DAILY ONLINE ACTIVITIES SUMMARY

Date:	**21-06-2020**	**Name:**	**Deepika K V**
Sem & Sec	**8th sem 'A' sec**	**USN:**	**4AL16CS030**

Online Test Summary

Subject	--		
Max. Marks	-	**Score**	-

Certification Course Summary

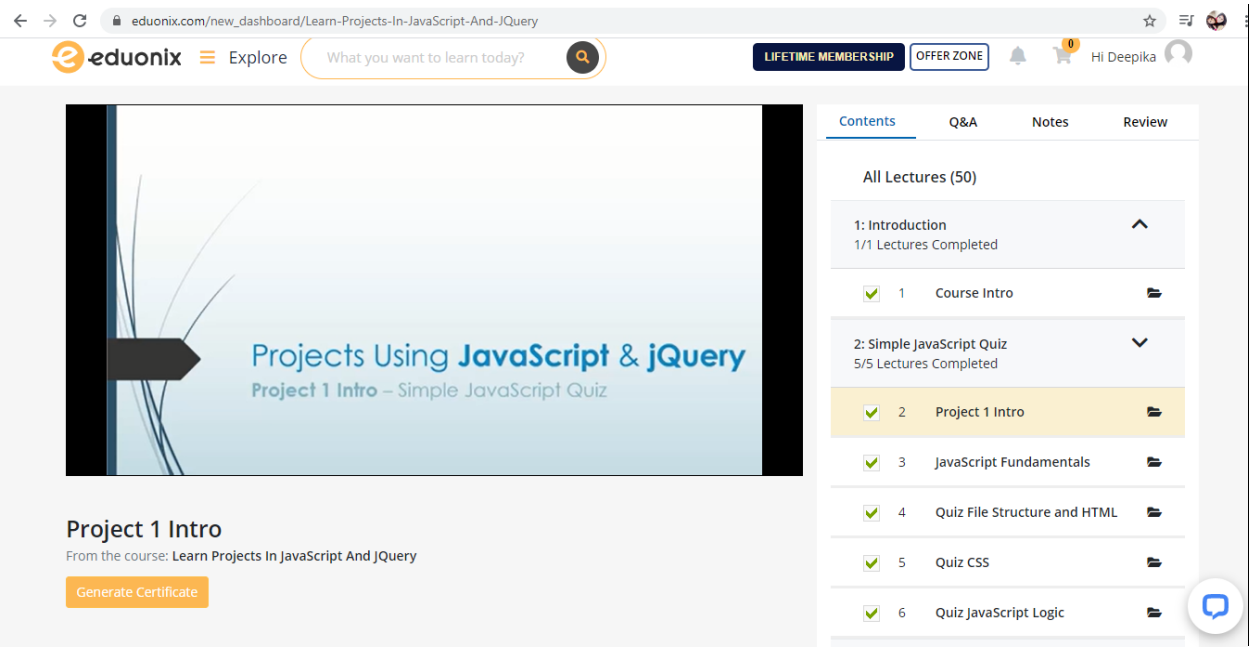
Course	Learn Projects in Javascript and JQuery		
Certificate Provider	**eduonix**	**Duration**	**9 hrs**

Coding Challenges

Problem Statement: Write a python program to check the given binary tree is a valid binary search tree(BST) or not.	
Status: SUBMITTED	
Uploaded the report in Github	**YES**
If yes Repository name	**Codes**
Uploaded the report in slack	**YES**

Online test details:

Certification Course Details:



Coding Challenge:

```
INT_MAX =
4294967296
            INT_MIN = -4294967296

            class Node:

                # Constructor to create a new node
                def __init__(self, data):
                    self.data = data
                    self.left = None
                    self.right = None


            # Returns true if the given tree is a binary search tree
            # (efficient version)
            def isBST(node):
                return (isBSTUtil(node, INT_MIN, INT_MAX))
```

```
# Retusn true if the given tree is a BST and its values
# >= min and <= max
def isBSTUtil(node, mini, maxi):

    # An empty tree is BST
    if node is None:
        return True

    # False if this node violates min/max constraint
    if node.data < mini or node.data > maxi:
        return False

    # Otherwise check the subtrees recursively
    # tightening the min or max constraint
    return (isBSTUtil(node.left, mini, node.data -1) and
          isBSTUtil(node.right, node.data+1, maxi))

# Driver program to test above function
root = Node(4)
root.left = Node(2)
root.right = Node(5)
root.left.left = Node(1)
root.left.right = Node(3)

if (isBST(root)):
    print ("Is BST")
else:
    print ("Not a BST")
```