

VoltGuard - Description Document

1. User Authentication

Purpose:

Secure user access to personal energy usage data.

Requirements:

- Use Auth0 for OAuth-based login via Google.
- Allow manual login using username/password.
- Store logged-in user sessions securely using Flask sessions.
- Prevent access to rate/device APIs if not logged in.

2. Electricity Rate Management

Purpose:

Let users set their local electricity cost per kilowatt-hour (\$/kWh), essential for accurate cost calculations.

Requirements:

- Provide a dropdown with prefilled state-by-state average rates.
- Allow users to manually enter a custom rate.
- Store the selected rate in the database (**User.rate**).
- Auto-load saved rate on page load using **/api/rate**.
- Recalculate energy costs in real-time when the rate changes.

3. Device Usage Management

Purpose:

Enable users to add, edit, and remove devices to track their electricity usage and cost.

Requirements:

- Input fields for: device name, power consumption (watts), daily usage hours, and category.
- Save new devices via POST to **/api/devices**.
- Update existing devices via PUT to **/api/devices/<id>**.
- Delete devices via DELETE to **/api/devices/<id>**.
- Automatically refresh and render updated device list and total costs.

- Validate user ownership of each device for updates/deletion.

4. Cost Calculations

Purpose:

Calculate the total daily energy cost based on user devices and selected rate.

Requirements:

- Calculate kWh/day = (watts × hours) / 1000 for each device.
- Calculate daily cost = total kWh × selected rate.
- Display per-device and total daily cost in UI.
- Update calculations in real-time when devices or rate change.

5. Pie Chart Visualization (Cost by Category)

Purpose:

Help users visually understand which categories consume the most energy.

Requirements:

- Render a doughnut chart using Chart.js.
- Group devices by category.
- Aggregate and display cost contribution of each category.
- Update chart dynamically when device list or rate changes.

6. Solar Savings Calculator

Purpose:

Show users how much they could save daily, monthly, and yearly using solar panels.

Requirements:

- Calculate daily savings = daily kWh × rate × 0.8 (assuming 80% efficiency).
- Show monthly = daily × 30; yearly = daily × 365.
- Include a UI section explaining calculation steps.
- Update dynamically with every change to device list or rate.

7. VoltBot Chat Assistant

Purpose:

Provide users with personalized energy-saving tips and support.

Requirements:

- Embed a floating chatbot UI (popup).
- Use Google Gemini API to process user queries.
- Pass current device list, rate, and cost as context for more relevant responses.
- Display responses and file/image uploads in chat interface.
- Include a recommendation shortcut button: “Get Energy-Saving Recommendations.”

8. Map Placeholder (Future Expansion)

Purpose:

Provide a visual location-based element for future features like solar potential, regional tips, or smart grid info.

Requirements:

- Currently displays a placeholder `<div>` with ID `#map`.
- Styled with modern container aesthetics.
- Optional Google Maps integration in future updates.