**DAILY REPORT**

**Student Name :SUSHMITHA.B.POOJARY**

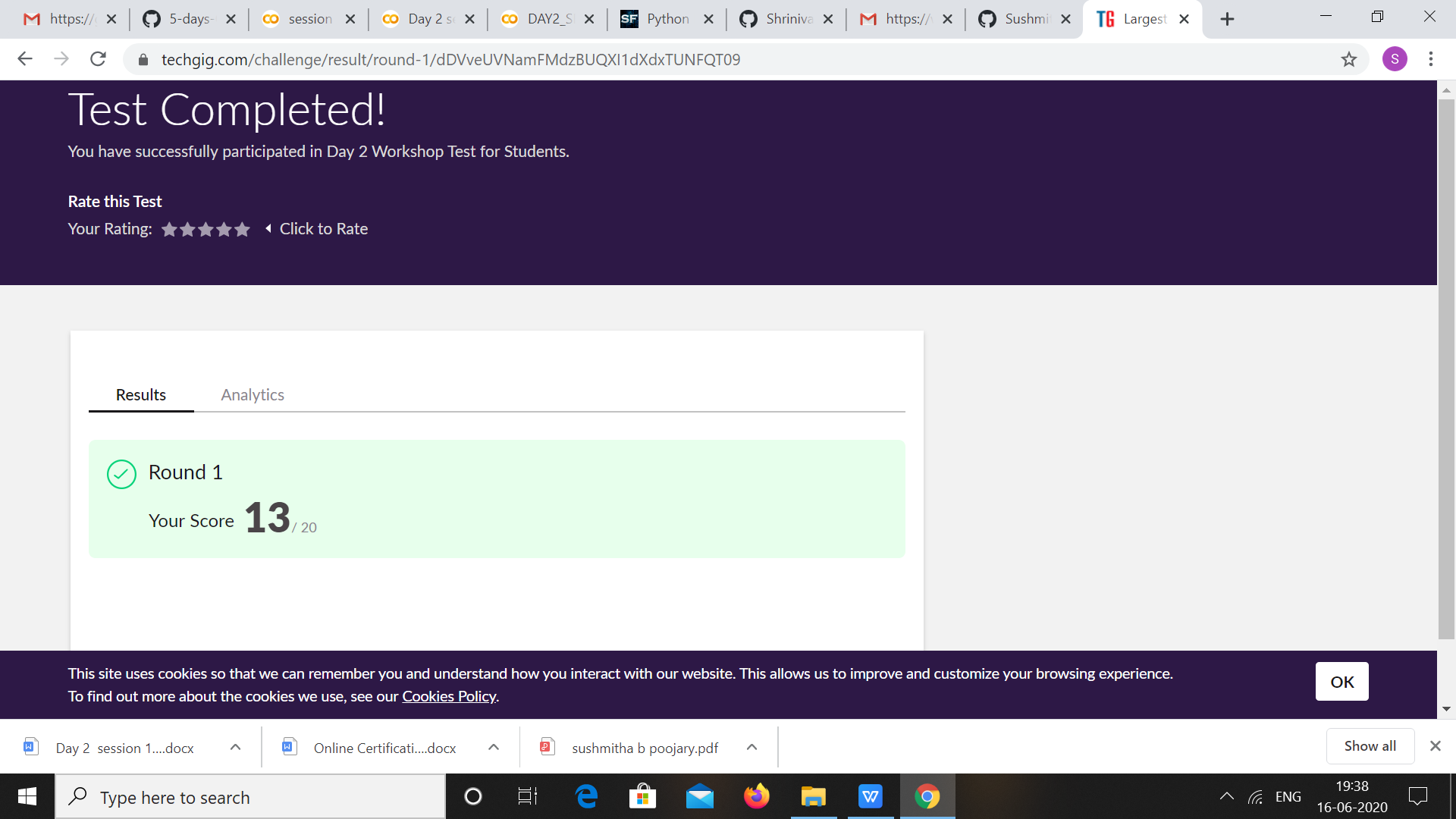
**Class and Sec : VI B**

**USN :4AL17CS103**

**DATE:16-06-2020**

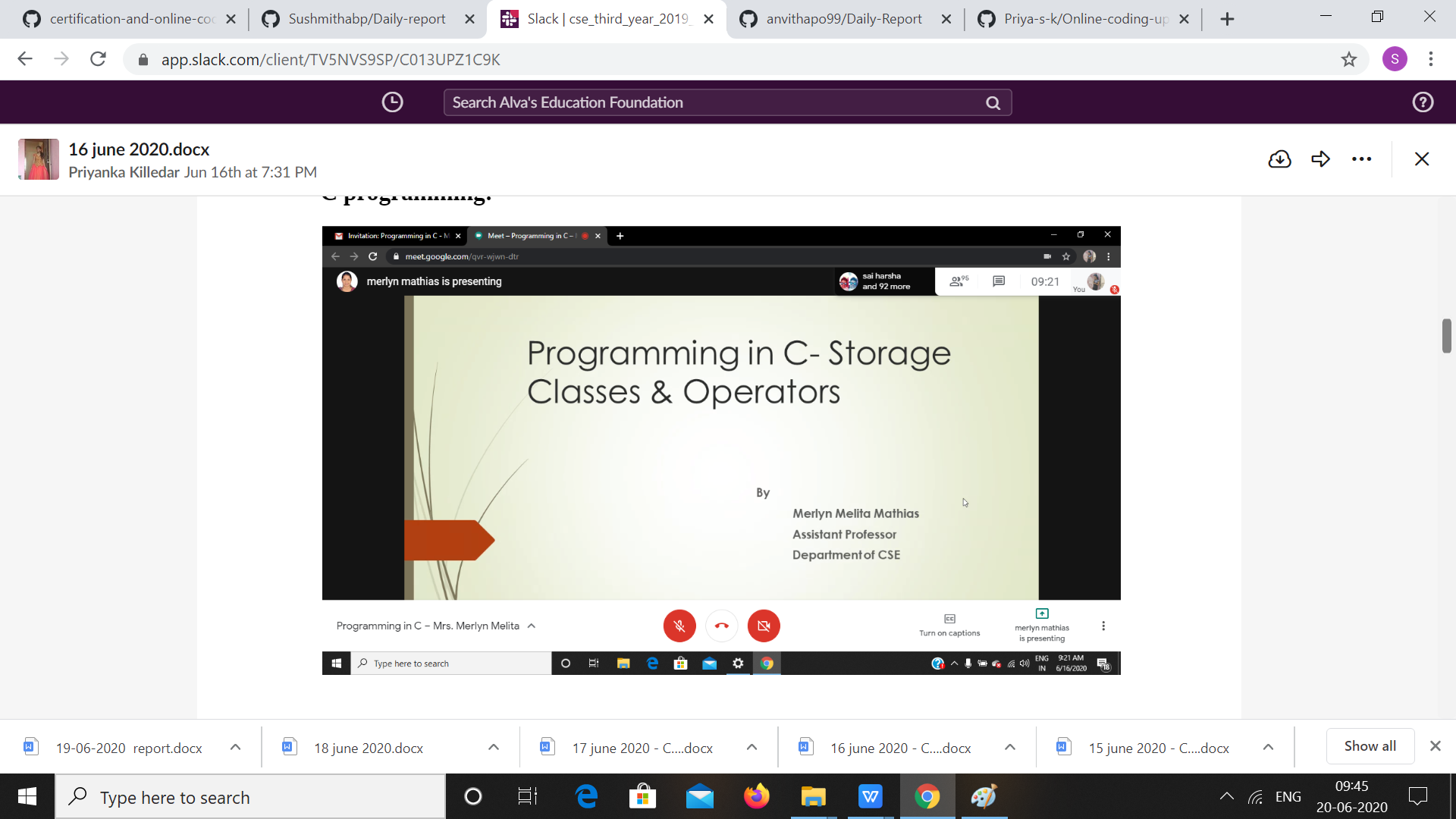
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Online Test Details** | | | | |
| **Subject** | **PYTHON WORKSHOP QUIZ** | | | |
| **Semester** | **VI -B** | | **Duration** | **30 Minutes** |
| **% of marks 20** | | **13** | | |

**snapshot of the test result**

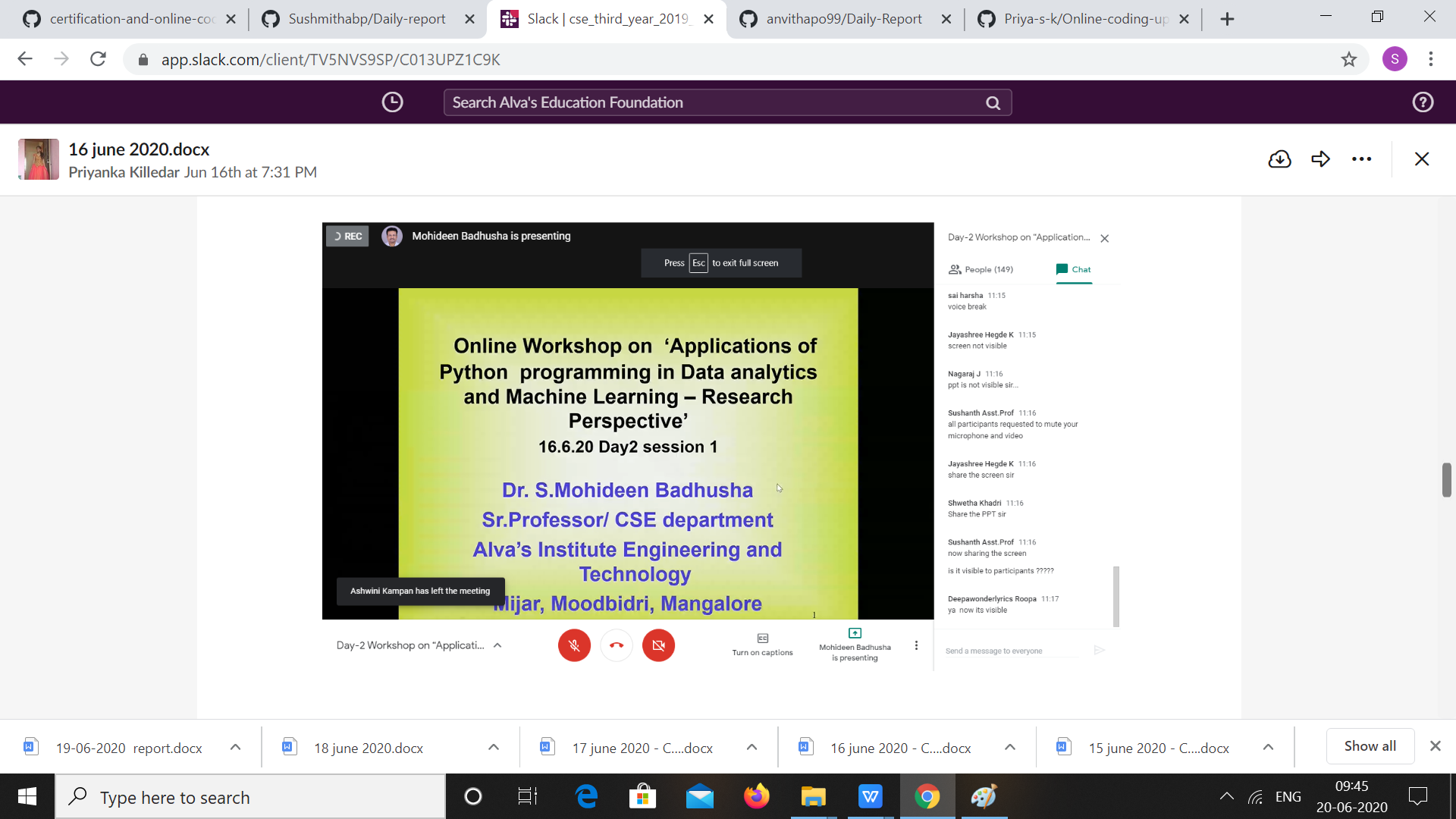
****

|  |  |  |  |
| --- | --- | --- | --- |
| **Pre-Placement Training** | | | |
| **Pre placement training** | **-** Programming in C(9:00 am to 11:00 am)  - Applications of python in DA and ML(11:00 am to 1:00pm) | | |
| **Faculty** | Mrs. Merlyn mathias,  Dr. Mohideen Badusha | **Duration** | 4 hours |

**snapshots of the daily class acitivities**

**PROGRAMMING INC **

PYTHON WORKSHOP

****

|  |  |
| --- | --- |
| **Coding Challenges** | |
| **Problem Statement:** 1 .Python program to check given tree is BST or NOT   1. Examples and Exercises on python Workshop. (Linked Google-colab to GitHub)   <https://github.com/Sushmithabp/Workshop-on-python-programming-in-DA-and-ML> | |
| **Status: Executed** | |
| **Uploaded the report both in Github & Slack** | **Yes** |

**Snapshots of your response to challenge.**

**Write a Python program to check whether a given a binary tree is a valid binary search tree (BST) or not?**

class TreeNode(object):

    def \_\_init\_\_(self, x):

        self.val = x

        self.left = None

        self.right = None

def is\_BST(root):

    stack = []

    prev = None

    while root or stack:

        while root:

            stack.append(root)

            root = root.left

        root = stack.pop()

        if prev and root.val <= prev.val:

            return False

        prev = root

        root = root.right

    return True

root = TreeNode(2)

root.left = TreeNode(1)

root.right = TreeNode(3)

result = is\_BST(root)

print(result)

root = TreeNode(1)

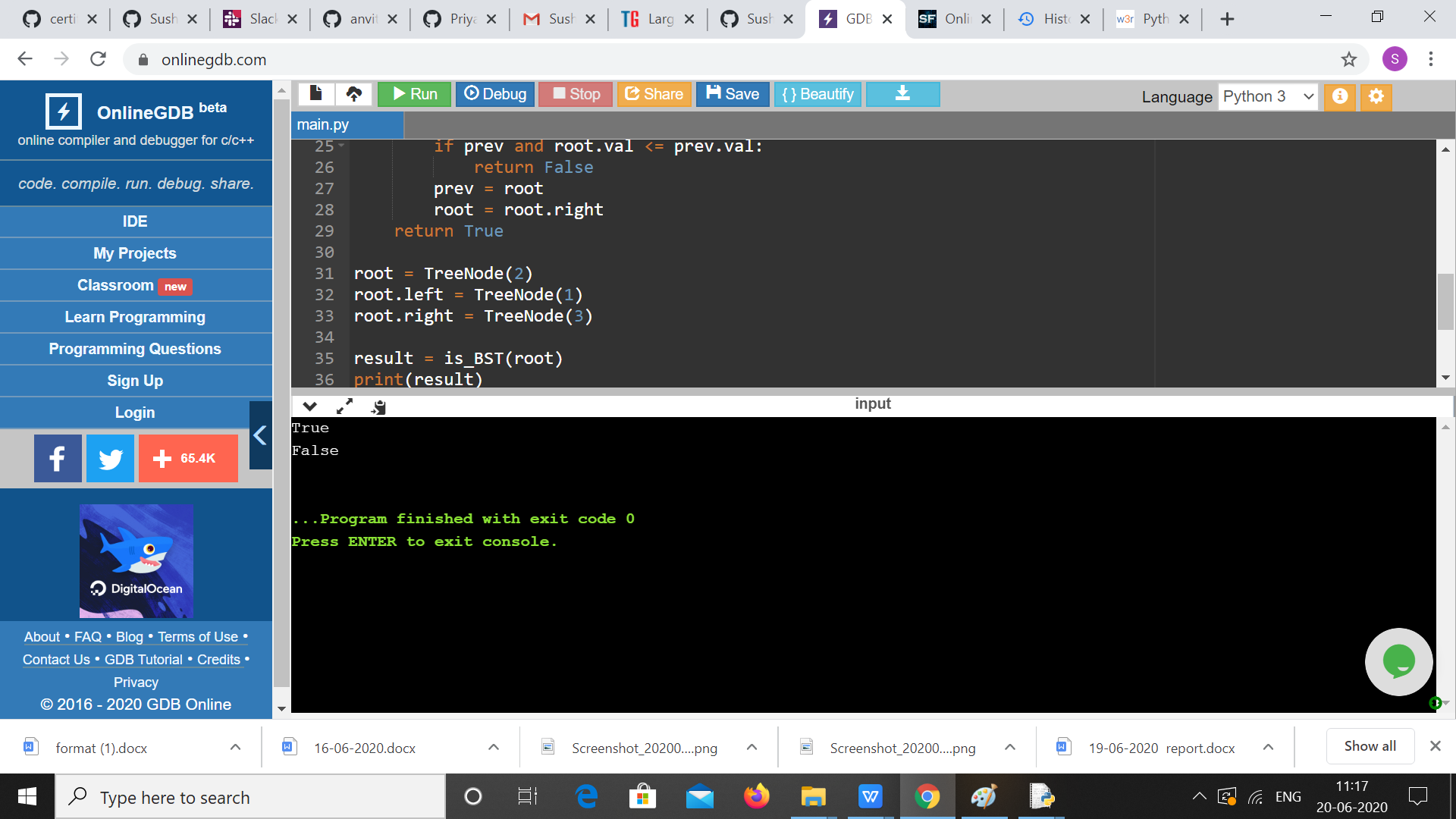
root.left = TreeNode(2)

root.right = TreeNode(3)

result = is\_BST(root)

print(result)

**Output:**

****