Mini Energy Management Dashboard –
Detailed Take-Home Assessment

1. Real-Time Power Flow (Live Chart)

Component Responsibilities

- Render a real-time Active/Reactive Power waveform using ECharts with smooth line animations, matching the red sinusoidal “Live Chart” UI.
- Maintain a rolling window of 100 data points on the chart at all times.
- Provide Start and Stop buttons below the chart to control polling instantly.

Data Flow & Polling Behavior

- Poll the mock API /api/power/live every 60 seconds (60,000 ms).
- Each API response returns exactly 20 new data points.
- When new data arrives:
    - Append the 20 new points to chartData.
    - If the length exceeds 100, remove the oldest 20 points so the array never grows beyond 100.

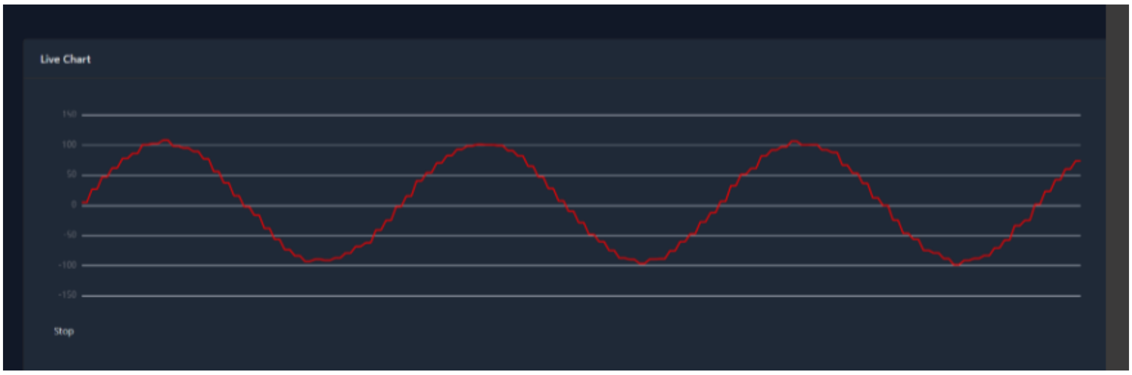
State Management – powerSlice

powerSlice must store:

- currentPower: Latest Active and Reactive power values returned from the live API.
- chartData: Array of up to 100 power data points used by the ECharts Live Chart.
- isPolling: Boolean flag to track whether the live polling loop is active.
- loading / error: Status flags for handling async API lifecycle and failures.

Custom Hook – usePolling(apiFn, interval)

- Internally manages setInterval and teardown logic for the live polling.
- Exposes startPolling(), stopPolling(), and a polling state flag to the Live Chart container.
- stopPolling() must clear the interval immediately so no further API calls occur once stopped.



2. Alarms Management Module (Event Table)

Component Responsibilities

- Display an alarms/events table similar to the dark "Events" UI, including severity, site name, event, times, and tags.
- Provide a real-time searchable interface above the table.
- Allow marking and unmarking alarms as favorites, visually indicated in the table (e.g., star icon).
- Provide a toggle or filter control to switch between:
  - "All Alarms" view (full list from API)
  - "Favorites" view (only favorited alarm IDs)

Search Behavior

- A text input above the table must drive server-side search.
- User input must be debounced by 400 ms so API calls do not fire on every keystroke.
- Debounced search logic must be implemented via the useDebounce() hook.

API Requirements

- Mock endpoint: GET /api/alarms.
- Must accept an optional ?search= query parameter to filter alarms by code, description, or site name on the backend mock.

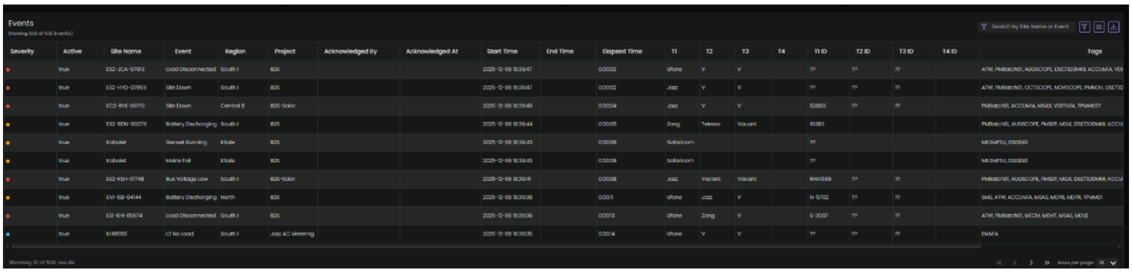
State Management – alarmSlice

alarmSlice includes:

- alarms: Full list of alarms fetched from /api/alarms.
- filteredAlarms: Derived list based on the current debounced search term and favorites filter.
- favorites: Array of favorited alarm IDs.
- loading / error: Flags for API fetch status and error handling.

Custom Hook – useDebounce(value, delay)

- Returns a debounced version of value after delay ms.
- Cancels the previous timeout when value changes so only the latest input is applied.



Events

Showing 500 of 500 Event(s)

Severity	Active	Site Name	Event	Region	Project	Acknowledged By	Acknowledged At	Start Time	End Time	Elapsed Time	T1	T2	T3	T4	T1 ID	T2 ID	T3 ID	T4 ID	Tags
●	true	ES2-JCA-07913	Load Disconnected	South I	B2S			2025-12-08 18:39:47		0:00:02	Ufone	V	V		??	??	??		ATW, PMBatch01, AUGSCOPE, DSE7320MKII, ACCUMIA, VE
●	true	ES2-HYD-07958	Site Down	South I	B2S			2025-12-08 18:39:47		0:00:02	Jazz	V	V		??	??	??		ATW, PMBatch01, OCTSCOPE, NOVSCOPE, PMNOV, DSE732
●	true	EC2-RYK-08718	Site Down	Central B	B2S-Solar			2025-12-08 18:39:45		0:00:04	Jazz	V	V		52808	??	??		PMBatch01, ACCUMIA, MSAS, VERTIVIA, TPMM08T
●	true	ES2-BDN-00078	Battery Discharging	South I	B2S			2025-12-08 18:39:44		0:00:05	Zong	Telenor	Vacant		61283				PMBatch01, AUGSCOPE, PMSEP, MSLK, DSE7320MKII, ACCU
●	true	Kobolet	Genset Running	Kitale	B2S			2025-12-08 18:39:43		0:00:06	Safaricom				??				MEGMFSU, DSE890
●	true	Kobolet	Mains Fail	Kitale	B2S			2025-12-08 18:39:43		0:00:06	Safaricom				??				MEGMFSU, DSE890
●	true	ES2-KSH-01748	Bus Voltage Low	South I	B2S-Solar			2025-12-08 18:39:41		0:00:08	Jazz	Vacant	Vacant		RAN1389	??	??		PMBatch01, AUGSCOPE, PMSEP, MSLK, DSE7320MKII, ACCU
●	true	EN1-ISB-04144	Battery Discharging	North	B2S			2025-12-08 18:39:38		0:00:11	Ufone	Jazz	V		N-5702	??	??		SMS, ATW, ACCUMIA, MSAS, MDPB, MDPR, TPMM01
●	true	ES1-KHI-05674	Load Disconnected	South I	B2S			2025-12-08 18:39:36		0:00:13	Ufone	Zong	V		S-3007	??	??		ATW, PMBatch01, MECM, MEMT, MSAS, MDSE
●	true	KHI8555	L3 No Load	South I	Jazz AC Metering			2025-12-08 18:39:35		0:00:14	Ufone	V	V		??	??	??		EMMIA

Showing 10 of 500 results

---

3. Site Details Page (Energy & Alarms)

Route

- Accessible via /site/:id.

Layout & Component Responsibilities

- Top “Alarm Summary” showing counts for:
  - Site Down, Critical, Major, Minor – styled as in the first screenshot.
- “Energy Report 1.0” section with ECharts daily energy bars (Solar vs Grid), matching the stacked bar chart UI.
- Maintenance Ticket Drawer that slides in from a side of the page to create a new ticket.

Daily Energy Consumption Chart

- Use ECharts to render a stacked bar chart similar to the “Energy Report 1.0 – Total Energy” view.
- X-axis: dates (e.g., daily timeline).

- Y-axis: energy in kWh, with separate series for Solar and Grid.
- Data source: siteSlice.energyStats.

Maintenance Ticket Drawer

- Opens from a button on the Site Details page (e.g., “Create Ticket”).
- Form fields:
  - Same as given in the screen shot
- Submission:
  - Simulate API call with setTimeout to mimic async behavior.
  - On success, close drawer (or reset form) and show a success notification (toast/snackbar).
  - On failure, show an error notification and keep the drawer open.
- All successfully submitted tickets must be stored in maintenanceSlice.

State Management – siteSlice

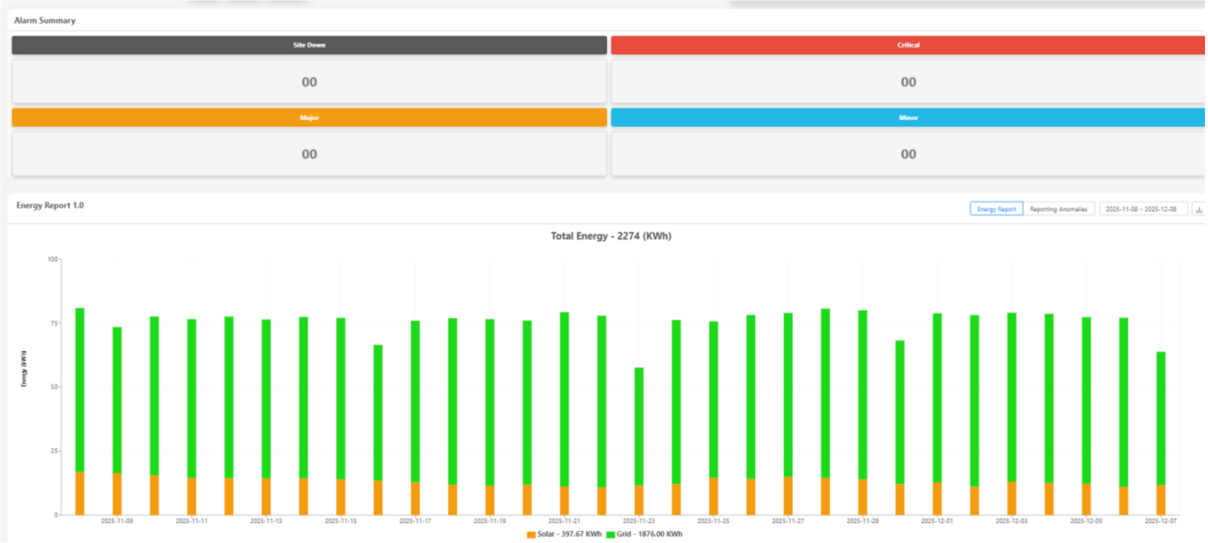
siteSlice must store:

- siteInfo: Basic metadata for the selected site (name, location, and current operating mode).
- mode: One of Grid Following, Microgrid, or Standalone, displayed in the header or summary area.
- energyStats: Daily energy data used for the ECharts Energy Report chart and total energy number.

State Management – maintenanceSlice

maintenanceSlice must store:

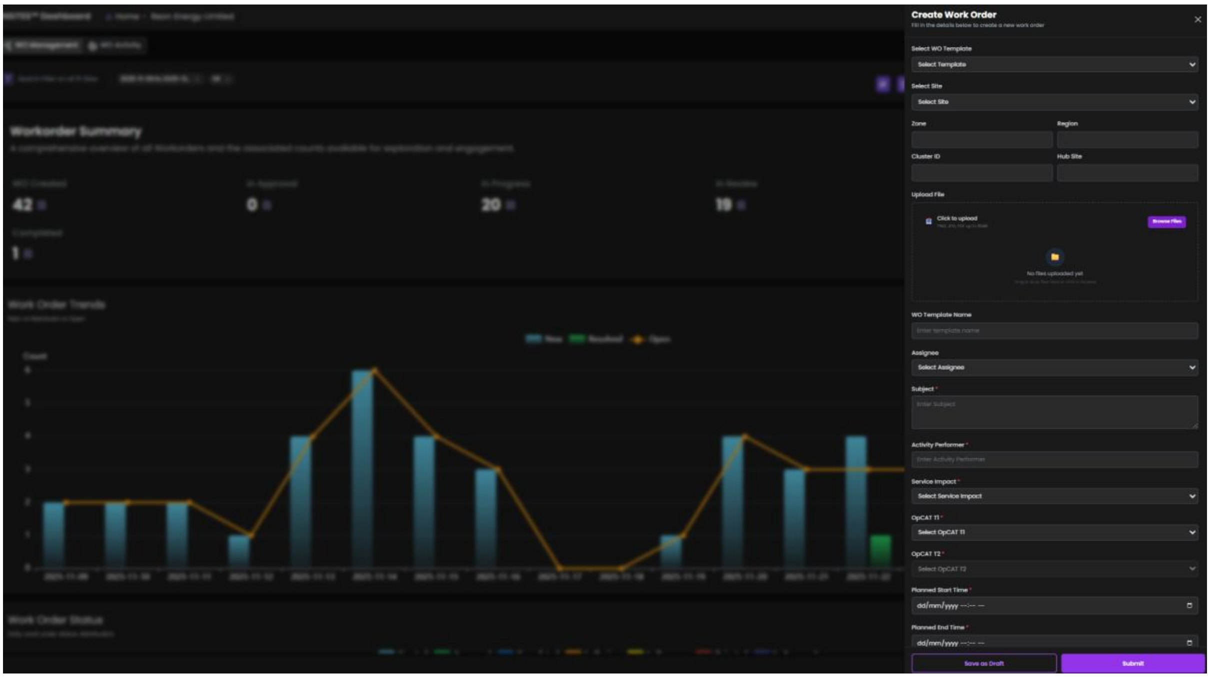
- tickets: Array of all maintenance tickets created from any site.
- submitting: Boolean that is true while a ticket is in-flight (used to disable the submit button and show a spinner).
- lastSubmitted: The most recently created ticket object for quick feedback or inline display.



Alarm Summary

Site Down	Critical
00	00
Major	**Minor**
00	00

Energy Report 1.0



Create Work Order

Fill in the details below to create a new work order

Select WO Template: Select Template

Select Site: Select Site

Zone:

Region:

Cluster ID:

Hub Site:

Upload File

Click to upload

No files uploaded yet

WO Template Name: Enter template name

Assignee: Select Assignee

Subject *: Enter Subject

Activity Performer *: Enter Activity Performer

Service Impact *: Select Service Impact

OpCAT T1 *: Select OpCAT T1

OpCAT T2 *: Select OpCAT T2

Planned Start Time *: dd/mm/yyyy --:-- --

Planned End Time *: dd/mm/yyyy --:-- --

4. Deliverables

Source Code & Documentation

- GitHub repository containing complete source code, including Redux slices, custom hooks, components, and routing.
- Clear README with:
  - Setup instructions
  - Environment variables and mock API description (/api/power/live, /api/alarms, ticket submission simulation)

    - Screenshots or links referencing the Energy Summary, Events table, and Live Chart views.

Deployment

- Live deployment on Vercel with:
    - Working Real-Time Power Flow Live Chart and Start/Stop controls.
    - Alarms Management page mirroring the dark “Events” table UI.
    - Site Details page reproducing the Alarm Summary, Energy Report 1.0 chart, Alarm History table, and Maintenance Ticket Drawer.

---

5. Data

You can create mock data according to your own understanding. You can use AI models to generate mock data but make sure not to use AI for core logics and functionalities.