Please complete the following to the best of your ability. Feel free to use other sources for guidance, but make sure to **only** use your **own** code. You are permitted to use a dll library such as *SnmpSharpNet* for SNMP operations.

You will need an appropriate version of Visual Studio (Get the free community edition if your copy won’t be sufficient), a local copy of MSSQL Express or MySQL. You will also need to set up an SNMP service on your local machine (or somewhere else), with read and write permission. Follow these instructions: <https://kb.paessler.com/en/topic/663-how-do-i-install-the-snmp-service-on-windows-systems> *(Please remember to set community strings!)*

You are free to use your own creativity, as long as you stick to the following requirements. Please don’t make it fancy - Keep it simple!

* Create a **Visual C# WPF** Application
* As part of the GUI, include any control to be used as a **log viewer** which displays all information being transferred
* As part of the GUI, **display all the information** asked for below. Please process and format the info first (Ex. we don’t want to see a full JSON response, just the required value).
* Create **two timers** with different intervals: **timer 1 - 30 seconds, timer 2 – 60 seconds**. You will use this to gather and update information on the specified frequency
* **Get** information from the following SNMP OIDs from your local SNMP server port 161, using **timer 1**:

|  |  |  |
| --- | --- | --- |
| Item | OID | Type |
| sysName | .1.3.6.1.2.1.1.5.0 | OctetString |
| sysLocation | .1.3.6.1.2.1.1.6.0 | OctetString |
| sysUptime | .1.3.6.1.2.1.1.3.0 | TimeTicks |
| Interface Type | .1.3.6.1.2.1.2.2.1.3.1 | Integer – Please see appendix A for enum |
| sysServices | .1.3.6.1.2.1.1.7.0 | Integer |

* As part of your GUI, include a **control to *Set*** a string to the following OID:
  + sysLocation: .1.3.6.1.2.1.1.6.0
* **Get** the following information from the local machine using **WMI** performance counters using **timer 1**.
  + CPU usage
  + Memory Usage
  + Disk usage
* **Get** the current time from the **SOAP** service using **timer 1**: http://webservice.theknot.com/Time/GetTime.asmx
* **Get** the local machine’s internet IP address from using **timer 2**:
  + https://api.ipify.org?format=json
* **Create sql file(s).** Use your own creativity when designing the database and tables for the storage of all the info
* As part of your WPF app, **store all of the information** above which were polled by the timers. Include **timestamps** of when they were polled in the database. Use the respective timer intervals to store the info.
* Create a new project – **Visual C# ASP.Net Web application**
  1. **Show latest values** of all the information above, obtained from the database
  2. Be able to show **historical data** per variable

When submitting, please make sure you attach all source code, sql files and anything we need to run your submission on our side.

**Appendix A – SNMP Enum for Interface Type**

{

other(1),

regular1822(2),

hdh1822(3),

ddn-x25(4),

rfc877-x25(5),

ethernet-csmacd(6),

iso88023-csmacd(7),

iso88024-tokenBus(8),

iso88025-tokenRing(9),

iso88026-man(10),

starLan(11),

proteon-10Mbit(12),

proteon-80Mbit(13),

hyperchannel(14),

fddi(15),

lapb(16),

sdlc(17),

ds1(18),

e1(19),

basicISDN(20),

primaryISDN(21),

propPointToPointSerial(22),

ppp(23),

softwareLoopback(24),

eon(25),

ethernet-3Mbit(26),

nsip(27),

slip(28),

ultra(29),

ds3(30),

sip(31),

frame-relay(32)

}