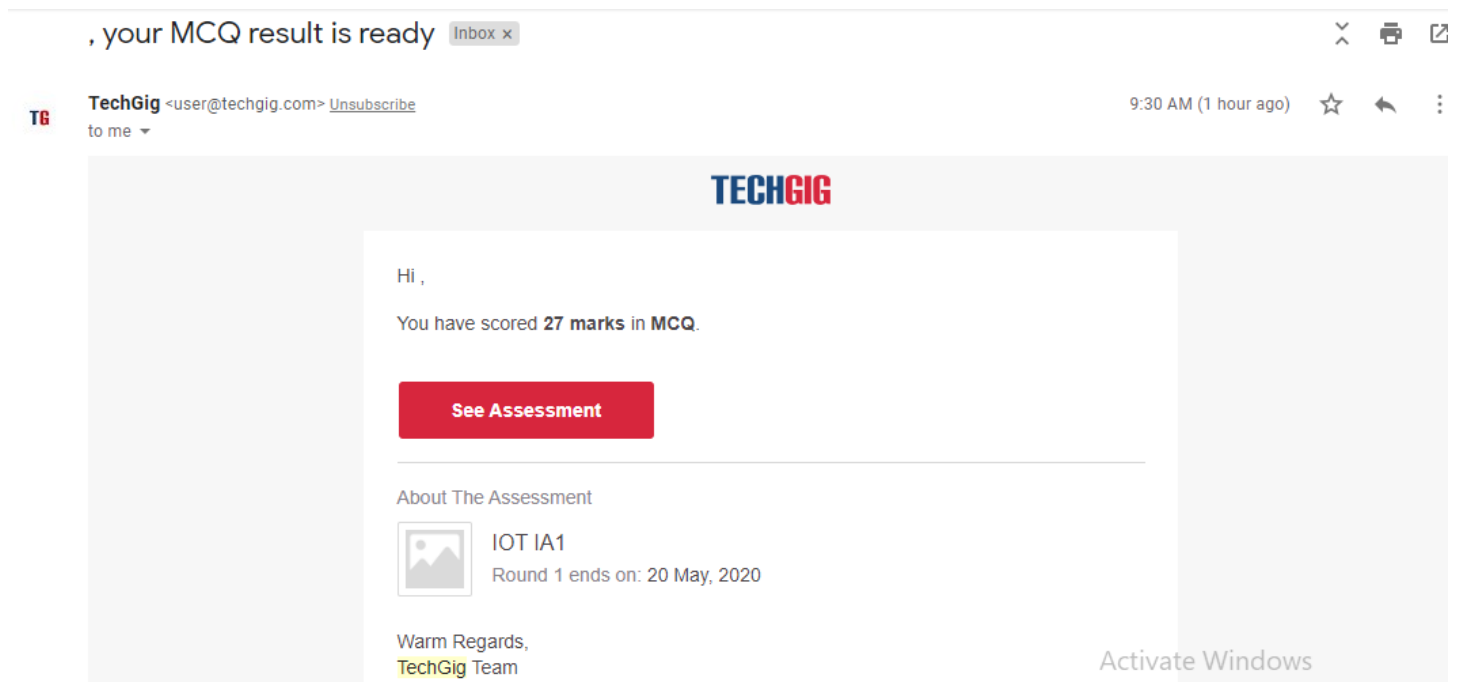


## **DAILY ONLINE ACTIVITIES SUMMARY**

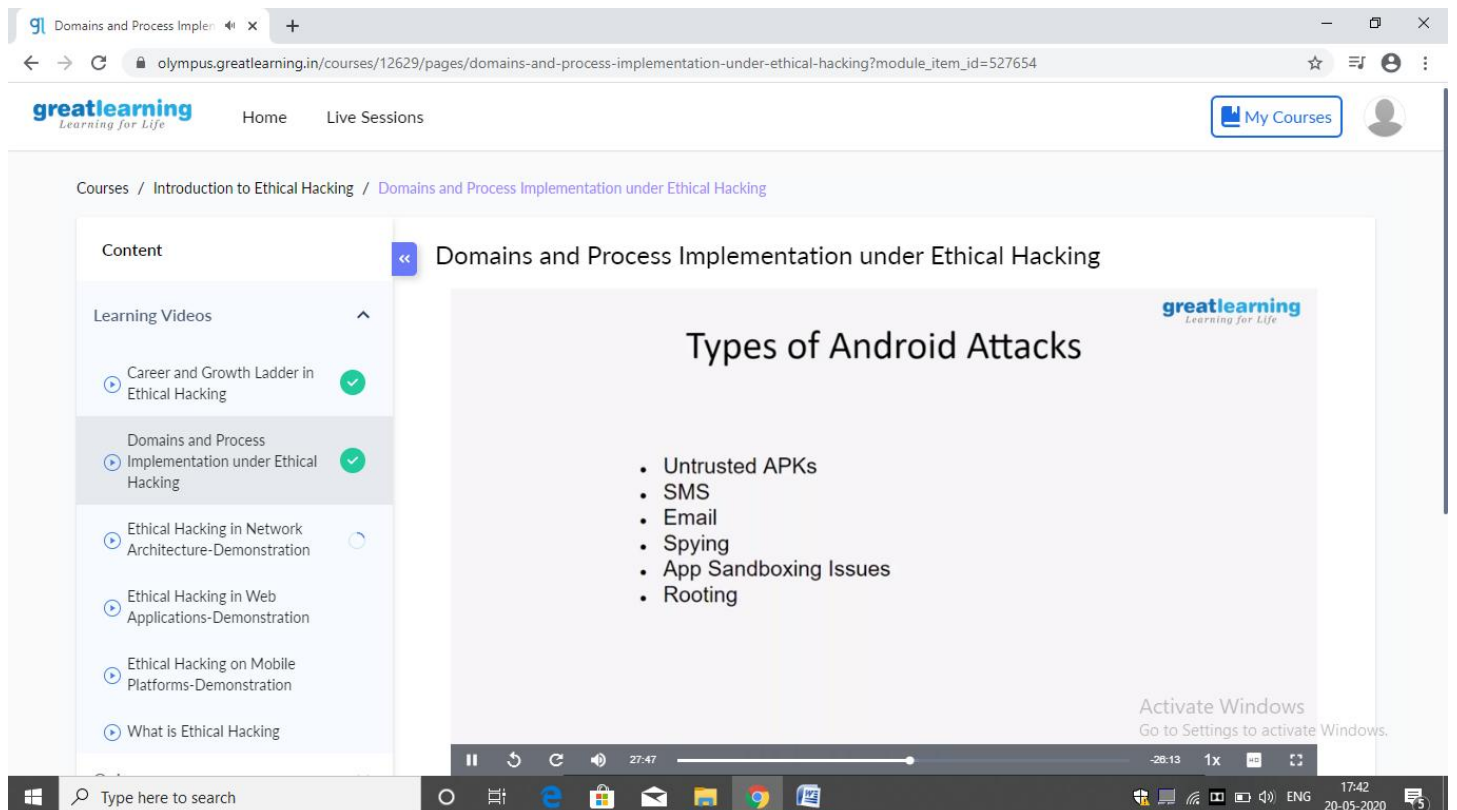
<b>Date:</b>	20/05/2020	<b>Name:</b>	Prathiksha
<b>Sem &amp; Sec</b>	8 <sup>th</sup> sem & B sec	<b>USN:</b>	4AL16CS070
<b>Online Test Summary</b>			
<b>Subject</b>	Introduction To Internet of Things(IOT)		
<b>Max. Marks</b>	30	<b>Score</b>	27
<b>Certification Course Summary</b>			
<b>Course</b>	Introduction To Ethical Hacking		
<b>Certificate Provider</b>	Great Learning Academy	<b>Duration</b>	6hrs
<b>Coding Challenges</b>			
<b>Problem Statement:</b> 1. Write a C Program to Reverse a Linked List in groups of given size 2. Generate Armstrong numbers using Python programming language			
<b>Status:</b> Solved			
<b>Uploaded the report in Github</b>		Yes	
<b>If yes Repository name</b>		Prathiksha	
<b>Uploaded the report in slack</b>		Yes	

## Online Test Details:



IA1 portion was Module 1 and 2.

## Certification Course Details:



**Topic :** Domains and progress implementation under ethical hacking.

## Coding Challenges Details:

### Program 1:

Test Case 1:

If a linked list is:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$

The value of size k is 2

Then the linked list looks like:  $2 \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 8 \rightarrow 7$

Test Case 2:

If a linked list is:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$

The value of size k is 3

Then the linked list looks like:  $3 \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 8 \rightarrow 7$

```
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);

```

---

## Program 2:

```

print("Enter 'x' for exit.");
print("Enter the interval (starting and ending number): ");
start = input();
if start == 'x':
    exit();
else:
    end = input();
    lower = int(start);
    upper = int(end);
    for num in range(lower, upper+1):
        tot = 0;
        temp = num;
        while temp != 0:
            dig = temp % 10;
            tot += dig ** 3;
            temp //= 10;
        if num == tot:
            print(num);

```