import pandas as pd  
import networkx as nx  
import matplotlib.pyplot as plt  
  
# Sample dataframe  
# Replace this with your actual data  
journey\_data = {  
 "Journey name": [  
 "Connectivity less",  
 "Dashboard",  
 "Deal Hub",  
 "Delete EE ID",  
 "For You",  
 "upgrade hub",  
 "view latest device info",  
 "wifi 7 learn page",  
 "your bills",  
 "£10 off protective kit",  
 ],  
 "Full path": [  
 "about you->exit modal->about you->EE 5.52.1",  
 "Your Wifi->BillingAndPaymentsRoute->Your Wifi",  
 "Deal Hub Home->your add ons->WeeklyDiscountsRoute",  
 "Manage->Springboard->Manage->Springboard",  
 "UpgradeProductsRoute->UpgradeProducts",  
 "Upgrade Hub->DeviceInfoRoute->Billing",  
 "DeviceInfo->LearnMore->LatestDeviceInfo",  
 "WiFi7->LearnMore->WiFi7LearnPage",  
 "Bills->PaymentsRoute->ViewBills",  
 "ProtectiveKit->Offers->ApplyDiscount",  
 ],  
}  
  
df = pd.DataFrame(journey\_data)  
  
# Clear any existing plots  
plt.close("all")  
  
# Preparing edge data  
edges = []  
for path in df["Full path"]:  
 nodes = path.split("->")  
 edges += [(nodes[i], nodes[i + 1]) for i in range(len(nodes) - 1)]  
  
# Counting the weights of each transition  
edge\_weights = pd.DataFrame(edges, columns=["source", "target"]).value\_counts().reset\_index()  
edge\_weights.columns = ["source", "target", "weight"]  
  
# Creating the graph  
G = nx.DiGraph()  
for \_, row in edge\_weights.iterrows():  
 G.add\_edge(row["source"], row["target"], weight=row["weight"])  
  
# Positioning nodes for visualization  
pos = nx.spring\_layout(G)  
  
# Drawing the graph  
plt.figure(figsize=(12, 12))  
nx.draw\_networkx(  
 G,  
 pos,  
 with\_labels=True,  
 node\_size=500,  
 node\_color="lightblue",  
 font\_size=10,  
 edge\_color="#888",  
 arrowsize=15,  
 font\_weight="bold",  
)  
  
# Adding edge labels for weights  
edge\_labels = {edge: G.edges[edge]['weight'] for edge in G.edges}  
nx.draw\_networkx\_edge\_labels(G, pos, edge\_labels=edge\_labels, font\_size=8)  
  
# Title and display  
plt.title("Directional Network Graph of Page Transitions", fontsize=16)  
plt.show()