**BEIA HQ TELEMETRY**

**TASK:** Practical test (please use google translate from Romanian to English): to be documented in a Gdoc in English and posted code in Github to connect IoT device (raspberry pi, esp, pycom, arduino, libelium, galileo, android, etc) through MQTT database, blockchain storage (Ethereum, IOTA, Hyperledger, etc.), orchestrating services with Arrowhead using docker and viewing in a mobile application, using Node-Red for data processing, viewing data in Grafana, setting notifications in Grafana , configuring local data storage in a folder with candidate name and auto-start resynchronizing local data with cloud in case of power outage or data network, data query via chatbot or voicebot.

link Grafana https://grafana.beia-telemetrie.ro/

user: interviu.practica

pass: beiapractica

server MQTT mqtt.beia-telemetrie.ro

topic MQTT / training / device / first-last-name /

link Arrowhead https://github.com/eclipse-arrowhead/core-java-spring

Arrowhead registry http://arrowhead.agile.ro:8443, orchestrator http://arrowhead.agile.ro:8441 and auth http://arrowhead.agile.ro:8445

Arrowhead tutorial https://docs.google.com/presentation/d/1YSk9fABJ7o5UNrHHwuLI-5dY92NqIQtvk9o4PoklqJ0/edit?usp=sharing

Example Arrowhead code https://github.com/beia/AHT\_wp4\_toolchain\_demo

Link Docker https://ocw.cs.pub.ro/courses/cc/laboratoare/01

More about this source textSource text required for additional translation information

**TECHNOLOGIES I USED:**

1. MQTT Data Store
2. JAVASCRITP
3. POSTMAN
4. BEIA HQ WEATHER AND TELEMETRY uRADMonitor SENSORS
5. IONIC JS FRAMEWORK
6. ANGULAR JS
7. SQL
8. POSTGREE
9. CORDOVA AND CAPACITOR JS
10. MAP JS

**URL OF APP:**

1. ANDROID APP: <https://drive.google.com/file/d/147EOW-VoL7ubcyl45MN4qRJJ8Kn5l-vW/view?usp=sharing>
2. iOS: Available on request

**VIDEO DEMONSTRATION:**

1. Video: <https://drive.google.com/file/d/1uMX0f3h6RKLdlCGk0H5dcxzOAgtce48e/view?usp=sharing>

**CODE REPOSITORY:**

1. GITHUB: <https://github.com/EnochOluwadamilare/beiahq-telemetry>

**PICTORIAL DEMONSTRATION:**

|  |  |  |
| --- | --- | --- |
| **S/N** |  |  |
| **Image** |  |  |
| **Note** | Welcome Page showing the button to the GitHub Repository | Menu Page showing the main menu that a user can navigate to. |
|  | | |
| **Image** |  |  |
| **Note** | Beia HQ List of all Telemetry Data |  |
|  | | |
| **Image** |  |  |
| **Note** | Sensor Results and Dataset | Sensor Results and Dataset. |

**DOCUMENTATION / EXPLANATION OF IMPLEMENTATION:**

* I searched and got programmatic access to the IOT uRADMonitor Sensors in BEIA Headquarters in Romania, and used Postman to test the payload and response of the list of IOT sensors.
* I used HTTP GET and POST requests to access the data from the IOT sensors through the bridging gateway of the sensors.
* I created a simple JavaScript CrossPlatform Application to create an attractive view and landing page for the data I got from the backend.
* I used Axios, Capacitor, Cordova, JavaScript and most importantly Java with other technology to package and bind the Angular JS Ionic App.
* I connected the Longitude and Latitude of the sensors to feed into a Google Maps on the App for you to know where these sensors are geographically placed.
* I used Android studio to finally sign the app and ready for a Debug controlled Test release.
* Data gathered were presented on MQTT dataset and graph.
* I pushed all the codes to GitHub for remote access and use.

**PUBLISH:** I would like BEIA to Publish the App so that it can be used to monitor the sensors of the company easily through the app. I can also work with other sensors to build a remote monitoring and remote-control system for the IOT Sensors and Devices if given the chance.