Assignment

Nunam Technologies India Pvt Ltd

Deadline: 3 Days from the date of receiving the assignment

Create a Simple two-fold Dashboard using any of the preferred frameworks of your choice. The task is to create a simple two-page dashboard using any standard framework for which data is already provided. Look at the section titled "*Deliverables*" to understand how the assignment is to be submitted.

Description:

For this task, the data is provided in a multi-sheet separated into 2 excel files where each file determines separate Li-ion cells and the file names without their extension determine the unique **cell-id** to identify them.
For example, if the file name is '**123.xls'**. Then the cell-id is **123** and must be determined in that way for identification purpose in the dashboard.

Description of each page are provided below:

Page 1

For the sub-page of the dashboard named '**Dashboard**': 2 pie charts represent the state of health for 2 cell-id represented in percentage as shown in the graphics below.

To calculate the State of Health (SoH) parameter:
*(Discharge Capacity / Nominal Capacity) *100*

Cell-id	Discharge Capacity	Nominal Capacity
5308	2992.02	3000
5329	2822.56	3000

Where Discharge Capacity and Nominal Capacity are mentioned above.

Page 2

Each Subsequent cell id page has an identical structure where 4 different block projects 4 different graphs/charts. To get data for the charts, look at the below table to find the data from specific sheets and columns.

Current Data:	Sheet no: 4, Column No: 6
Voltage Data:	Sheet no: 4, Column No: 7
Capacity Data:	Sheet no: 4, Column No: 8
Temperature Data:	Sheet no: 6, Column No: 5
Time Data:	Sheet no:4,Column No: 11

The whole dashboard should have 3 pages where the first page is an overview and the next 2 pages are drop-down menus identifying each Li-ion cell by cell-id where each subpage provides 4 charts for each cell-id.

Deliverables:

1. Add all the provided data into a relational database of your choice (MySQL, PostgreSQL)
2. Develop REST API endpoints to get the data from the database instead of direct database connection.
3. Use any charting library of your choice (Chart.js, Plotly for Python etc.) for the graphs and visualizations.
4. Serve the UI dashboard over port 8080.
5. Add a README file on how to run the project including step-by-step instructions.
6. Add a small GIF of the UI dashboard in a working situation to show its workings and usability.
7. Please make sure that the whole project is git tracked and available in public Git repositories (GitHub, Gitlab, Bitbucket)

Bonus Points:

1. Add unit test coverage for at least 70% of your codebase and add a text file to indicate the success/failure rate of all the functions.
2. Add basic authentication for the REST API endpoints.
3. Add pagination and filtering in the REST API endpoints.
4. Create OpenAPI specification-based API documentation for the created API endpoints.

How to Submit:

A separate assignment submission link is mentioned along the email. Use that link to submit the git repository link.

Do keep in mind to add documentation of some form

In case you are not able to submit the whole assignment within the given time, submit with as much as you have completed. We are primarily looking at how you are solving the given problem set and your technical expertise and your ability to learn things quickly, not at a full-fledged software developer.

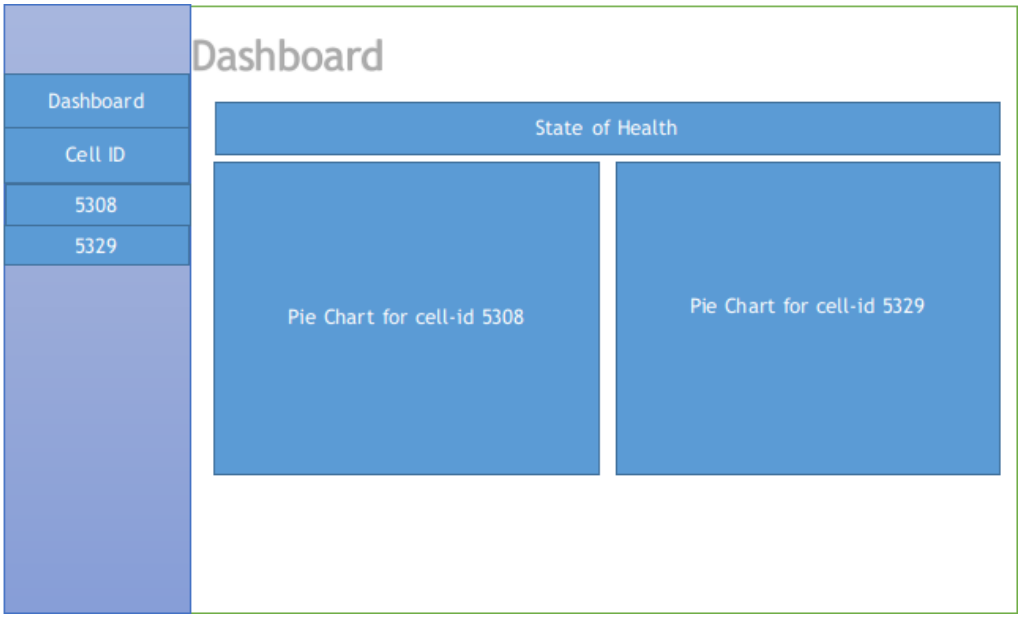
We are also not attached to any specific tech-stack, so use whatever you are familiar with.

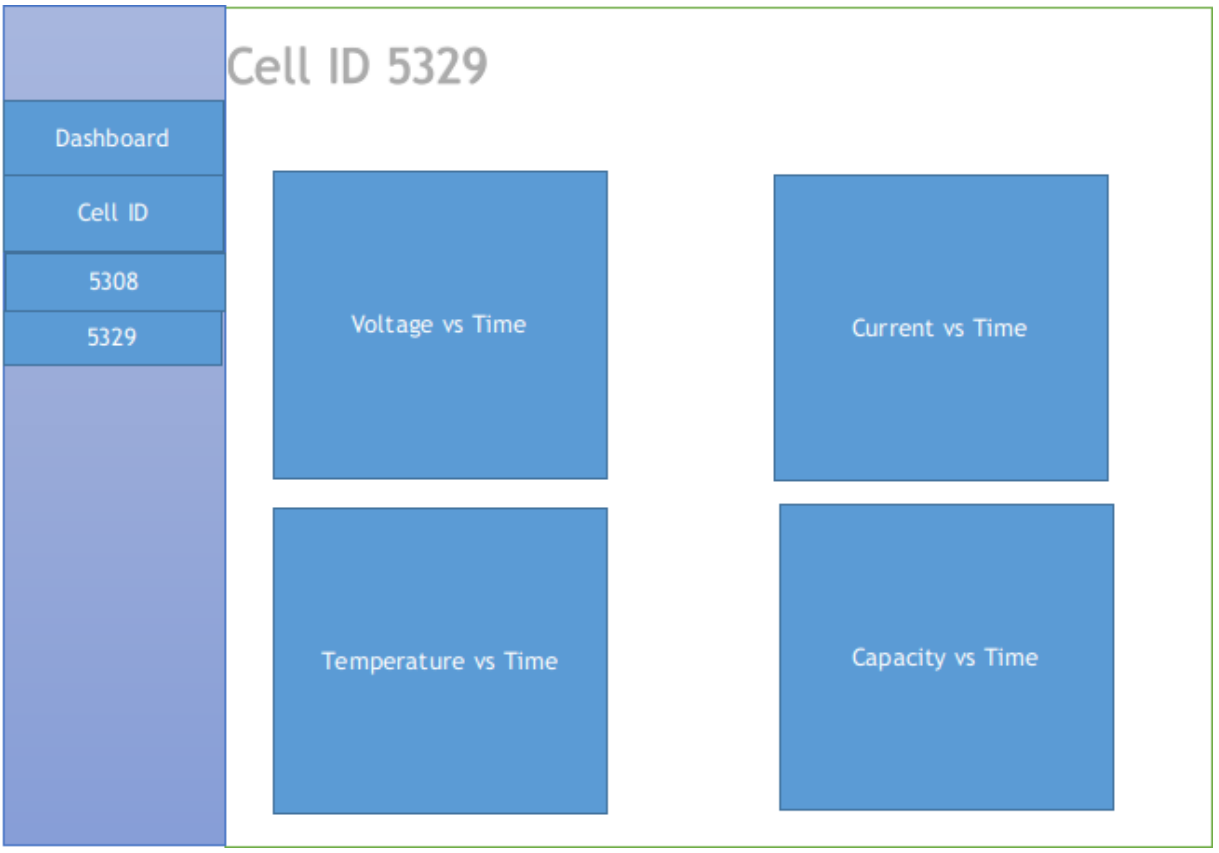
In case of any doubt, email at saradindu@nunam.com

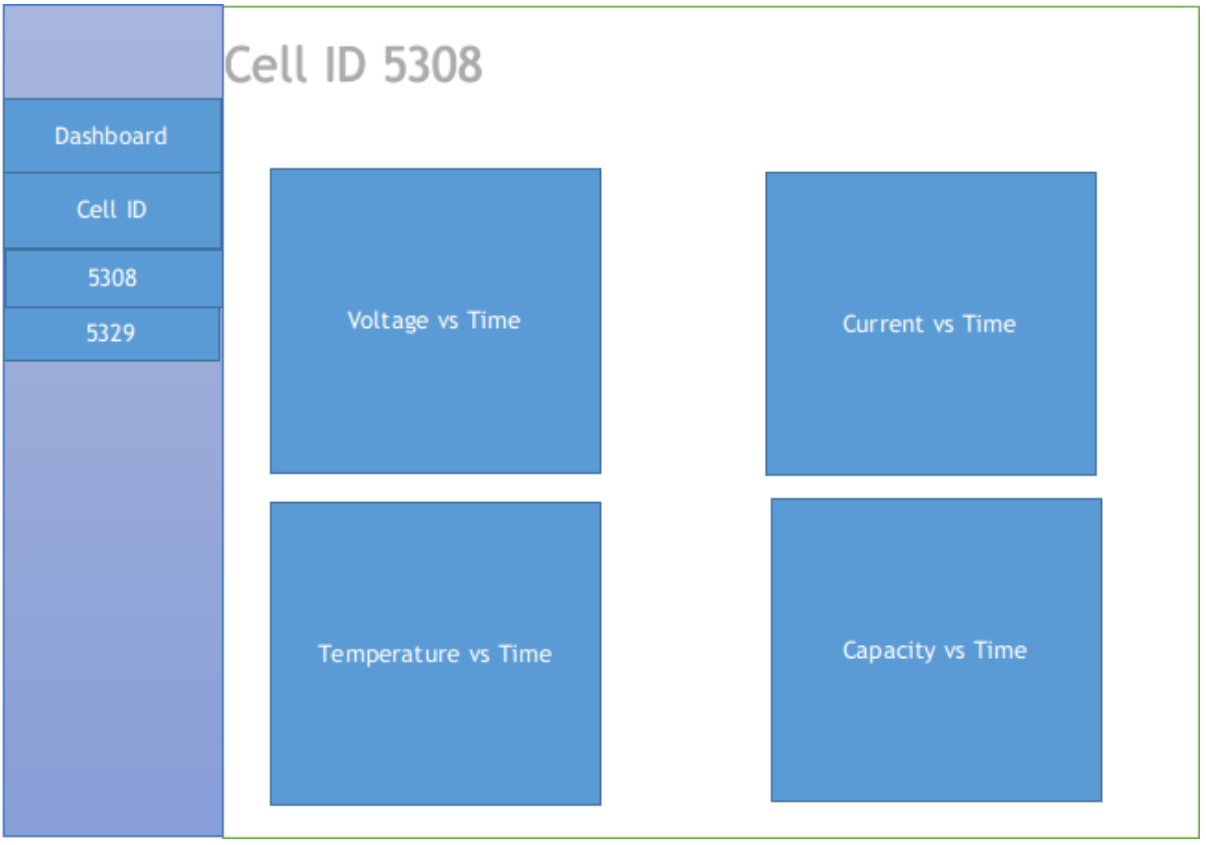
Below are a set of sample dashboard to help you.

Sample Expected Result:

The basic overview design of the dashboard is provided below.




Cell ID 5329
Dashboard
Cell ID
5308
5329
Voltage vs Time
Current vs Time
Temperature vs Time
Capacity vs Time


Cell ID 5308
Dashboard
Cell ID
5308
5329
Voltage vs Time
Current vs Time
Temperature vs Time
Capacity vs Time