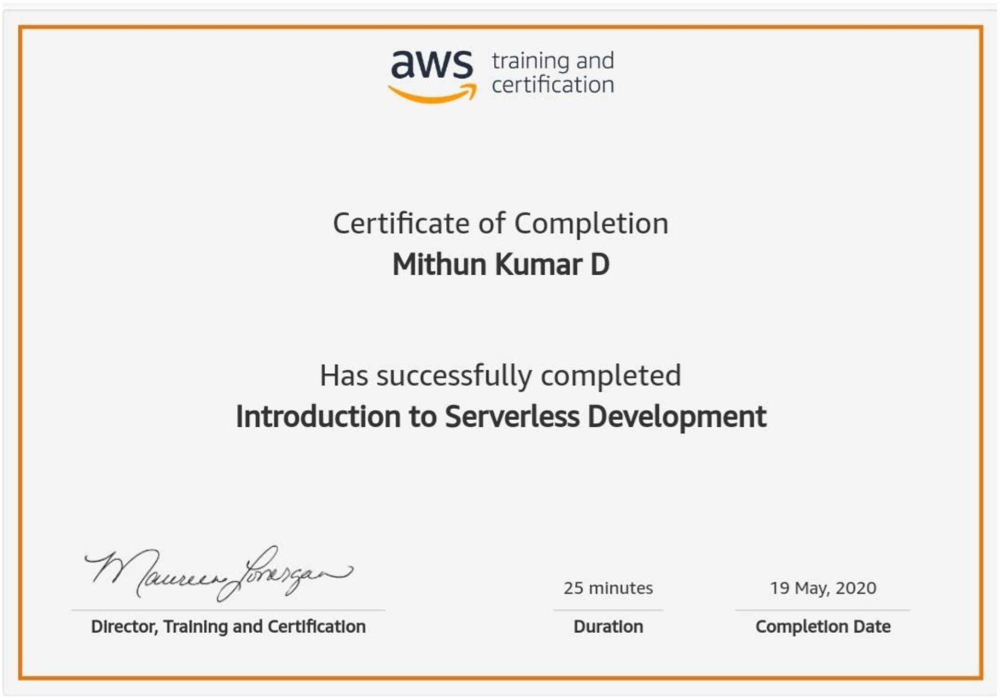
DAILY ONLINE ACTIVITIES SUMMARY

Date:	**19/05/2020**	**Name:**	**Mithun Kumar D**
Sem & Sec	**VIII Semester & A section**	**USN:**	**4AL16CS053**

Online Test Summary			
Subject	**BDA (Couldn’t able to attend test due to network issues)**		
Max. Marks	**30**	**Score**	**00**
Certification Course Summary			
Course	**Introduction to Serverless Development**		
Certificate Provider	**AWS**	**Duration**	**(25 mins)**
Coding Challenges			
Problem Statement: 1) To find out the shortest palindrome 2) To identify is given list is palindrome or not using stack			
Status: COMPLETED			
Uploaded the report in Github	**YES**		
If yes Repository name	**mkd18**		
Uploaded the report in slack	**YES**		

Certification Course Details:


training and certification

Certificate of Completion
Mithun Kumar D

Has successfully completed
Introduction to Serverless Development

[illegible]
Director, Training and Certification

25 minutes
Duration

19 May, 2020
Completion Date

Coding Challenges Details:

1) We have a Letter or a word then we need add some letters to it and need to find out shortest palindrome

For example we take "S": S will be the shortest palindrome string.

If we take "xyz": zyxyz will be the shortest palindrome string

So we need to add some characters to the given string or character and find out what will be the shortest palindrome string by using simple java program.

```
package shortestpalindromeexample.java;

import java.util.Scanner;


public class ShortestPalindromeDemo {


public static String shortestPalindrome(String str) {
```

```
int x=0;

int y=str.length()-1;


while(y>=0){

if(str.charAt(x)==str.charAt(y)){

x++;

}

y--;

}


if(x==str.length())

return str;


String suffix = str.substring(x);

String prefix = new StringBuilder(suffix).reverse().toString();

String mid = shortestPalindrome(str.substring(0, x));


return prefix+mid+suffix;

}


public static void main(String[] args) {


Scanner in = new Scanner(System.in);
```

```
System.out.println("Enter a String to find out shortest palindrome");


String str=in.nextLine();


System.out.println("Shortest palindrome of "+str+" is "+shortestPalindrome(str));
 }
}
```

2) Write a simple code to identify given linked list is palindrome or not by using stack.

First take a Stack. Traverse through each node of the linked list and push each node value to Stack.

Once the traversal & copying is done, iterate through linked list from head node again.

In each iteration, pop one stack element and compare with node value in respective iteration. It is expected to match stack popped value with node value.

In case of all matches, its a palindrome. Any one element mismatch makes it not a palindrome.

```
import java.util.Stack;


class Node {
int data;
Node next;


Node(int i)
{
```

```
this.data = i;
this.next = null;
}
};

class Main
{
public static boolean isPalindrome(Node head)
{
Stack s = new Stack<>();

Node node = head; // push
while (node != null) {
s.push(node.data);
node = node.next;
}

// traverse
node = head;
while (node != null)
{
int top = s.pop(); //pop

if (top != node.data) {
return false;
}

node = node.next;
}

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.");
}
}
}
```