Smart Home Energy Monitor ? Implementation Report

1. Backend/middleware/auth.cjs

- Replaced the custom HMAC token logic with jsonwebtoken, keeping PBKDF2 password hashing but delegating cookie signing/verification to a single, audited library. This reduces security drift and makes rotation/expiration easier to manage.

2. Backend/utils/jwt.cjs

- Removed the unused alternate JWT helper so there is exactly one auth implementation to maintain, eliminating confusion about which module the server actually uses.

3. src/utils/energyContextTypes.ts

- Extended the context contract with a ‘refreshBrokerConfig‘ callback so components such as Settings/Onboarding can notify the provider whenever broker/onboarding data in localStorage changes.

4. New modular hooks

- src/hooks/useEnergyState.ts: centralizes persisted state \(devices, alerts, tariffs,) Tj 0 -14 Td (etc.\) p
than storage plumbing.
- src/hooks/useOnboardingConfig.ts: tracks broker URL + onboarded flag with stable dependencies and storage-event syncing, preventing repeated MQTT reconnect loops.
- src/hooks/useMqttLifecycle.ts & useMqttSubscriptions.ts: isolate connection/resubscribe logic and topic handling, making reconnect behavior and message processing easier to test.
- src/hooks/useAutomationRunner.tsx: moves automations, standby kill, and offline-health monitoring into a dedicated runner; keeps EnergyContext readable while preserving toasts/commands.
- src/hooks/useBudgetAlerts.ts: debounces budget toasts so users only see 75/90/100% alerts once per day instead of every minute.

5. src/contexts/EnergyContext.tsx

- Refactored to consume the new hooks, drastically shrinking the file and giving each responsibility \(state, MQTT, automations, budgets\) its own module. The provider still exposes device CRUD plus the new refresh helper, but now renders more predictably and is simpler to maintain.

6. src/pages/Onboarding.tsx

- After a successful ‘connectAndWait‘, trims and saves the broker URL, calls ‘refreshBrokerConfig‘, and otherwise behaves as before. This ensures the provider immediately sees the new broker without relying on unstable ‘localStorage.getItem‘ dependencies.

7. src/pages/Settings.tsx

- Broker saves are now validated: tariff/TOU settings persist immediately, but the broker URL is only stored \(and the provider refreshed\) if ‘connectAndWait‘ succeeds. Users get inline error text and a disabled button while a save is in progress, stopping ?locked out? scenarios caused by a bad host.

8. docs/codebase-overview.pdf & docs/improvement-suggestions.pdf

- Added shareable write-ups summarizing the architecture and the agreed-on improvements. These help teammates or stakeholders review the system without needing the CLI transcript.