\*head\_ref) = new\_node;

}

void printList(struct Node \*node)

{

while (node != NULL)

{

printf("%d ", node->data);

node = node->next;

}

}

int main(void)

{

struct Node\* head = NULL;

push(&head, 8);

push(&head, 7);

push(&head, 6);

push(&head, 5);

push(&head, 4);

push(&head, 3);

push(&head, 2);

push(&head, 1);

printf("\nGiven linked list \n");

printList(head);

head = reverse(head, 2);

printf("\nReversed Linked list \n");

printList(head);

return(0);