# Data Upload And Validation & Tariff Configuration

Reuse md\_shaving\_v2.py  
  
Peak Events Timeline  
Monthly Target Calculation Summary Table (reuse v2)  
Power Consumption With Peak Events Highlighted (reuse v2)  
Peak Event Detection Result (reuse v2)  
Monthly Summary (TOU MD excess and TOU Required Energy) (reuse v2)

Battery Selector (reuse v2)Recommended battery size  
Dropdown table to allow user select battery

Financial Analysis Table

# Forecasting

Checkbox to enable method  
Dropdown table to let users select which method to use   
P90 calculations

Table to showcase error   
  
MD ShavingToggle: Historical/Actual

1. General Shaving   
2. Battery Conservation Mode   
3. Hybrid (switch ON/OFF)

* Include dummy weather data, future: integration with Msia MET API or temperature sensor

4. Tail aware shaving

* **Idea:** detect “events,” then keep shaving through the tail with adaptive rules rather than a fixed on/off trigger.
* **Event detection:** Start when actual(t) (or forecast P90(t) in forecast mode) exceeds MD\_target + deadband **or** ROC(t) > threshold for N consecutive mins.
* End when the *moving max* over the last W\_end mins drops below the target and ROC is small.
* **Tail control:** Maintain a **tail window** that *extends automatically* if any of these fire:

1. ROC(t) still positive above a small floor,
2. short spikes (< W\_spike) reoccur within the window,
3. temperature or CDD is high (cooling inertia).

* Discharge shape in tail: **exponential decay cap**

req\_tail(t) = req\_peak \* exp(-(t - t\_peak)/τ), clipped by SOC/power limits.

* If **SOC is tight**, switch to **pulse shaving** in tail: small bursts to clip just the micro-peaks.
* **One liner for devs**

“Keep shaving until *both* intensity (moving max) and velocity (ROC) are quiet, and taper with a decay curve so you don’t stop too early.”

Forecast Data Display:

Main chart (keep it simple)

* Actual y(t)y(t)y(t) (solid line).
* One chosen control horizon y^h(t)\hat y\_h(t)y^​h​(t) (e.g., 5-min ahead), with its P10–P90 band if available.
* Shaving overlay (required shave & applied shave) for that horizon.
* Toggle to switch hhh (1–10). Default to the horizon you intend to control with in production.

This matches real-time usage (you’ll act on a specific lead time), and keeps stakeholders focused.

Side panel / small multiples (how horizon changes things)

* Mini-cards or a small grid: for each hhh, show MAE/MAPE/RMSE and tail-miss rate.
* Optional line chart: error vs lead time (1…10 min). Easy to grasp: “longer lead, more error.”

Peak/event zooms (what you care about most)

* Event-aligned strips: for each detected peak window, show Actual vs y^h\hat y\_hy^​h​ (selected horizon) with tail shading and where the controller acted/missed.
* A carousel of events or a dropdown to jump to a specific event.

Heatmaps (for pattern discovery)

* Day × hour (or day × lead time) heatmap of MAPE or tail-miss rate. Highlights systematic patterns (e.g., Fridays 14:00 are tough).