Create sections in front end to be divided into

1. System
2. Data
3. Products

# **System Level**

1. Create a python script which will be installed in all machines (clients)
   1. Script should make post request to database
      1. Hostname, ip, timestamp, success/failure
      2. Should log all requests
2. Python should run a cronjob to check the database and see if any systems have not responded in the given time slot (Server)
   1. Should capture success and failure count.

Logic: if the system does not send a response in a given time then it should show as issue

````

*Every 10 seconds, the daemon:*

1. Gets all active services *from* database

2. For each service, calculates "when should this run next?"

3. If a service is due *for* checking, runs the check

4. Updates the database *with* results

5. Calculates the next run time *for* that service

```

# **DATA SET LEVEL**

# Ocean Systems Monitoring

Systems

Alerts

Product

Data

There will be 4 levels of monitoring.

* Systems
  + Able to register a system to monitor
  + Test 200ok
  + Able to post and update system status from non spc machine which isn’t connected to the internet, e.g. post from remote sites.
  + For spc machines, we can test LAN 200ok and public IP 200ok
* Data
  + For now we will only monitor data ingested in ocean portal (type:oceanportal)
  + Sample api <https://ocean-middleware.spc.int/middleware/api/task_download/1/>
    - Here we see last\_runtime and last\_download file, check datetime from last download file to see what time the file was downloaded, it’s best to get time from file name, api below has information on how the file name is generated. Can replace the infix to the datetime of the file.
  + Parent api to above: <https://ocean-middleware.spc.int/middleware/api/dataset/1/>
    - Here we see frequency in which the dataset should be updated, so for this one it is 12 hourly, we give a buffer of 1 day (db field), if it does not update in 1 day time, we should change the status to Poor
  + Here is full list of tasks <https://ocean-middleware.spc.int/middleware/api/task_download/>
  + For each entry in the table, we should be able to tell the health of the datadownload, based on daily, yearly, etc.
* Products -TBC
* Alerts - TBC

**All Testing to happen once a day this we will schedule in crontab**

**Main Dashboard visualized in the big TV**

**Last updated: 20-June-2025 12:00:00UTC**

Systems

Alerts

Product

Data

4 main cards to show on the very top

* Inside each card to show Excellent and Poor counts
* E.g. System.excellent = 10
* System.poor = 10
* Then if we see poor then we go and check and reset the poor and excellent, else keep updating the excellent.

**Data set 1**

Since this

1. Check if all tasks exists in the monitoring table
   1. If doesn’t exists add it DBeaver Sample Database (SQLite)
2. Update status in monitoring table
3. Update logs in monitoring log table status, message

task\_name = concat (id + : + task\_name)

ip = “ocean-middleware.spc.int/middleware/api/”

port: 80

protocol: oceans portal

check interval based on when

interval type based on when

interval units based on when

last status based on valid/invalid

comments to have this two results from json

"next\_download\_file":

    "last\_download\_file":