***PROJECT***

***ON***

***Basic Social Network Analysis Tool***

**Overview:**

This document outlines a basic social network analysis tool implemented in Python. The tool focuses on graph representation, degree centrality calculation, and basic visualization.?

**Code Structure:**

Python:

import networkx as nx

# Create a new directed graph

G = nx.DiGraph()

# Add nodes

G.add\_node("chandana")

G.add\_node("venky")

G.add\_node("sreeram")

G.add\_node("sushma")

# Add edges

G.add\_edge("chandana", "venky")

G.add\_edge("venky", "sreeram")

G.add\_edge("sreeram", "sushma")

G.add\_edge("sushma", "chandana")

G.add\_edge("venky", "chandana")

in\_degree\_centrality = nx.in\_degree\_centrality(G)

print("In-degree Centrality:")

for node, centrality in in\_degree\_centrality.items():

print(f"{node}: {centrality:2f}")

out\_degree\_centrality = nx.out\_degree\_centrality(G)

print("\nOut-degree Centrality:")

for node, centrality in out\_degree\_centrality.items():

print(f"{node}: {centrality:2f}")

UG = G.to\_undirected()

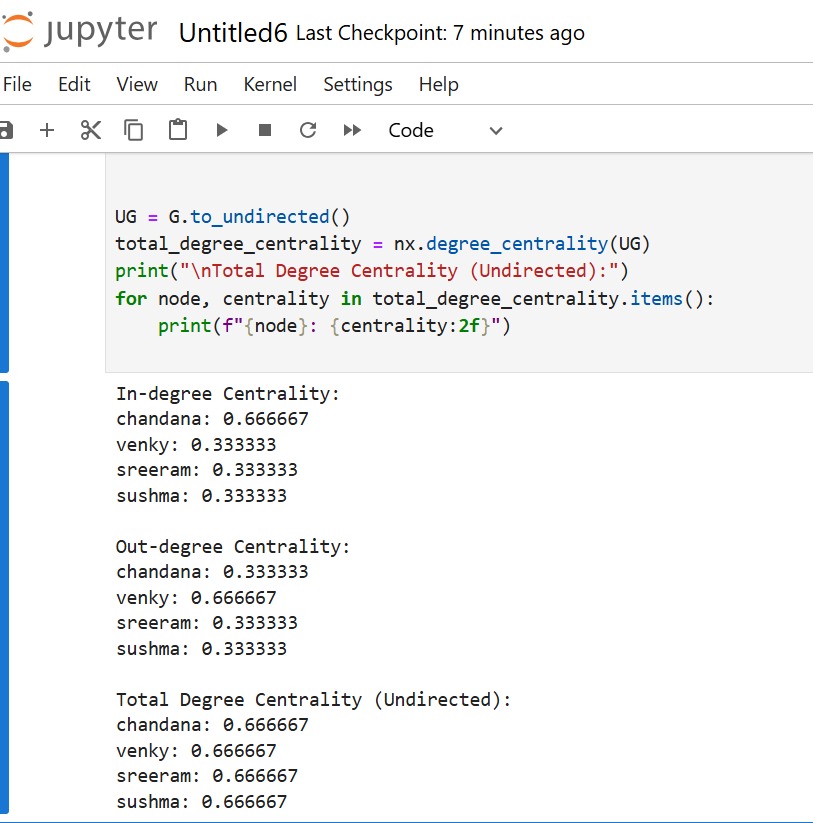
total\_degree\_centrality = nx.degree\_centrality(UG)

print("\nTotal Degree Centrality (Undirected):")

for node, centrality in total\_degree\_centrality.items():

print(f"{node}: {centrality:2f}")

**OUTPUT:-**



EXPLANTION OF THE PROJECT : -

This project calculates and prints the centrality measures of a directed graph. Centrality measures indicate the importance of nodes in a graph.

Here's a breakdown of the project:

1. Directed Graph Creation: A new directed graph G is created using NetworkX.

2. Node Addition: Four nodes - "chandana", "venky", "sreeram", and "sushma" - are added to the graph.

3. Edge Addition: Five directed edges are added between the nodes:

- "chandana" to "venky"

- "venky" to "sreeram"

- "sreeram" to "sushma"

- "sushma" to "chandana"

- "venky" to "chandana"

4. In-Degree Centrality Calculation: The in-degree centrality of each node is calculated using nx.in\_degree\_centrality(G). In-degree centrality measures the number of edges pointing to a node.

5. Out-Degree Centrality Calculation: The out-degree centrality of each node is calculated using nx.out\_degree\_centrality(G). Out-degree centrality measures the number of edges pointing from a node.

6. Undirected Graph Conversion: The directed graph G is converted to an undirected graph UG using G.to\_undirected().

7. Total Degree Centrality Calculation: The total degree centrality of each node in the undirected graph is calculated using nx.degree\_centrality(UG). Total degree centrality measures the total number of edges connected to a node**.**

8. Printing Centrality Measures: The centrality measures (in-degree, out-degree, and total degree) are printed for each node.