Problem: Identify Isolated Nodes in a Graph

Description:

Write a function `identify_isolated_nodes(graph)` that takes an adjacency list representation of a graph and returns a list of isolated nodes. An isolated node is a node that has no connections (its adjacency list is empty).

Function Signature:

```
def identify_isolated_nodes(graph: dict) -> list:
```

Input:

- `graph` (dict): A dictionary where each key is a vertex, and its corresponding value is a list of adjacent vertices (the adjacency list representation).

Output:

- A list of nodes that are isolated. If there are no isolated nodes, return an empty list.

Example:

```
identify_isolated_nodes({"A": ["B"], "B": ["A"], "C": []})  # Output: ["C"]
identify_isolated_nodes({"A": ["B"], "B": ["A"], "C": ["D"], "D": ["C"]})  # Output: []
```

Constraints:

- The graph will have at least one vertex.

Notes:

- Iterate through each vertex and check if its adjacency list is empty to determine if it is isolated.