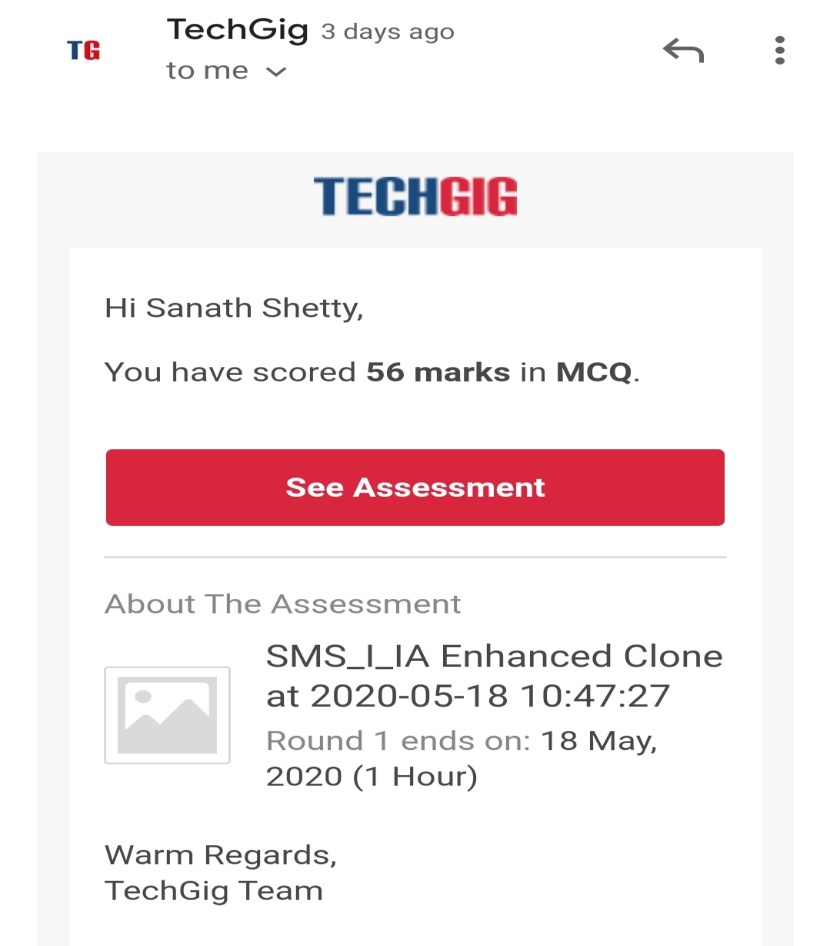
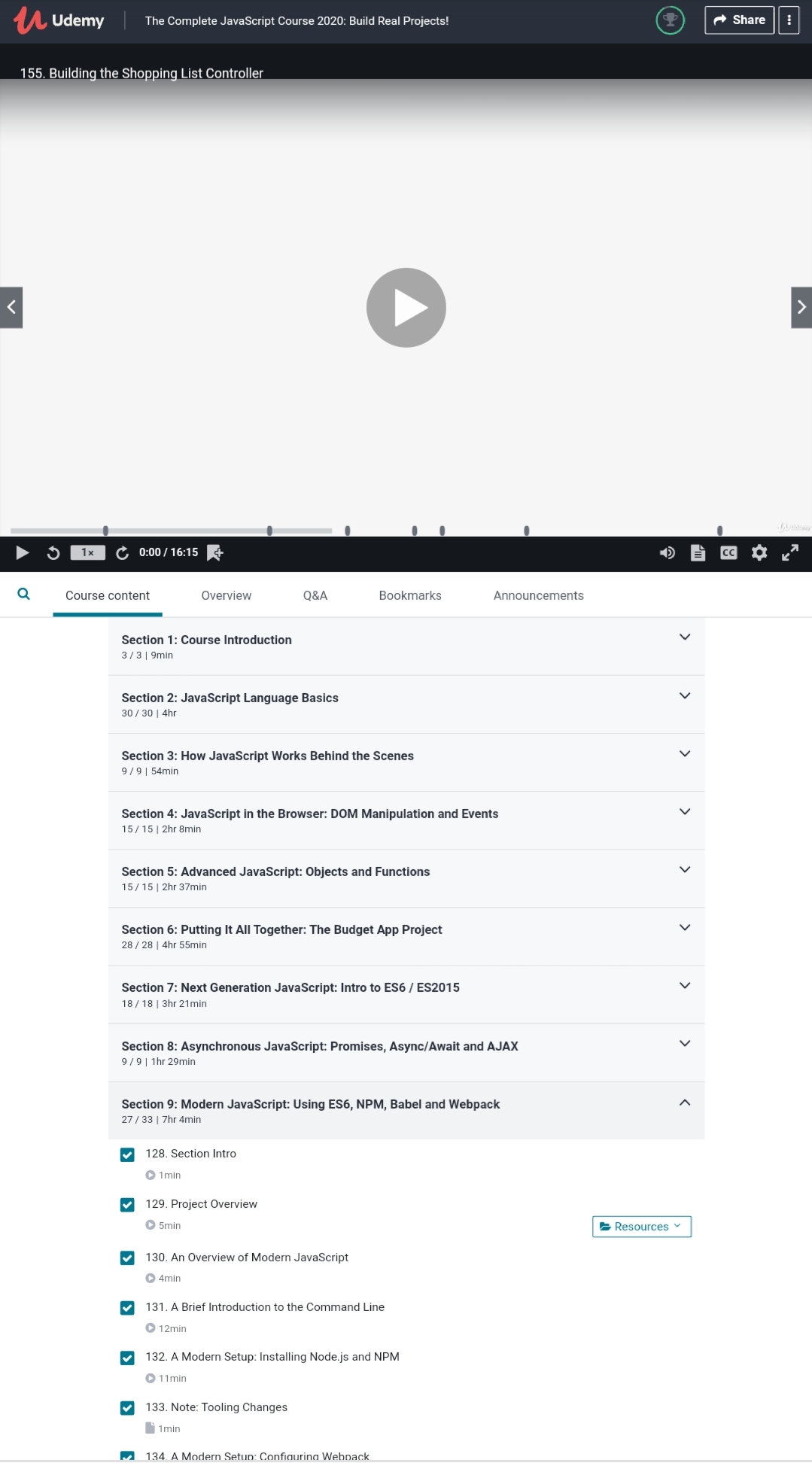
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **18-05-2020** | | | | | **Name:** | **Sanath Shetty** | |
| **Sem & Sec** | **8th sem B sec** | | | | | **USN:** | **4AL16CS094** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **SMS** | | | | | | |
| **Max. Marks** | | **60** | | **Score** | | | **56** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **The Complete javascript course 2020** | | | | | | | |
| **Certificate Provider** | | | **Udemy** | | **Duration** | | | **28 hours** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement: java coding problem** | | | | | | | | |
| **Status: completed** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | **alvas-education-foundation/Sanath-Shetty** | | | |
| **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)

Watched the previous pending videos of The complete javascript course

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

Coding was given n it was uploaded for github and slack

**PROGRAM 1**

package pk;

import java.util.Scanner;

public class StringOperators

{

public static void main(String args[])

{

int i;

String str;

int counter[] = new int[256];

Scanner in = new Scanner(System.in);

System.out.print("Enter a String : ");

str=in.nextLine();

for (i = 0; i < str.length(); i++) {

counter[(int) str.charAt(i)]++;

}

// Print Frequency of characters

for (i = 0; i < 256; i++) {

if (counter[i] != 0) {

System.out.println((char) i + ":-" + counter[i] + " times");

}

}

}

**}Bottom of Form**

**PROGRAM 2**

import java.util.Stack;

// Data Structure to store a linked list node

class Node {

int data;

Node next;

Node(int i)

{

this.data = i;

this.next = null;

}

};

class Main

{

// Function to determine if a given linked list is palindrome or not

public static boolean isPalindrome(Node head)

{

// construct an empty stack

Stack s = new Stack<>();

// push all elements of the linked list into the stack

Node node = head;

while (node != null) {

s.push(node.data);

node = node.next;

}

// traverse the linked list again

node = head;

while (node != null)

{

// pop the top element from the stack

int top = s.pop();

// compare the popped element with current node's data

// return false if mismatch happens

if (top != node.data) {

return false;

}

// advance to the next node

node = node.next;

}

// we reach here only when the linked list is palindrome

return true;

}

public static void main(String[] args)

{

Node head = new Node(1);

head.next = new Node(2);

head.next.next = new Node(3);

head.next.next.next = new Node(2);

head.next.next.next.next = new Node(1);

if (isPalindrome(head)) {

System.out.print("Linked List is a palindrome.");

} else {

System.out.print("Linked List is not a palindrome.");

}

}