APIwiz Assignment - Approach Document

- Given a **DAG (Directed Acyclic Graph)** representing a workflow,
- Traverse from the root node (`Node-1`) such that:
    - A node is executed **only after all its parent nodes** have executed.
    - If a node has **multiple children**, they should be **executed in parallel**.

⚙ Tools & Technologies

- **Python**
- **Libraries**

    collections.defaultdict – For adjacency list & in-degree tracking
    threading.Thread – To simulate parallel execution

Design &Implementation:

Read node count N, then parse N lines to build vertexMap:Map<Integer,String>.

Read edge count M then parse M lines to form edgeSet:List<Pair<Integer,Integer>>

Execution Logic

- Start traversal from **Node 1**.
- For each node:
    - **Print its name.**
    - For each child:
        - Decrement its in-degree.
        - If in-degree is now 0, execute it in a **new thread**.
- Use Thread.join() to ensure all child executions finish before the parent finishes.

Output:

Node names printed in order of valid execution.

Final line prints total number of unique node.