Snehita · S
1BM18CS109

```
from collections import defaultdic
class Graph()
    def__init__(self):
        self.edges=defaultdic(list)
        self.weights={}
    def addEdge(self,from_node, to_node, weight):
        self.edges[from_node].append(to_node)
        self.weights[(from_node, to_node)] = weight
    def dijsktra(graph, initial, end):
        shortest_paths = { initial: (None,0)}
        current_nod = initial;
        while currentnode != end;
                visited.add (current_node)
                dest = graph.edges[current_node]
        for next in dest:
            weight = graph.weights[(current_node,next_node)] + weight to cur_node.
            if next not in shortest path:
                shortest_path[next] = (cur_node, weight)
            else
                cur_shortest_wght = shortest_paths[next_nod].
        next_dest = {node: shortest_path_node] for node in shortest paths
                    if node not in visited}.

        path =[].
        while cur_node not None:
                path.append (cur_node)
                next = shortest_paths[cur_node][0]
                cur_node = next.
        path = path[::-1]
        print (path).
```