dk

> "In India, energy usage in homes is rapidly increasing. But most households still don’t use smart systems to manage power.
>
> At the same time, we face:
>
> - High electricity bills
> - Unreliable power supply
> - Poor usage of renewable sources like solar

> Our project aims to solve these real problems faced by Indian homes using a smart solution that combines:
>
> - Machine learning
> - IoT sensors
> - And a simple Android app.

> It helps reduce costs, survive power cuts, and use solar/wind power more effectively.
>
> That’s why we designed this system, keeping **Indian households and their energy challenges** in mind.”

2. App Structure Overview

> “Our system has three main components:
>
> 1. **ESP32-based hardware system** to sense energy flow, battery level, and control switching.
> 2. **Machine learning models** to predict energy usage and generation.

> 3. **Android app** to show real-time data and suggestions to the user.

> These are connected through Wi-Fi, cloud, and local Flask server, enabling real-time energy decisions."

🔍 3. Prediction Module – What We Predict and Why

> "We trained ML models to predict next day's energy usage, solar output, and wind generation, based on weather and past data.
>
> These predictions help us plan battery usage and switching decisions in advance — especially useful if solar is low tomorrow or a power cut is expected."

🔁 4. Switching Logic – How Power Flows Are Controlled

> "Our ESP32 runs logic to control power sources like:
>
> - If solar generation is high → use solar first, then charge battery.
> - If battery is full and load is high → use from battery to reduce grid usage.
> - If battery is low and no solar → switch to grid.

> This logic works both in real-time and based on next-day predictions — ensuring cost-saving and energy efficiency."

⚠️ 5. Power Cut Prediction – Special Feature

> “We fetch power cut info daily from news websites and official portals using web scraping and RSS feeds.
>
> If a power cut is expected tomorrow, the system **stores more battery today** by delaying non-urgent loads and stopping battery discharge.
>
> This improves **resilience** during outages.”

💡 6. Smart Suggestions – Helping the User

> “Based on usage predictions and solar forecasts, our app suggests:
>
> - Best time to run washing machine or AC
> - Alerts user if solar is producing well
> - Tells user if tomorrow power cut is likely, so they can plan”

🤖 7. Automated Switching – Based on Real-Time + Predicted Data

> “Our app and microcontroller work together to:
>
> - Monitor sensors in real-time
> - Compare with ML-predicted needs
> - Automatically switch between **grid, battery, or solar**
> - Maintain battery reserve based on **predicted power cut info**

SOC (State of Charge) is like a fuel gauge for batteries.

It tells us how much battery is left — example: 100% = fully charged, 30% = low.

1.If battery goes to 0% repeatedly,
battery lifespan reduce aagum

2.Overcharging heats up the battery → **safety issue**
Battery chemicals degrade →
capacity reduce aagum

Charging power gets wasted (no efficiency)

- **Voltage levels** (most common method)
- **Coulomb counting** (measuring charge in/out — accurate)
- **ESP32 + Current sensor** → Estimate how much battery used and charged

- SOC is estimated using voltage sensor data
- Prevents deep discharge & overcharge
- Maintains battery health and backup
- Used in switching logic to decide power source
- SOC + predicted usage + power cuts → controls charging/discharging