Mini Social networks analysis tool

It is a desktop software for network analysis and visualization. It is a tool used by data analysts and researchers to explore graphs of different networks. It enables users to interact with the graphs in many ways like coloring and adjusting nodes and edges based on calculated metrics or existing features, filtering the graphs based on any criteria, applying different community detection methods, link prediction techniques, and detecting influential users.

Your task is to implement such an interactive system using any programming language you like (we recommend using Python (NetworkX)). It is a GUI desktop application, but you can implement it as a web application.

Requirements:

1- **Node and Edge Attributes:** Allow users to define and visualize custom node and edge attributes, such as size, color, label, and shape.
2- **Layout Algorithms:** Implement various layout algorithms to visualize network structures differently. This could include force-directed algorithms (e.g., Fruchterman-Reingold), and hierarchical layouts (e.g., tree or radial layouts).
3- **Graph Metrics and Statistics**: Integrate a wide range of graph metrics and statistics to analyze network properties and characteristics. This could include basic metrics such as degree distribution, clustering coefficient, and average path length.
4- **Filtering Options**: the filtering capabilities to include:
    I. Filtering nodes based on centrality measures (use at least three centrality measures).
    II. filtering nodes based on their membership in specific communities or their centrality scores falling within certain ranges.
5- **Community Detection Comparison**: compare the results of different community detection algorithms (e.g., Girvan Newman and Louvain algorithm) side by side. This could involve displaying metrics such as the number of communities detected, modularity scores, or other evaluation criteria for each algorithm.
6- **Graph Partitioning and Clustering**: Implement algorithms for partitioning the graph into clusters or communities based on various criteria.
7- **Clustering Evaluation**: Apply at least 3 community detection evaluations (internal and external evaluation)

8- **Basic Visualization**: Basic network structure visualization.
9- **Link Analysis Techniques**: Implement various link analysis techniques to analyze the relationships between nodes in the network. This could include algorithms such as PageRank, and Betweenness Centrality. These techniques can help identify important nodes and relationships within the network.

For loading the network, two CSV files will be uploaded nodes file and the edges file.

You should also handle both directed and undirected graphs.