Implementation Approach for SLTPower Prox Dashboard

Project Title: Energy Track 24/7 – Energy Monitoring Dashboard
Role: React Developer Intern
Duration: 3 Days
Branch Name: Janaka-Task

Implementation Approach for SLTPower Prox Dashboard

The **SLTPower Prox Dashboard** was developed using modern front-end technologies to provide a responsive and interactive interface for power asset and energy consumption monitoring. The key objective was to build a modular, scalable, and visually consistent dashboard based on the provided UI specifications.

1. Project Initialization

The project was bootstrapped using **Vite** for fast development and optimized builds. **React** with functional components and hooks (useState, useEffect) was used to manage UI state and interactivity.

2. Styling and UI Framework

Tailwind CSS was integrated for utility-first styling, ensuring rapid development of responsive layouts. For UI components, **shadcn/ui** was used to implement accessible and customizable elements like tabs, cards, and buttons.

3. Component Structure

A clean, component-based architecture was adopted. Components were organized into reusable blocks (e.g: Navtabs), each responsible for rendering specific data or interface elements.

4. Data Visualization

Charts and graphs were integrated using mock data to simulate real-time monitoring. Data was represented using bar charts, line graphs, and summary cards to display metrics like energy usage, Energy Distribution, and performance.

5. Responsiveness and Accessibility

The layout was built with responsiveness in mind using Tailwind’s grid and flex utilities. shadcn/ui ensured accessible components that align with ARIA standards.

6. Testing and Finalization

The application was tested across screen sizes to ensure responsiveness. Components were refined for consistency with the UI reference. Code was organized and pushed to GitHub under a dedicated branch.

Technologies Used

This project leverages modern front-end technologies and frameworks to ensure maintainability, responsiveness, and development efficiency. The following tools and libraries were utilized:

Technology	Description
Vite	A fast frontend build tool that improves the development experience with lightning-fast hot module replacement (HMR) and optimized builds.
JSX	JavaScript XML, a syntax extension that allows HTML-like code within JavaScript, primarily used in React for defining UI components.
React	A powerful JavaScript library for building interactive and modular user interfaces through component-based architecture.
shadcn/ui	A modern UI component library built on top of Radix UI and Tailwind CSS. It provides accessible, unstyled components that are fully customizable.
Tailwind CSS	A utility-first CSS framework that enables rapid styling through predefined class utilities, improving consistency and reducing custom CSS overhead.

References

1. Vite – Official Site
2. React Documentation
3. shadcn/ui Documentation
4. Tailwind CSS Documentation