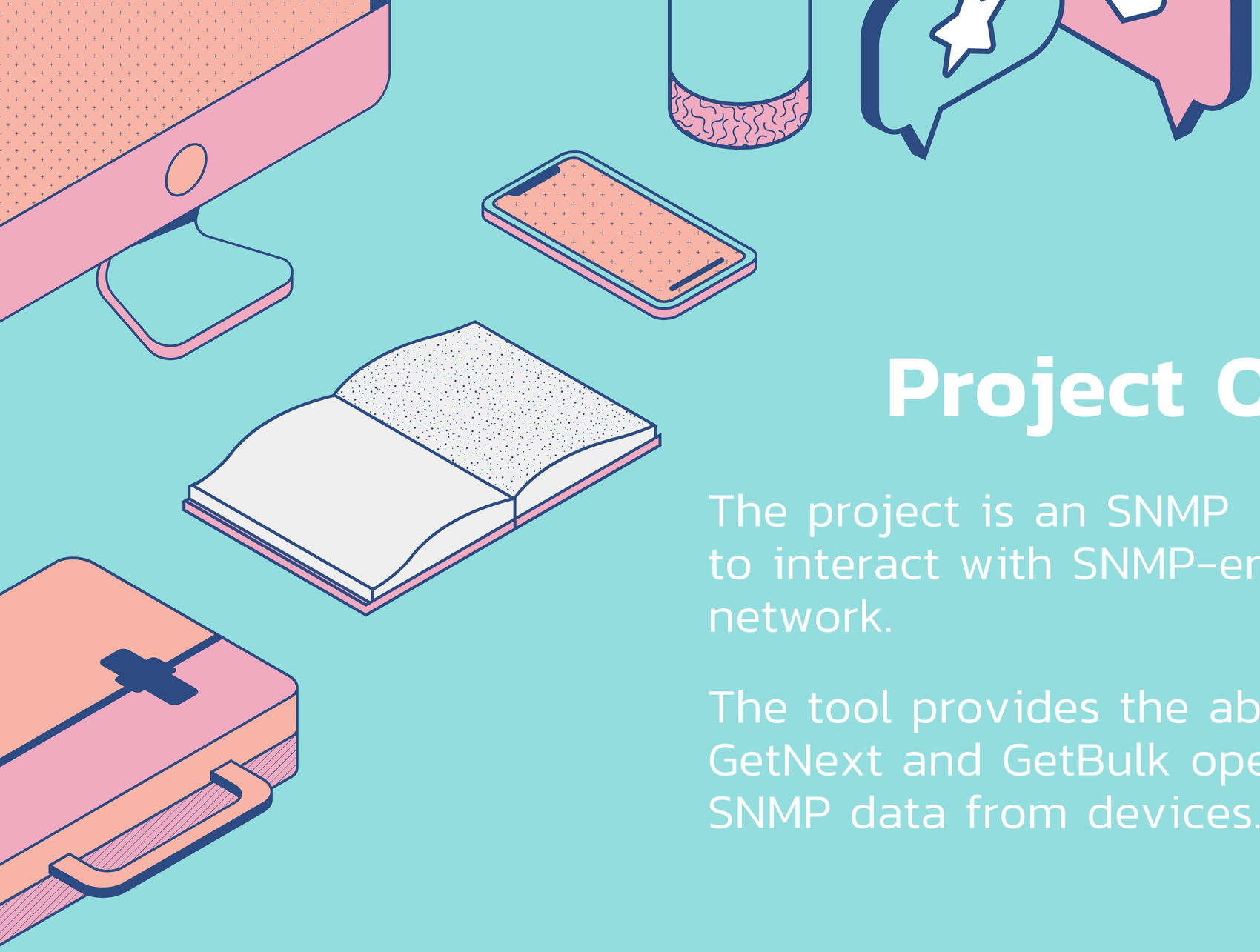
Project 4

SNMP-Based Network Management Tool with Python

ITCS465 NETWORK MANAGEMENT

NAME:		ID:
Sirapath	Thainiyom	6488108
Suphavadee	Cheng	6488120
Jidapa	Moolkaew	6488176



Project Overview

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

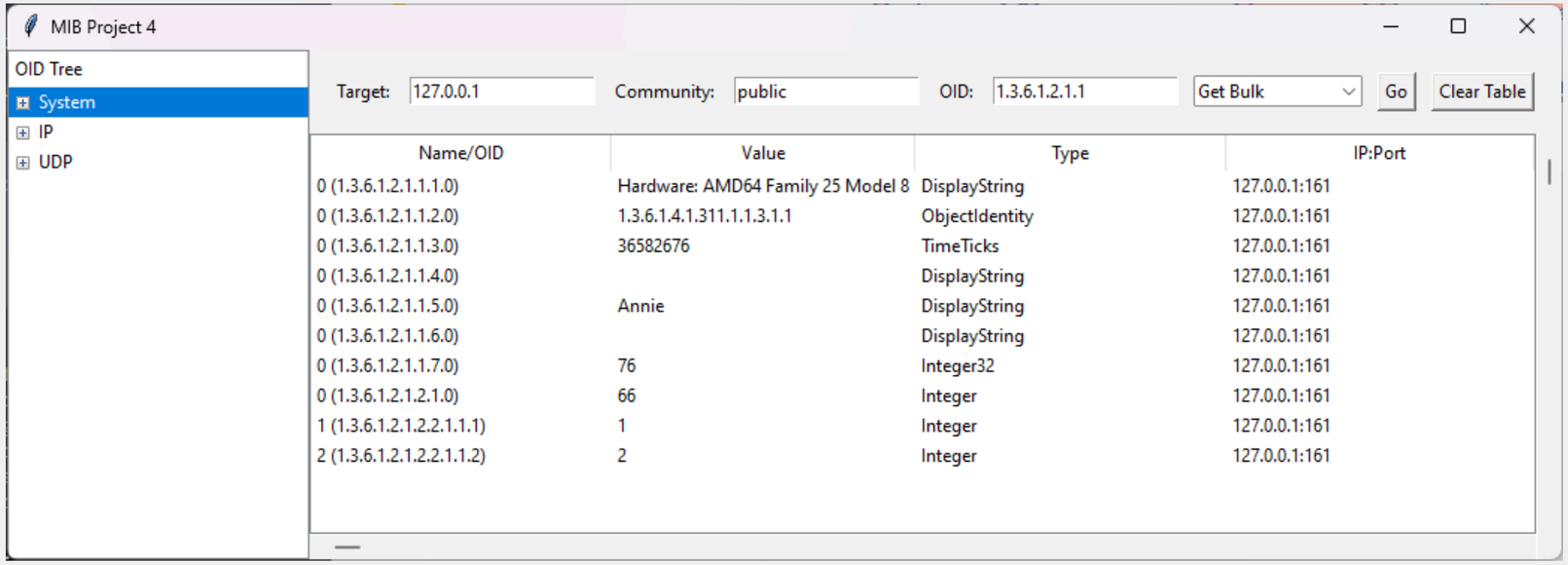
The tool provides the ability to perform SNMP GetNext and GetBulk operations to retrieve SNMP data from devices.

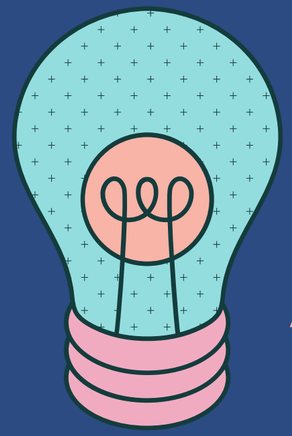
About the SNMP Management Tool

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





System Architecture

Tools Used

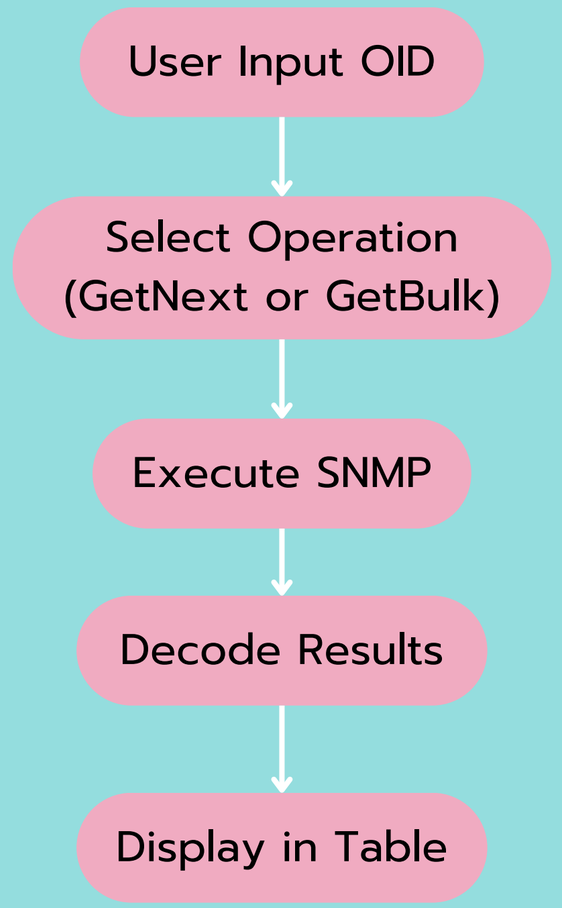
Python Programming Language

- 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.

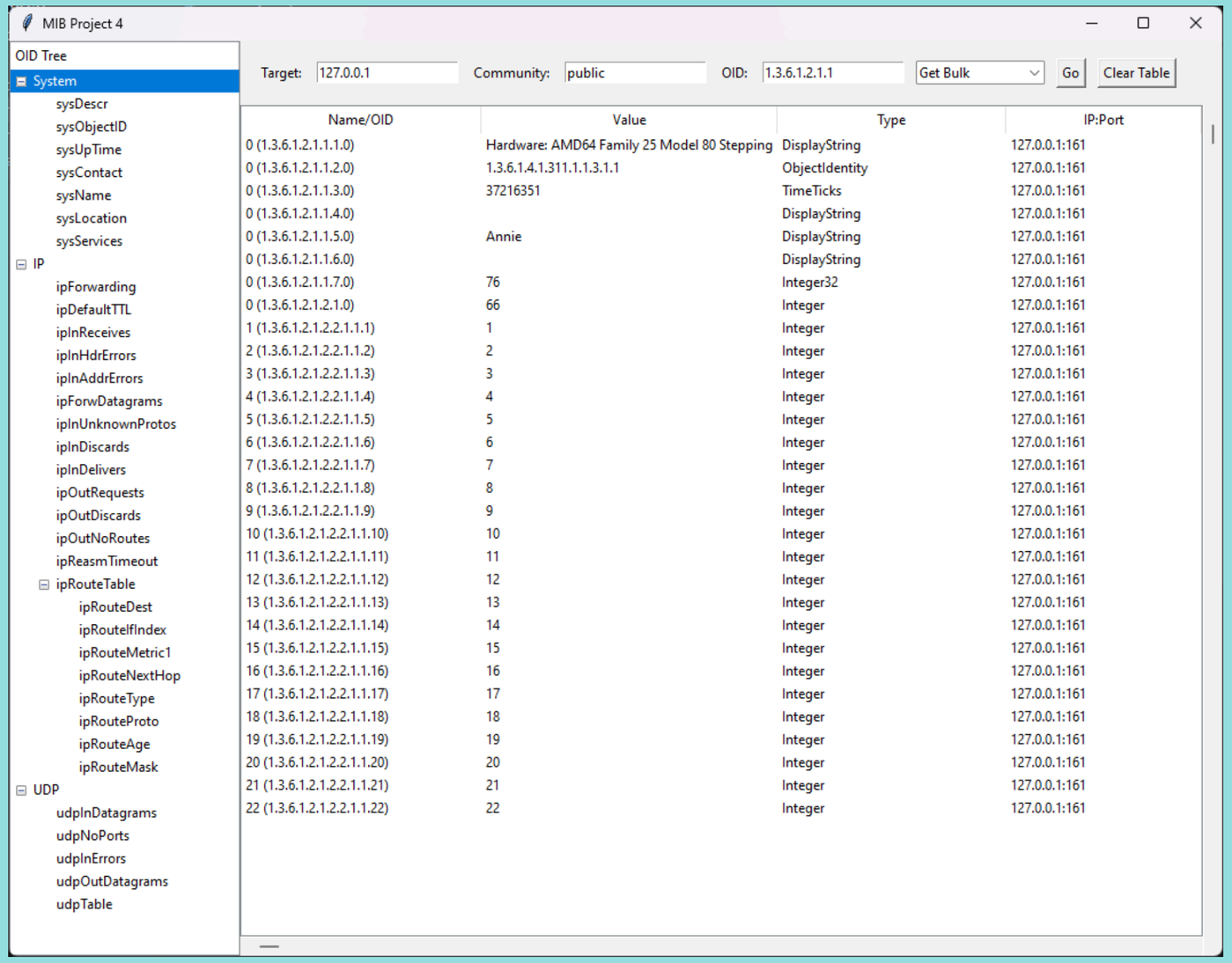
Get Bulk Function

snmp_get_bulk()

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

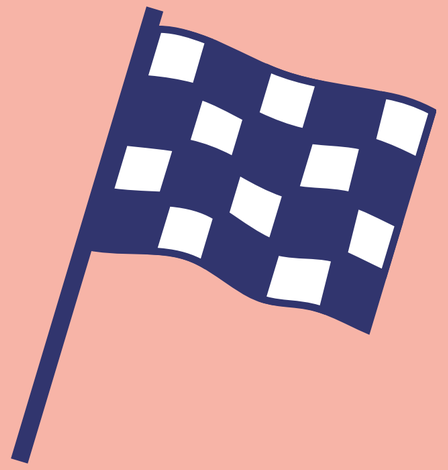
Error Handling: Checks for SNMP errors like timeout, access issues, or invalid OIDs.

Graphical User Interface (GUI)



- 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).

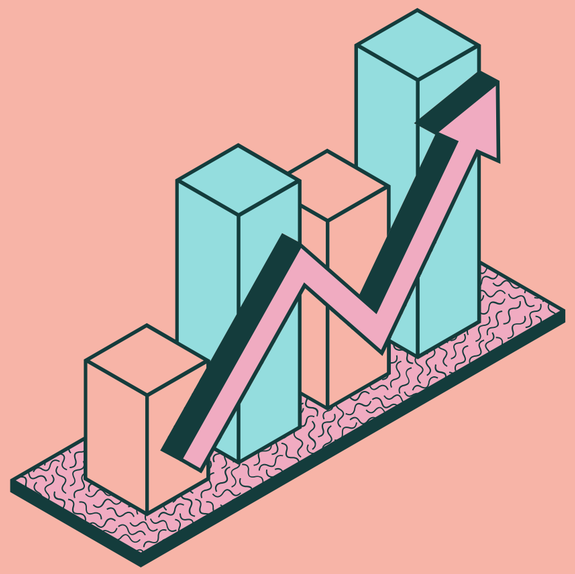
Implemented specific decoding for IPAddress and byte data types.

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.

Integrate SNMP Traps

Allow real-time alerts for network events.

Data Visualization

Include graphs and charts for better insights.

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