Next Gen Bitaxe Dashboard

# Frameworks & Deployment Method

* VueJS framework, using composition API
* Single Page Application
* All CSS will be placed in a CSS file, no embedding of CSS into code, JavaScript, HTML, etc.
* Dockerfile, docker build and docker deployment on Linux/AMD64 and Linux/ARM64

# Style Guidelines

* White: #f5f5f5
* Red: #ff1744
* Dark background: #121212
* Dark Header: #1f1f1f
* Green: #28a745

# Application Architecture

* Offload as much rendering and data fetching as possible to web browser
* Use proxy method to call device and mining core APIs
  + No direct exposure of device & mining core APIs to application frontend
  + Example: Application calls it’s API for device information, which in turn calls device’s API for system information.
  + Allows internal, private network API to be proxied to a external public internet call by the dashboard application.
* Use JWT authentication, can be enabled and disabled.
* Use SHA256 hash for passwords in transit

# Features

* Real time data from each device and mining core
* Can disable & enable mining core data pulls, in case user doesn’t have mining core installed
* All fields / Attributes mapped from JSON attribute names to display friendly names
* Format values to friendly formats
  + Hashrate: Mh/s, Gh/s, Th/s, Ph/s format
  + Voltage: convert millivolts to Volts
  + Power: expressed in watts
  + Frequency: Mhz
  + Difficulty (Network, Best & Session): M, G, T format
  + Response Time: Milliseconds (ms)
  + Uptime: Days, Hours, Minutes, Seconds.
  + For Over Heat Mode & Using Failover Stratum, displayed as Disabled, Enabled
* User can configure application parameters, which when saved will cause a real time reload of the application
* User can change configuration for each AxeOS device, using proxy calls to AxeOS System API
* User can restart each AxeOS device, using proxy calls to AxeOS restart API
  + Confirmation modal before executing the restart request to the AxeOS device
* Application will launch into bootstrap mode on first use, missing configuration files
  + Basic configuration will include
    - Application Title
    - Enable / Disable Authentication
      * If enabled,
        + input for username & password, with validation and unmask option
        + Ability to generate 32 Character JWT Token Key
        + Ability to set JWT age (i.e. 1day, 1hr, etc.)
    - Input for one or more AxeOS device Name & URL
      * Example: Bitaxe1, <http://192.168.7.220>
    - Enable / Disable Mining Core integration
      * If enabled, input for mining core URL
        + Example: <http://192.168.7.149:4000>
    - Submit button should say, “Create Configuration”
    - URL’s for AxeOS and Mining Core will be verified as accurate on submit
    - URL fields validated as valid URL formats
    - Once submit is successful, application will real time reload configuration and send user to application (or login if login enabled)
* Real time graph charts for each AxeOS Device, showing hashrate and ASIC Temp over time, using AxeOS API, called by application proxy.
* Refresh intervals for each AxeOS device’s data stream should be configurable and set to 25 seconds initially.
* Provide a logout icon that when clicked logs the user out (aka, invalidates their session).
* Configurations, Settings and Charts will be shown in popup modal windows.
* Loading, and loading errors will be displayed for each real time data stream.
* AxeOS and Mining Core availability status will be shown with a green (available) or red (unavailable) icon next to the friendly name.
* Data loading should happen independently for each data call, thereby not blocking the entire site while loading data, data appears as it becomes available.
* Use icons for Configuration, settings, chart and logout buttons. Icons located in the /icon folder
* All configurations should be in configuration files
  + Application Configurations
  + AxeOS field mappings
  + AxeOS URLs
  + Mining Core field mappings
  + Mining Core URL
  + Field types (Boolean, progress bar, number, string, etc.)
  + Progress thresholds for green, yellow and red status.
* If authentication is enabled
  + All API calls will validate the session before executing the call (aka backend security)
  + The logout icon will appear
* Allow the user to configure if configurations should be allowed in the web UI.
* Allow the user to configure if AxeOS settings can be changed in the web UI.
* A modal window to show more details for each AxeOS Device, beyond what is shown on the main page for each device.
* The ability to configure what fields are shown on the main page for each AxeOS device
* The ability to configure what fields are shown for each Mining Core pool on the main page.

# Folder structures

* Place source code into a /src folder
* Separate backend and frontend code by folder /backend & /frontend
* Separate, logically, backend components into folders under /backend
* Put all public facing code into a /public folder.
* Put configurations into a /config folder, above the /src. It has to be only configurations so it can be mapped to a docker -v mapping setting.