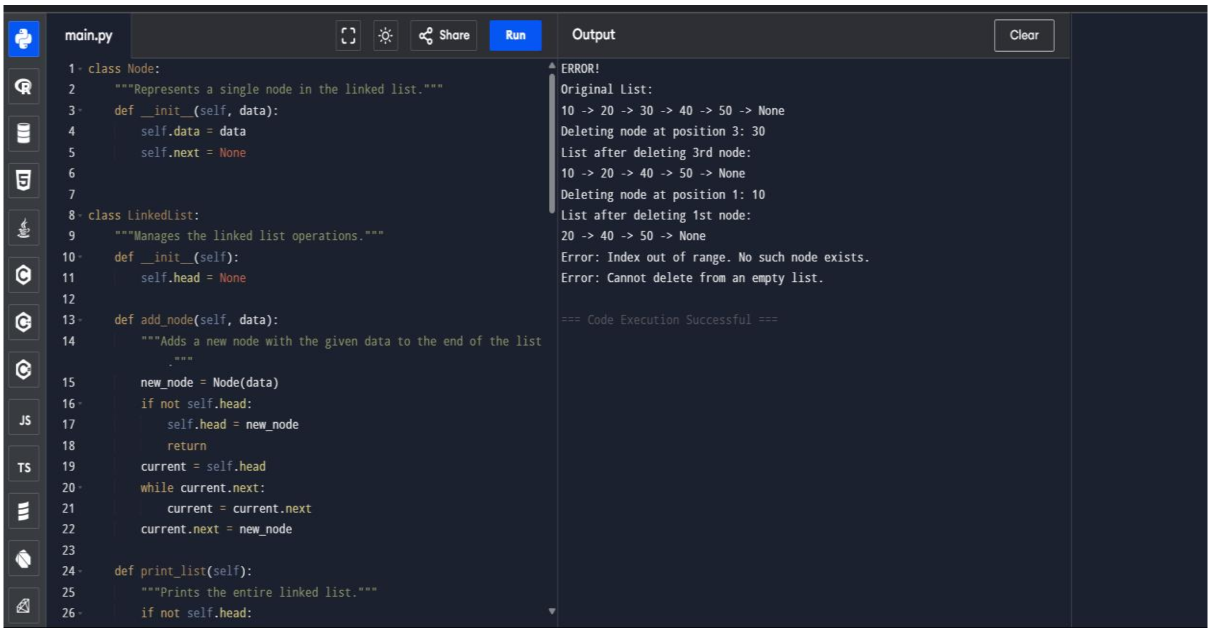
```
class Node:
    """Represents a single node in the linked list."""
    def __init__(self, data):
        self.data = data
        self.next = None



class LinkedList:
    """Manages the linked list operations."""
    def __init__(self):
        self.head = None

    def add_node(self, data):
        """Adds a new node with the given data to the end of the list."""
        new_node = Node(data)
        if not self.head:
            self.head = new_node
            return
        current = self.head
        while current.next:
            current = current.next
        current.next = new_node

    def print_list(self):
        """Prints the entire linked list."""
        if not self.head:
            print("List is empty.")
            return
        current = self.head
        while current:
            print(current.data, end=" -> ")
        current = current.next
    print("None")

def delete_nth_node(self, n):
    """Deletes the nth node (1-based index) from the linked list."""
    if not self.head:
        raise Exception("Cannot delete from an empty list.")
    if n <= 0:
        raise Exception("Invalid position. Index should be 1 or greater.")

    if n == 1:
        print(f"Deleting node at position {n}: {self.head.data}")
        self.head = self.head.next
        return

    current = self.head
    prev = None
    count = 1

    while current and count < n:
        prev = current
        current = current.next
        count += 1

    if not current:
        raise Exception("Index out of range. No such node exists.")

    print(f"Deleting node at position {n}: {current.data}")
    prev.next = current.nextif __name__ == "__main__":
ll = LinkedList()
```

```
# Add sample nodes
ll.add_node(10)
ll.add_node(20)
ll.add_node(30)
ll.add_node(40)
ll.add_node(50)

print("Original List:")
ll.print_list()

try:
    ll.delete_nth_node(3)  # Delete the 3rd node (30)
    print("List after deleting 3rd node:")
    ll.print_list()

    ll.delete_nth_node(1)  # Delete the 1st node (10)
    print("List after deleting 1st node:")
    ll.print_list()

    ll.delete_nth_node(10)  # Try to delete out-of-range node
except Exception as e:
    print("Error:", e)

try:
    empty_list = LinkedList()
    empty_list.delete_nth_node(1)  # Deleting from empty list
except Exception as e:
    print("Error:", e)
```



```python
class Node:
    """Represents a single node in the linked list."""
    def __init__(self, data):
        self.data = data
        self.next = None


class LinkedList:
    """Manages the linked list operations."""
    def __init__(self):
        self.head = None

    def add_node(self, data):
        """Adds a new node with the given data to the end of the list
        ."""
        new_node = Node(data)
        if not self.head:
            self.head = new_node
            return
        current = self.head
        while current.next:
            current = current.next
        current.next = new_node

    def print_list(self):
        """Prints the entire linked list."""
        if not self.head:
```

```
ERROR!
Original List:
10 -> 20 -> 30 -> 40 -> 50 -> None
Deleting node at position 3: 30
List after deleting 3rd node:
10 -> 20 -> 40 -> 50 -> None
Deleting node at position 1: 10
List after deleting 1st node:
20 -> 40 -> 50 -> None
Error: Index out of range. No such node exists.
Error: Cannot delete from an empty list.

=== Code Execution Successful ===
```