|  |  |
| --- | --- |
| **Name** | Sahil Mehta |
| **UID no.** | 2023300143 |
| **Experiment No.** | 1 (Home Assignment) |

|  |  |
| --- | --- |
| **AIM:** | To develop a user-friendly Streamlit-based network utility application that enables users to execute essential network diagnostic commands, such as Ping, NSLookup, Ifconfig, Netstat, and Traceroute |
| **Program 1** | |
| **PROBLEM STATEMENT:** | Develop an application which allows users to access essential network utility commands without opening terminal |
| **PROGRAM:** | import streamlit as st  import subprocess  import platform  def ping\_website(website):      try:          result = subprocess.run(              (                  ["ping", "-n", "4", website]                  if platform.system() == "Windows"                  else ["ping", "-c", "4", website]              ),              stdout=subprocess.PIPE,              stderr=subprocess.PIPE,              text=True,          )          if result.returncode == 0:              return result.stdout          else:              return f"Error: Unable to reach {website}. Please check the website address or your internet connection."      except Exception as e:          return f"An error occurred: {str(e)}"  def nslookup\_website(website):      try:          result = subprocess.run(              ["nslookup", website],              stdout=subprocess.PIPE,              stderr=subprocess.PIPE,              text=True,          )          if result.returncode == 0:              return result.stdout          else:              return f"Error: Unable to resolve {website}. Please check the website address."      except Exception as e:          return f"An error occurred: {str(e)}"  def ifconfig\_info():      try:          command = (              ["ipconfig"] if platform.system() == "Windows" else ["ifconfig"]          )          result = subprocess.run(              command, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True          )          if result.returncode == 0:              return result.stdout          else:              return f"Error: Unable to retrieve network information."      except Exception as e:          return f"An error occurred: {str(e)}"  def netstat\_info():      try:          command = (              ["netstat", "-a"]              if platform.system() == "Windows"              else ["netstat", "-an"]          )          result = subprocess.run(              command, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True          )          if result.returncode == 0:              return result.stdout          else:              return f"Error: {result.stderr}"      except Exception as e:          return f"An error occurred: {str(e)}"  def traceroute\_info(website):      try:          command = (              ["tracert", website]              if platform.system() == "Windows"              else ["traceroute", website]          )          result = subprocess.run(              command, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True          )          if result.returncode == 0:              return result.stdout          else:              return f"Error: {result.stderr}"      except Exception as e:          return f"An error occurred: {str(e)}"  st.markdown(      """      <style>      .stApp {          background-color: #1e1e1e;          color: #ffffff;      }      .stButton button {          background-color: #4CAF50;          color: white;          border-radius: 5px;          padding: 10px 20px;          font-size: 16px;      }      .stButton button:hover {          background-color: #4CAF50;      }      .stSelectbox div {          font-size: 18px;      }      .stTextInput input {          font-size: 16px;          background-color: #2d2d2d;          color: #ffffff;      }      .stTextArea textarea {          font-family: monospace;          font-size: 14px;          background-color: #2d2d2d;          color: #ffffff;      }      .stMarkdown h1 {          color: #4CAF50;      }      .stMarkdown h2 {          color: #ffffff;      }      .stSidebar {          background-color: #2d2d2d;      }      .stSidebar .stMarkdown h1 {          color: #4CAF50;      }      .footer {          text-align: center;          padding: 10px;          background-color: #2d2d2d;          color: #ffffff;      }  </style>  """,      unsafe\_allow\_html=True,  )  st.title("🌐 Network Utility Tool")  st.markdown("\*\*Select any command to analyze network details.\*\*")  with st.sidebar:      st.header("⚙️ Command Selection")      command = st.selectbox(          "Choose a Command",          [              "Ping (Check if a website is reachable and measure response time)",              "NSLookup (Find the IP address of a website or domain)",              "Ifconfig (Display your device's network settings)",              "Netstat (Show active network connections)",              "Traceroute (Show the path data takes to reach a website)",          ],      )  if "Ifconfig" in command:      st.write("🔍 Retrieving network details...")      with st.spinner("Fetching network information..."):          output = ifconfig\_info()      st.text\_area("📄 Ifconfig Output", output, height=300)  elif "Netstat" in command:      st.write("🔍 Retrieving network connection details...")      with st.spinner("Fetching connection details..."):          output = netstat\_info()      st.text\_area("📄 Netstat Output", output, height=300)  else:      with st.form("website\_form"):          website = st.text\_input("🌍 Enter Website Name")          submit\_button = st.form\_submit\_button("🚀 Execute")      if submit\_button:          if website:              with st.spinner(                  f"Executing {command.split('(')[0]} on '{website}'..."              ):                  if "Ping" in command:                      output = ping\_website(website)                  elif "NSLookup" in command:                      output = nslookup\_website(website)                  elif "Traceroute" in command:                      output = traceroute\_info(website)              st.text\_area(                  f"📄 {command.split('(')[0]} Output", output, height=300              )          else:              st.warning("⚠️ Please enter a website name.")  st.markdown("---") |
| **RESULT:**   1. Ping      1. NSLookup      1. Ifconfig      1. Netstat      1. Traceroute | |
| **CONCLUSION:** | Through this project, I gained a deeper understanding of building a network utility application using Streamlit. I explored how to use Python's subprocess module to execute system commands like Ping, NSLookup, Ifconfig, Netstat, and Traceroute, ensuring compatibility for Windows. This experience also helped me improve my skills in designing an interactive and user-friendly interface with Streamlit, incorporating features like dropdown menus, text inputs, and dynamic output displays. Working on this project not only enhanced my knowledge of network diagnostics but also demonstrated how to seamlessly integrate backend functionality with a clean and responsive frontend for real-time data visualization. |