Gram Power (India) Pvt. Ltd.
Email: info@grampower.com
Website: www.grampower.com

# GRAM POWER TECHNICAL ASSIGNMENT ROUND

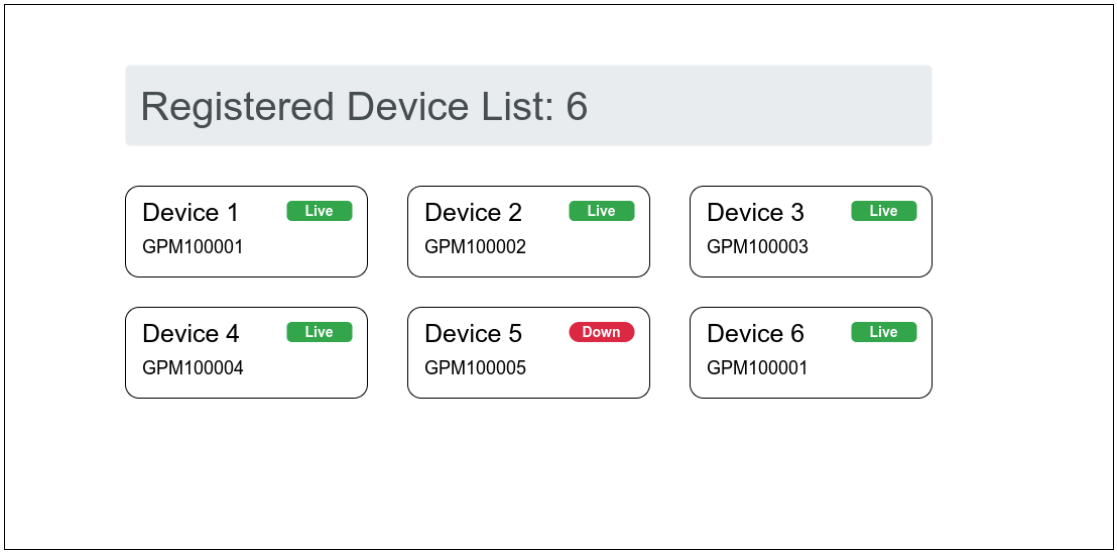
## Programming Assignment

At Gram Power, we push our engineers and design team to build beautiful and highly intuitive user interfaces for our consumers and at the same time our team is expected to build software that scales. We handle gigabytes of data on a daily basis and visualizing, analysing and managing this data is a core requirement of our technology. This assignment will test your design skills, code quality, quality of understanding of technology tools. While you design and implement your solution, we expect you to think carefully about how your software will behave when it is used at scale, i.e. will it be able to support large number of users or end points interacting concurrently with your software.

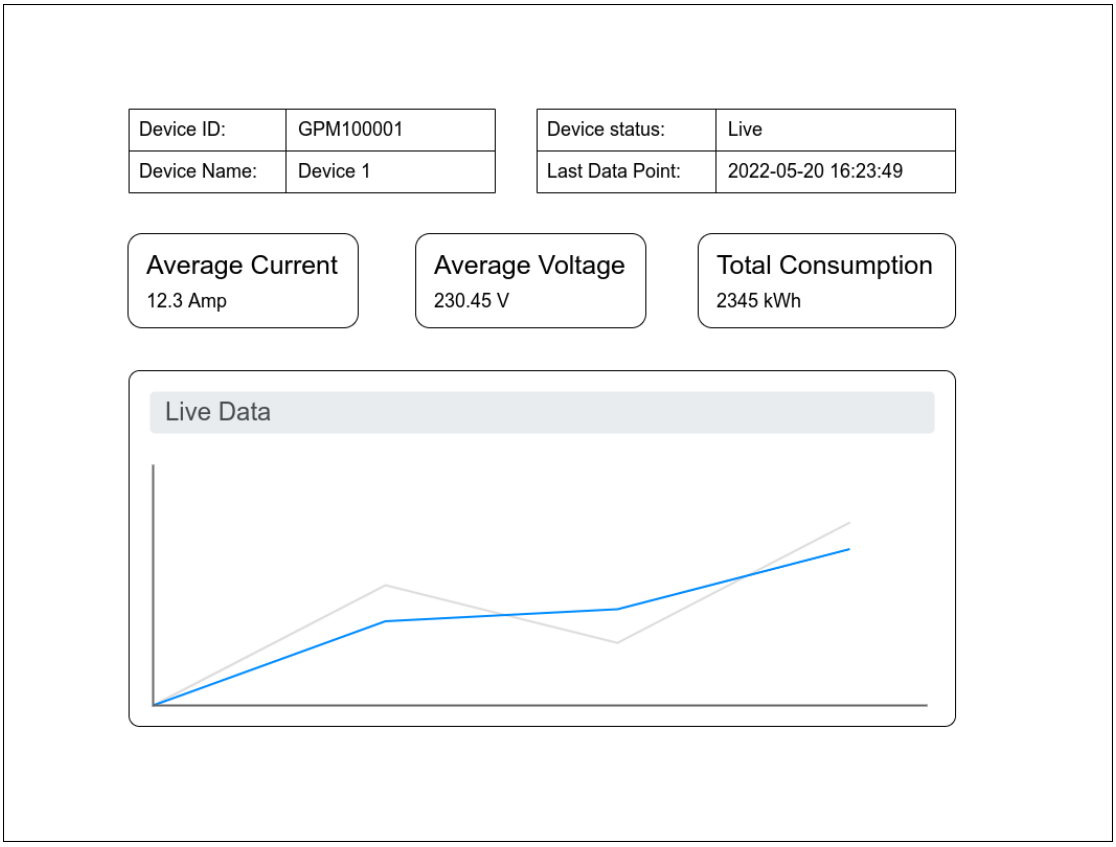
The task at hand is to create a simple web app in which a user can login and get redirected to a IoT devices dashboard page. There will essentially be three types of pages in the app:

1. User login page.
2. Page to view list of all devices.
3. Page to view details of a particular device live and aggregated data.

The final deliverable is a project which you publish on a free cloud hosting service such as Heroku or anything else of your choice. We should be able to view your project through a password protected URL. You can then give us access to the github repository to view the source code.



Gram Power (India) Pvt. Ltd.
Email: info@grampower.com
Website: www.grampower.com




We'd prefer you design your own wireframes for all three pages. If you want, you can use the wireframes above as a skeleton.

The web interface should:

1. Allow a user to login and redirect to dashboard
2. Show a list of all devices
3. In device detail page contain device's aggregated values in different cards
4. The chart on the device page should show **live data ingestion without refreshing the page**
5. If the device's last data point is more than an hour old, then it should show the status as 'down' instead of 'live'.

Your backend stack should:

1. Ingest data in a database of your choice coming from an MQTT connection (connection endpoint details are shared in the assignment email).
2. Create a websocket to show live data on frontend.

## Implementation Guidelines

Gram Power (India) Pvt. Ltd.
Email: info@grampower.com
Website: www.grampower.com

Pay special attention to the following while implementing the assignment:

1. Preferably use a Python based web framework (such as Django, Flask). But you are free to choose any other technology platform you are comfortable with and feel would be the most appropriate for this assignment.
2. Try to follow standard coding practices and comment down codes wherever necessary.
3. Document the code and also document the steps required to build and run the project.
4. Maintain the software in a version controlled repository (preferably **git**). The final zip file that you submit, should have the meta data (.git) folder which would allow us to look through your commits and branches. You can give us access directly to the git repository if that's easier.
5. Maintain a consistent casing for variables, functions, casing.
6. Unit Testing is highly advisable and implementation will add brownie points

**--All the Best!--**