Q1: A linked list is a collection of nodes, where each node contains data and a reference to the next node. It’s different from an array because it doesn’t require contiguous memory allocation, and elements can be dynamically added or removed.
Q2: A singly linked list is a structure where each node points to the next node, and the last node points to null. This list can only be traversed in one direction.
Q3: A doubly linked list has nodes that point to both the next and previous nodes, allowing traversal in both directions. Unlike a singly linked list, which only allows forward traversal, a doubly linked list is more flexible.
Q4: A circular linked list is a variation of a linked list where the last node points back to the first node, forming a loop. There are no null references in this structure.
Q5: A sentinel node is a placeholder node that simplifies boundary conditions like handling empty lists or inserting nodes at the beginning or end of a list. It simplifies operations on linked lists by avoiding special cases for empty lists.