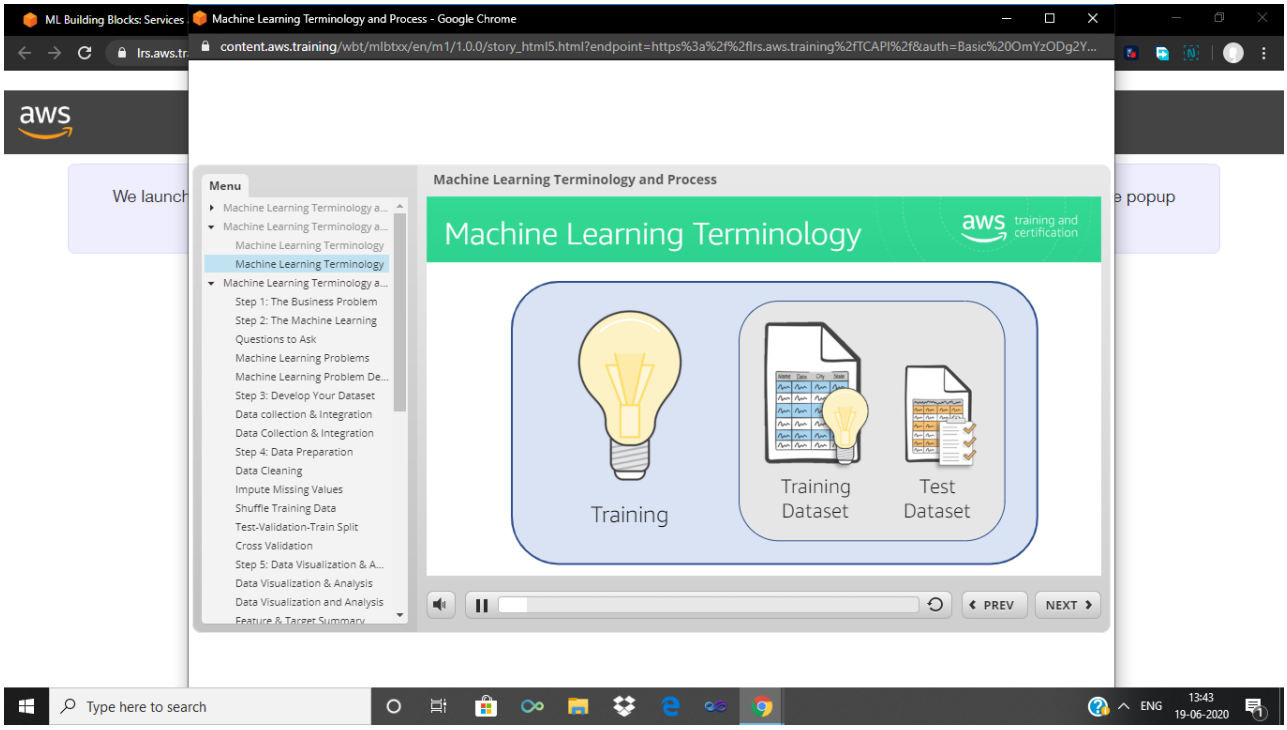
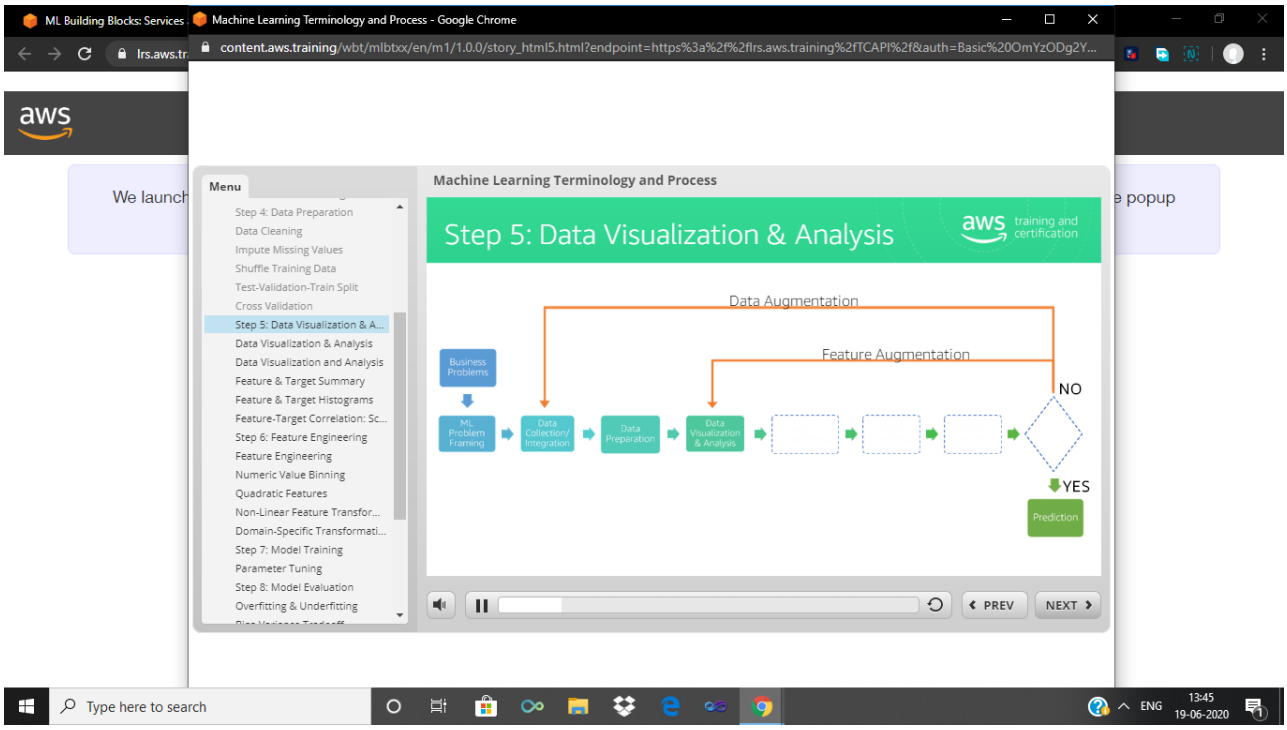
DAILY ONLINE ACTIVITIES SUMMARY

Date:	***17-06-2020***	***Name:***	**SAFNAAZ**
Sem & Sec	**8th B**	**USN:**	**4AL16CS081**
Online Test Summary			
Subject	-		
Max. Marks	-	***Score***	-
Certification Course Summary			
Course	**Amazon web service**		
Certificate Provider	**Aws**	***Duration***	***3 Hours***
Coding Challenges			
Problem Statement: **c program for triply linked list**			
Status: COMPLETED			
Uploaded the report in Github	**YES**		
If yes Repository name	**Safnaazsheikh**		
Uploaded the report in slack	***YES***		

Certification Course Details:





Coding challenges online details:

c program for triply linked list

```
#include<stdio.h>
struct SLL;
struct TLL {
struct TLL *top;
struct TLL *bottom;
struct SLL *next;
};
typedef struct TLL tnode;
typedef struct SLL {
char ch;
struct SLL *link;
};
typedef struct SLL snode;
snode *newnode, *ptr, *prev, *temp;
snode *first = NULL, *last = NULL;
tnode *newt, *tlast = NULL, *ttemp;
//--- TLL node---
tnode* create_tnode()
{newt = (tnode *)malloc(sizeof(tnode));
if (newt == NULL)
{
printf("\nMemory was not allocated");
return 0;
}
else
{
newt->top = NULL;
newt->bottom = NULL;
newt->next = NULL;
return newt;
}
}
//---SLL---
snode* create_node(char c)
{
newnode = (snode *)malloc(sizeof(snode));
if (newnode == NULL)
{
printf("\nMemory was not allocated");
return 0;
}
else
{
newnode->ch = c;
newnode->link = NULL;
```

```
return newnode;
}
}
//--- insert SLL---
void insert_node_first(char c)
{
newnode = create_node(c);
if(tlast->next == NULL)
tlast->next = newnode;
if (first == last && first == NULL)
{
first = last = newnode;
first->link = NULL;
last->link = NULL;
}
else
{
temp = first;
first = newnode;
first->link = temp;
}printf("\n----INSERTED %c TO SLL----", c);
}
//---insert TLL---
void insert_Tnode()
{
newt = create_tnode();
if (tlast == NULL)
{
tlast = newt;
tlast->next = NULL;
tlast->top = NULL;
tlast->bottom = NULL;
}
else
{
ttemp = tlast;
tlast = newt;
tlast->next = NULL;
tlast->top = ttemp;
tlast->bottom = NULL;
ttemp->bottom = tlast;
}
printf("\n----CREATED NEW TLL----");
}
void main()
{
char s[100], n;
int i;
scanf("%[^;]s",s);
insert_Tnode();
for(i = 0; s[i] != '\0'; i++)
```

```
{
n = s[i];
if(n == '\n')
insert_Tnode();
else
insert_node_first(n);
}
printf("\n%s\n",s);
}
```