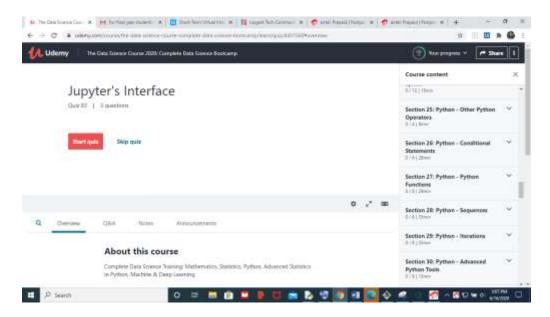
## **DAILY ONLINE ACTIVITIES SUMMARY**

Date:	17/06/202	20	Name:	Vishwa	Vishwas Acharya	
Sem & Sec	8 <sup>th</sup> - A		USN:	4AL16	CS002	
		Online Te	st Summary	7		
Subject						
Max. Marks			Score			
Certification Course Summary						
Course The Data Science Course 2020:Complete Data Science Bootcamp						
Certificate Provider		Udemy	Duration		29hours	
Coding Challenges						
Problem Statement: Python program to check if a binary tree is bst or not						
Status: Exec	cuted					
Uploaded th	e report i	n Github	Yes			
If yes Repository name			vishwas_acharya			
Uploaded th	e report i	n slack	Yes			

## **Online Test Details:**

## **Certification Course Details:**



## **Coding Challenges Details:**

```
program34.py - D/Useis/lennin/Desktop/vishiers, acharya/oxiding_solutions/program34.py (3.6.1)
File Edit Format Pun Options Window Help
# Python program to check if a binary tree is but or not
                                                                                    Python 3.8.1 Shell
INT MAX = 4294967296
INT MIN = -4294967296
Class Node:
                                                                                     File Edit Shell Debug Options Window Help
                                                                                    Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AM
      def _init (self, data):
    self.data = data
    self.left = None
    self.right = Mone
                                                                                    D64)] on win32
                                                                                    Type "help", "copyright", "credits" or "license()" for more information.
                                                                                    = RESTART: C:/Users/lenovo/Desktop/Vishwas_acharya/coding_solutions/grogram3e.py
Is BST
>>>>
def isBIT(node);

return (isBITUT1(node, IST_HIM, INT_MAX))

def isBITUT1(node, mini, maxi);

if node is Nume;
      If node.data < mini == node.data > maxi:
      isBSTUtil(node.left, mini, node.data -1) and
isBSTUtil(node.right, node.data+1, maxi))
root = Node(4)
root.left = Node(2)
root.right = Node(5)
 soot.left.left = Bode(1)
soot.left.right = Bode(3)
 of (isBST(root)):
print ("Is BST")
=lre:
      print ("Not a Bet")
```