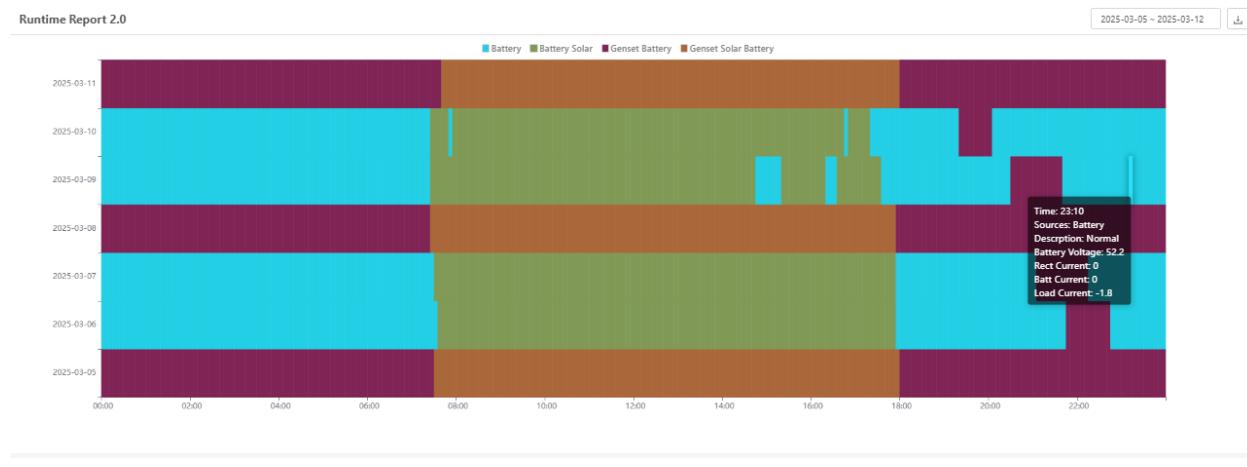


# Frontend Developer Assessment: Heatmap Visualization

**Objective:** Develop an interactive heatmap visualization that represents runtime data based on a JSON file. The heatmap should be similar in appearance and functionality to the provided example image.

Example Heatmap:



## Requirements:

### 1. Data Handling

- The heatmap should accept and parse a JSON file that contains runtime data.
- Ensure efficient handling of large datasets without performance issues.
- The “rtsources” field in the data object represents the current source.
- The “value” field in the meta object corresponds to the rtsources in the data.
- Use the color field in the meta object to determine the heatmap cell color.
- Include a tooltip displaying the desc field from the meta object when hovering over a heatmap cell.

Note : rtsources field in data object and and “value” field in the meta , ensuring the heatmap color and tooltip display the appropriate metadata

### 2. UI Design & Implementation

- Implement a user-friendly interface that closely resembles the provided example.
- Display a title "Runtime Report" at the top.
- Use a grid-based heatmap to represent data, where different colors indicate different statuses.
- Add a legend to indicate color representations (e.g., Battery, Battery + Solar, Genset +Battery, Genset+Solar+Battery).
- Display a date range selector at the top right corner.
- Provide an option to export/download the chart.

### **3. Functionality**

- Enable users to hover over heatmap cells to view details.
- Allow zooming or filtering by date range.
- Ensure real-time updates if new data is provided.
- Implement smooth transitions between state changes.

### **4. Tech Stack**

- Use React.js
- Utilize ECharts for visualization, specifically using the example provided in the link: [ECharts Heatmap Example](#)
- CSS or Tailwind for styling and responsiveness.
- Fetch API for data fetching.

### **5. Performance Considerations**

- Optimize rendering for large datasets.
- Implement lazy loading or virtual scrolling if necessary.
- Ensure minimal re-renders for smooth interactions.

### **6. Deliverables**

- A fully functional heatmap UI that matches the example.
- Source code hosted in a GitHub repository.
- A README file with setup instructions.

- A live demo link (optional but preferred).
- 

**Evaluation Criteria:**

- Accuracy of the visualization compared to the example.
- Code quality, structure, and best practices.
- UI/UX quality and responsiveness.
- Proper documentation.