Reverse a Linked List

Create a class called Node with the following properties:

data: to store the data of the node
next: to store the reference to the next node in the linked list
Create a class called LinkedList with the following properties:

head: to store the reference to the head node of the linked list
Implement the insert method in the LinkedList class:

Create a new node with the given data
If the linked list is empty (head is None), set the new node as the head
Otherwise, traverse the linked list until the last node is reached
Set the next of the last node to the new node
Implement the reverse method in the LinkedList class:

Initialize two pointers, prev and current, to None and the head of the linked list, respectively
Iterate through the linked list using a while loop until the current pointer becomes None
Inside the loop:
Store the next node in a temporary variable (next_node) before modifying the next pointer of the current node
Set the next pointer of the current node to the previous node (prev)
Move prev to the current node and current to the next node
After the loop, update the head of the linked list to prev, which is now the new head
Implement the display method in the LinkedList class:

Start with the head of the linked list and traverse through each node
Print the data of each node
Repeat until reaching the end of the linked list (current node becomes None)

Create an instance of the LinkedList class called linked_list

Insert elements into the linked list using the insert method

Print "Original linked list"

Display the linked list using the display method

Reverse the linked list using the reverse method

Print "Reversed linked list"

Display the reversed linked list using the display method