



# Project 4

## SNMP-Based Network Management Tool with Python

ITCS465 NETWORK MANAGEMENT

NAME:

Sirapath  
Suphavadee  
Jidapa

Thainiyom  
Cheng  
Moolkaew

ID:

6488108  
6488120  
6488176



# Project Overview

The project is an SNMP Manager Tool designed to interact with SNMP-enabled devices on a network.

The tool provides the ability to perform SNMP GetNext and GetBulk operations to retrieve SNMP data from devices.



# About the SNMP Management Tool

To create an interactive SNMP management tool to fetch and display network device information.

## Key Features

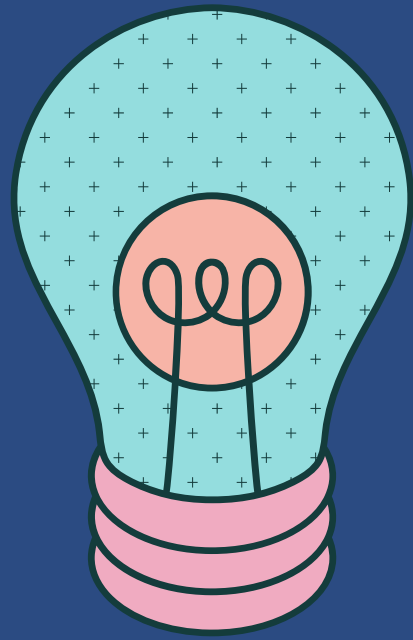
Browse OIDs using a tree structure

Perform SNMP operations

Display fetched data in a table

Pagination for bulk requests

MIB Project 4				
OID Tree				
System				
IP				
UDP				
Target: 127.0.0.1 Community: public OID: 1.3.6.1.2.1.1 Get Bulk Go Clear Table				
Name/OID Value Type IP:Port				
0 (1.3.6.1.2.1.1.1.0) Hardware: AMD64 Family 25 Model 8 DisplayString 127.0.0.1:161				
0 (1.3.6.1.2.1.1.2.0) 1.3.6.1.4.1.311.1.1.3.1.1 ObjectIdentity 127.0.0.1:161				
0 (1.3.6.1.2.1.1.3.0) 36582676 TimeTicks 127.0.0.1:161				
0 (1.3.6.1.2.1.1.4.0) DisplayString 127.0.0.1:161				
0 (1.3.6.1.2.1.1.5.0) Annie DisplayString 127.0.0.1:161				
0 (1.3.6.1.2.1.1.6.0) DisplayString 127.0.0.1:161				
0 (1.3.6.1.2.1.1.7.0) 76 Integer32 127.0.0.1:161				
0 (1.3.6.1.2.1.2.1.0) 66 Integer 127.0.0.1:161				
1 (1.3.6.1.2.1.2.2.1.1.1) 1 Integer 127.0.0.1:161				
2 (1.3.6.1.2.1.2.2.1.1.2) 2 Integer 127.0.0.1:161				



# System Architecture

## Tools Used

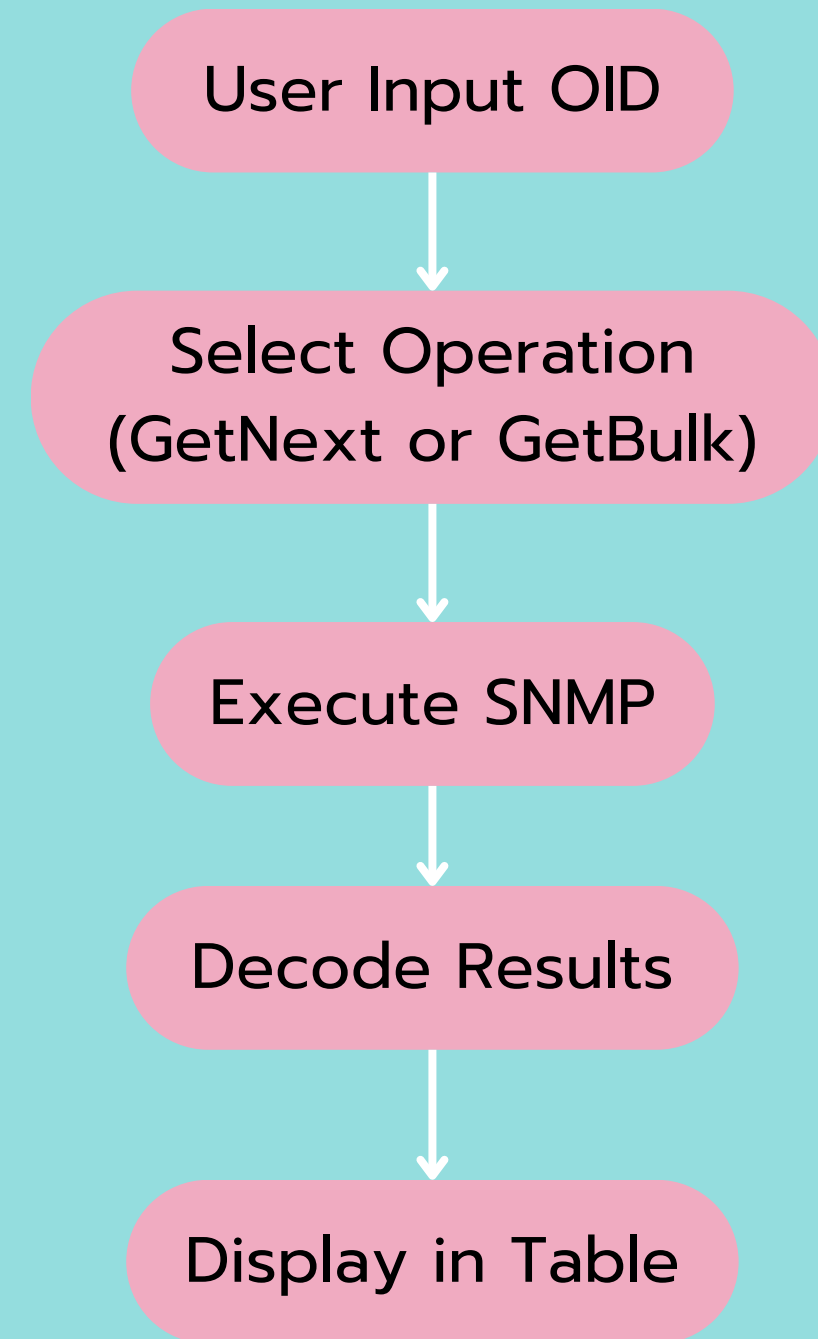
Python Programming Language

- PySNMP Library: A Python library for SNMP operations.
- Tkinter Library: Python's standard GUI toolkit

## Modules

- GUI built with Tkinter (TreeView, Table, Input Fields).
- Backend SNMP operations using PySNMP (snmp\_get\_next, snmp\_get\_bulk).

## Workflow of the SNMP Tool



# Implementation





# SNMP Operations

Used PySNMP's **nextCmd** for GetNext and **bulkCmd** for GetBulk.  
Properly decoded SNMP values (IP, bytes, strings).

## Get Next Function

`snmp_get_next()`

- Fetches the next object in lexicographical order from the MIB.

## Get Bulk Function

`snmp_get_bulk()`

- Fetches multiple objects in one request.
- Useful for retrieving large datasets.

**Error Handling:** Checks for SNMP errors like timeout, access issues, or invalid OIDs.



# Graphical User Interface (GUI)



MIB Project 4

OID Tree

System

sysDescr  
sysObjectID  
sysUpTime  
sysContact  
sysName  
sysLocation  
sysServices

IP

ipForwarding  
ipDefaultTTL  
ipInReceives  
ipInHdrErrors  
ipInAddrErrors  
ipForwDatagrams  
ipInUnknownProts  
ipInDiscards  
ipInDelivers  
ipOutRequests  
ipOutDiscards  
ipOutNoRoutes  
ipReasmTimeout  
ipRouteTable  
ipRouteDest  
ipRouteIfIndex  
ipRouteMetric1  
ipRouteNextHop  
ipRouteType  
ipRouteProto  
ipRouteAge  
ipRouteMask

UDP

udpInDatagrams  
udpNoPorts  
udpInErrors  
udpOutDatagrams  
udpTable

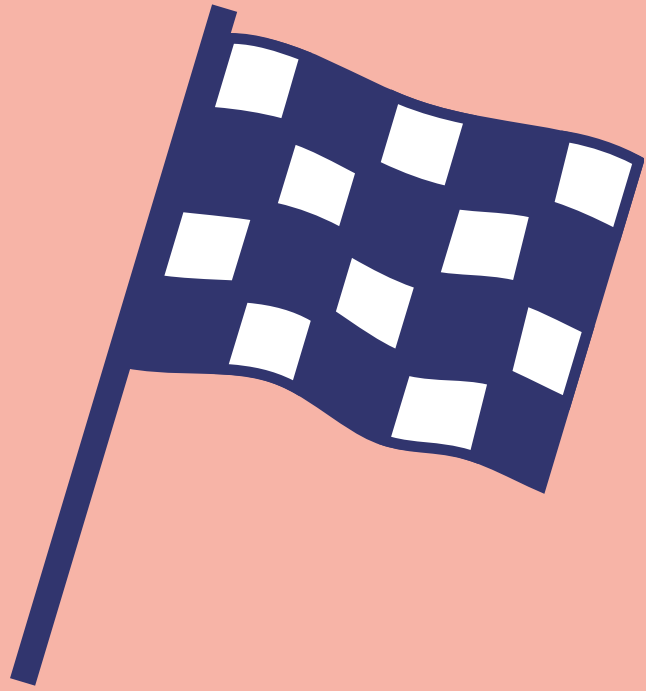
Target: 127.0.0.1 Community: public OID: 1.3.6.1.2.1.1 Get Bulk Go Clear Table

Name/OID	Value	Type	IP:Port
0 (1.3.6.1.2.1.1.1.0)	Hardware: AMD64 Family 25 Model 80 Stepping	DisplayString	127.0.0.1:161
0 (1.3.6.1.2.1.1.2.0)	1.3.6.1.4.1.311.1.1.3.1.1	ObjectIdentity	127.0.0.1:161
0 (1.3.6.1.2.1.1.3.0)	37216351	TimeTicks	127.0.0.1:161
0 (1.3.6.1.2.1.1.4.0)		DisplayString	127.0.0.1:161
0 (1.3.6.1.2.1.1.5.0)	Annie	DisplayString	127.0.0.1:161
0 (1.3.6.1.2.1.1.6.0)		DisplayString	127.0.0.1:161
0 (1.3.6.1.2.1.1.7.0)	76	Integer32	127.0.0.1:161
0 (1.3.6.1.2.1.2.1.0)	66	Integer	127.0.0.1:161
1 (1.3.6.1.2.1.2.2.1.1.1)	1	Integer	127.0.0.1:161
2 (1.3.6.1.2.1.2.2.1.1.2)	2	Integer	127.0.0.1:161
3 (1.3.6.1.2.1.2.2.1.1.3)	3	Integer	127.0.0.1:161
4 (1.3.6.1.2.1.2.2.1.1.4)	4	Integer	127.0.0.1:161
5 (1.3.6.1.2.1.2.2.1.1.5)	5	Integer	127.0.0.1:161
6 (1.3.6.1.2.1.2.2.1.1.6)	6	Integer	127.0.0.1:161
7 (1.3.6.1.2.1.2.2.1.1.7)	7	Integer	127.0.0.1:161
8 (1.3.6.1.2.1.2.2.1.1.8)	8	Integer	127.0.0.1:161
9 (1.3.6.1.2.1.2.2.1.1.9)	9	Integer	127.0.0.1:161
10 (1.3.6.1.2.1.2.2.1.1.10)	10	Integer	127.0.0.1:161
11 (1.3.6.1.2.1.2.2.1.1.11)	11	Integer	127.0.0.1:161
12 (1.3.6.1.2.1.2.2.1.1.12)	12	Integer	127.0.0.1:161
13 (1.3.6.1.2.1.2.2.1.1.13)	13	Integer	127.0.0.1:161
14 (1.3.6.1.2.1.2.2.1.1.14)	14	Integer	127.0.0.1:161
15 (1.3.6.1.2.1.2.2.1.1.15)	15	Integer	127.0.0.1:161
16 (1.3.6.1.2.1.2.2.1.1.16)	16	Integer	127.0.0.1:161
17 (1.3.6.1.2.1.2.2.1.1.17)	17	Integer	127.0.0.1:161
18 (1.3.6.1.2.1.2.2.1.1.18)	18	Integer	127.0.0.1:161
19 (1.3.6.1.2.1.2.2.1.1.19)	19	Integer	127.0.0.1:161
20 (1.3.6.1.2.1.2.2.1.1.20)	20	Integer	127.0.0.1:161
21 (1.3.6.1.2.1.2.2.1.1.21)	21	Integer	127.0.0.1:161
22 (1.3.6.1.2.1.2.2.1.1.22)	22	Integer	127.0.0.1:161

- OID Tree Navigation.
- Input fields for SNMP parameters (Target, Community, OID).
- Dropdown for selecting SNMP operation (GetNext or GetBulk).
- Results displayed in a TreeView table

**DEMO**





# Challenges and Solutions

HOW CHALLENGES WERE ADDRESSED?

**Decoding SNMP data types (e.g., IP address, byte strings).**



Implemented specific decoding for IPAddress and byte data types.

**Managing OID pagination with GetBulk.**

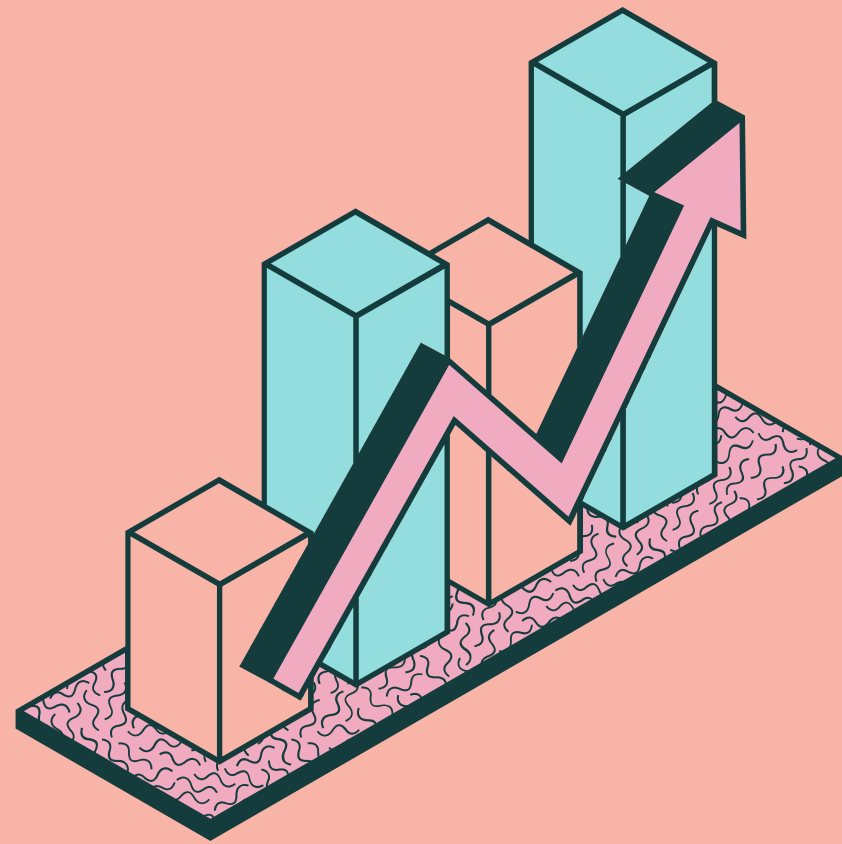


Used max\_repetitions to limit fetched records and ensure responsiveness.

**Handling SNMP errors (invalid OIDs, timeouts).**



Added error handling and user-friendly alerts.



# Future Enhancements

POTENTIAL IMPROVEMENTS

## Add SNMPv3 Support

Implemented specific decoding for IPAddress and byte data types.

## Data Visualization

Include graphs and charts for better insights.

## Integrate SNMP Traps

Allow real-time alerts for network events.

## Export Results

Allow exporting fetched data to CSV or Excel.

# Conclusion



## What Was Achieved

- Successfully built an SNMP management tool with Python.
- Simplified the process of fetching and visualizing SNMP data.

**“Tools like this are crucial for network monitoring and troubleshooting.”**

**THANK YOU**